Qualitative Principles of Visual Information Encodings

Vom Fachbereich Informatik der
Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften
(Dr. rer.nat.)

genehmigte Dissertation

von

Benjamin Karer

Datum der wissenschaftlichen Aussprache: 5. April 2019

Dekan: Prof. Dr. Stefan Deßloch

Berichterstatter: Prof. Dr. Hans Hagen

Berichterstatter: Prof. Dr. Gerik Scheuermann

D 386

About the Author

Benjamin Karer started his scientific career in 2008 studying physics at the University of Karlsruhe which was later renamed the Karlsruhe Institute of Technology (KIT). In October 2011, he relocated to TU Kaiserslautern and changed his major subject to computer science, keeping physics as his minor subject. In computer science, he specialized in scientific visualization and the visual analysis of complex high-dimensional data. He obtained his Bachelor degree in March 2014 after submitting his Bachelor's thesis titled "Manifold Learning and Projection of High-Dimensional Data Using Graph Abstraction". In April 2016, he graduated receiving his Master degree at TU Kaiserslautern with the Master's thesis titled "Drall-Based Simulation of Inextensible Elastic Strips". He became a PhD-candidate in June 2016, after obtaining a stipendship at TU Kaiserslautern. In addition to the stipendship, he continued to work as a research assistant at TU Kaiserslautern's Computer Graphics and HCI Group. In July 2018 he left TU Kaiserslautern and began working as an instructor and lecturer for cybercrime investigations and computer forensics for public service authorities.

Abstract

As visualization as a field matures, the discussion about the development of a theory of the field becomes increasingly vivid. Despite some voices claiming that visualization applications would be too different from each other to generalize, there is a significant push towards a better understanding of the principles underlying visual data analysis. As of today, visualization is primarily data-driven. Years of experience in the visalization of all kinds of different data accumulated a vast reservoir of implicit knowledge in the community of how to best represent data according to its shape, its format, and what it is meant to express. This knowledge is complemented by knowledge imported to visualization from a variety of other fields, for example psychology, vision science, color theory, and information theory. Yet, a theory of visualization is still only nascent. One major reason for that is the field's too strong focus on the quantitative aspects of data analysis. Although when designing visualizations major design decisions also consider perception and other human factors, the overall appearance of visualizations as of now is determined primarily by the type and format of the data to be visualized and its quantitative attributes like scale, range, or density. This is also reflected by the current approaches in theoretical work on visualization. The models developed in this regard also concentrate primarily on perceptual and quantitative aspects of visual data analysis. Qualitative considerations like the interpretations made by viewers and the conclusions drawn by analysts currently only play a minor role in the literature. This Thesis contributes to the nascent theory of visualization by investigating approaches to the explicit integration of qualitative considerations into visual data analysis. To this end, it promotes qualitative visual analysis, the explicit discussion of the interpretation of artifacts and structures in the visualization, of efficient workflows designed to optimally support an analyst's reasoning strategy and capturing information about insight provenance, and of design methodology tailoring visualizations towards the insights they are meant to provide rather than to the data they show. Towards this aim, three central qualitative principles of visual information encodings are identified during the development of a model for the visual data analysis process that explicitly includes the anticipated reasoning structure into the consideration. This model can be applied throughout the whole life cycle of a visualization application, from the early design phase to the documentation of insight provenance during analysis using the developed visualization application. The three principles identified inspire novel visual data analysis workflows aiming for an insight-driven data analysis process. Moreover, two case studies prove the benefit of following the qualitative principles of visual information encodings for the design of visualization applications. The formalism applied to the development of the presented theoretical framework is founded in formal logics, mathematical set theory, and the theory of formal languages and automata. The models discussed in this Thesis and the findings derived from them are therefore based on a mathematically well-founded theoretical underpinning. This Thesis establishes a sound theoretical framework for the design and description of visualization applications and the prediction of the conclusions an analyst is capable of drawing from working with the visualization. Thereby, it contributes an important piece to the yet unsolved puzzle of developing a visualization theory.

Organization

Chapter 1 identifies important limitations in the currently predominant data-centric purely qualitative approach to visual data analysis. Based upon these findings, it motivates the concept of qualitative visual analysis and derives the three fundamental research questions to be addressed in this Thesis. Chapter 2 introduces a theoretical framework for the qualitative visual analysis process based on a formal treatment of domain information, data, the visualization, and the mental model determining an analyst's reasoning strategy. The findings in the first two chapters also yield the three qualitative principles of visual information encodings central to this Thesis. In chapter 3, workflows inspired by qualitative visual analysis are discussed. In particular, these are a tight integration of insight provenance information with the data installing a feedback mechanism making the results of previous analysis steps accessible to further analysis and a feasibility proof together with an algorithm for the automatic generation of visual analytics pipelines based on queries for insight rather than for views on data. Chapter 4 discusses the influence of qualitative considerations on visualization design. Two case studies prove the benefit of following the identified principles for visualization design. Chapter 5 concludes this Thesis with a summary of the achievements made and a brief prospect of potential further developments in the direction of the work discussed in thes Thesis.

