Chapter 2
Information Visualization

This chapter introduces information visualization as a canonical foundation of this thesis. We will first try to differentiate information visualization from related areas. The goal is to have a common understanding of the term information visualization in context of this work. Thereafter we outline the interdisciplinary character of information visualization. For this we start with the human and introduce various models and research outcomes on visual perception. We will continue with our human centered view on information visualization and describe classifications for interaction with information visualization. Based on an appropriate classification for our purposes, we will describe the interaction with application examples. Thus interactive visualizations leads to solving tasks, the next chapter will introduce visual task classifications. We will find a common understanding on the way how tasks are classified in literature in contrast to interactions. Therefore an abstraction of the task classification will be performed. Based on this abstracted task classification, we will describe the task and classify them in order to have a more concrete understanding of visualization tasks. This will be important for our conceptual model. With this procedure we will have a view how human is involved in the visualization process and which tasks can be solved. Further it will be necessary to investigate the aspect of data in and for information visualization. We will continue with the same procedure and introduce classifications of data. Further we will slightly change an existing classification and introduce the data types based on this classification. The chapter will conclude with a section about technique and methods for visualizing information. This section will follow the same procedure and introduces first various existing classifications. Here again we will see that the proposed classifications are not appropriate for our purposes and will combine existing classifications to have a baseline for introducing the visualization techniques. The visualization techniques and methods will be introduced exemplary and do not claim to cover the state of the art. The main goal of this chapter that was partially published in [1, 2] is to have a common understanding about the terms, methods, and techniques of information visualization. Therefore we chose the view from human side, the tasks, and the data to describe information visualization. Figure 2.1 illustrates an abstract view on the structure of this chapter.

K. Nazemi, *Adaptive Semantics Visualization*, Studies in Computational Intelligence 646, DOI 10.1007/978-3-319-30816-6_2

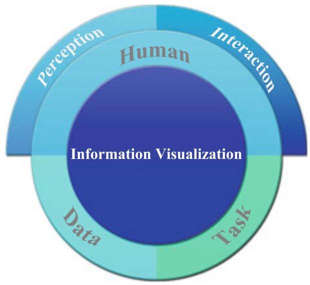

Fig. 2.1 Abstract view on the structure of the chapter information visualization

2.1 Terminological Distinction

The most common definition for information visualization in computational systems was brought by Card et al. [3]. They started with a more general definition of *visualization* in computational systems and defined visualization as *The use of computer-supported, interactive, visual representations of data to amplify cognition* [3, p. 6], whereas the *cognition* is further proposed as "acquisition or use of knowledge" [3, p. 6]. With this definition they worked out that the main goal of visualizations is to provide insights (*discovery, decision making, and explanation*) and not only pictures. Visualizations may represent different types of data. In case of visualizing physical data, Card et al. tends to the term *scientific visualization*. [3, p. 6] Based on the type of data to be visualized they define *information visualization* as:

> *The use of computer-supported, interactive, visual representations of* ***abstract*** *data to amplify cognition.* [3, p. 6]

The main difference in this definition is the term "abstract data", which is related to the fact that no obvious spatial mappings can be assigned to the data. Without a spatial abstraction, one challenge is the problem of rendering the data into an effective visual [3, p. 7]. To face the mapping problem of raw data to visual forms Card et al. proposed a reference model for visualization [3, p. 17], using outcomes of previous works on non-computational visualization of abstract data [4]. The proposed reference model for visualization counts today as the most influential reference model for information visualization. It provides a data transformation process from raw data to views involving the human in the interaction processing. The reference model is an excellent groundwork to understand, define and distinct information visualization. Figure 2.2 illustrates the reference model with its transformation steps.

The series of transformations begins with *raw data* and ends after three transformation steps with the human, who gains insights from the visual presentations. Vice versa the human is enabled to operate and thereby manipulate and adjust the transformation steps (user interaction on different level). The first step of transformation is *data transformation*, with the diverse raw data formats to relations or sets

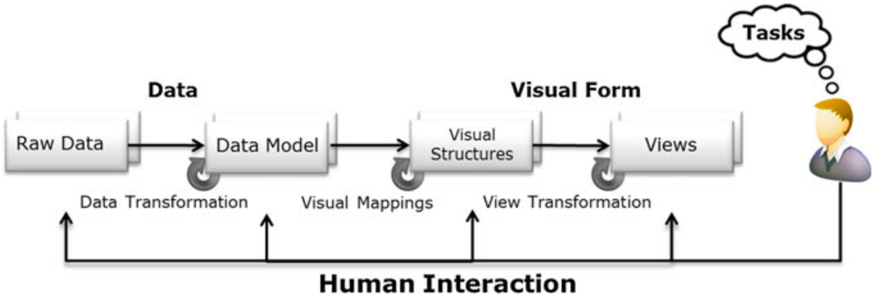

Fig. 2.2 Reference model for visualization (adapted from [3, p. 17] *with kind permission of B. Shneiderman*)

of relations (*data table*) that are structured and easier to visualize [3]. Card et al. define these relations mathematically as a set of tuples (see Eq. 2.1).

$$\left\{< Value_{ix}, Value_{iy}, \ldots >, < Value_{jx}, Value_{jy}, \ldots >, \ldots\right\} \tag{2.1}$$

A *Data Table* combines relations with their describing metadata [3]. A data table is represented by rows, which contains variables as set of values in the tables and cases as set values for each variable. In context of data tables they introduce a categorization of the data variables and their possible sequences. They propose that there are three basic types of variables, *nominal*, *ordinal*, and *quantitative*. Nominal variables are unordered sets (are only $=$ or $\neq$ to other values), ordinal variables are ordered sets (obeys a $<$ relation), and quantitative variables are numeric ranges (can do arithmetic on them) [3, pp. 17–23].

The next step in the transformation process of the reference model is the mapping of the data tables to *Visual Structures*. Here the work of Bertin [4] builds the foundation of visual variables and structures to provide an effective mapping [3, pp. 23–31]. The reference model proposes that two main factors are important to provide an effective mapping to visual structures. The mapping should preserve the data with their type of variables and emphasize the important information to be perceived well by the human. The visualization should enable the human to interpret faster, distinct graphical entities, or make to fewer errors [3, p. 23]. In today's evaluation methods the two main factors for measuring the efficiency of visualizations are *task completion time* (faster interpretation) and *task completion correctness* (fewer errors). The visual structures of the reference model are enhancements of Bertin's work on graphical semiology [3, 4]. While Bertin subdivided the visual variables into retinal variables and layout, the reference model does not propose such a differentiation [3, p. 26]. It enhances the model of Bertin and consists of spatial substrates, marks, and graphical properties. Although the authors propose that some visual encodings are more appropriate for *uncontrolled processing* (or *preattentive*) (see Sect. 2.2.1) in tasks like search or pattern detection and others for *controlled* *processing* (see Sect. 2.2.2) [3, p. 25] the reference model itself does not propagate this separation. It focuses more on a general transformation of data tables and their sequential characteristics to visual structures. Visual structures may appear as *Spatial Substrates*, *Marks*, *Connection and Enclosures*, *Retinal Properties*, and *Temporal Encodings*, whereas the transformation encloses the entire spectrum of visual structures.

The final step of the reference model completes the loop between human and visualizations (visual forms) [3, p. 31]. It transforms static graphical presentation by incorporating humans' interaction to create different views of visual structures and provide an interactive visual environment. Card et al. lists three main view manipulations: (1) Location probes use location to reveal additional information from data tables, (2) Viewpoint controls magnify or change the viewpoint, e.g. by zooming or panning, and (3) Distortion provides a modification of the visual structure by creating a context plus focus view [3, p. 31]. The view manipulation techniques will be investigated in more detail in Sect. 2.3.2. The introduced reference model describes in a comprehensible way the transformation processes from raw data to visual structures, the view manipulations, and human operations on different levels back to the transformation steps. These steps focus on the how abstract data can be visualized interactively with computational systems and provide a well-established explanation of information visualization.

In recent years, the research field of *Visual Analytics* evolved from Information Visualization and other areas to emphasize the knowledge generation aspect. Visual Analytics were often used synonymous to information visualization, although both terms gained established definitions. The early and most influential definition of Visual Analytics was proposed by Thomas and Cook [5]:

> *Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces.* [5, p. 4]

Their definition emphasizes the "overwhelming amounts of disparate, conflicting, and dynamic information" [5, p. 2] in particular for security related analysis tasks. One of the main focuses of Visual Analytics is to "detect the expected and discover the unexpected" [5, p. 4] from massive and ambiguous data. They outlined that the main areas of the interdisciplinary field of Visual Analytics are:

- *Analytical reasoning techniques*: for obtaining insights and support analytical tasks such as decision making.
- *Visual representations and interaction techniques*: for enabling users to explore and understand large amounts of data, and interact with them with their visual perception abilities.
- *Data representations and transformations*: to convert all types of data, even conflicting and dynamic, to support visualization and analysis.
- *Production, presentation and dissemination*: to provide a reporting ability for a broader audience and communicate the analysis results. [5, p. 4]

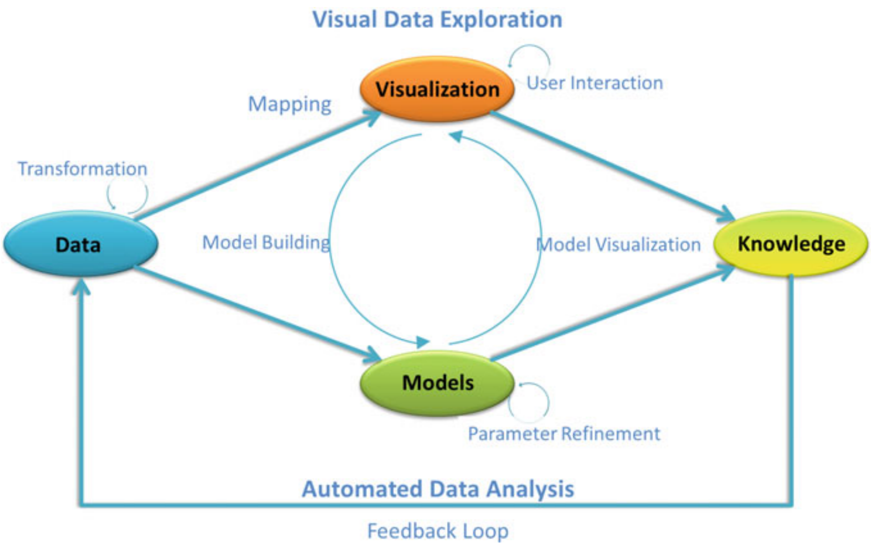

Fig. 2.3 The visual analytics process (adapted from [11, p. 10])

The definition of Visual Analytics gained a series of revisions to precise the abstract formulation [6–11]. Keim et al. commented that the definition of such an interdisciplinary field is not easy [11]. A more precise definition is:

> *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 datasets.* [11, p. 7]

This definition stated more precisely the interdisciplinary nature of Visual Analytics by introducing and outlining the combined use of analysis techniques and interactive information visualizations. In addition, it emphasizes the challenge of data amount, thus this confines Visual Analytics to "very large" data-sets. The main characteristics of solving analytical tasks with interactive information visualizations still remain. This definition of Visual Analytics is illustrated by a model for the Visual Analytics process. Figure 2.3 illustrates the process that targets on providing a tight coupling of visual and automatic analysis methods through human interaction to enable human to gain insights and knowledge [11, p. 10].

The visual analytics process models the different stages represented by oval forms and their transitions with arrows. The process starts with the data that may need to be preprocessed and transformed to an adequate way (indicated with the transformation arrow). After the transformation stage the "analyst" may choose to visualize the data or to use automatic analysis methods [11]. Keim et al. does not use the term "user" in their process. It may indicate that the Visual Analytic model is a dedicated design for "analysts" with the necessary of previous knowledge about the processes or tasks (analysis). If the automatic analysis is chosen, techniques from data mining are applied to generate models from the underlying data. These models can further be

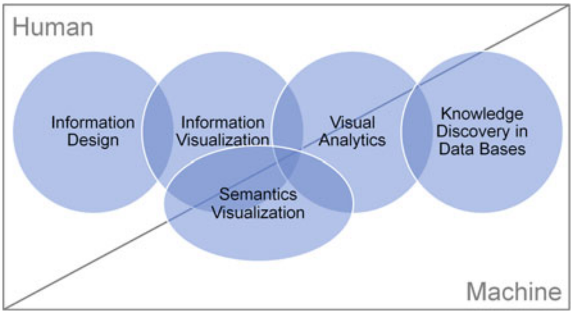

Fig. 2.4 The computer's and human' role in visualization in context of policy modeling (from [12, p. 85])

evaluated, refined, or specified by interacting with data [11]. Visualizations are used to interact with the models and manipulate and refine the parameters. Further the selection of alternating models can be visualized to evaluate the findings out of the generated model. If the analyst decide to visually explore the data first, the underlying model has to be confirmed based on this hypothesis. The visual representations reveal insights, which can further be refined by interactions on the visualizations [11]. The entire Visual Analytics process tightly couples the visualization and automatic data modeling (data analysis) methods. It provides an interactive process to make use of both, the interactive visual representations and data modeling approaches for acquiring knowledge and insights, which build the last stage of the process [11]. The role of human and the possibilities to interact in the stages of the visual analytics process remains as they are proposed in the reference model for visualization [3]. The main difference is the interactively combined techniques for visualizing and analyzing data.

Kohlhammer et al. proposed a differentiation of visualizations in context of policy modeling [12]. Their differentiation proposes a classification based on the role of human and machine in the data processing pipeline. Thus Visual Analytics make use of more automatic processing and modeling techniques than information visualization, the model distinguishes visual analytics based on the role of the involvement of automatic (computer-based) methods. Figure 2.4 illustrates the differentiation and introduces further the field of *Semantics Visualization*, which will be described more detailed in Chap. 3.

In this work we use the definition of information visualization as defined by Card et al. [3]. Thus Visual Analytics makes use of information visualization for the visual stages [11], we use the term information visualization for the visualization aspects of visual analytics too. When describing Visual Analytics systems, our focus will be the way how information is visualized and human are interacting with and perceiving the visualizations. Amplifying cognition and acquiring knowledge [3, 11] with the use of human's visual perception is essential for this work, whereas the automated data processing and data analytics methods are not in scope of this work.

2.2 Visual Perception and Processing

Visualization is strongly related to the way how human perceive and process visual information. Physiological and psychological studies showed that vision processing consist of two main stages of attention, preattentive and attentive processing. Understanding these stages is essential for the identification of those visual attributes and variables that should be considered for the visual representation of information. This section introduces the terms preattentive and attentive visual processing and summarizes some of the most common theories. It further builds the foundation for the adaptation of the visual attributes. Physiological aspects of human image and vision perception will not be discussed in this section. For further readings in physiological aspects of vision perception the work of Hubel [13] is recommended.

2.2.1 Preattentive Processing

The process how human perceives visualizations were investigated in research for several years [14]. A fundamental result was the discovery of a limited set of visual properties, which is rapidly detected by the low-level visual system [14]. The so called preattentive features are detected by human in less than 250 milliseconds, which suggests that certain information can be processed in parallel [13–15]. A unique visual property allows identifying an object preattentively. This unique visual property might be length, width, size, curvature, number, terminators, intersection, closure, hue, intensity, flicker, direction of motion, binocular luster, stereoscopic depth, 3D depth cues, and lighting direction. All this variables are associated with the four primitive variables luminance and brightness, color, shape and texture, [14] which provide processing of visual information prior to selection [16]. This visual stimulus is called ‘pop-out effect’, an uncontrolled movement of eyes to visual features. Ward et al. name four tasks, which uses the pop-out effect in psychological experiments for performing tasks [14]:

- **Target detection**: The task is to detect presence or absence of a target with unique visual features within a field of distractors.
- **Boundary detection**: Users have to detect a texture boundary between two groups of objects, where each group has common visual features.
- **Region tracking**: Users track one element as it moves in time and space.
- **Counting and estimation**: The task is to count or estimate the number of objects with different visual attributes [14].

Treisman’s ***Feature Integration Theory*** [17] has become one of the most influential theories in the area of preattentive visual information processing. It gives insights into the preattentive detection of boundary and targets in fields of distractors. To evidence the preattentive perception of visual features, she designed a set of tests. Therefore a target with a unique visual attribute (*target detection*) or a group

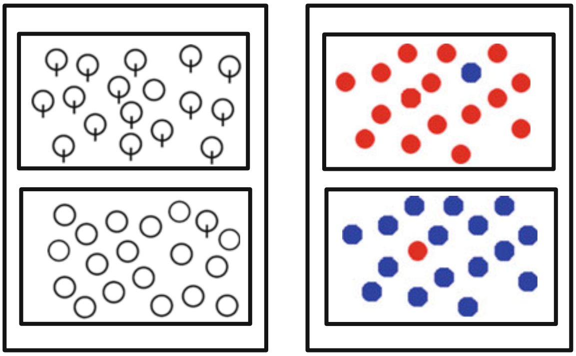

Fig. 2.5 Asymmetric and symmetric preattentive visual features (adapted from [14, 20])

of target elements with unique visual features (*boundary detection*) was placed in a field of distractors. The subjects had to communicate as fast as possible, if the target is absent or present, while the amount of distractors was increased. Treisman and other researchers tested the accuracy and the time of the responses. They assumed, if the visual information would be processed serially, the subjects would need more time, when the amount of the distractors increases. And if the amount of the distractors plays no role for the measured time and accuracy, the visual task was processed in parallel, according to that preattentively [17]. In a further test (*accuracy model*) a screen with a target or a group of targets was shown to the subjects just for 200 to 250 msec. In this time frame the subjects had no time to focus attentively to a certain object. So if they give the accurate answer to the presence or absence of the visual targets, the task was solved preattentively. Treisman and others used this test to identify a list of preattentive visual features [17–19]. Further they detected that some of the visual features are asymmetric, while others are symmetric. A circle with a line (as a visual feature) in a sea of circles can be processed preattentively, while a circle without a line in a sea with circle with lines is not preattentively processed [15, 17, 19]. Figure 2.5 illustrates the difference between symmetric and asymmetric visual features.

Treisman and Souther explained the phenomenon of preattentive visual processing using a model of low-level human vision made up of a feature map and a master map of locations [18]. They proposed to use a manageable set of features, consisting of the main visual attributes. The feature map therefore consisted of the visual variables, color, size, orientation, luminance and contrast. Whereas each of the features had their own map and for the color the four primaries red, green, blue and yellows [18] and the three primaries red, yellow, blue [19]. The feature map was expanded [20], in which the features luminance and contrast were replaced by *stereo distance*.

The *master map of locations* in their *theoretical framework* [19] is a medium in which the attention operates. This map "specifies *where* in the display things are, but not *what* they are" [19, p. 17]. With a unique visual feature or unique visual features compared to the distractors, a localization of the target or boundary is enabled with the master map of locations. The more an object differs from the distractors, the better it can be processed. A green square, for example, in a sea of red circles can be better recognized preattentively than a red square. This phenomenon shows that there exist differences between the preattentive processing of visual features. For that reason Treisman expanded her model in later works [19, 21], not only proposing a strict dichotomy of features being processed serial or parallel. These are more two ends of a spectrum [19–21].

Treisman proposed in her theory two main stages of visual perception, the preattentive and the focused attention stage. The preattentive stage is strongly related to one unique visual feature that stimulates a 'pop-out effect'. In this stage neither the target is localized, nor is it identified. One main finding of the *Feature Integration Theory* was that the localization of a target object is processed serially on the master map of locations. She evidenced that the presence and absence of a target object with unique visual features can be processed preattentively, but the identification and localization of the object on the master map of locations requires focused attention [17]. She evidenced her theory with the *illusionary conjunction*, where subjects identified not existing target objects in a sea of distractors with more than one unique visual feature.

The strict bisection of serial and parallel low-level visual processing based on the conjunction of visual attributes is not advocated by all researchers. Quinlan and Humphrey for instance propose that the search time for visually detecting objects depends on two other factors. Firstly on the number of items of information required to identify the target and secondly on how easily a target can be distinguished from the distractors, whereas unique visual features play no role on their '*Similarity Model*' [14, 22]. The model introduces the criteria *target to non-target similarity (T-N similarity)* and *non-target to non-target similarity (N-N similarity)*. Visual search time is based on *T-N similarity*, which defines the similarity between target and distractors and *N-N similarity*, the similarity between the distractors. The proposed model assumes that as *T-N similarity* increases, the search time decreases. Further as *N-N similarity* decreases, the search time increases and the search efficiency decreases. *T-N* and *N-N similarity* are related and comprehend each other. If *T-N* decreases and *N-N* similarity decreases too, a preattentive perception of the target will not be registered. If both similarities increase, the effect of a preattentive perception will get lower [14, 22]. The *Similarity Theory* preaches that the more an object distinguishes from the distractors and the more the distractors are similar, the better and faster it can be perceived, regardless of any unique visual attributes.

A more recent model of a two stage paradigm of preattentive and attentive visual perception was proposed by Wolfe in his ***Guided Search*** model [23–26]. In his first attempt his Guided Search model had a preattentive and an attentive stage, based on Treisman's Feature Integration Theory and explaining more the Similarity Theory. He further proposed that the information from the first stage could be used to guide the attention to the attentive stage [23]. An object with unique visual variables would lead in the preattentive stage the focus of the subject to the visual object, this attention is further present in the second attentive stage. The future versions of the Guided Search model, including the recent version *Guided Search 4.0* [26], proposed more smooth transition between the two strictly bisected stages of attention. One main finding of Wolfe was that the preattentive visual activation is not only stimulus-driven (bottom-up), like proposed in the Feature Integration Theory, but also user-driven (top-down). The Guided Search model argues the differentiation with its *feature maps*. Stimuli are assumed to be in parallel across the entire visual field. "At some point, independent parallel representations are generated for a limited set of basic visual features" [24, p. 204]. These sets of limited visual features are feature maps. The feature maps or independent maps for each visual attribute, e.g. color, size and orientation. Each of these maps may contain further maps, e.g. the color map may contain a map for green, red etc. Wolfe listed in his second model a set of visual features containing *orientation, color, motion, size, stereoscopic depth, other depth cues, binocular lusture, vernier offset, curvature, terminators* and *intersection*. [24] In case of localizing a target object, the feature maps are activated. And this activation can be either bottom-up or top-down [24].

The bottom-up activation is stimulus driven and thereby not depended to the subject's knowledge or preferences in a visual task. This activation is based on the differences between a target object and the neighboring distractors. The neighborhood of the target can be bounded in a 5×5 array around the identified object. Guided Search assumes that the bottom-up activation is calculated separately for each feature in the feature map. The bottom-up activation guides attention to a distinctive item in a field of distractors, if the visual features of the object are unusual. In contrast to the bottom-up activation, the top-down activation is user-driven and depends strongly to the task, knowledge and preferences of the user. [26] For instance, if a red circle is placed in a field of distractors of mixed color circles, the bottom-up activation will not be registered. But if the user is instructed to search for a red circle within the field of the heterogeneous distractors, the knowledge of the task will guide him to the red circle. This user-driven activation can be registered in a similar time-frame to the stimulus-driven activation [24].

Wolfe proposes that the strict dichotomy of parallel and serial visual processing does not hold. [24, 26] The Guided Search model assumes that the information from the first preattentive stage is forwarded to the second stage. The direct attention is guided through the preattentive processing, whereas the region of the target object is in the attentive processing further the region of interest. Wolfe evidences his model with triple conjunction of color, size and form (orientation) [23, p. 430]. The fact that three visual attributes are forwarded to the serial process leads to faster search process and reaction time.

The active involvement of users and the consideration of their pre-knowledge, preferences and tasks play an important role in the Guided Search model. If a user has an imagination of the searched target object, the reacting time decreases. The involvement of users' pre-knowledge played more and more a key-role in further works of computational modeling of visual attention in both stages. The challenge for the implementation of a comprehensive computational model of visual attention is the consideration of both activation types, [27] and consequently the involvement of users' pre-knowledge and visual tasks.

2.2.2 Attentive Processing

Human are able to detect certain visual features in parallel and thereby preattentively. The preattentive processing of information depends on visual features of targets and distractors. The 'pop-out effect' in this stage guides the attention of human to certain visual features, whereas this guidance is in most cases uncontrolled (see top-down and bottom-up activation in Sect. 2.2.1). The 'pop-out effect' does not include the localization or the target detection.

The attentive processing of visual information (or *postattentive vision* [14], *directed attention* [26]) begins, when we stop attending to the out-popped target (assuming there exists one) and look at something else [14]. Although, the strict dichotomy of parallel and serial processing is still disputed, Ware has proposed a three stage model, subdividing the attentive processing of visual information into a serial stage of *Pattern Recognition* and a further stage of *Sequential Goal-Directed Processing*, beside a preattentive stage [28].

Ware's model of perceptual processing is a simplification of several methods and models. The first of his three staged model is the preattentive stage, based on the proposed models of Treisman and Wolfe. Here information is processed in *parallel to extract low-level properties of the visual scene* [28, p. 20]. Similar to the described models, the parallel information processing cannot be consciously controlled by the user, is rapid and extracts basic visual features. The visual features that are investigated in this model are *orientation, color, texture*, and *movement patterns*. Based on the original Feature Integration Theory the parallel activation is bottom-up. Instead of using termini like stimulus-driven or feature-driven bottom-up activation, Ware introduces a *data-driven model of processing*. At the second *Pattern Recognition* stage of his model, rapid but active processes divide the visual field into regions and simple patterns. In this serial stage, regions and localizations can be identified, e.g. regions of similar or same colors. The flexibility of this stage can be influenced by both, the bottom-up activation from the previous parallel stage and the top-down activation. The top-down activation is driven by visual queries in this model. The visual queries are analog to Treisman's *feature maps*. Ware characterizes the second pattern perception stage including slow serial processing, with more emphasis on arbitrary aspects of symbols and the fluent combination of the bottom-up and top-down feature processing [28, p. 22].

The last stage of the three-stage model, the *Sequential Goal-Directed Processing*, is the highest level of perception involving active attention. The use of external visualizations let us "construct a sequence of visual queries that are answered through visual search strategies" [28, p. 22]. At this stage only few objects are in focus of attention, which are constructed by the subject from available patterns to solve a given visual query task. One main aspect in this stage is the use of the term *construct* that leads to the assumption that knowledge from the long-term memory (pre-knowledge) is associated to the visual patterns and new knowledge is constructed by human. In the context of knowledge construction it is necessary to introduce two terms that are essential for gathering knowledge through visualization, namely ***recall*** and ***recognition***. Ware proposes that *recall* "consists of the activation of particular pathway" [29, p. 388] of associations stored in the long-term memory. Recall makes use of visual or verbal-propositional information to activate the traces of the long-term memory. It is necessary to describe (verbal or visual) some patterns and traces of our memory without the use of an indicator. Ware constitutes that *recognition* is superior to *recall*, thus in *recognition* a visual memory trace is reawakened [29, p. 388].

This phenomenon is one main reason, why visual system should consider in their design the knowledge of users. With the use of *recognition* instead of *recall* the efficiency of the problem-solving process in visualizations can be improved.

The aspect of post-attentional processing in terms of dynamic generation of visual representation was investigated by Rensink in his ***Triadic Architecture*** [30]. He argues based on the *Coherence Theory* [30, p. 19] that focused attention is needed to see changes at the time they occur and only one object in a scene (screen) can be given a coherent representation [30]. Moreover, the representation is limited in the amount of displayed information. So it is necessary to shift the attention to the appropriate objects at the right time. He discards the assumption that all visual processing pass a single attention locus (*attento-centric*) and proposes a triadic architecture with independent information processing systems. The first system, the *low-level vision*, makes use of the preattentive features to shift the attention to the location of interest. This level creates a high-detailed, volatile structure [30, p. 34]. In this system of early processing the resultant structures *(proto-objects)* may be sophisticated, the spatial coherence is limited and simply replaceable by new stimuli [31, p. 262].

The second system, the *Object (attentional)*, investigates the spatial arrangement (*Layout*) in the scene and activates a focused attention [31]. This provides a non-volatile representation of the locations of various structures on limited-capacity attentional system. This is used when attention is already directed. The third system, the *setting (nonattentional)* facilitates the perception via *gist* (meaning) and *layout*. Rensink proposed that "the most abstract aspect of a scene is its meaning" (gist) [30, p. 36]. It is a result from the context of an object and is used to refer to the properties of the long-term memory to *recognize* an image. The most important aspect in this context is the unification of *Layout* in terms of spatial arrangement of objects. Rensink proposes that one important aspect of the scene structure is *Layout*, "without regards to visual properties or semantic identity" [30, p. 36]. *Layout* is used to support the problem solving process as knowledge about the relationships of coherent-objects is needed [29]. The associated collection of representations is *scene schema*. Rensink proposes that gist and layout involve short-term or working representations, whereas the scene schema is long-term structures [30].

2.3 Visual Interaction

Today's information visualization systems do not just offer a static picture. Most of the existing visualizations provide different interaction techniques that allow solving the given visualization task through graphical interaction. The provided interaction method is one of the key-features of the visualization system and the issue of interacting with visualization was already investigated by various researchers. Several classifications, concepts and techniques were introduced to affirm the importance of interaction in visualizations. This section gives an overview about some of the classifications interactive visualizations. The section does not claim to be complete and aims to give an overview of the idea of *interaction* in information visualization systems.

2.3.1 Classifications of Visual Interactions

An abstract classification of interaction in visualizations was brought by Ware, who proposes a classification of interlocking feedback loops of *data manipulation*, *exploration and navigation*, and *problem-solving* [28]. At the lowest level, the *data manipulation loop*, objects are selected and moved using the basic skill of eye-hand coordination. In this loop the system and human reaction delay is an important factor for efficient interaction with visualizations. Ware introduces several measurements criteria and rules for measuring reaction time, e.g. reaction time (*Hick-Hyman law* [32]), selection time (*Fitts' law* [33]), and path tracing [34]. The selection and reaction time in the lowest level of interaction is constraint to the users' knowledge in interacting with the systems. Over time, people become more skilled in operating low-level interactions with visualizations. The informal learning of interaction with systems is introduced by Ware with the simple expression, known as *power law of practice*. This law describes the users' learning curve as:

$$\log(T_n) = C - \alpha \log(n) \tag{2.2}$$

where $C = \log(T_1)$ is the first performance of the user with a system, T_n is the time required to perform the nth trail, and α is a constant that represent the learning curve.

At the intermediate loop of *exploration and manipulation*, the way through large visual data space is found. In this interaction level the known similarities are "recognized" to find the way through the data. The differentiation between *knowledge types*, e.g. declarative, procedural, and topological knowledge [35] plays an essential role to find the path to the targeted knowledge and build a *cognitive spatial map* [28]. The highest level of the model the *problem-solving loop* provides the ability to form hypotheses. The augmented visualization process provides refinements and reformulations until a possible solution is identified. The iterative character of this level can further be enhanced by replacing and revising visualizations.

A classification of visual interaction methods is the *Visual Information Seeking Mantra* proposed by Shneiderman [36]. His mantra is not explicitly declared as a classification for interaction methods. It is far more a starting point for designing advanced graphical user interfaces and the foundation of the *Task by Data Type Taxonomy* of information visualization [36]. Shneiderman proposes in his mantra *overview first, zoom and filter, then details on demand*. The interactions on visual environments are ordered sequentially and have an iterative character. This classification is according to Ware's model on the highest *problem solving loop* of visual interactions. The mantra further correlates seven data types to seven tasks on the highest level of abstraction.

Cockburn et al. enhanced the interaction aspects of the *Visual Information Seeking Mantra* to survey and categorize visualizations [37]. They defined "overview plus context" as *Spatial Separation* between focused information entities and contextual information. The "zooming" interaction was reduced to the temporal separation of entities, whereas "focus plus context" minimizes the seam within the contextual information. Further proposed "cue-based" techniques selectively highlight information within the information context [37].

Keim enhanced and refined the *Visual Information Seeking Mantra* too and introduced the following interaction classification of information visualization [36, 38, 39]: *1. Data-to-Visualization Mapping, 2. Overview, 3. Zoom, 4. Filter, 5. Details on Demand, 6. Relate, 7. History, 8. Extract,* and *9. Linking & Brushing*. In this model the *interaction techniques* (Mapping, Projection, Filtering Link&Brush and Zooming) [38, p. 81] were categorized to *distortion techniques* and *data visualization techniques*. In this enhanced model *distortion* is categorized as an interaction technique. Further the *standard* interaction technique is introduced to conclude the whole spectrum of possible interaction techniques.

Hearst lists the following main techniques for interacting and navigating with information visualization within abstract data space: *brushing and linking*, *panning and zooming*, *focus-plus-context*, *magic lenses* and *animation* to retain the context [40, p. 260]. Further the combination of interactions as for example *overview plus detail* are proclaimed for solving tasks with interaction methods. The techniques are seen as foundations for the design and implementation of visualization techniques. In contrast to Keim's classification, Hearst proclaims a more *context-oriented* interaction. *Zooming* is mentioned in combination with *panning*, where the panning-action enables to view the overview-context before zooming into the visual area of interest. A similar procedure is proclaimed for all the interaction techniques, classified by Hearst. In the classification of Ware the identified interaction techniques can be positioned vertical to the whole spectrum of the interaction-loops. The main target of the interaction in this classification is problem-solving, whereas the context-orientation is addicted to the exploration and manipulation loop as well as to the data manipulation loop. Data entities, relations and attributes are visualized dynamically through the visualization techniques that access the data directly.

Ward and Yang introduced in [41] a classification of interaction in visualizations that distinguishes between *interaction operators* and *interaction operands*. They proposed that there is a significant difference if an interaction is *operated* to different objects or spaces. These objects or spaces are the operands of the interaction procedure. "To determine the result of an interactive operation, one needs to know within what space the interaction is to take place" [41, p. 2]. In their first attempt they classified three *interaction operations* and thereby *interaction operators*. With *navigation*, *selection* and *distortion* a significant percentage of the interaction operations in visualization systems was identifies [41, p. 1]. The interaction operands were classified in section of spaces upon which an interactive operator is applied. Their proposed framework contained following spaces as *operands*: *screen-space (pixels)*, *data value-space*, *data structure-space*, *attribute-space*, *object-space (3D surfaces)* and *visualization structure-space*.