This Thesis is based upon a number of scientific articles published during the time of the author's PhD-candidateship that started in June 2016. Whereever it is not explicitly stated otherwise, the work presented in this Thesis is either original to the Thesis or extracted from one of the author's scientific publications cited as core references at the end of each chapter.

Contents

List of Figures

Chapter 1

Insight Beyond Numbers

Motivating the major research questions in focus of this thesis, the first chapter reviews some well-known and discusses some more implicit limitations of the current standard approach to data analysis focusing only on measurable and quantitative aspects of the data. It is pointed out that those limitations can be alleviated by explicitly taking into account the qualitative aspects of data analysis, especially the reasoning process executed by the analyst and the knowledge about the general context the data is investigated in. Qualitative visual analysis is proposed as an addition to the currently dominant purely quantitative analysis. It does not focus on qualitative data but rather on the qualitative aspects of data analysis, explaining visualization from the perspective of the viewer's reasoning and interpretations rather than only by the data displayed. The discussion concludes with the derivation of three major research questions to be answered in the remaining chapters: How to formally capture the interconnection and interweaving of quantitative and qualitative aspects and considerations in visualization and visual analytics, what kind of workflows need to be implemented to support or perform qualitative visual analysis, and how qualitative visual analysis can be integrated into visual analytics solutions for information and scientific visualization.

1.1 Introduction

The support of data analysis is one of the earliest applications of visualization. When computers became powerful enough to allow the generation of graphical data representations from available digital data, early applications were dominated by science and engineering. Although these fields have been joined by a myriad of other applications, science and technology are still major drivers for the field. It is perhaps due to this historical development that there is a certain tendency among visualization scientists to prefer measurable quantitative characterizations of data and analysis results over nonmeasurable qualitative considerations. Numbers are being trusted as unbiased, exact, neutral, and objective. Human reasoning instead is conisered imprecise and prone to error. Quite remarkably, the very fact of recognizing the risk of misinterpretation and

error already emphasizes the importance of qualitative considerations for data analysis. There is indeed a collection of work showing that this is actually a concern. However, the discussion either remains implicit or focuses only on certain aspects of the analysis process. This work advocates the open and explicit consideration of qualitative aspects from a holistic perspective as necessary and integral component of the visual data analysis process. In its investigation of qualitative aspects of the analysis process as a general principle rather than a necessary adaptation to domain- or task-specific requirements, this work follows a different approach than, for example, design studies. The focus on general applicability rather than finding individual problem-specific solutions also sets the proposed ideas clearly apart from application- or purpose-driven visualization. The discussion reveals that an open and explicit consideration of qualitative aspects is a necessary component of a holistic perspective on the visualization and data analysis process. Towards such a more holistic perspective, this chapter attempts to identify qualitative aspects of significant influence on the analysis result and subsumes them under the term qualitative visual analysis. Being an addition to the currently predominant data-centric perspective on data analysis, qualitative visual analysis has a solid embedding in both empirical findings and theoretical models about reasoning with visualization.

The discussion is founded in three core ideas determining the major components of the notion of qualitative visual analysis presented in this Thesis. The first concept is concerned with the interpretation of data. Different perspectives on the same data will yield different analysis results. There is no absolute meaning to the data since every analysis result at some point requires evaluation. Evaluation in turn either implies human judgment or an automatic inference of meaning. In the former case, the interpretation is inherently subjective. In the latter case, the interpretation is predetermined by a programmer with respect to the programmer's background and by derivation schemes either implicitly or explicitly defined by the domain context. In both cases, the conclusions drawn and thus the analysis result will differ between individuals, especially between different domains. Therefore, qualitative analysis makes a difference in practice. Domains do not only have their own interpretations of data but also bring their own deduction and inference principles and mechanisms. Supporting analysis within a domain thus does not only mean to reflect the data based on its shape and structure, but also to support the reasoning. Reasoning is not reading off information. Too often, it is assumed that insight is obtained as soon as a certain structure is identified in the visualization. Yet, studies show that there is much more to this process. Artifacts and structures in the display do not only need to be recognized but also to be combined and understood correctly, and to be evaluated in the analysis context. Insight therefore is an emergent property of the viewer's cognitive processing rather than a feature of the data that can be mined or otherwise extracted. Consequently, qualitative analysis is about reasoning, not about perception. In many applications, the insights to be obtained have to be inferred or derived by reasoning about the data rather than just reading off the result from the graphical representation. Capturing the reasoning process is an aspect of information provenance. Since conclusions depend on interpretations, supporting the deduction of insight is a qualitative consideration. The knowledge necessary to infer insight about the domain is not part of the data and differs between viewers and between domains. The same

holds for the inference structures and reasoning systems applied to interpret the data. The knowlede outside the data thus has a significant impact on the interpretation of data. It does not only determine what is read from a visualization but also how it is understood. To render analysis workflows comparable, detailed provenance information is required to document the reasoning process and to identify the influence of outside knowledge. Qualitative Analysis relates the insight to the outside. The idea of qualitative analysis is to explicitly relate the information to be found inside the data to the ouside knowledge applied to obtain it by reasoning about the data.