The *screen-space* consists of actions directly on the screen with no impacts on the data. This contains transformation on screen-level, such as *panning*, *zooming*, *rotation* or *pixel-level* operation, e.g. *transformation*, *sampling* or *replication*. Interactions on the *data-value space* involve the data values for view specification. On this space *panning and zooming* or other interaction operations change the data values being displayed. An interaction operation on this space is similar to a database query for specifying data values. Interaction and navigation operations on the *data structure-space* involve view transformation along the structure of the data. Operation on this space allows identifying regions of interest in the data structure, e.g. selection of data in a cluster hierarchy [42]. Operations on the *attribute-space* are similar to that on *data value-space*; they involve a view transformation based on the attributes of the graphical objects. Whereas interactions on the *object-space* are defined as direct manipulation of graphical object, primarily 3D-objects, which can be turned transformed etc. Interaction operations on the *visualization structure-space* involve the view transformation of the visualized structure. The data are not manipulated on this level, but the user is able to rearrange the visual structure.

The classification of Ward and Yang is part of their unified framework for interactions on visualizations. This framework further proposed the parameterization of the operands [41, pp. 6–8] to define an extensive assortment of interaction operators. Ward et al. extended their framework in their recent work [14, pp. 315–354]. In this extended classification they enhanced the interaction operators by *filtering*, *reconfiguring*, *encoding*, *connection* and *abstraction/elaboration*. Further the *distortion* operation is not considered as a class of interaction operations. The interaction operands and the parameterization remain in the new version of their framework and classification.

This section gave an overview of some classifications of interaction methods in visualization systems. We presented heterogeneous classifications that investigated visual interaction in different levels of abstractions. The introduced classifications were rarely published as interaction classifications; they are rather evolved from design guidelines for visualizations or from classifications of visualizations. Nevertheless we could conclude that interactions in visual environments are investigated in various abstraction levels. Interactions transform the view on the visualized data by manipulating the data or the visual representation. The manipulation on both, data and visualization can be further classified as the model of Ware showed.

2.3.2 Visual Interaction Techniques

This section introduces the most common interaction techniques based on the classification of Hearst [40]. This classification describes the interaction techniques at a lower level of abstraction similar to Keim's classification. It further considers the context of users' interactions and is therefore adequate to explain the interaction techniques in the context of this thesis, which will further describe semantics visualizations. We added the interaction techniques *semantics zoom*, *dynamic queries* and *direct manipulation* to the model of Hearst for covering a wider range of possible interaction techniques.

Brushing and Linking
In multiple-visualization user interfaces, different visual representations of the same data give a view on various perspectives to the data. To not lose the visual context, "brushing and linking" provides a highlighting or selection of visual objects between different views [40]. The highlighting may occur in various ways, e.g. by changing the color or size of the *brushed* objects. The main target is to provide a visual differentiation to the displayed objects and distractors. The work on preattentive perception described in Sect. 2.2.1 provides important visual features to perform this interaction and provide visual features to distinguish brushed objects in linked visualizations.

Panning and Zooming
Panning and zooming provides the change of the viewpoint to the visualized data [40]. Card et al. use the term "panning and zooming" in their listing of interaction techniques as an equivalent to *camera movement and zooms* [3]. Panning and zooming targets to refine the visual area of interest by moving the screen or the view on the screen (pan) and zooming into the area of interest. Furnas and Bederson introduced an analytical framework by *space-scale diagrams* for a direct visualization and analysis of important scale relates issues. [43] They represented the panning and zooming interaction as space-scale diagrams by trajectories [43].

Focus plus Context
Zooming leads to the problem of getting more details about the zoomed part and losing the surrounding information. The higher the zoom-factor is, the more details can be shown about particular items, but the overall structure and the information context get lost with the increasing zoom-factor. To face this problem the interaction metaphor *focus plus context* technique offers more details in the zoomed part but keeps the context in a lower level of detail [3, p. 307].

One of the earliest techniques for focus plus context is the *fisheye view* [44, 45]. The model of "Degree of Interest" [46] was the pioneering foundation for the work on the fisheye views. In contrast to zoom, which is a transformation on the view level, the focus plus context interaction is categorized as transformation on data-level [3].

Beside the distortion techniques (fisheye views), Card et al. list *filtering*, *selective aggregation*, *micro-macro readings* and *highlighting* as selective information reduction methods for keeping the contextual area [3].

Semantic Zooming

Traditional zooming techniques operate on the visual level of a graphical data representation and manipulate the view. According to Ware's model [28] the zooming interaction occurs on the *problem solving loop*, e.g. by changing the size of a zoomed object. In contrast to the ordinary zoom, semantic zoom uncovers detailed information to encompass the context and meaning of a zoomed target [47]. Semantic zoom is for example recently used in ontology visualization for reducing the complexity of large ontologies [48].

Animation

Interactions in information visualization manipulates the visual representation of the underlying data on different levels [28], e.g. by data transformations, visual mappings, and view transformations. In these types of interactions users acts directly with visualizations and change the view. In contrast to the manipulating interaction techniques, animation does not provide the user with manipulation functionalities. The animation is more an implication of the users' interaction [49]. The literature in visual perception suggests the use of animation for the improvement of interaction and understanding [50, 51]. Attracting attention, perceiving in peripheral vision and comprehending the visual changes are the most argued reasons for the implementation of animation as consequences of interactions [50–52]. Further the continuous changes of visual parameters in visualization can be easily followed and understood [52].

Overview plus Detail

Keeping the information context, while gathering detailed information about a specific area of interest, is the main goal of *overview plus detail*. It displays the information with different levels of details in two or more linked visualizations. Card et al. differentiate between time- and space-multiplexed overview plus detail [3]. Time-multiplexed overview plus detail is conceptually similar to panning and zooming or just zooming, thus the interaction is processed serially. Here the main attribute of overview plus detail can be named as the fact, that the time-dependent serial interaction steps provide two main views, an overview of the information and a detailed view of the area or objects of interest. In contrast to that, space-multiplexed overview plus detail conveys both information detail-levels at the same time, in two separated areas of the display (views). This is most common way, how overview plus detail is used in visualization systems [3]. In contrast to panning and zooming, semantic zoom or focus plus context, the detailed and overview information are visualized at the same time in mostly separated display or display areas (space-multiplexed).

Dynamic Queries
The access of information in a human-understandable way is one main goal of information visualization. Today's information databases contain huge amount of data. The visualization of the whole data-set on a single visualization is often overcharging the human perception and information acquisition abilities. Dynamic queries can help to access the required information interactively, which can satisfy users' heterogeneous needs of information acquisition [53, 54]. Dynamic queries provide a well-known and successful approach for exploring [55] and visually seeking vast amount of data [53]. Visual interactive query formulations and refinements enable the reduction of the visualized information to a comprehensible and relevant scale [55].

Direct Manipulation
All the introduced visual interaction methods manipulate the visualization, either on a data-transformation level or on the level of visual transformation. Direct manipulation provides a direct interaction with the user interface or visualization without the need of commands. It bridges the gap between human and machine with a more intuitive graphical metaphor of interaction and avoids the barrier to translate ideas into commands [56]. Golbeck introduces the idea of direct manipulation with the example of deleting items through the trash [57]. The physical selection of an item and putting it into the trash is more intuitive an obvious interaction as a command-line expression for the same action Shneiderman defined various criteria such as visibility of objects and actions of interest; rapid, reversible, incremental actions; and replacement of command-language syntax for direct manipulation [56, 58]. In information visualization, where the data are represented with graphical metaphors or representatives, direct manipulation is essential for a natural interaction. Each interaction, which substitutes a command line expression, can be defined as direct manipulation.

2.4 Visualization Tasks

Interaction with visualizations enables the dialog between user and the visual representation of the underlying data. The interactive manipulation of the data, the visual structure or the visual representations provides the ability to solve various tasks and uncover insights. The term "task" in the context of information visualization is often used ambiguously. Often, interactions and tasks are not distinguished for visualization design, whereas the knowledge about the task to be solved with the visualization is of great importance for its design and thereby for the adaptation. This section starts with the introduction of taxonomies and classifications of tasks in visualization systems. The classifications will enlighten the heterogeneous view on visualization tasks and enable getting an overview of the differences. The classifications will enable to

Table 2.1 Task classification by Shneiderman [36, p. 337]

Task	Description
Overview	Gain an overview of the entire collection
Zoom	Zoom in on items of interest
Details-on-demand	Select an item or group and get details when needed
Relate	View relationships among items
History	Keep a history of actions to support undo, replay, and progressive refinement
Extract	Allow extraction of sub-collections and of the query parameters

investigate high-level tasks in more detail. These high-level tasks will be introduced in the second part of this section and conclude the section.

2.4.1 Classifications of Visual Tasks

One classification of tasks in visualization is the already mentioned *Task by Data Type Taxonomy* of Shneiderman (see Sect. 2.3) [36]. With the assumption that users are viewing collections of data with multiple attributes, he proposes that a basic search task is the selection of items that satisfies the search intents. This classification enhances Shneiderman's *Visual Information Seeking Mantra* with the tasks *relate*, *history*, and *extract*. Table 2.1 illustrates the seven tasks.

The overall tasks in this classification can be abstracted to the high-level tasks *exploration* and *search* and leads to finding (relevant) information.

Buja et al. proposed in their early work [59] a classification concept that investigates the interaction with visualizations (*view manipulation*) and the tasks that are supported by these interactions. They supposed that the purpose of the view manipulations is to support the search for structures in data [59]. For this search they identified three fundamental tasks for data exploration, namely *finding gestalt*, *posing queries*, and *making comparisons*. Finding certain patterns of interest, e.g. clusters, discreteness or discontinuities, are classified in the task *finding gestalt*. *Posing queries* is the next step after gestalt features of interest were found and further information are desired to get an comprehensible view on the chosen parts of the data. For the task *making comparisons* they distinguish between two types of comparisons. First the comparison of variables or projections and second the comparison of subsets of data. The comparison of variables enables the "view from different sides" [45, 59], which illustrates the data from different perspectives, whereas the data subset comparison provide a "view of different slices" and thereby of different subset of data [59].

Further they proposed that the identified tasks are optimally related to three manipulation views. For *gestalt finding* they identified the *focusing individual views*. Here

Table 2.2 Task classification by Buja et al. [59, p. 80]

Task	Manipulation view	Interaction
Finding gestalt	*Focusing individual views*	Choice of projection, aspect ratio, zoom, pan, order, scale, scale-aspect ratio, animation, and 3-D rotation
Posing queries	*Linking multiple views*	Brushing as conditioning/sectioning, database query
Making comparisons	*Arranging many views*	Arranging scatter plot matrix and conditional plot

focusing provide any operation of that manipulates the subset of data or view. The choice of projection, for viewing or the choice of aspect, ratio, and zoom are examples of focusing. For *posing queries* they identified *linking multiple views*. The linking contains view manipulation as brushing or query issuing by highlighting. *Making comparisons* is related to *arranging many views*. They propose that the arrangement of large numbers of related plots for simultaneous comparison is a powerful informal technique [59].

With this tasks and manipulation views they further propose a set of low-level interaction techniques that are related to each high level task. Table 2.2 provides an overview of the proposed task, manipulation views and interactions that are related to each other.

Another approach, which correlates low-level interactions with visualization tasks, was proposed by Chuah and Roth [60]. They summarized their "basic visualization interactions" as a set of low-level-interactions with the attributes input, output, and operation and abstracted them to three basic visualization tasks [60, p. 31]. At the lowest level they propose "Data Operations", which contains interactions affecting the elements within visualizations, e.g. add, delete or derive attributes. The higher level considers "Set Operations", which refers to operations on sets, which may have group characteristics. The highest level investigates "Graphical Operations", which are divided into encode-data, set-graphical-value, and manipulate-objects. While the classes encode-data and set-graphical-value change graphical attributes or the mapping between graphical objects and data, the class manipulate objects operates on graphical objects as a unit of manipulation [60, p. 33–36]. The investigated tasks in this classification focus on comparison and finding patterns as graphs or in data. The high-level task of this classification can be abstracted as "analysis". The aspect of analysis was investigated in various works. One early example is the classification of Wehrend and Lewis [61]. They proposed a taxonomy with ten analytical tasks: *location*, *identity*, *distinguish*, *categorize*, *cluster*, *distribution*, *rank*, *compare within entities*, *associate*, and *correlate*. Zhou and Feiner [62] proposed an approach by considering not only the interaction and manipulation abilities of visualizations. They

Table 2.3 Visual task classification by Keller and Keller (adapted from [14, p. 380])

Task	Description
Identify	Recognition of objects based on the presented characteristics
Locate	Identification of the position of an object
Distinguish	Determination the difference of objects
Categorize	Classification of objects into distinct types
Cluster	Grouping of objects based on similarities
Rank	Ordering objects by intended relevance
Compare	Examination of similarities and differences of objects
Associate	Drawing relationships between two or more objects
Correlate	Finding causal or reciprocal relationships between objects

investigated the human perception and the intended task of the visual presentation method in their classification to provide a more user centered task-classification.

Based on various existing classifications, they characterized visual tasks along two dimensions. In the dimension *Visual Accomplishments* the focus lies on the intention of the visual presentation [62, p. 394]. They assumed that a presentation intends either to convey the presenter's message or to help user solving a perceptual task. Based on this assumption, visual tasks are distinguished at the highest level between tasks that *inform* users by *elaborating* or *summarizing* and those, which *enables* users to perform a visual *exploration* or *computation*. Their second dimension *Visual Implications* considers research outcomes of the human visual perception. Based on these outcomes they summarize three types of visual perception and cognition principles: (1) the *visual organization* principle investigates how people organize and perceive a visual presentation, (2) the *visual signaling* principle investigates the manner how people interpret visual cues and infer meanings and (3) the *visual transformation* principle explains how people perceive visual cues and switch their attention. This incorporates the outcomes of the preattentive visual perception too (see Sect. 2.2.1). Zhou and Feiner use these principles to infer visual tasks and assign them to the first dimension of *Visual Accomplishments*.

A more user-centered approach for classifying task was proposed by Keller and Keller [63]. Their classification considers the goals and intentions of the users and suggest based on these certain visual representations [14, p. 164 and p. 380]. They classify the user-intended tasks in nine task categories (see Table 2.3). The main characteristic of their classification is that only analytical aspects play a role for users interacting with visualizations. Previous general tasks like exploration or search does not play any role.

Table 2.4 Visual task categorization by Yi et al. (adapted from [64, p. 1226])

Category	Description
Select	Mark something as interesting to enable the following of the object
Explore	Show something else e.g., different subsets of data
Reconfigure	Provide a different view or arrangement of the underlying data
Encode	Provide a different fundamental view by selecting another visualization technique
Abstract/elaborate	Provide a different level of detail on the data e.g., by details-on-demand techniques (see Sect. 2.3.2)
Filter	Provide a view with certain (predefined) criteria
Connect	Provide a visual connection (e.g. by brushing) between the same objects on different views

A comprehensive classification of users tasks based on user intentions and the interaction role in information visualization was provided by Yi et al. in [64]. Their classification attempts to abstract the most used interaction possibilities with users' intentions to provide categories of interaction. They classify the user tasks based on the role of interaction in information visualization in seven categories (see Table 2.4).

Although the identified categories are abstract views on the interaction roles, the level of abstraction differs enormously. The category *select* for example, can be defined as simple and low-level interaction. Here a user marks an object of interest to be able to follow this object in changed views [64]. In contrast to *select* the category *explore* provide a real abstraction of interaction to a user task. Here the user is able to view on various subset of data to see different characteristics and perform a various number of low-level task e.g., comparing subsets or identifying relevant objects.

Pike et al. extends the proposed approach of Yi et al. [64] by differentiating between low-level and high-level interactions intending to meet high- and low-level user tasks and goals and propose a mutual feedback between user goals and tasks and the affordance of interactive visualizations [65]. They define seven categories of high-level tasks, which can be achieved by a number of low-level tasks and interactions respectively. Further they relate the representation and interaction intents of interactive visualizations, similar to the proposed classification of Zhou and Feiner [62] to low-level representation and interaction techniques. The proposed approach relates the classifications of user goals and tasks with the abilities and goals of interactive visualization in a "mutual feedback" [65]. The relationship of the proposed techniques and the user's goals and tasks is the "analytical discourse", which investigates the low-level interaction and user goals to form a feedback between them [65, p. 265].

The classification of Pike et al. considers the interaction value and user's goal and tasks from both perspectives, information visualization and Visual Analytics and gives a good overview of the high-level tasks intended by users and provided by interactive visualizations. Nevertheless, the differentiation of high- and low-level tasks is not clearly defined. A "compare" task could be a part and therefore a low-level task of "assess" or "analyze", while important tasks like "decision making" [66, 67] are not considered at all.

Fluit et al. proposed in [68] a very simple classification of visualization tasks in the special domain of ontology visualizations in the categories *Analysis*, *Query*, and *Navigation*. Therefore they define the *Analysis* task for getting a global view on data, the *Query* task for finding a narrow set of items, and the *Navigation* task for graphically navigating through the data [68]. In their revised work [69] the last category *Navigation* was replaced by *Exploration*. They propose that *Analysis* can be performed within a single domain with various perspectives, in various sets of data, and by monitoring the changes of data over time. The category *Query* is divided into the processes of query formulation, initiation of actions, and review of results. The task category *Exploration* is defined as finding information that are loosely of interest for the users [69]. Here a further subdivision is not proposed.

2.4.2 High-Level Visual Tasks

The previous section could work out that visual tasks are defined and classified in various levels of granularity. The described approaches are mostly using similar tasks or interactions to describe the problem solving process in visual interfaces. This section targets on the identification of high-level visual tasks as foundation for the visualization design and the adaptation process. We define, in this work, high-level tasks as tasks that are a summarization of visual tasks and provide a higher level of abstraction.

The described classifications consider in various ways the aspect of *search*. Shneiderman's *Visual Information Seeking Mantra* proposes a top-down information seeking approach [36]. Buja et al. propose searching information as the main task, which can be solved by manipulating the view [59]. Zhouh and Feiner investigate the way from information to user [62]. They elaborated the enabling and informing users. Informing users by elaboration and summarization, premises the information searching task. They further propose that *search* is a sub-task of *exploration* in their *enabling* category [62]. Fluit et al. propose in their simplified task classification *search* as one of the three high-level tasks and call it *query* [68, 69]. As the most of the presented classifications consider *search* as an important and fundamental task in visualizations, *search* is considered in this thesis as one of the high-level visual tasks. Beside *search* the high-level visual task *explore* plays an essential role. The task classifications show that exploring information plays a key-role for each classification. Shneiderman's model proposes a top-down seeking model with the characteristics of exploration [36]. From overview to detailed information can be assigned as an exploration task [36]. Zhou and Feiner explicitly name the task *explore* is a higher-level task of search and verify [62]. Yi et al. have their own categorization for *explore*, although their classification is not considering the high-level tasks [64]. The classification of Pike et al. [65] assigns the task *explore* as the high-level task on the user-goal and tasks level. In particular the classification of Keller and Keller [63] proposes a different view on solving visual task. They propose that the main visual interactions are solving more analytical task (see Table 2.3. Their task classification can be abstracted to *analyze*. Zhouh and Feiner differentiate in their model different aspects of *analyze*. In particular, the task category enable and verify leads to the higher level task *analyze*, whereas some aspects of inform and summarize are addressing the analysis task [62]. Pike et al. identified the task *analyze* already as a high-level task in their model [65]. Further they assigned the task "compare" as a high-level task too, whereas other works (e.g., [63]) assigns compare as a sub-task of the analytical visual problem solving process. As the analytical tasks plays an important role in all presented classification, *analyze* should be assigned as a high-level visual task.

On balance, the classification of Fluit et al. seems to be a well elaborated high-level task definition, whereas the definition of each task cannot be intermeatable with other definitions. In this work we investigate *search*, *explore* and *analyze* as high-level tasks. Further we assign the identified tasks as sub-tasks of the identified three high-level tasks. This is to ensure that the major tasks are categorized to the high-level tasks. This classification and task definition will be applied in this work. In Fig. 2.6 the classification and the assignment of the lower-level tasks are illustrated as applied in this work.

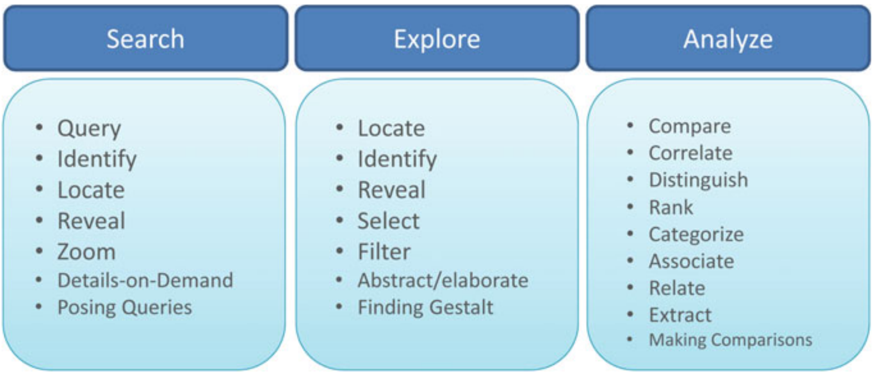

Fig. 2.6 High-level tasks with their sub-tasks

2.5 Data Foundations

A fundamental component in information visualization is data. As the reference model for visualization [3] and the visual analytics process [11] already illustrated (see Sect. 2.1), the process of visualizing information starts with the underlying data. It is essential to process the data in order find the adequate way for visualization. Therefore different aspects of data play a role in visualization. This section introduces some of the most common aspects of the data that should be considered in visualizations. First a number of common classifications on data will be introduced. Afterward different types of data will be described based on an established classification.

2.5.1 Classifications of Data

The starting point of the visual transformation is the data, thus the classification of data is essential for visualizations [3, 11, 70, 71]. In this section various classifications of data will be presented. Some of these classifications were already mentioned in context of interaction and tasks. Further, most of the visualization techniques are described based on the data classes and the way of their categorization [14]. Altogether the classifications of data can be abstracted to three main ways of categorizing data: by data values (level of measurement), by the transformation steps of data, and the data dimensions.

Card and Mackinlay introduced [71] and enhanced [3] a classification based on the value of data. This considers the level if measurement of data values and their ability to order. As already introduced (Sect. 2.1) they propose that data values can be:

- nominal: without any value that can be ordered
- ordinal: possess a value that can be ordered by relations between the values
- quantitative: numerical values and provide thereby a natural order [3, 71].

Ward et al. define the ability of numerical order of data value as "ordinal" [14]. They define that ordinal data values can be binary (e.g. 0 and 1), discrete, or continuous [14, p. 46]. Both, discrete and continuous data types may have numerical values. Further they introduce the mathematical concept of *scale* [14]. They define the data type *scale* as values with ordering relation with distance metric, with which the distances between the values can be computed, and with the existence of absolute zero for the definition of a fixed lowest value [14, pp. 46–47].

Chi introduced a taxonomy for visualizations [72] by using their *Data State Reference Model* [73]. Although, the taxonomy was proposed for classifying visualizations, the aspect that the data transformation and data types are the baseline, is interesting for data classification. The Data State Reference Model [72, 73] proposes three types for data transformation, four data stages and four types of operations within the model [72, pp. 1–2]. The model starts with the *value* (raw data) and

Table 2.5 Data type classification by Shneiderman [36]

Data type by Shneiderman	
1-dimensional	Linear data types
2-dimensional	Planar or map data
3-dimensional	Real world objects
Temporal	1-dimensional data with start and finish time
Multi-dimensional	Data in relational and statistical data-bases with *n* attributes
Tree	Data with a link to (one) parent
Networks	Data items linked to an arbitrary number of other items

generates some form of *analytical abstraction* (*data transformation*). The analytical abstraction contains information about the data (meta-data). In the transformation operations *visualization transformation* and *visual mapping transformation*, appropriate visualizations are chosen (visualization abstraction) and the visual *picture* is generated [72]. The main aspect of data categorization is the differentiation between the raw data (value) and the meta-data (analytical abstraction), that contain structured information about the raw data [72, 73].

Another way of classifying data is by their dimensionality [3, 14, 36, 74]. This classification that was already introduced in context of classifying tasks and interactions (Sects. 2.3 and 2.4), is the most common way to differentiate data and their mapping to visualizations, tasks, and interactions [36, 74, 75]. This classification was proposed by Shneiderman in context of tasks to be solved with visualizations and his *Visual Information Seeking Mantra* [36]. The classification subdivides data based on their dimension in seven categories (illustrated in Table 2.5). The purpose of the classification was not to cover all types of data. Shneiderman proposed that there may exist further data types, e.g. $2\frac{1}{2}$-, 4-dimensions or multitrees. His categorization "reflects an abstraction of reality" [36, p. 339] and various visualizations may use combinations of them [36].

Keim et al. proposed based on Shneiderman's classification a "data type" classification [74], which defines the number of data variables as the dimensionality of data [74, p. 4]. In their classification One-dimensional data have one dense dimension, but they count temporal data in this category. They propose that each point of time may have further variables assigned (and can be multi-dimensional). Further they consolidated Shneiderman's 3- and multi-dimensional data into the category "Multi-dimensional data" and the categories Tree & Networks into the category of "Hierarchies & Graphs". Further they put two new categories into their classification "Text & Hypertext" and "Algorithms & Software" [74]. Table 2.6 illustrates their classification.

Table 2.6 Data type classification by Keim et al. [74]

Data types by Keim et al.	
One-dimensional	Data with one dense dimension, e.g. temporal data
Two dimensional	Data with two dense dimensions, e.g. X-Y-plots or geographical data
Multi-dimensional	Data with more than three dimension, also called multivariate data (We assume that three-dimensional data are investigated here too), e.g. relational databases
Text & Hypertext	Data with unknown dimensions and number. In particular the interlinked (hyperlinked) data (text, multimedia content in the World Wide Web)
Hierarchies & Graphs	Data with relationships to other information entities. The relation can ordered, arbitrary or hierarchical, e.g. e-mail relationships of persons, hyperlink relations on web
Algorithm & Software	Written representation (program code) of complex algorithms

The introduced classification investigated different aspects of data from different viewpoints. The ability to order data values plays an important role for classifying data in [3, 71]. The stages and transformation of data, in particular the differentiation between data and metadata, plays a role in [72, 73]. The main differentiation aspects for Shneiderman [36] and Keim et al. [74] were the dimensions of the data variables, which further may have an order too.

2.5.2 Data Types

The introduced classifications showed that the way how data types are distinguished is tightly coupled to visualization design. We assume that all data variables and values can be distinguished based on their level of measurement according to [36]. Let us assume for example that we have data-set with two variables, time and books. Let us further assume that there is no more information about the variable book, so that we are not able to categorize (order) it. We can now categorize this set of data as two-dimensional (or one-dimensional according to [74]) data with one quantitative (or ordinal according to [14]) and one nominal variable. Based on this information a fitting visualization could be chosen or designed. According to our example all variables can be categorized according to [36], therefore we introduce the data types based on their dimensionality according to Shneiderman [36] and Keim et al. [74]. It is important to have a common understanding of dimension in context of this

Table 2.7 Data types in context of this work (adapted from [36, 72, 74]

Data types in context of this work	
One-dimensional	Data with only one variable, e.g. list of words or temporal data without any associated variables
Two-dimensional	Data with two exactly two variables, e.g. X-Y-plots of time and books
Multi-dimensional	Data with more than two variables, e.g. relational databases
Hierarchies & Graphs	Data with relationships to data. The relation can be ordered, arbitrary or hierarchical
Metadata	Structured data with associations (links, identifiers) to other unstructured data, e.g. markup descriptions of textual webpages

work. Further some data types are not of interest, e.g. "Algorithms & Software", and other can be included to another data type. The aspect of metadata is of interest in context of this work, therefore we enhance the categorization with metadata. The data types will be described according to a slightly different categorization as illustrated in Table 2.7.

One-Dimensional Data
The variable of these data can be ordinal, nominal or quantitative. The way how the values of the variable are ordered (sequential manner as proposed in [36]) is an important factor for the visualization. One example for one dimensional data can be a data-set of countries with the variable "country name". As defined by Shneiderman [36], this variable or data value is nominal and does not own a "natural" order. It may be obvious from our experience with such data, that the nominal variable in this case is ordered alphabetically to provide an appropriate order for searching the relevant information in a more efficient way. This data-set can be visualized as list using the alphabetical order. One-dimensional data can be represented as a table with one column (variable) and a set of rows.

Two-Dimensional Data
Two-dimensional data have exactly two variables that are associated with each other. The values can be represented in a data table as two columns. The two variables may be ordinal, nominal or quantitative. An example could be a data-set consisting of countries with the variables "country name" and "population". As the nominal variable "country name" may be ordered alphabetically, the variable "population" owns a natural quantitative order. Two-dimensional data can be visually represented by X-Y-plots, whereas many chart visualizations are designed to illustrate these kinds of data. In the mentioned example a pie-chart may be an adequate way of illustration; based on the fact that one dimension has a quantitative order and the other nominal. Another example for two-dimensional data may be a data-set of "events" with the variables "quantity" and "time". The variable quantity represents the amount of

events associated with time as a second variable. The main difference between the data-sets is their order.

Multi-dimensional
Data-sets with three and more variables are investigated in this thesis as multi-dimensional data. Shneiderman differentiates in his classification between three and multi-dimensional data [36], while Keim et al. do not classify three-dimensional data at all [74]. The reason is quite simple, thus visualizing the third dimension on a two-dimensional screen is easy. The third dimension or variable can be presented by an adequate visual variable, e.g. color or size. The choice of the visual variable depends on the ability to order the data variable [3, 76]. If the data variable or dimension is for example quantitative, the size of the icons may be the right visual variable to illustrate this issue, in case of a nominal variable the differentiation may be performed by color. Data-sets with more than three dimensions are called in the literature multi-dimensional or multivariate data. There are many data-sets that consist of more than three variables [74]. With each variable or dimension the complexity of visualizing the data increases. Each of these variables may be nominal, quantitative or ordinal. Their visualization can overwhelm and confuse even experts, if hundreds of dimensions are visualized at the same time. Shneiderman proposes the use of buttons, if the cardinality is small, further he introduces a slider to control two-dimensional *scattergrams* [36, 77]. Different visualization approaches focus on multi-dimensional data, the interaction with the visualizations and the control of their dimensions [78–81].

Hierarchies & Graphs
Data entities in data-sets may have different relations to each other and provide thereby a hierarchical or network structure. Keim et al. differentiates between arbitrary, ordered and hierarchical relationships of data entities [74, p. 4]. The relationships of the entities may have different structures. Shneiderman lists as examples, *acyclic, lattices, rooted, unrooted, directed, and undirected* as examples [36, p. 339], but proposes to investigate them all as network data. Keim et al. include the hierarchical structure as the same data type and do not differentiate anymore between hierarchies and networks [74]. Example of these relations may be the interlinking of web-pages, the correspondence of emails or relations in relational data bases. [36, 74] There are various visualization techniques that face in particular the problem of large graphs and huge amount of entities [82–88].

Metadata
Metadata can be simply defined as data or information about data. Chi introduced in his Data State Reference Model a transformation step that produces some form of *analytical abstraction* from the underlying data [72]. The analytical abstraction is one way to generate a data model that provides information about the underlying data. Thus metadata depends in general on the underlying data; their dimensionality can depend to the dimensions of the data too. It is further possible that metadata have less or more dimensions as the raw data. It depends what the purpose of the metadata is. In general, metadata is represented in a structured form using an annotation (markup)

language. The developments in World Wide Web and mobile devices fostered the use and application of metadata in the recent past. There exist a number of common languages for describing metadata, e.g. the Extensible Markup Language (XML) or the Hypertext Markup Language (HTML).

2.6 Methods and Techniques in Information Visualization

There exist many classifications of visualization techniques that use various criteria for categorizing visualizations and provide different views [89]. This chapter already introduced some of the most common classifications. These are using data, interactions, tasks, or the stages of data processing for classifying the visual representation. The following part of this section will focus on a classification of the visual representations and convey correlations to the introduced classifications.

2.6.1 Classifications of Visualization Techniques

The most common classifications or taxonomies for information visualization use one or more of the introduced criterion, namely data, interactions, tasks, or the stages of data processing for classifying the visual representations. Although, most of the visualization aspects were introduced, the step from data to a visual mapping [3] is also essential to understand information visualization as foundation of this work. This section will outline the aspect of visualization techniques from introduced classifications and will further enhance with those classification that use other as the introduced criterion.

Card and Mackinlay classified visualizations based on data value types [71]. Their classification uses the ability to order data values (see Sect. 2.5.1) as criterion for categorizing visualizations. With the use of graphical properties [4] and the mapping to these properties they classified visualizations based on data value types (ordered, nominal, and quantitative) into *scientific visualization*, *GIS*, *multi-dimensional plots*, *multi-dimensional tables*, *information landscapes and spaces*, *node and link*, *trees*, and *text transformation*.[71] In their revised work [3], as they defined *information visualization*, they do not classify scientific visualization as a class or category of information visualization [3]. This definition was applied and is today valid too, that the scientific visualization builds an own class of visualizations [5, 14, 90].