1.2 State of the Art

Because the interpretation of data is subject to the analysis context, the results obtained from data analysis are automatically subject to qualitative consideration. It is therefore not surprising that there is a variety of work already considering qualitative aspects. Yet, these aspects are often only applied implicitly or even abstracted away by attempts to measure visualization performance. The following discussion lists a variety of of applications of qualitative visual analysis based on work that either implicitly or explicitly employs qualitative considerations.

It is widely accepted in the community that visualizations should be tailored to the domain context [118, 128]. As a result, there is a plethora of techniques trying to incorporate the domain's perspective into the design. Tory and Möller point out expert interviews to be useful especially in the early phase of design but also remark that they should be complemented by user studies evaluating the resulting application against the design goals [138]. Understanding of the domain context can be obtained for example from involving users in participatory design, applying structured interviews [128, 131] or by field studies [47, 64, 67]. Domain-analysis is a specific technique useful when working in environments with existing software solutions [50]. Task-oriented design aims at the characterization of analysis tasks and their decomposition into interaction workflows [4, 149]. Being directly concerned with the solution of specific low-level or high-level analysis questions, task-based design primarily captures the analysis context. In this regard, a recent crowdsourcing study reports that the same visualization performs differently in the contexts of different tasks [122]. From the opposite perspective, this is reflected by different interaction patterns users of visual analytics applications show while performing different tasks [57]. These findings suggest to dynamically adapt the visualization not only to the domain context but also to the analysis and the user context. In this direction, Golemati et al. proposed a context-adaptive visualization environment extending the typical focus of automatic visualization on the domain and data by an explicit consideration of user profiles and preferences [55]. In the intelligence sector, user models for adaptive visualization have been reported to support improving the distinguishing of relevant and non-relevant information which in turn allows optimizing relevance-based visualizations [1].

Derived information can only be as trustworthy as the data it is derived from. Da Silva et al. point out the problem for automatically inferred data in the se-

mantic web [36, 37]. From the perspective of visualization, insight provenance has been addressed from different angles. Two major directions are automatic tracking of user interaction (e.g. [4,12,61]), and manual documentation of analysis decisions and interpretation steps (e.g. [60, 83]). Gotz and Zhou combine these approaches and organize them around a taxonomy of provenance aspects with different granularity [58]. Based on this taxonomy, they introduce HARVEST [56], a versatile visual analysis framework that can be applied for example in collaborative analysis, where a group of analysts can share insights with each other or with other groups. The visualization used to obtain an insight is obtained is combined with an action trail to make the insight comprehensible for other users who might then apply these insights (i.e. extend the trail) to develop their own. A similar approach but with a focus more on the inference aspect is followed by ProveML [146], an extension of the open provenance model [95]. A survey by Hall et al. identifies guidelines for the implementation of provenance in geospatial visualization by reviewing the corresponding literature with respect to a model for human reasoning about spatiotemporal data [65]. Although the paper is focused on geospatial visualization, the derived guidelines are general enough to be adapted for other applications. The review can thus serve as a starting point for more research in this direction. If cooperation across domains is to be established, each domain might introduce its individual language and interpretations. For such semantically heterogeneous environments, a graph-based visualization has been proposed allowing the viewer to navigate through the multiple possible interpretations [73].

Studies report that global tasks, involving inference from a number of observations are harder to perform then local tasks where information can be read directly from the depiction [62, 115]. Casner applies a similar argument to motivate the transformation of cognitive tasks into perceptive tasks that provably yield the same results but are substantially easier to perform [23, 24]. Rather then directly optimizing the visualization towards insight emergence a viewer can also be provided with proper guidance. In this regard, Demiralp et al. propose a tool for what they call insight-queries [39]. Although their definition of insight based on purely quantitative properties is rather unusual, their approach enables the search for abstract information with a focus on data correlation.

1.3 Motivation and Approach

The aim of this chapter is to outline the importance of qualitative considerations like possible interpretations of data, the general analysis context, and applied reasoning procedures for the discussion of visual information encodings. A review of the currently predominant purely quantitative approach to data analysis reveals several limitations resulting from an insufficient consideration of qualitative aspects. From these observations, the definition and scope of qualitative visual analysis are derived which is presented as an additional perspective on the visualization and visual data analysis process. The third section reflects the role of qualitative visual analysis in the general reasoning process based on findings reported in the literature. Qualitative visual analysis addresses a gap in the existing work concerning the combination of several qualitative aspects

of analysis towards a holistic perspective on visual data analysis. This gap motivates the three major research questions to be addressed in this Thesis: visualization validation, workflows supporting reasoning, and the influence of qualitative considerations on visualization design.