Data as criterion for classifying information visualization played an important role and are still today an important factor for classifications. The most common classification beside the classification based on data value order is the *type by data taxonomy* [36, p. 337] (see Sects. 2.4.1 and 2.5.1). This classification investigates both, the data types as described in Table 2.5 and correlates them to the visualization tasks introduced in Table 2.1 [36, 76]. The correlation of tasks and data types leads according to Shneiderman to solve visualization problems. He proposes to

Table 2.8 Visualization techniques (adapted from [74]

Visualization techniques according to Keim [74]	
Standard 2D/3D display	X-Y and X-Y-Z plots for standard visualization, e.g. barcharts, linecharts, or piecharts
Geometrically-transformed display	Techniques with exploratory statistics to find interesting transformation and patterns in multi-dimensional data-sets
Iconic display	Techniques that map attributes of data values from multi-dimensional data-sets to the features of icons, which may appear as *little faces*, *needle icons*, *star icons*, *color icons* etc.
Dense pixel display	Techniques that divide the screen into multiple subwindows based on the amount of the dimensions in the data-sets. Each data value of the dimensions is mapped to one pixel of the screen
Stacked display	Tailored techniques to present data partitioned in hierarchical manner. Therefore a coordinate system is embedded to another one and this may have further embedded coordinate systems. Each of the coordinates visualizes two attributes and provides with their stacked character the visualization of multi-dimensional data-set

arrange the data types in this taxonomy on the left side, which describes the task-domain. The information objects are correlating to users intentions for solving the tasks. [36] With this matrix a taxonomy is proposed that focuses on data types but investigates the problem-solving process (visual information seeking mantra) by the categorized tasks. [36, 76] The correlations of tasks (interactions) to data and this kind of mapping were enhanced by Keim et al. [74]. They used a slightly different data type classification (see Table 2.6), the interaction techniques (see Sect. 2.3.2) and provided a correlation to visualization techniques. They defined five visualization techniques that correspond to the display mode. They proposed that the visualization classes are "basic visualization principles" [74, p. 5] and can be combined to provide efficient visualizations. Table 2.8 illustrates the visualization classes identified by Keim et al. [74].

The classification of Keim et al. focuses on the visualization of multi-dimensional data. The identified data types include graphs and hierarchies, graph-layout algorithms are not mentioned at all. This type of data is correlated to geometrically-transformed data, which may contain graph-layouts, but this type of visualization is not mentioned [74].

Table 2.9 Visualization techniques of Keim in contrast with each other

Visualization techniques according to Keim [38, 74, 75]		
[38, 75]	[74]	Description
Graph-based	–	Structured visualization of graphs and networks, e.g. basic graphs
Hierarchical	–	Subdividing the k-dimensional space and presenting the subspaces as hierarchies, e.g. treemap
Pixel-oriented	Dense pixel display	Techniques that map data values to the features of screen pixels
Icon-based	Iconic display	Techniques that map attributes of data values to the features of icons
Geometric	Geometrically-transformed	Techniques with exploratory statistics to find interesting transformation and patterns in multi-dimensional data-sets
–	Standard 2D/3D display	X-Y and X-Y-Z plots for standard visualization, e.g. barcharts, linecharts, or piecharts
Hybrid	–	Arbitrary combinations of the introduced techniques

Previous works of Keim [38, 75, 91] proposed a similar classification of visualizations, whereas a dedicated class for Graph-Layout was identified. This classification used more the interaction techniques rather than the data dimensions for identifying the categories of visualization techniques [38, p. T6-6]. Further this classification proposes more the combination of the different visualization classes and classes for 3D-, Dynamic- and Hierarchical techniques. Table 2.9 illustrates this classification and compares it to the visualization techniques in Table 2.6. The descriptions of the visualization techniques are used based on the work in [38, 75, 91] and may slightly be different. Just in case of the standard visualizations 2D and 3D Displays no description were given, so this is according to [74].

A high-level taxonomy to categorize visualization (both: scientific and information visualization) was proposed by Tory and Möller [90]. They investigated the visual space as whole and included factors like user models to get ideas about the "object of study". The involvement of a user model that affect the understanding of what data represent was novel in contrast to the work that was previously focused just on data, interactions and tasks [90]. The classification of visualization provides a high-level classification on data level. The authors differentiate between "discrete" and "continuous" character of "design models", "the conceptualization of a system that the designer has in mind" [90, p. 3] based on Norman's definition [92]. The differentiation leads to the choice of various display attributes, e.g. color or transparency [90]. In the next step of their classification they introduce low-level taxonomies based on the dependent and independent variables (data values) in visualizations. The continuous model visualization correlates the type of the dependent variables to the number of the independent variables. Dependent variables can occur as scalar (color gradients, isolines), vector (glyphs, particle traces), and tensor (ellipsoid-shaped glyphs) [90, p. 5]. The discrete model visualization first differentiates between data structure and data value. Value is defined by the dimension of the underlying data. Tory and Möller differentiate between 2D, 3D, and nD [90]. The structure may occur as node-link, hierarchies, and space-filling mosaics [90].

However, other criteria have been proposed to classify visualizations: e.g. by space, by changes of the data over time or their transformation steps, by number of visual attributes, by tasks and interactions, by the several aspects of data, or by human factors [37, 67, 71, 72, 89]. Further some special criteria were investigated for the classification of visualizations. Grimstead et al. for example investigated visualizations in context of collaboration [93]. Some visual classifications appeared in context of the application domain. Gelernter for example investigated visualizations in context of digital libraries and proposed a categorization in hierarchical lists, concept maps, tree maps, and self-organizing maps [94]. The main criterion for classification still remained the data based classification. Ward et al. presented a taxonomy of visualizations based on the data structure and subdivide each class again by the data types (data value) or visual attributes [14]. This more recent example amplifies the assumption that the visual transformation of the data type, structure and value is an adequate way to categorize visualization types. The classification of Ward et al. is illustrated in an abstract way in Table 2.10.

Table 2.10 Visualization classification by Ward et al. (adapted from [14])

Spatial data	Geospatial data	Multivariate data	Trees, networks and graphs	Text and documents
One-dimensional data	Visualization of point data	Point-based technique	Hierarchical structures	Single document visualizations
Two-dimensional data	Visualization of line data	Line-based techniques	Arbitrary graphs and networks	Document collection visualizations
Three-dimensional data	Visualization of area data	Region-based technique		Extended text visualizations
Dynamic data		Combination of techniques		
Combining technique				

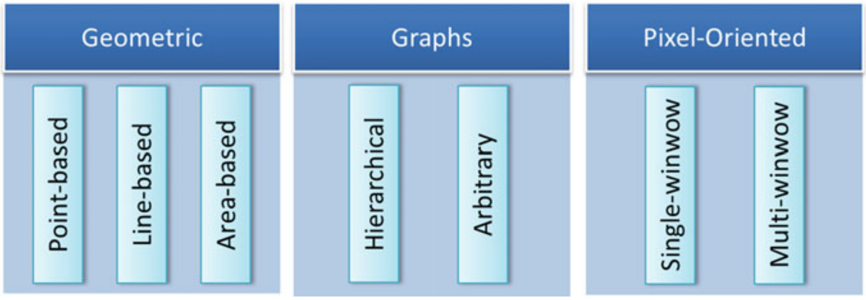

Fig. 2.7 Visualization classification used in this work (adapted and combined from [14, 74, 75]

2.6.2 Visualization Techniques

The classifications of the visualization techniques could outline that many criteria can be used for categorizing visualization. The main criteria still remain the data structure and data value [14, 38]. This section will give an overview about the visualization techniques that are classified above. For introducing the visualizations, a combination of the most relevant classifications will be used. The baseline of the classification used in this work, is the visual categorization by Keim et al. [38, 74, 75, 91]. Thus the different classifications use various terms for similar classes of visualization (see Table 2.9), the introduced techniques will be introduced by recently used terms in the research literature. Further some categories are either obvious (e.g. hybrid), not of interest in context of this work, or not anymore in context of information visualization. These classes will not be considered. Based on these factors a slightly different classification will be introduced that considers the research goals of our thesis. Figure 2.7 illustrates the visualization classification as a combined result of the works of Keim et al. and Ward et al. The introduced visualization techniques serve as examples and do not claim to present the state-of-the-art in information visualization.

Geometric Visualizations

Geometric visualization techniques [38] (Geometric Projection Techniques [91], Geometrically transformed Techniques [74]) transform data values of one- to multi-dimensional data into graphically transformed object. Every transformation that goes beyond the mapping of data values to pixel or a transformation to graph-layout can be count as geometric visualization technique. Keim et al. propose that this transformation aims at finding interesting patterns, in particular in multi-dimensional data-sets [74, p. 5]. In context of this thesis simple transformation in one and two dimensional data-sets are counted as geometric visualization techniques. Thus the "standard 2D /3D" visualization techniques are investigated as part of this visualization technique. The geometric transformation can be performed in various ways. In two dimensional data, e.g. the transformation is a mapping to geometric objects. In multi-dimensional data-sets the use of exploratory statistics, e.g. Principal Concept Analysis (PCA) [95] or Factor analysis and multidimensional scaling, enables such geometric projections [74]. The outcome of the transformation is geometric objects that may appear as different object types. After the review of recent and traditional visualization techniques that use a geometric transformation, the classification of Ward et al. for multivariate data [14, p. 237] is partially applied to outline the main "geometric objects". Therefore we classify the geometric visualization techniques into their output geometry that commonly appears as "points", "lines", and "areas".

Point-Based Visualizations

The geometric transformation to point-based techniques includes the visual projection of data values to graphical representations as points, marks or other "aesthetic entities" [14, p. 237]. According to given attributes the visual representations of each record are placed on the screen to derive a visual representation of the whole data set. Depending on the selected attributes and the chosen layout method, point-based techniques are suitable to compare certain data characteristics, identify outliers or irregularities in the data, recognize relationships among data entities, and identify unexpected or previously unknown clusters and patterns. Usually each data record is projected from its n-dimensional space to a k-dimensional space (usually two- or three-dimensional) and the visual representation of the record is represented at the k-dimensional point on the screen.

One of the most common and established approaches for point-based visualizations are the use of Cartesian coordinates for positioning points [96]. The most common way perform such a positioning are scatterplot (two- or three-dimensional scatterplots [97]). Each axis of the coordinate system represents one dimension of the underlying data. The type of each dimension can either be ordinal or nominal. With the use of linear interpolation or other interpolation and projection methods [98] the data values are mapped to the dimensions of the plot. The representations of each data record are drawn at the location in the coordinate system that corresponds to the attribute values for the dimension in the record. So, different records can be compared according to the chosen dimensions. Another important feature of scatterplots is that the amount of the visualized dimensions is not limited to the number of axes of the coordinate system. Current approaches [97–99] make use of visual properties of the depicted marks to include further dimensions into the representation. One common method for including further dimensions in a scatterplot is mapping of data values to visual variables, e.g. color, size or shape.

The projection of points to Cartesian coordinates is just one way for the geometric transformation of data values as points on the screen. Further examples for point-projections are *Barycentric mappings* that consider coordinates as weighted sums of the anchor positions [100] or projections to circles, constructing multiple dimensions by a flattening process [101]. A recent example of Dinkla et al. visualizes network data with strong structural characteristics as point-based matrices (Compressed Adjacency Matrices) [102]. These matrices allow an easy detection of sub networks with specific structures and motifs [14, 102].

Line-Based Visualizations

Line-based visualizations project data values as lines on the screen. Therefore the vertical axis represents traditionally the value of a data variable and the horizontal axis the order. In many cases this order is a temporal (quantitative) variable that visualizes a continuous value. Line-based visualizations are the most common way to graphically represent the continuous value, e.g. in stock markets and financial sectors. Due to the high distribution and the high familiarity, these visualization techniques are effective for users to analyze and explore data. Each line represents one data value in correlation to two dimensions. With this characteristic the basic form of line-based visualizations are univariate. In contrast to point-based visualization techniques, line-based techniques provide also visual patterns for slopes, curvature, crossing, and further line patterns [14].

The most common way of line-based visualizations is line-graph. Similar to the traditional scatterplots, line-graphs visualizes data values on the axes of Cartesian coordinates. The line-graph is originally a univariate data visualization that can be easily extended for visualizing multivariate data [14]. According to Ward et al. there are four main strategies for providing visualizations of multivariate data with line-graphs: (a) superimposition (b) stacking, (c) ordered superimposition, and (d) ordered stacked [14, p. 246]. The idea behind each of these strategies is to represent each dimension with an own line in the same coordinate system using one of the four strategies:

- *Superimposing*: Superimposing multiple lines in one coordinate system allows directly comparing different dimensions and records in a single visualization. Crossings and shared trends can be easily recognized with this strategy. However if the dimensionality increases this approach becomes unclear.
- *Stacking*: Stacking lines on top of each other avoid crossings of lines. The idea is to start with the first line and use it as base line for the following. The quantitative value of each record is reflected by the distance of the line at a specific point to the line beneath it. So it is difficult to recognize the actual value for each record but this strategy is well suited to explore aggregated values of multiple records.
- *Ordered superimposition*: The ordering or sorting of records based on a specific dimension is another strategy for visualizing multivariate data in a line-graph. The ordering of the records has direct impact to the expressiveness of the visualization. Adequate orderings alleviate the recognition of patterns and relations among the data set.
- *Ordered stacking*: Analogue to the superimposition strategy, the ordering can also be applied to stacked line-graphs. The ordering of records according to a certain dimension may reveal interesting patterns and is a direct factor for the expressiveness of the visualization technique [14, p. 264]

Another well-known and established method of the geometric transformation to line-based visualizations is “parallel coordinates”. This projection technique was introduced by Inselberg for studying high-dimensional geometry [103] and found its way through many applications and enhancements to multi-dimensional data. In contrast to line-graphs, parallel coordinates do not make use of orthogonal axes. They order the axes, which may represent the data dimensions parallel to each other. Spaced vertical and horizontal lines represent the ordering or value of the data. Traditionally each data record is plotted as a polyline across the parallel arranged axes. The polyline crosses each axis at the position proportional to its value to represent the characteristics of the data record. Parallel coordinates can be used to identify clusters in the data by means of similar curve shapes of the visualized records and to identify correlations and outliers [14, 104, 105]. Furthermore parallel coordinates allow two basic ordering methods that can be controlled by the user in most of existing implementations: (1) order of axes and (2) ordering of values [14]. The order of axis determines which dimensions are arranged next to each other and the ordering of values the position of values according to each axis. Analogue to line charts the ordering of axes as well as the ordering of the values in each dimension influences the expressiveness of the visualization. If an unfavorable ordering is chosen relevant patterns in the underlying data may remain hidden from the user.

Since the development of parallel coordinates many approaches extended the idea to provide more efficient visual pattern recognition. Fua and Ward introduced a hierarchical approach for visualizing aggregated information of large data-sets [106]. Miller and Wegemann introduced the concept of line densities to replace the raw data with a density plot that reveals clusters in multivariate data [107]. Yang proposed an interactive hierarchical dimension ordering, spacing and filtering approach that is based on dimension hierarchies derived from similarities among dimensions and introduced an approach for visualizing association rules in parallel coordinates [108]. Peng et al. presented an approach for reducing visual clutter by using dimension reordering strategies [109]. Novotny and Hauser proposed an approach for visualizing context at several levels of abstraction in parallel coordinates for representing outliers and trends [110] [14, p. 248]. Yuan et al. enhanced the line-based approach of parallel coordinates with point based techniques of scatterplots to enhance the usability in large multi-dimensional data [104]. Zhou et al. proposed a splatting approach to reduce clutter in parallel coordinates and reveal patterns [105]. Pilhöfer et al. presented ordering and optimization algorithms based on Bertin's classification [4] to reveal cluster with visual variables [111].

Line-graphs and parallel coordinates were example of line-based visualization, using orthogonal or parallel axes to visualize data values and dimensions, their correlations and visual patterns. The idea of visualizing lines across axes or parallel to axes was applied on different further approaches. Ward et al. outline in particular the radial axes techniques that project a given range of values to a circular scale [14]. They propose that each record is plotted as a line offset from the circular base for representing the data set. This technique is especially useful for analyzing periodic or cyclic data.

Area-Based Visualizations

Area-based (region-based, space-based, space-filling) visualization techniques make use of filled polygons or spaces on the screen to project data values and dimensions on the screen [14]. Usually instantiations of area-based techniques incorporate different properties of the given data into the visual design of the polygons to convey additional values and data characteristics to offer the possibility of comparing different features of data. For instance the size, the shape or the color of the visual representation of a data record can be utilized for visually representing additional dimensions of the data set. Due to the ability of the human perception which enables an effective differentiation of the length or the size of presented polygons, area-based visualizations are successfully applied for representing and analyzing information encoded in the data [14].

The most common representative of area-based visualization technique is the bar chart that was successfully integrated in many different applications for comparing and analyzing data. Similar to scatterplots or line-graphs, bar charts are based on a Cartesian coordinates that include commonly two or three dimensions. Usually the vertical axis represents the range of the values, whereas the horizontal axis represents an ordering of the given records [14]. Each data record is represented as a bar and the length visualizes the data value. According to Ward et al. there are two different strategies to present each data record in a bar chart, stacked or clustered [14]. In a stacked bar chart the values of each dimension and record are stacked together in an aggregated bar. In a clustered bar chart the value of each record for each dimension is represented in a bar and positioned next to each other [14]. Other approaches of bar chart visualizations utilize a three dimensional coordinate system to separate the different dimensions into a new coordinate.