1.4 Related Work

One of the early attempts to study the actual process of reasoning with graphics is the work of Scaife and Rogers. In their 1996 article on external cognition, they discuss research questions in the cognition of data visualization, regarding the analysis of data (external structure) with respect to a mental model (internal structure) [123]. Today, more than twenty years later, some of the research questions they ask are still open. For example, the actual processes of data (mis-)interpretation and integrating the understanding of observations with knowledge are not well understood. Getting a better understanding of these issues is one aim of qualitative analysis. Cleveland and McGill define graphical perception as the act of decoding the quantitative and qualitative information encoded in visual data representations [34]. Quantitative information covers the measurable aspects of data – its numerical or otherwise deterministic values and the measuarable properties that can be derived from this information. Qualitative observations cannot be completely captured based only on numerical and nominal data. Due to this fact, they are often regarded as less exact than numbers or otherwise deterministic data values [102]. Yet, this is a grave misunderstanding. Traditionally, science is driven by theory modeling behavioral properties of a system. This theory has to be validated against observations in a controlled experimental environment. Although the theory's formulas will often be applied to predict numerical values or estimates, the actual description is obtained from the values' qualitative characteristics. The prediction is only correct, if the model reproduces the relations and dependencies between the parameters and variables correctly. Qualitative analysis thus is not an imprecise comment but rather a precise and deterministic prediction of observations determined by the functions and predicates of the corresponding domains' applied logics and deduction techniques.

Anscombe's quartet is a popular example demonstrating how visualization can aid the formulation of an initial theory or descriptive model for the data [6]. Discussing the value of visualization for practitioners, Fekete et al. emphasize the value of visualization's capability to support identifying models as a starting point for analysis [51]. They propose a combination of automatic data analysis and visualization, arguing that automatic analysis and (exploratory) visual analysis answer different analysis questions and thus should attempt to support each other rather than establish concurrency. Their arguments are in line with the notion of qualitative analysis proposed below. The combination of visual and automatic data analysis is the central idea of visual analytics [74]. A holistic perspective on data analysis benefits from synergies between data-centric and user- or task-centric considerations [112]. While the added value of such a holistic perspective is significant, the qualitative aspects of data analysis are underrepresented in the discussions of visualization performance.

Determining the effectiveness of visualization remains a challenge. There is some consensus within the community that design benefits from expert reviews while evaluation should be performed by user studies [138]. However, there is less agreement on whether user studies should be quantitative or qualitative. Quantitative studies are restricted to the assessment of measurable variables, limiting them to benchmark tasks. These benchmark tasks, however, tend to be rather low-level [110]. Qualitative user studies ask open questions providing the users with the opportunity to find their own answers and to report their own insight. On the downside, the insights reported from qualitative studies are generally hard to compare. In this regard, it has been proposed to identify coding schemes transforming the qualitative results to quantitative data [102].

Regarding general qualitative considerations, Kosslyn identifies a set of acceptability criteria allowing to evaluate how well visualization conveys information on the semantic level [82]. Ratwani, Trafton, and their colleagues remark that these findings do not generalize to situations involving the inference of meaning where it cannot simply be read off from the depiction [116,140]. Studies performed by Guthrie et al. also account for this observation [62]. Petre and Green report evidence, that obtaining information from visualization also depends on experience [107]. These results emphasize the importance of qualitative aspects of data analysis for the reasoning process. Only few models for the processes underlying the inference of meaning from graphical representations exist for cases where it cannot be read off directly. Examples are the prediction of visual saliency for different regions of a visualization and attempts to estimate the amount of mutual information that can be established between a visualization and a viewer based on visual reproducibility [71, 72]. In combination, quantitative and qualitative models allow the simultaneous consideration of optimal data representation and reasoning support. It is therefore a concern of qualitative visual analysis to promote additional research in this direction. The explicit consideration of qualitative aspects regarding the inference of meaning and information by data interpretation requires precise but intuitive models to reason about this process. It is therefore also a concern of qualitative visual analysis to encourage deeper investigation of these processes and to promote the development of additional models to capture domain-specific closure and reasoning schemes.

In his "Views on Visualization", Van Wijk proposes different high-level models for the visualization process [143]. One of his key findings is that the visualization itself – not only the interpretation by its viewer – is subjective: It does not only depend on the data but also on the algorithms and data structure used as well as other factors determined by the programmer. Remarkably, this finding generalizes to automatic analysis. Qualitative analysis as proposed in this Thesis recognizes this inherent limitation of all algorithmic quantitative treatment of data and alleviates it by requiring the provision of provenance information that at least allows the analyst to comprehend the motivation and consequences of the programmer's choice of a specific algorithm. Bresciani and Eppler review 51 papers to identify pitfalls in visualization design [21]. Among these issues they remark that accuracy is a general problem of visualization because graphical representations are less exact than for example numbers or tables. Other than a lack of accuracy in the direct depiction, there is also an interpretative inaccuracy which might mislead the viewer or analyst to false conclusions although

the visualization actually contains all necessary information. This problem is even more severe if the information cannot just be read off but has to be inferred or derived from the depiction. In this case, the fact that information is inferable or derivable does not automatically determine the viewer to actually apply the rules revealing it. The standard work in this direction are the books of Huff and Tufte on "How to lie with Statistics" and "Visualization Lies" [69,141]. Extending the consideration and discussion by the qualitative aspects causing these problems, these issues can be resolved – or at least mitigated.

1.5 The Limits of Quantitative Data Analysis

Quantitative approaches have always dominated the development of new data analysis techniques. Quantitative data analysis is well suited for all problems that only require only limited qualitative understanding and are commonly interested primarily in a quantitative characterization of the domain based on measurement and valuation. Rather than to derive a set of models and theories, their aim is to obtain descriptions. The focus is more on the system's appearance than on its behavior. Visualization, in contrast, is typically concerned with the derivation of insight from observations made in some domain and thus focused on the generation of theories and models. This discrepancy results in certain limitations of a purely quantitative perspective on visualization which are discussed in the following.

1.5.1 Foundational Considerations

All data is expressed in terms of mathematics and logics. There is a set of symbols or numbers and a collection of combining operations as well as a precise framework determining how to apply these operations in order to implement transformations between symbols or even between operations. Interestingly enough, the mathematics applied to the description of the data is not always capable of describing the knowledge an analyst is interested in learning from data analysis. Informally speaking, the attempt to obtain an image of an observation in terms of data might very well loose relevant information about the surrounding environment – just like a photograph focusing on a single object looses detail about the object's surrounding. Thus, just like looking at this photograph will not reveal the complete picture, it appears that the same applies to analyzing data since, essentially, data is just a snippet of some limited view on an observation. In some sense, just like what is in the picture is an interface to connect the viewer with the depicted scene, the context spanned by the data and the information it provides is an interface connecting the analyst to the observation. The specification of this interface is encoded by the relations between the outside domain context and the inside data context. Analysis of the inside of some data context more often than not requires information from the outside domain context to obtain meaningful results. Even worse, there also are cases where the outside domain context actually dominates the analysis, rendering an approach only focusing on the data completely infeasible. An example for this would be the analysis of crime scene information during the first response to a

crime that has just been reported. In this situation, available reliable data is sparse and will only gradually be aggregated during the investigation process. The initial considerations therefore have to rely more on general domain knowledge than on available data. Summarizing and generalizing these considerations indicates the following principle:

Principle: Inside-Outside Principle of Data Analysis

Towards the extraction of meaningful information from data, it is necessary to consider both, the data context as a coherent unity of of quantifiable or otherwise expressible encoded information and its surrounding domain and general context determining the system's behavioral aspects and the viewer's capability of understanding and interpreting data analysis results based on knowledge outside the data.

Within the visualization community, it is commonly agreed that a qualitative perspective on the data is a good way to derive information, relations, and finally obtain insight and knowledge about the domain. This is also recognized by widely known visualization models such as in Bertin's distinction between presentation and layout [16] and Rensink's triadic architecture for the description of visual data representations [117]. Both agree on the distinction between information that can simply be read off from a graphical representation and information needing to be inferred involving a reasoning process. While the former represents the answers presented by quantitative data analysis, the latter category involves a qualitative reasoning process. Yet, the discussion in most articles concentrates almost entirely on quantitative aspects of the analysis. The central proposition of this chapter hence is to establish an explicit link between the inside and the outside of the data context by adding a qualitative component to the discussion of the visual data analysis workflow. Further motivating this idea, the following discussion investigates some limitations of purely quantitative data analysis.

1.5.2 Scales and Units

Depending on the chosen scale, the appearance of the data can change entirely, potentially misleading to wrong interpretations [52, 87]. Improper scaling can cause apparent correllations to appear in the data where actually there are none. Even worse, this can be achieved quite easily by proper data manipulations [69]. Especially for complex high-dimensional systems, there is a risk of implying observable but actually non-present correllations in the visualization [145]. Another well-known issue with high-dimensional data is what is often called the curse of dimensionality. Distance and similarity measures that are very intuitive and descriptive in low dimensions, quickly lose their discriminative power in higher dimensions [17, 66]. Many techniques for dimension-reduction also tend to summarize dimensions by variance rather than by units. Consequently, the meaning of distances in the lower-dimensional representation is ambiguous or even totally unclear. Besides this fact, multidimensional distance

measures are a popular method to group data into clusters for further analysis. It can be questioned, whether the structures found this way are actually meaningful, especially since there are cases where a semantically meaningful measure capturing all data dimensions simultaneously might not even exist at all [77]. Perhaps even worse, the application of measures without proper respect to the units they operate on might again imply structure in the visualization where there actually is none and thus yield misinterpretations.