The geometric transformation of data in area-based visualization techniques do not need to be arranged to certain axes. Thus the polygons and spaces provide various visual variables for projection, the polygons themselves may contain information represented by appropriate visual variables. For instance the data in tables that already owns relationships are predestinated to be visualized as polygons [14]. Multivariate data are often stored in tables (see also Sect. 2.1) where each row represents a data record containing the values for each of the dimensions represented by the columns of the table. Tabular displays like heatmaps [112, 113] or table lens visualizations [114] directly exploit this tabular structure to visualize data. Heatmaps utilize a color gradient to map each value of the given table to a specific color and fill the corresponding cell of the table with the derived color [14]. For instance such an approach visualizes higher values with a more intensive color for visually separating these values. The approach of visualizing multivariate data with heatmaps is especially useful for the task of identifying outliers in the data by means of strongly deviating colors.

The Table Lens approach introduced by Rao and Card [114] represents each data record in a row of the table and encodes the values for each dimension and record respectively with a visual representation. For instance a numerical data entry is represented as a horizontal bar and the length represents the value. Usually implementations of Table Lens include different ordering functions that allow users to order the data records in the table with respect to a selected dimension. Another commonly integrated feature of table lens is the ability to expand specific rows by selection and to inspect the data records in its textual form. Table Lens visualizations are especially useful for inspecting a large amount of data and to get an overview of its characteristic as well as analyzing the data distribution of specific dimensions.

Beside tabular and multidimensional data, various data projections were developed with area-based visualizations. Shneiderman proposed a visualization approach that transforms hierarchical data structures in nested rectangular areas by splitting the screen into vertical and horizontal rectangles [115]. This *Treemap* visualization is a classic representative of area-based visualization techniques that is not constrained to any data structure. It is obvious that hierarchies can be conveyed with this visualization in an effective way, but further enhancements of the Treemap proofed its applicability for various data [116–122]. Brunetti for instance applied the Treemap visualization for semantic data [122]. Cox applied a circular Treemap to visualize the customer price index of New York [118] and used color of the spaces to visualize the temporal changes.

The introduced techniques for area-based visualizations should be considered as examples that may illustrate the effectiveness of these visualization techniques. There exist many further approaches that make use of the transformation of data values to polygons and their visual variables. Further examples could be the *Docuburst* introduced by Collins et al., an approach for a radial space-filling tree visualization to explore textual documents [123], Zhou et al. introduces an approach that make use of a splatting framework to reduce the clutters and transforms lines to spaces [105], and we have proposed a superimposed stacked-graph using polygons for visualizing trends and the occurrence of related documents for detecting latent trends over time [124]. Shin et al. proposed a hierarchical tree visualization for mobile devices by integrating focus plus context [125]. Their Tablorer system integrates space-filling and indented lists to visualize hierarchies.

Many applications and methods used combination of the above introduced techniques. A famous example is the *Table Lens* proposed by Rao and Card [114]. Table Lens make use of all three mapping methods based on the value type of the data, e.g. quantitative variables are presented by bars [114]. This approach was applied and enhanced by further applications (e.g. [126, 127]).

Graph Visualizations

Many data sets provide a kind of relationship between data entities. These relationships may be stored in tables, as metadata in structured documents, or appear as a result of data processing. The result of the data entity relationships can be called graphs. A graph consists of data entities that represents as a set of nodes and their connections to each other, called edges or links [74]. Diestel describes a graph as a pair $G = (V, E)$ of sets, where $E \subseteq [V^2]$, $V \cap E = 0$ and the elements of V are vertices or nodes and the elements of E are edges or links [128, p. 2]. The resulting relationships may occur in different ways and can consequently be visualized in various ways. Keim et al. proposes for visualizing graphs ordered, hierarchical, and arbitrary network relation visualizations [74, p. 4]. von Landesberger differentiates graph visualization techniques between node-link, matrix, and combined techniques [129]. This thesis differentiates between those relationships that describe a parent to child relation (hierarchical) and relations that have not any hierarchical correlations

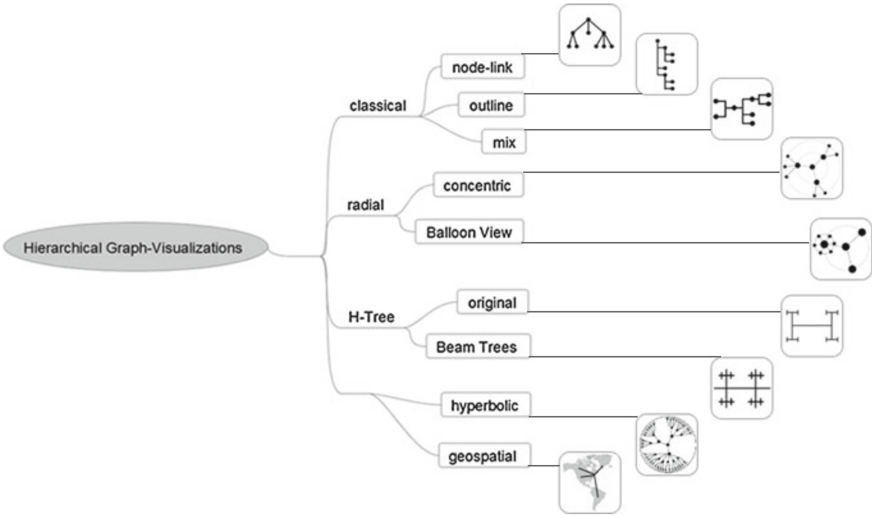

Fig. 2.8 A taxonomy for visualizing hierarchies with graph-layouts (adapted from [130, p. 579] and based on [84, p. 4])

(arbitrary). For differentiating the graph-layouts Herman et al. proposed a survey of different graph-layout algorithms and their application scenarios [84].

Hierarchical Graph-Visualizations
Hierarchical visualizations (or Tree visualizations [14]) aim at interactively visualize the data relationships that can be described as parent to child relation. Graph-visualizations are predestinated for visualizing these kinds of relations. Although the previously introduced Treemaps and further space-filling visual techniques are used to interactively present the hierarchical structure, one main way of visualizing hierarchical relations still remains the use graphs and their node-link diagrams in different ways. We elaborated the literature on graph-visualization and worked out a taxonomy of graph-layouts for visualizing hierarchies [130, 131]. Figure 2.8 illustrates an adapted version of our taxonomy. Although the list of graph-layouts for visualizing hierarchies is much longer, the illustration gives an overview of the how graph-layout could be used to present hierarchies.

One prominent way of using node-link diagrams for visualizing graph-based hierarchies are *dendrograms* [132]. Dendrograms are binary three structures that are characterized by the fact that all nodes of a hierarchy level are in the same line. This attribute improves the visual arrangement of the hierarchical structure. The simple graph-structure of dendrograms allows complex information presentation, whereas huge numbers of entities may overcharge users. Chen et al. investigated this aspect and proposed an overview plus context approach for dendrograms that separates a dynamically-linked overview and detail-view dendrogram to allow more complex information visualization [132].

D'Ambros and Lanza proposed an approach of visualizing hierarchies of discrete "time figures" [133]. Their approach investigated the problem of bug-finding and

-reporting in software systems and visualized it in a hierarchical graph-diagram. Therefore they used a heatmap similar approach for coloring time periods in rectangles, which are then visualized in a hierarchical node-link diagram. Holten and van Wijk introduced a visualization approach for comparing different hierarchies [134]. They proposed a visual clutter reduction method (Hierarchical Edge Bundles) that provides an easy interaction with the complex hierarchical structures. Dinkla et al. proposed a Visual Analytic system that supports comparisons of hierarchies by including node-link diagrams for the hierarchy representation of the weights of the related instances [135]. With a further linked-visualization they provided a detailed view on hierarchy structure, weights and metadata with a user-customizable analysis algorithm for ordering the weights as heatmap rectangles and find interesting nodes [135].

Arbitrary Graph-Visualizations
Hierarchical graph-visualizations are one specific type of graph-visualizations. They premise that at least a parent-child dependency exists. Even, if these hierarchies are just one subgroup of graph-visualizations, the placement algorithms could achieve high complexity as we described in the previous pages. Compared to hierarchical graph-visualizations, the placement of nodes in arbitrary graph layouts that fulfill certain optimization constraints is more complex brandes03. Arbitrary graph-visualizations illustrate information that does not contain a known class or structure [14]. There exist many ways to visualize graph structures, e.g. as node-link diagrams or matrices [136]. Ward et al. subdivides the use of node-link diagrams for graph-visualization into planar and force-directed graph drawing [14]. Force-directed [137, 138] graph drawing makes use of mass-spring-simulations, the so called “spring-embedders”, and model the optimality criteria as an energy function [137]. Each pair of nodes is connected with two forces, one caused by the spring between them and the other a repulsion that keep nodes from getting too close to each other [14, p. 278]. Planar graph drawing makes the assumption that the underlying graph is planar and contains for instance no crossing edges. The research work on graph-drawing contains various methods for drawing graphs and visualizing information and their relationships as visual patterns. Each of the classes proposed by Ward et al. may rise in various ways and provide slightly different views on data and information. We worked out a kind of taxonomy for arbitrary graph-visualization [130]. Figure 2.9 illustrates an adapted version of our taxonomy. Similar to the hierarchical approaches graph-drawing methods, the list of graph-layouts for visualizing non-structured networks is much longer, the illustration aims to give just an overview of some algorithms.

There exist many approaches for visualizing graphs and these were applied in various ways to visualize information. One of the main problems of arbitrary graph-visualization is the complexity of huge amounts of nodes and edges. In particular, in arbitrary graph visualizations, where the structure and classes are unknown [14], the graph-visualization may become difficult for human to understand. Therefore many approaches faced this problem from different point of views. Abello et al. for instance proposed an approach for navigating in large graphs by displaying an overview of the

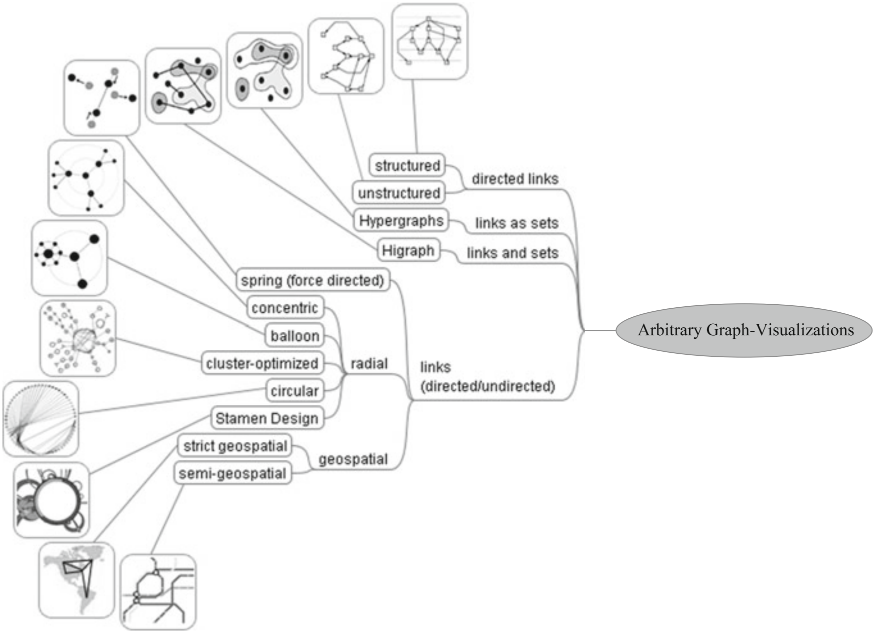

Fig. 2.9 A taxonomy for visualizing arbitrary graphs with graph-layouts (adapted from [130, p. 579] and [84, p. 4])

graph and provide with this overview a navigation support [86]. This linked overview further enables a filter functionality to collapse or expand sub-graphs in their detailed graph visualization. van Ham and Perer [87] faced the same problem from an opposite viewpoint. They proposed that the procedure of getting first an overview and then detailed information (see Sect. 2.4.2 for *visual information seeking mantra*), may not be appropriate for all visualization tasks or groups. They applied the “Degree of Interest” concept of Furnas [45] to graph-visualizations and proposed an interaction model that starts with a user interest-based *search* on initial nodes [87, pp. 954–955]. The second step of their interaction model is *show context* that provides a sub-graph of the focused node. Their model concludes with *expand on demand step*, where users can decide to expand this sub-graph and get more contextual information or an overview. They applied their model to a massive data-set of legal document citations in order to provide a comprehensible view on complex court decisions.

May et al. used the degree-of-interest approach of van Ham and Perer and enhanced it with a multiple focal node selection based on an enhanced focus plus context metaphor [88]. Their approach uses a symbolic (arrows) representation to point along the shortest-path to regions of interest in the graph that might be worth exploring. Further they added landmarks as graphical cues to give information on the context of the visible sub-graph.

Graph visualizations were applied in various domains, e.g. social networks [139, 140] for heterogeneous tasks, e.g. threat detection [141] and with various methods [142, 143]. This thesis does not claim to give a comprehensive view on graph visualizations or their algorithmic optimization methods. There exist a huge number of literature that investigate the aspect of graph visualization, graph drawing [82, 85, 144] and statistics [96] in depth. Further various surveys [84, 136, 145, 146] exist that give an excellent overview on these graph visualization techniques.

Pixel-Oriented Visualizations

Pixel-oriented visualization techniques project each data value of a data-set to one pixel of the screen and present them related to the dimensions [74, 147, 148]. This visualization approach is appropriate for massive data, thus the screen provides a huge amount of pixels and thereby huge visualization capabilities for an excellent overview of massive data. The visualization of data values as pixel has limitations too. One pixel may have the visual attributes of color, including brightness, hue, and saturation or make use of the Grey-scale and its values [148]. Further variables, e.g. shape or texture cannot be used on the pixel level. Massive data can be visualized with the pixel-oriented visualizations with two main limitations: Firstly the amount of visual variables, in most cases color, is limited to a certain range [149]. The second limitation is about arranging the pixels related to the data set. The visualization approach can be seen as a function that projects values from high-dimensional space to a two-dimensional screen [148].

To face the problem of the limitation of visual variables in pixel-oriented visualizations, Oelke et al. has investigated various visual variables for their "boosting" effect [150]. For their work they investigated the visual properties of Ware [28, 29] with respect to their applicability in pixel-oriented visualizations [150]. Whereas the restriction of one data value per pixel could not be applied to many of the proposed visual variables. Thus the variable *halo* for instance needs the space surrounding the pixel, which visualizes the data value [150, p. 9]. They introduced a differentiation of pixel-oriented visualization based on *image-driven* and *data-driven* boosting, which was further subdivided into *parse* and *dense* data sets. They evaluated eight visual variables based on the mentioned classification.

Keim proposed a differentiation of pixel-oriented visualization in *Query-Independent* and *Query-Dependent* visualization techniques [147, p. 2]. While query-independent visualizations visualizes data values by mapping them directly to color, query-dependent visualizations consider the user intention, as her query on the data set. Therefore the distances of attribute values to the query are mapped to colors. The mapping of color in query-independent visualization can be performed by naturally orders of data values [148] or by any other attribute of interest, e.g. by the quantity of data value appearance [80]. Keim introduced the question of using subwindows in various ways [148], e.g. as circle segment techniques or rectangular techniques. The use of subwindows in pixel-oriented visualization is common, but there exist various techniques that make use of a single window for visualizing the colored pixel for conveying information. In this thesis we investigate some examples and separate the pixel-oriented visualization techniques from the human point of view in *single-window visualizations* and *multi-window visualizations*. For perceiving visual information it is in our opinion important how information is visualized.

Single-Window Pixel-Oriented Visualizations
According to Keim, single-window pixel-oriented visualizations make use of the screen to project data values as colored pixel [148]. The order of the pixel depends on the intention of the visualization, if the data value contains a natural order, e.g. quantitative values. The mapping is assigned to this order. May introduced an approach for a single-window pixel-oriented visualization by applying the expressions of the *Karnaugh map* [151] to the visual appearance [152]. His approach enables the visualization of multi-dimensional data on a two-dimensional single-screen display and provides the recognition and detection of visual patterns in the multi-dimensional data-set [152, p. 227]. Another example for such a single-window pixel-oriented visualization was brought by Stein et al. [153]. They used the single-window pixel-oriented approach for visualizing social networks.

Multi-window Pixel-Oriented Visualizations
Pixel-oriented visualizations make maximum use of the screen. Thus each data value can be mapped to one single pixel; the most common approaches are dividing the screen into subwindows [148]. The multi-window approach provides more possibilities to detect and recognize clusters, visual patterns, and correlations [148]. The most common approach for subdividing the screen into multiple windows is rectangular subwindows. Andrienko et al. introduced such an approach for visualizing spatio-temporal data in a pixel-oriented multi-window visualization by using self-organizing maps [154]. Although the rectangular subwindows are the most common way of visualization in this context, there exist further approaches for sub-windowing the visual area. Keim et al. proposed a multi-resolution approach for pixel-oriented visualization in circle segments [155]. Their approach focused on improving the scalability of pixel-oriented visualization by introducing a multi-resolution pixel-oriented visual exploration approach for large data-sets. Therefore they combined clustering techniques with pixel-oriented projections to preserve local clusters by using circle segmentation, an enhanced type of *CircleView*.