1.5.3 Features vs. Entities

Much less often than scaling or unit issues, the problem of comparability between the investigated data objects is addressed directly. Although similarity measures are a popular approach to structure data sets for visualization, their semantic interpretation is rarely discussed explicitly in visualization applications. However, although this is typically implied, not everything that appears closer in the visualization is automatically more similar. First, distance and similarity measures often combine multiple dimensions without regarding the units they represent. This mixture of units renders the actual distance meaningless if the units are not compatible, no matter whether proper scaling is applied or not. The reason is that for incompatible units, the summary of features implied by the distance or similarity measure introduces ambiguity to the interpretation. As a simple three-dimensional example, consider a data set of objects of different color at different positions, say traffic lights at certain crossings in a city. Now, let any distance measure be applied to this data set. The question now is, what exactly some distance x between two traffic lights reveals about them. The same distance might be measured for traffic lights because they show the same color or because they are close to each other. Consequently, the traffic lights showing opposing colors at the same crossing may well appear unrelated while the traffic lights at another crossing might appear more similar only because they show the same color. While such an interpretation might actually be useful, it is a result of a certain semantic ambiguity rather than of a careful choice of features to compare the different entities.

Visualization often tends to compare entities by the totality of their features rather than by the discriminative property determined by the analysis question. The very fact that visualization is about entities and objects rather than about features is neglected if the visualization is only computed on the features rather than on the entities. A scatterplot matrix, for example, is not a comparison of entities in different features but rather a comparison of the variation of different features among a set of entities. Consequently, it is unsuitable for the comparison of entities simply because in their representation as points in a scatterplot, it is very hard to identify the same entity in the different plots. Proper representations of similarities and differences determining the comparison between different entities and objects in the data hence need to be designed with careful consideration of the analysis question and general domain context rather than only by statistical properties of the data features.

1.5.4 Context-Sensitivity of Interpretations

The meaning of data is not absolute but bound to different levels of context. Different domains and even different users will obtain completely different results from analyzing the exact same data set. For example, an ecologist working for an environment-protection agency will interpret data on the mileage and motorization of cars differently than a mechanical engineer analyzing trends in motorization for the automobile industry, just because of their individual profession (individual context). Although the example is fictional, it is plausible that in the same data set other aspects are relevant for the environment-protection agency than for the automobile industry. These general perspectives determine the analysis' focus and hence influence its results (domain context). Assuming both analysts are tasked with finding structure in the data, both will likely apply similar techniques. For example, they might both attempt to classify the data based on its values (data context). Assuming the ecologist analyzes the data with a focus on pollution, different classes will be obtained than from the mechanical engineer's classification with respect to motorization features. The differing motivation behind the classification yields different results (analysis context). Figure 1.1 illustrates the example. Even though both analysts investigate the same data and apply the same techniques for classification, the obtained results are different ((A) and (B)). Note that this is not a result of an ambiguity in the graphical representation. Different domains will likely interpret the same data differently. Overlaying the ecologist's classification with the subspace considered by the mechanical engineer shows an interesting correlation between low motorization and high pollution (red) as well as strong motorization and low pollution (green) (C). Showing all five dimensions concerned by the two analysis (D) and highlighting the correlation to low weight (orange crosses), it turns out that this is due to two specific choices in the class definitions. The ecologist excluded heavy-weight cars from the high-pollution category since their stronger engines naturally show higher combustion. The mechanical engineer considered the specific power (the quotient of horsepower and displacement) for the class definition and distinguished between necessarily (brown) and unnecessarily strong (green) motorization. Due to these considerations, the relative pollution of heavy-weight vehicles with strong motors is only considered to be among the medium group (cf. A and C).

In practice, incomplete or incorrect qualitative context thus provokes misunderstanding or misinterpretation of the data.Without knowledge of the ground truth – the data and the class definitions – a third analyst matching cars between the two classifications, would not be able to resolve the aforementioned observations and probably be misled to the conclusion that heavier cars or cars with stronger motors cause less relative pollution compared to smaller cars with weaker motors. While to some extend the data reveals this to actually be the case, this information is unreliable as it is a result of the distortions applied to the class definitions.

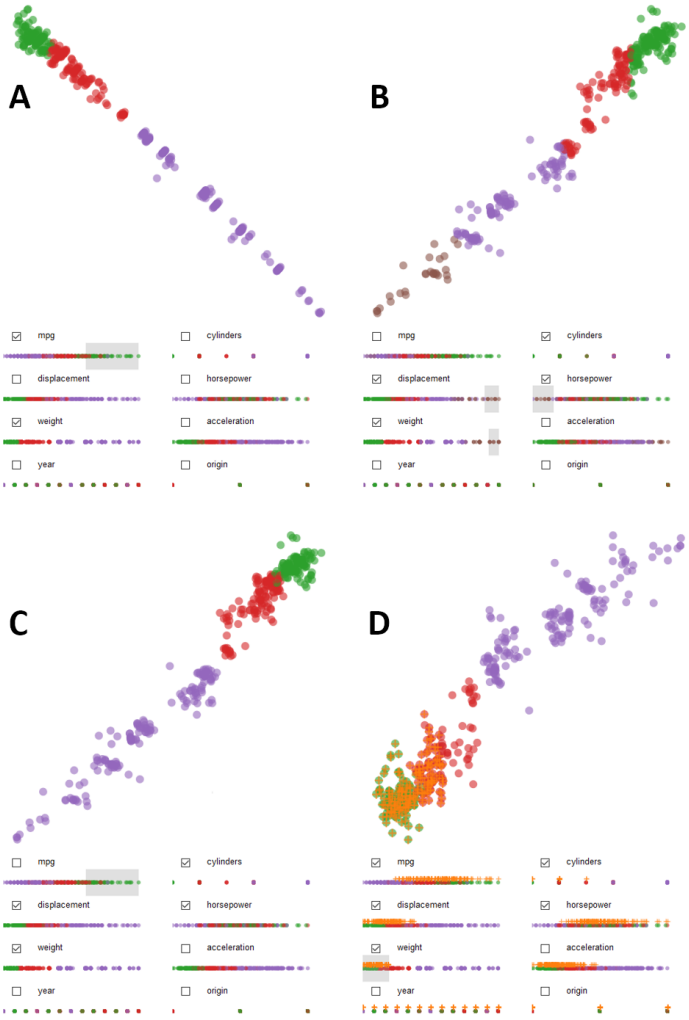


Figure 1.1: *The analysis result depends on the general context. In the example, the auto-mpg dataset [88] capturing different features of cars from the 1970's has been investigated through the perspectives of an ecologist interesting in relative pollution caused by the cars and a mechanical engineer interested in their motorization. Despite being fictional, the example illustrates the effects of different analytical foci due to different domain contexts. Both analysts classify the data in the subspaces reflecting their respective analysis context (marked checkboxes). Classes are obtained by nearest-neighbor matching according to the Euclidean distance to seed points characteristic for each class as defined by the analysts. Although the method is the same, the ecologist (A) and the mechanical engineer (B) obtain different classes. For comparison, the engineer's perspective is overlayed with the ecologist's clustering (C) and all dimensions considered by the two analysis are shown with an emphasis on the correlation by weight as the common dimension between the two classification definitions (orange marks in (D)).*

1.5.5 Locality and Complexity

The interpretation of data becomes significantly harder if comparisons are to be made and where different parts of the visualization need to be related. As a simple example, consider a pie chart. By simple read-off, the viewer can determine the contribution of each indivdual group to the total. Comparing the contributions of different groups however requires to compare diagram slices that might not be neighbors. In cases where the difference is not entirely obvious, instead of a simple size comparison, the two sectors need to be evaluated by their contribution to the whole to make them comparable. This becomes even more complex when the aggregation of several sectors is considered. If those sectors are not neighbors, the aggregation requires to relate objects from across the depiction to the whole in order to obtain their combined influence. If those sections are neighbored, the sectors can instead simply be interpreted as one.

This problem is actually well known in visualization and solutions to the problem described here appear quite obvious. One possibility would be to allow the viewer to reposition the pie chart's slices to enable more direct comparison between slices and accumulation of values over a number of slices. However, even this simple example points out the quickly increasing complexity of analysis if the analysis requires the combination of components localized in different parts of the visualization. Despite the common awareness of this problem, the efficiency of visualizations is rarely discussed explicitly on a theoretical basis reviewing the analyst's anticipated reasoning procedures. The common approach to the assessment of visualization effectiveness and efficiency is to conduct user studies. Yet, the validation by typically small scale user studies can hardly account for the optimality of the visualization. At best, those studies show that a well-defined and typically small group of participants performs a given task more or less well. Surprisingly, complete descriptions of the anticipated reasoning involved with the solution of tasks are quite rare in the literature. Without such a clear description of the anticipated reasoning structure to be applied by the viewer to interpret the visualization, it is hard to predict the complexity for the viewer to solve a given task with a given visualization. As a consequence, it is considerably hard to assess whether some visualization technique can be applied to a different problem than the one it has originally been developed for.

1.5.6 Uncertainty

Limited measurement precision is a common source for uncertainty in the data. This type of uncertainty typically takes the forms of tolerances and measurement error and is commonly characterized by an interval indicating the range of possible values around the measured value. In multiple dimensions, each dimension has its own interval width. This interval width is independent of statistic effects and is therefore usually fixed. Due to the nature of this type of uncertainty, effects involving the passing of thresholds may occur slightly sooner or later than suggested by the data. Nevertheless, no matter where exactly within the interval an event occurs, the resulting effects – and thus their interpretation – are the same. A dependency between exact value and occurring event would

require the kind of statistical analysis that is performed in ensemble visualization. However, to evaluate the observed scenario qualitatively, the measurement error is only of minor relevance as long as no value within the tolerance interval exceeds the threshold. In other words: As long as all values within the intervals' boundaries show the same behavior, a single or few representatives are sufficient to completely describe the observation on a qualitative level. For example, even the worst, average, and best case of a path of an asteroid crossing the course of Earth can be depicted as simple lines as long as no other uncertainty than a tolerance is involved. Qualitatively, the line will cross the Earth's course in all cases. The uncertainty only needs to be made explicit if the intervals indicate that a collision cannot be ruled out. Even then, explicit depiction of uncertainty would only be necessary close to where a collision can be expected. For the rest of the asteroid's path, the uncertainty does not add information relevant for the potential collision event. In conclusion, from the qualitative perspective, measurement error and tolerance only become relevant if they indicate multiple alternative interpretations of the results. Even then, it is sufficient to show the uncertainty only in those regions where it actually makes a difference.