2.7 Summary and Findings

This chapter introduced information visualization as canonical foundation of this thesis. The terminological distinction aimed at clarifying the term information visualization in contrast to visualization, scientific visualization and the recently rising term of Visual Analytics. We defined information visualization in context of this work as interactive visualization of abstract data that includes the visualization of data models and provide an interactive character with interlinking to data and their operands to amplify cognition and provide insights and knowledge. In context of this work the human with his ability to perceive and process visual information is mainly focused. We investigated human perception and human visual processing to give an insight how human perceives visualizations. We could outline that beside heterogeneous classification of human visual processing, the so called parallel and sequential (or serial) processing plays an important role, in particular for choosing the proper visual variables. Different research outcomes of studies in cognitive sciences were introduced that prove at least a continuous differentiation of visual variables and the way how and when they are perceived. Further the results of the studies can be used to improve in particular the visual appearance of abstract data.

With the interactive character of information visualization, we could depict that information visualization is more than only pictures. We introduced various classifications of interaction in information visualization and selected one classification to describe interaction on data and visual level. Thus the human interaction leads to solving tasks with information visualization, different classifications of visualization tasks were introduced. We could illustrate that a clear bisection between tasks and interactions is not possible with the existing classifications. We introduced a high-level tasks classification and tried to categorize the different existing classifications into the abstract model, which was partially published in [2] and assigned categorized the existing types of tasks into the abstract model. With visual perception, tasks and interactions, we covered the human-interaction with visual information systems. The data level completed the process of data transformation to an interactive visual representation. In this context various classifications of data provided different views on data, their value, and their dimensions or variables. For describing the data types the three introduced classifications were merged into a slightly different classification. The goal was to give a common understanding of the terms that was often used ambiguously in context of data and visualizing data. We described the most common appearances of data in context information visualization based on our classification. The outcomes of the data structures, values and variables were used to introduce information visualization methods and techniques. Therefore various classifications were introduced that give different views on visualization techniques. We chose one common classification and changed it slightly to introduce an overview of possible visualizations. Based on the introduced classification we introduced exemplary visualization techniques and methods.

The main goal of this chapter was to give an overview of the various disciplines, techniques, goals, and approaches that are coupled to interactive information visualization: cognitive scientists investigates the perception of visual illustrations, algorithmic methods optimize layouting, data models and visualization techniques, the area of human computer interaction investigates the behavior of human and appropriate reactions of computer systems, and further research areas, e.g. Visual Analytics work on the optimization for coupling these methods. Further this chapter worked out that information visualization as an interdisciplinary research area makes use of ambiguous terms and classifies their approaches in various ways. Therefore it is essential to have a common understanding of the terms at least in context of this work.

The next two chapters will focus on specific domains of information visualization and its applications. First we will introduce a general view on semantic technologies and data. The goal is to give a more detailed view on the state of the art in semantics visualization. Thereafter we introduce the general idea of adaptive systems and survey the existing approaches on adaptive information visualizations.

References

1. K. Nazemi, M. Steiger, D. Burkhardt, J. Kohlhammer, in *Handbook of Research on Advanced ICT Integration for Governance and Policy Modeling*, ed. by P. Sonntagbauer, K. Nazemi, S. Sonntagbauer, G. Prister, D. Burkhardt (Business Science Reference (IGI Global, Hershey PA, USA, 2014)
2. K. Nazemi, J. Kohlhammer, in *1st International Workshop on User-Adaptive Visualization (WUAV 2013). Extended Proceedings of UMAP 2013, CEUR Workshop Proceedings*, vol. 997 (2013). ISSN:1613-0073
3. S.K. Card, J.D. Mackinlay, B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*, 1st edn. (Morgan Kaufmann, San Francisco, 1999). http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1558605339
4. J. Bertin, *Semiology of Graphics* (University of Wisconsin Press, Madison, 1983)
5. J.J. Thomas, K.A. Cook, *Illuminating the Path: The Research and Development Agenda for Visual Analytics* (National Visualization and Analytics Ctr, 2005). http://www.worldcat.org/isbn/0769523234
6. D. Keim, G. Andrienko, J.D. Fekete, C. Görg, J. Kohlhammer, G. Melançon, in *Information Visualization*. Lecture Notes in Computer Science, vol. 4950, ed. by A. Kerren, J. Stasko, J.D. Fekete, C. North (Springer, Berlin/Heidelberg, 2008), pp. 154–175
7. D.A. Keim, F. Mansmann, J. Schneidewind, H. Ziegler, J. Thomas, *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, Lecture Notes in Computer Science (LNCS) (Springer, Berlin, 2008)
8. J. Thomas, J. Kielman, Challenges for visual analytics. *Information Visualization* (2009). doi:10.1057/ivs.2009.26. http://www.cs.uml.edu/~grinstei/InfoVisJournal-2009-2011/Information%20Visualization-2009-Thomas-309-14.pdf
9. J. Thomas, in *2007 6th International Asia-Pacific Symposium on Visualization*. APVIS'07 (2007), p. ix
10. J. Thomoas, J. Kielman, Inf. Vis. J. **11**, 309 (2009). Special Issue: Foundations and Frontiers of Visual Analytics
11. D. Keim, J. Kohlhammer, G. Ellis, F. Mansmann, *Matering the Information Age Solving Problems with Visual Analytics* (Eurographics Association, Goslar, 2010)
12. J. Kohlhammer, K. Nazemi, T. Ruppert, D. Burkhardt, IEEE Comput. Graph. Appl. **32**(5), 84 (2012). doi:10.1109/MCG.2012.107. http://bibcd.igd.fraunhofer.de/bibcd/INI_Science/papers/2012/12p079.pdf
13. D.H. Hubel, *Scientific American Library* (W. H. Freeman, San Francisco, 1988)
14. M. Ward, G. Grinstein, D. Keim, *Interactive Data Visualizations Foundations, Techniques, and Applications* (A K Peters, Ltd., Natick, 2010)
15. A.M. Treisman, Comput. Vis. Graph. Image Process. **31**(2), 156 (1985)
16. J. Wolfe, A.M. Treisman, What shall we do with the preattentive processing stage use it or lose it? Poster presented at the Third Annual Meeting of the Vision Sciences Society, Sarasota, FL (2003)
17. A.M. Treisman, G. Gelade, Cogn. Psychol. **12**(1), 97 (1980)
18. A.M. Treisman, J. Souther, J. Exp. Psychol.: Hum. Percept. Perform. **12**, 107 (1986)
19. A.M. Treisman, S. Gormican, Psychol. Rev. **95**(1), 15 (1988)
20. A.M. Treisman, Sci. Am. **255**, 106 (1986)
21. A.M. Treisman, J. Exp. Psychol.: Hum. Percept. Perform. **17**(3), 652 (1991)
22. P. Quinlan, G. Humphreys, Attention Percept. Psychophys. **41**, 455 (1987)
23. J.M. Wolfe, J. Exp. Psychol. Hum. Percept. Perform. **15**(3), 419 (1989)
24. J.M. Wolfe, Psychon. Bull. Rev. **1**(2), 202 (1994)
25. J.M. Wolfe, G.G. Gancarz, Lakshminarayanan, V. (ed.), *Basic and Clinical Applications of Vision Science* (Springer, Netherlands, 1999), pp. 189–192
26. J.M. Wolfe, W. Gray (ed.), *Integrated Models of Cognitive Systems*, pp. 99–119 (2007)
27. L. Itti, C. Koch, Nat. Rev. Neurosci. **2**, 1 (2001)

28. C. Ware, *Information Visualization Perception for Design* (Morgan Kaufmann Publishers, San Francisco, 2004)
29. C. Ware, *Information Visualization Perception for Design* (Morgan Kaufmann (Elsevier), Amsterdam, 2013)
30. R.A. Rensink, Vis. Cogn. **7**, 17 (2000)
31. R.A. Rensink, Annu. Rev. Psychol. **53**, 245 (2002)
32. R. Hyman, J. Exp. Psychol. **45**, 423 (1953)
33. P. Fitts, J. Exp. Psychol. **47**, 381 (1954)
34. J. Accot, S. Zhai, in *Proceedings of CHI 1997* (ACM, 1997), pp. 295–302
35. A.W. Siegel, S.H. White, H.W. Reese (ed.), *Advances in Child Development and Behavior*, vol. 10 (1975), p. 9
36. B. Shneiderman, in *VL* (1996), pp. 336–343
37. A. Cockburn, A. Karlson, B.B. Bederson, ACM Comput. Surv. **41**(1), 1 (2008)
38. D.A. Keim, Visual techniques for exploring databases, in *Invited Tutorial, International Conference on Knowledge Discovery in Databases (KDD'97)* (1997)
39. M. Leissler, A generic framework for the development of 3d information visualisation applications. Ph.D. thesis, Technische Universität Darmstadt (2004)
40. M.A. Hearst, User interfaces and visualization, in *Modern Information Retrieval*, ed. by R.A. Beaza-Yates, B. Ribeiro-Neto (Addison Wesley Longman, ACM Press, 1999), pp. 257–324
41. M.O. Ward, J. Yang, in *Joint Eurographics/IEEE TCVG Symposium on Visualization*, Konstanz, Germany, ed. by P. Brunet, D. Fellner (2004), pp. 137–145
42. E.A. Rundensteiner, Y. Fua, M.O. Ward, IEEE Trans. Vis. Comput. Graph. **6**, 150 (2000)
43. G.W. Furnas, B.B. Bederson, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'95* (ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995), pp. 234–241. doi:10.1145/223904.223934, http://dx.doi.org/10.1145/223904.223934
44. R. Spence, M. Apperley, Behav. Inf. Technol. **1**(1), 43 (1982). doi:10.1080/01449298208914435. http://www.tandfonline.com/doi/abs/10.1080/01449298208914435
45. G.W. Furnas, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'86* (ACM, New York, NY, USA, 1986), pp. 16–23. doi:10.1145/22627.22342
46. G.W. Furnas, The fisheye view: A new look at structured files. Technical report, Murray Hill, NJ (AT&T Bell Laboratories) (1981)
47. K. Boulos, Int. J. Health Geograph. **2**, 1 (2003)
48. J. Garcia, R. Theron, F. Garcia, in *Highlights in Practical Applications of Agents and Multiagent Systems*. Advances in Intelligent and Soft Computing, vol. 89, ed. by J.B. Pérez, J.M. Corchado, M.N.M. García, V. Julián, P. Mathieu, J.C. Bago, A. Ortega, A. Fernández-Caballero (Springer, Berlin/Heidelberg, 2011), pp. 85–92. http://dx.doi.org/10.1007/978-3-642-19917-2_11
49. L. Bartram, in *1997 IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation*, vol. 2 (1997), pp. 1686–1692. doi:10.1109/ICSMC.1997.638254
50. S. Palmer, *Vision Science: Photons to Phenomenology* (MIT Press, Cambridge, 1999)
51. G. Robertson, M. Czerwinski, D. Fisher, B. Lee, Rev. Hum. Factors Ergon. **5**(1), 41 (2009). doi:10.1518/155723409X448017, http://rev.sagepub.com/content/5/1/41.abstract
52. G.G. Robertson, J.D. Mackinlay, in *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology, UIST'93* (ACM, New York, NY, USA, 1993), pp. 101–108. doi:10.1145/168642.168652, http://doi.acm.org/10.1145/168642.168652
53. B. Shneiderman, IEEE Softw. **11**(6), 70 (1994). doi:10.1109/52.329404, http://dx.doi.org/10.1109/52.329404
54. C. Ahlberg, B. Shneiderman, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating Interdependence, CHI'94* (ACM, New York, NY, USA, 1994), pp. 313–317. doi:10.1145/191666.191775, http://doi.acm.org/10.1145/191666.191775

55. C. Ahlberg, C. Williamson, B. Shneiderman, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'92* (ACM, New York, NY, USA, 1992), pp. 619–626. doi:10.1145/142750.143054, http://doi.acm.org/10.1145/142750.143054
56. B. Shneiderman, C. Plaisant, *Designing the User Interface*, 4th edn. (Pearson Education, Toronto, 2005)
57. J. Golbeck, Direct manipulation (Univerity of Maryland, 2002). http://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/dirman.html. Accessed June 2012
58. B. Shneiderman, Direct manipulation: a step beyond programming languages, in *Human-Computer Interaction*, ed. by R.M. Baecker (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987), pp. 461–467. http://dl.acm.org/citation.cfm?id=58076.58115
59. A. Buja, D. Cook, D.F. Swayne, J. Comput. Graph. Stat. **5**, 78 (1996)
60. M.C. Chuah, S.F. Roth, in *INFOVIS'96: Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS'96)* (IEEE Computer Society, Washington, DC, USA, 1996), p. 29
61. S. Wehrend, C. Lewis, in *VIS'90: Proceedings of the 1st Conference on Visualization'90* (IEEE Computer Society Press, Los Alamitos, CA, USA, 1990), pp. 139–143
62. M.X. Zhou, S.K. Feiner, in *CHI'98: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1998), pp. 392–399
63. P.R. Keller, M.M. Keller, *Visual Cues: Practical Data Visualization* (IEEE Computer Society Press, Los Alamitos, CA, 1994)
64. J.S. Yi, Y. ah Kang, J. Stasko, J. Jacko, IEEE Trans. Vis. Comput. Graph. **13**(6), 1224 (2007). doi:10.1109/TVCG.2007.70515
65. W.A. Pike, J. Stasko, R. Chang, T.A. O'Connell, Inf. Vis. **8**(4), 263 (2009). doi:10.1057/ivs.2009.22, http://dx.doi.org/10.1057/ivs.2009.22
66. J. Kohlhammer, Knowledge representation for decision-centered visualization. Ph.D. thesis, Technische Universität Darmstadt (2005)
67. R. Amar, J. Stasko, in *IEEE Symposium on Information Visualization. INFOVIS 2004* (2004), pp. 143–150. doi:10.1109/INFVIS.2004.10
68. C. Fluit, M. Sabou, F. van Harmelen, *Handbook on Ontologies*. International Handbooks on Information Systems, ed. by S. Staab, R. Studer (Springer, Berlin, 2004), pp. 415–434
69. C. Fluit, M. Sabou, F. van Harmelen, in *Visualizing the Semantic Web: XML-Based Internet and Information Visualization*, 2nd edn., ed. by V. Geroimenko, C. Chen (Springer-Verlag, London, 2006)
70. D.A. Keim, Commun. ACM **44**, 38 (2001). doi:http://doi.acm.org/10.1145/381641.381656, http://doi.acm.org/10.1145/381641.381656
71. S.K. Card, J.D. Mackinlay, in *INFOVIS* (1997), pp. 92–99
72. E.H. Chi, in *INFOVIS'00: Proceedings of the IEEE Symposium on Information Vizualization 2000* (IEEE Computer Society, Washington, DC, USA, 2000), p. 69
73. E.H. Chi, J. Riedl, in *INFOVIS'98: Proceedings of the 1998 IEEE Symposium on Information Visualization* (IEEE Computer Society, Washington, DC, USA, 1998), pp. 63–70
74. D.A. Keim, C. Panse, M. Sips, in *Exploring Geovisualization*, ed. by J. Dykes, A. MacEachren, M.J. Kraak (Elsevier, Oxford, 2003)
75. D.A. Keim, in *SIGMOD Conference* (1996), p. 543
76. B. Shneiderman, Olive: On-Line Library of Information Visualization Environments (1999). http://otal.umd.edu/Olive/, Accessed July 2013
77. C. Ahlberg, B. Shneiderman, in *Proceedings of ACM CHI94 Conference on Human Factors in Computing Systems* (1994), pp. 365–371
78. A. Inselberg, B. Dimsdale, in *Proceedings of the 1st Conference on Visualization'90, VIS'90* (IEEE Computer Society Press, Los Alamitos, CA, USA, 1990), pp. 361–378. http://portal.acm.org/citation.cfm?id=949531.949588
79. M. Gahegan, Comput. Environ. Urban Syst. **22**(1), 43 (1998). doi:10.1016/S0198-9715(98)00018-0, http://www.sciencedirect.com/science/article/pii/S0198971598000180

80. T. May, J. Kohlhammer, Comput. Graph. Forum **27**(3), 911 (2008). http://bibcd.igd.fraunhofer.de/bibcd/INI_Science/papers/2008/08p049.pdf
81. T. May, J. Davey, in *IEEE Conference on Visual Analytics Science and Technology 2010. Proceedings. IEEE Computer Society Visualization and Graphics Technical Committee (VGTC)* (IEEE Press, New York, 2010), pp. 239–240. http://bibcd.igd.fraunhofer.de/bibcd/INI_Science/posters-etc/10dp008.pdfbibcd.igd.fraunhofer.de/bibcd/INI_Science/posters-etc/10dp008a.pdf
82. P. Mutzel, C. Gutwenger, R. Brockenauer, S. Fialko, G.W. Klau, M. Krüger, T. Ziegler, S. Näher, D. Alberts, D. Ambras, G. Koch, M. Jünger, C. Buchheim, S. Leipert, in *Whitesides* [118], pp. 456–457
83. S. Whitesides (ed.), *Graph Drawing, 6th International Symposium, GD'98*, Montréal, Canada, August 1998. Lecture Notes in Computer Science, vol. 1547 (Springer, Berlin, 1998)
84. I. Herman, G. Melançon, M. Marshall, IEEE Trans. Vis. Comput. Graph. **6**(1), 24 (2000)
85. C. Gutwenger, M. Jünger, G.W. Klau, S. Leipert, P. Mutzel, R. Weiskircher, in *Graph Drawing* (2001), pp. 473–474
86. J. Abello, F. van Ham, N. Krishnan, IEEE Trans. Vis. Comput. Graph. **12**(5), 669 (2006). doi:10.1109/TVCG.2006.120
87. F. van Ham, A. Perer, IEEE Trans. Vis. Comput. Graph. **15**, 953 (2009)
88. T. May, M. Steiger, J. Davey, J. Kohlhammer, Comput. Graph. Forum **31**(3), 985 (2012). doi:10.1111/j.1467-8659.2012.03091.x, http://bibcd.igd.fraunhofer.de/bibcd/INI_Science/papers/2012/12p037.pdf
89. Q. Chengzhi, Z. Chenghu, P. Tao, in *Asia GIS Conference* (2003)
90. M. Tory, T. Möller, in *Proceedings of the IEEE Symposium on Information Visualization, INFOVIS'04* (IEEE Computer Society, Washington, DC, USA, 2004), pp. 151–158. doi:10.1109/INFOVIS.2004.59, http://dx.doi.org/10.1109/INFOVIS.2004.59
91. D.A. Keim, H.P. Kriege, IEEE Trans. Knowl. Data Eng. **8**, 923 (1996)
92. D.A. Norman, *The Design of Everyday Things* (Doubleday, New York, 2002). First published in 1988
93. I.J. Grimstead, D.W. Walker, N.J. Avis, in *DS-RT'05: Proceedings of the 9th IEEE International Symposium on Distributed Simulation and Real-Time Applications* (IEEE Computer Society, Washington, DC, USA, 2005), pp. 61–69
94. J. Gelernter, Knowl. Organ. **34**(3), 128 (2007)
95. I.T. Jolliffe, *Principal Component Analysis*, 2nd edn. (Springer, New York, 2002)
96. S.B. Jarrell, *Basic Statistics* (Brown (William C.) Co., USA, 1994)
97. H. Sanftmann, D. Weiskopf, IEEE Trans. Vis. Comput. Graph. **18**(11), 1969 (2012). doi:10.1109/TVCG.2012.35
98. S. Bachthaler, D. Weiskopf, in *Proceedings of the 11th Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'09* (Eurographics Association, Aire-la-Ville, Switzerland, 2009), pp. 743–750. doi:10.1111/j.1467-8659.2009.01478.x, http://dx.doi.org/10.1111/j.1467-8659.2009.01478.x
99. F. Jourdan, A. Paris, P.Y. Koenig, G. MelanÃSon, in *Pixelization Paradigm*. Lecture Notes in Computer Science, vol. 4370, ed. by P. LÃvy, B. Grand, F. Poulet, M. Soto, L. Darago, L. Toubiana, J.F. Vibert (Springer, Berlin, 2007), pp. 202–215. doi:10.1007/978-3-540-71027_18, http://dx.doi.org/10.1007/978-3-540-71027-1_18
100. F. Piekniewski, L. Rybicki, in *Artificial Intelligence and Soft Computing–ICAISC 2004*. Lecture Notes in Computer Science, vol. 3070, ed. by L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L. Zadeh (Springer, Berlin, 2004), pp. 247–252. doi:10.1007/978-3-540-24844-6_33, http://dx.doi.org/10.1007/978-3-540-24844-6_33
101. J. Sharko, G. Grinstein, K. Marx, IEEE Trans. Vis. Comput. Graph. **14**(6), 1444 (2008). doi:10.1109/TVCG.2008.173
102. K. Dinkla, M. Westenberg, J. van Wijk, IEEE Trans. Vis. Comput. Graph. **18**(12), 2457 (2012). doi:10.1109/TVCG.2012.208
103. A. Inselberg, Vis. Comput. **1**(2), 69 (1985). doi:10.1007/BF01898350, http://dx.doi.org/10.1007/BF01898350

104. X. Yuan, P. Guo, H. Xiao, H. Zhou, H. Qu, IEEE Trans. Vis. Comput. Graph. **15**(6), 1001 (2009). doi:10.1109/TVCG.2009.179
105. H. Zhou, W. Cui, H. Qu, Y. Wu, X. Yuan, W. Zhuo, in *Proceedings of the 11th Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'09* (Eurographics Association, Aire-la-Ville, Switzerland, 2009), pp. 759–766. doi:10.1111/j.1467-8659.2009.01476.x, http://dx.doi.org/10.1111/j.1467-8659.2009.01476.x
106. Y.H. Fua, M. Ward, E. Rundensteiner, in *Proceedings of the Conference on Visualization'99: Celebrating Ten Years, VIS'99* (IEEE Computer Society Press, Los Alamitos, CA, USA, 1999), pp. 43–50. http://dl.acm.org/citation.cfm?id=319351.319355
107. J.J. Miller, E.J. Wegman, Construction of line densities for parallel coordinate plots, in *Computing and Graphics in Statistics*, ed. by A. Buja, P.A. Tukey (Springer, New York, NY, USA, 1991), pp. 107–123
108. L. Yang, IEEE Trans. Knowl. Data Eng. **17**(1), 60 (2005). doi:10.1109/TKDE.2005.14
109. W. Peng, M. Ward, E. Rundensteiner, in *IEEE Symposium on Information Visualization. INFOVIS 2004* (2004), pp. 89–96. doi:10.1109/INFVIS.2004.15
110. M. Novotny, H. Hauser, IEEE Trans. Vis. Comput. Graph. **12**(5), 893 (2006). doi:10.1109/TVCG.2006.170
111. A. Pilhofer, A. Gribov, A. Unwin, IEEE Trans. Vis. Comput. Graph. **18**(12), 2506 (2012). doi:10.1109/TVCG.2012.207
112. L. Wilkinson, *SYSTAT for DOS: Advanced Applications* (SYSTAT Inc., Evanston, IL, 1994)
113. D. Borland, R. Taylor, IEEE Comput. Graph. Appl. **27**(2), 14 (2007). doi:10.1109/MCG.2007.323435
114. R. Rao, S.K. Card, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating Interdependence, CHI'94* (ACM, New York, NY, USA, 1994), pp. 318–322. doi:10.1145/191666.191776, http://doi.acm.org/10.1145/191666.191776
115. B. Shneiderman, ACM Trans. Graph. **11**(1), 92 (1992). doi:10.1145/102377.115768, http://doi.acm.org/10.1145/102377.115768
116. B. Shneiderman, M. Wattenberg, in *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01), INFOVIS'01* (IEEE Computer Society, Washington, DC, USA, 2001), pp. 73. http://dl.acm.org/citation.cfm?id=580582.857710
117. B. Shneiderman, C. Plaisant, Treemaps for space-constrained visualization of hierarchies. University of Maryland (2009). http://www.cs.umd.edu/hcil/treemap-history/index.shtml. http://www.cs.umd.edu/hcil/treemap-history/index.shtml, Accessed July 2013
118. A. Cox, All of inflation's little parts. The New York Times (2008). http://www.nytimes.com/interactive/2008/05/03/business/20080403_SPENDING_GRAPHIC.html, http://www.nytimes.com/interactive/2008/05/03/business/20080403_SPENDING_GRAPHIC.html, Accessed July 2013
119. K. Nazemi, M. Breyer, C. Hornung, in *Universal Access in Human-Computer Interaction. Applications and Services*. Lecture Notes in Computer Science, vol. 5616, ed. by C. Stephanidis (Springer, Berlin, 2009), pp. 83–91. doi:10.1007/978-3-642-02713-0_9, http://dx.doi.org/10.1007/978-3-642-02713-0_9
120. K. Nazemi, C. Stab, M. Breyer, D. Burkhardt, T. May, T. von Landesberger, THESEUS CTC-WP5-Innovative Benutzerschnittstellen und Visualisierungen: Alleinstellungsmerkmale der CTC-WP5 Technologien. (THESEUS Core Technology Cluster for Innovative User Interfaces and Visualizations: Unique Features of the CTC-WP5 Technologies). THESEUS Programme Deliverable of the Fraunhofer Institute for Computer Graphics Research (IGD). Nazemi K. (ed.) (2010)
121. M. Rios-Berrios, P. Sharma, T.Y. Lee, R. Schwartz, B. Shneiderman, Gov. Inf. Q. **29**(2), 212 (2012). doi:10.1016/j.giq.2011.07.004, http://www.sciencedirect.com/science/article/pii/S0740624X11001055
122. J.M. Brunetti, in *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics, WIMS'13* (ACM, New York, NY, USA, 2013), pp. 37:1–37:8. doi:10.1145/2479787.2479824, http://doi.acm.org/10.1145/2479787.2479824

123. C. Collins, S. Carpendale, G. Penn, in *Proceedings of the 11th Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'09* (Eurographics Association, Aire-la-Ville, Switzerland, 2009), pp. 1039–1046. doi:10.1111/j.1467-8659.2009.01439.x, http://dx.doi.org/10.1111/j.1467-8659.2009.01439.x
124. C. Stab, M. Breyer, D. Burkhardt, K. Nazemi, J. Kohlhammer, in *Proceedings of SIGRAD 2012. SIGRAD, Linköping Electronic Conference Proceedings; 81* (Linköping University Electronic Press, Linkõping, 2012), pp. 83–86. http://bibcd.igd.fraunhofer.de/bibcd/INI_Science/papers/2012/12p121.pdf
125. H. Shin, G. Park, J. Han, in *Proceedings of the 13th Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'11* (Eurographics Association, Aire-la-Ville, Switzerland, 2011), pp. 1131–1140. doi:10.1111/j.1467-8659.2011.01962.x, http://dx.doi.org/10.1111/j.1467-8659.2011.01962.x
126. M. John, C. Tominski, H. Schumann, Visual and analytical extensions for the table lens. in *Proceedings of SPIE*, vol. 6809 (2008), pp. 680907–680912. doi:10.1117/12.766440, http://dx.doi.org/10.1117/12.766440
127. G. Albuquerque, M. Eisemann, D. Lehmann, H. Theisel, M. Magnor, in *2010 IEEE Symposium on Visual Analytics Science and Technology (VAST)* (2010), pp. 19–26. doi:10.1109/VAST.2010.5652433
128. R. Diestel, *Graph Theory* (Springer, New York, 2010). http://diestel-graph-theory.com/basic.html
129. T. von Landesberger, Visual analytics of large weighted directed graphs and two-dimensional time-dependent data. Ph.D. thesis, Darmstadt, TU, Dissertation (2010), 265p
130. K. Nazemi, M. Breyer, A. Kuijper, in *Human Centered Design*. Lecture Notes in Computer Science, vol. 6776, ed. by M. Kurosu (Springer, Berlin, 2011), pp. 576–585. doi:10.1007/978-3-642-21753-1_64, http://dx.doi.org/10.1007/978-3-642-21753-1_64
131. M. Breyer, K. Nazemi (supervisor), D.W. Fellner (supervisor), Benutzerzentrierte Visualisierung von Informationsempfehlungen basierend auf Recommender Systemen. Masterthesis at the Technische Universität Darmstat. Supervised by K. Nazemi (2010), 171 p
132. J. Chen, A. MacEachren, D. Peuquet, IEEE Trans. Vis. Comput. Graph. **15**(6), 889 (2009). doi:10.1109/TVCG.2009.130
133. M. D'Ambros, M. Lanza, in *Proceedings of the 10th European Conference on Software Maintenance and Reengineering, 2006. CSMR 2006* (2006), pp. 10 pp.-238. doi:10.1109/CSMR.2006.51
134. D. Holten, J.J. van Wijk, in *Proceedings of the 10th Joint Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'08* (Eurographics Association, Aire-la-Ville, Switzerland, 2008), pp. 759–766. doi:10.1111/j.1467-8659.2008.01205.x, http://dx.doi.org/10.1111/j.1467-8659.2008.01205.x
135. K. Dinkla, M.A. Westenberg, H.M. Timmerman, S.A.F.T. van Hijum, J.J. van Wijk, in *Proceedings of the 13th Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'11* (Eurographics Association, Aire-la-Ville, Switzerland, 2011), pp. 1141–1150. doi:10.1111/j.1467-8659.2011.01963.x, http://dx.doi.org/10.1111/j.1467-8659.2011.01963.x
136. T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. van Wijk, J.D. Fekete, D.W. Fellner, Comput. Graph. Forum **30**(6), 1719 (2011). doi:10.1111/j.1467-8659.2011.01898.x, http://bibcd.igd.fraunhofer.de/bibcd/INI_Science/papers/2011/11p118.pdf
137. S.G. Kobourov, Spring Embedders and Force Directed Graph Drawing Algorithms (2012). arXiv:abs/1201.3011
138. W.T. Tufte, Proc. London Math. Soc. **3**, 743–768 (1963)
139. L. Shi, N. Cao, S. Liu, W. Qian, L. Tan, G. Wang, J. Sun, C.Y. Lin, in *Visualization Symposium, PacificVis'09. IEEE Pacific* (2009), pp. 41–48. doi:10.1109/PACIFICVIS.2009.4906836
140. B. Gretarsson, J. O'Donovan, S. Bostandjiev, C. Hall, T. Höllererk, in *Proceedings of the 12th Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'10* (Eurographics Association, Aire-la-Ville, Switzerland, 2010), pp. 833–842. doi:10.1111/j.1467-8659.2009.01679.x, http://dx.doi.org/10.1111/j.1467-8659.2009.01679.x

141. T. Crnovrsanin, I. Liao, Y. Wuy, K.L. Ma, in *Proceedings of the 13th Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'11* (Eurographics Association, Aire-la-Ville, Switzerland, 2011), pp. 1081–1090. doi:10.1111/j.1467-8659.2011.01957.x, http://dx.doi.org/10.1111/j.1467-8659.2011.01957.x
142. M. Zinsmaier, U. Brandes, O. Deussen, H. Strobelt, IEEE Trans. Vis. Comput. Graph. **18**(12), 2486 (2012). doi:10.1109/TVCG.2012.238
143. A. Telea, O. Ersoy, in *Eurographics/IEEE-VGTC Symposium on Visualization* (Eurographics Association and IEEE, 2010)
144. G.D. Battista, P. Eades, R. Tamassia, I.G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs* (Upper Saddle River, NJ: Prentice Hall, 1998)
145. M. Graham, J. Kennedy, Inf. Vis. **9**(4), 235 (2010). doi:10.1057/ivs.2009.29, http://dx.doi.org/10.1057/ivs.2009.29
146. S. Jürgensmann, H.J. Schulz, in *Poster at the IEEE InfoVis 2010* (2010)
147. D.A. Keim, SIGMOD Rec. **25**, 35 (1996). http://0-doi.acm.org.millennium.lib.cyut.edu.tw/10.1145/245882.245896
148. D. Keim, IEEE Trans. Vis. Comput. Graph. **6**(1), 59 (2000). doi:10.1109/2945.841121
149. M. Wijffelaars, R. Vliegen, J.J. van Wijk, E.J. van der Linden, in *Proceedings of the 10th Joint Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'08* (Eurographics Association, Aire-la-Ville, Switzerland, 2008), pp. 743–750. doi:10.1111/j.1467-8659.2008.01203.x, http://dx.doi.org/10.1111/j.1467-8659.2008.01203.x
150. D. Oelke, H. Janetzko, S. Simon, K. Neuhaus, D.A. Keim, in *Proceedings of the 13th Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'11* (Eurographics Association, Aire-la-Ville, Switzerland, 2011), pp. 871–880. doi:10.1111/j.1467-8659.2011.01936.x, http://dx.doi.org/10.1111/j.1467-8659.2011.01936.x
151. M. Karnaugh, Trans. Am. Inst. Electr. Eng. **9**, 593 (1953)
152. T. May, Modelle und Methoden für die Kopplung automatischer und visuell-interaktiver Verfahren für die Datenanalyse. Models and Methods for Coupling automatic and visual-interactive Approaches for Data Analysis. Ph.D. thesis, Technische Universität Darmstadt (2012)
153. K. Stein, R. Wegener, C. Schlieder, in Advances in *2010 International Conference on Social Networks Analysis and Mining (ASONAM)* (2010), pp. 233–240. doi:10.1109/ASONAM.2010.18
154. G. Andrienko, N. Andrienko, S. Bremm, T. Schreck, T. von Landesberger, P. Bak, D. Keim, in *Proceedings of the 12th Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'10* (Eurographics Association, Aire-la-Ville, Switzerland, 2010), pp. 913–922. doi:10.1111/j.1467-8659.2009.01664.x, http://dx.doi.org/10.1111/j.1467-8659.2009.01664.x
155. D.A. Keim, J. Schneidewind, M. Sips, in *Proceedings of the 1st First Visual Information Expert Conference on Pixelization Paradigm, VIEW'06* (Springer-Verlag, Berlin, 2007), pp. 12–24. http://dl.acm.org/citation.cfm?id=1759363.1759367

http://www.springer.com/978-3-319-30815-9

Adaptive Semantics Visualization
Nazemi, K.
2016, XVIII, 422 p. 139 illus., 123 illus. in color.,
Hardcover
ISBN: 978-3-319-30815-9