Incomplete or erroneous input data can be considered to introduce uncertainty in the missing or wrong entries taking the form of local discontinuities or holes in the data. Especially if this problem only affects a single dimension in a vector, ignoring the whole vector potentially results in a loss of important information. On the other hand, filling the hole or replacing an obviously erroneous value introduces an estimation error. Many algorithms are only stable if there are no missing data entries. The uncertainty here results from the problem that if the exact values are not known, the estimation can be arbitrarily wrong. While the assumptions being made to generate the estimation typically yield sound results, especially where sampling is sparse the obtained results cannot be trusted with perfect confidence. At the same time, it is hard to estimate the potential error if there is no additional information available for the estimated data. As a consequence of this uncertainty, the results obtained from analysis are unreliable if conclusions are drawn from within regions affected by such an estimation. Highlighting the corresponding regions in the visualization indicates them as uncertain and therefore less trustworthy.

Combinations of these effects occur when approximating or interpolating continuous fields based on discrete samples. Even if theoretically the sampling is dense enough to perfectly reconstruct the continuous signal, the estimate will be uncertain due to the influence of measurement error on the sampled data. For example, depending on the data and the estimation method, multiple stationary values of some function will be introduced where there should be only one such value. Yet, from the qualitative perspective, the relevance of this uncertainty depends on the analysis situation and task. If the exact position of a stationary value is not important, the uncertainty is not relevant. If instead, for example, potential collisions between two objects need to be analyzed, the effects of an uncertain shape of each estimated object and the sampling tolerance accumulate. The prediction is not perfectly reliable due to the estimation and the tolerances might indicate possible collisions at different places than the measurement may suggest. Again, combining these uncertainties and highlighting them only in those regions where the prediction is uncertain or a critical event like the collision occurs guides the attention towards the important information.

1.5.7 Subjectivity and Provenance

Probably the most important part of data analysis is the actual process of drawing conclusions resulting in the construction of a model explaining the observation. Assumptions, presuppositions, and assertions are often implicit and not directly communicated or documented. Yet, their influence on the analysis result is decisive. Especially in interactive analysis setups involving human interpretation of observations, documentation of results often does not cover the deduction steps taken to obtain them. The major problem here is that all data analysis is inherently subjective. There is nothing like neutral or objective data because data alone do not provide an interpretation by themselves. Thus, the interpretation is either the result of the analysis conducted by the viewer or hardcoded into the visualization to be read off directly from the display. In either case, it is important to document the provenance information for a given interpretation such that the reasoning behind the interpretation can be reconstructed and verified. As a result of this inherent subjectivity, the conclusions drawn from visualizations can change drastically with different contexts. As a simple example consider a bar chart showing the calories of certain dishes from which a diet is to design for some patient. If the task is to evaluate which is the best dish for the patient, the choice will be different for a patient suffering from obesity than for a patient suffering from starvation. As another example, consider a data set of performance values measured for a number of entities capable of performing some task. In this example, a number of entites performes significantly less than the others. The available options are a relatively low cost solution removing the weak performing entities and replacing them by the other type or to apply costly measures to attempt to increase the performance of the weaker group. Given only this information without further context, the solution appears to be quite straight forward. For a number of machines in a production complex, replacement might actually be a good solution. If, instead, the performance data was gathered for the workers rather than the machines or for a set of school children, the decision is likely to be entirely different even though the raw, uninterpreted numerical data is exactly the same. In order to reproduce the findings and the external knowledge applied by the analyst to draw the conclusions resulting in those findings, careful documentation of provenance information is crucial. This provenance information is also a neccessary prerequisite for the reconstruction of an analysis result by a different user who might only be looking at the data visualization but is not informed about the context.

1.5.8 Contingency, Coherence, and Emergence of Insight

Contingency of information as a qualitative attribute is not necessarily reflected inside the data context. Figure 1.2 shows an example for such a situation. The images have been transformed until the data alone does not allow recognizing them as faces. The lack of contingency is in this case compensated by the human perception's closure capabilities.

The example illustrates that insight is not some set of objects in data that

Figure 1.2: *Example data from Mooney's visual closure experiment [94]. The pictures have been manipulated until a viewer without knowledge about characteristics of human facial expressions cannot identify the shapes clearly as faces. Mooney's experiment shows that viewers are capable of mapping their mental image of faces to the shapes, enabling them to correctly determine estimates of age, sex, and even the emotion shown by the facial expression. The performance in this closure is almost equal over a wide range of ages. Contingency of information in visual data analysis is thus not limited by what is actually depicted but can also be achieved by projecting coherent structures from the viewer's mental model to the displayed information – an aspect commonly not covered by quantitative characterizations of the information conveyed by visualization.*

can be extracted or mined in some way. Instead, it is an emergent property – a qualitative interpretation of the structure of relations between different objects in the data. Emergent properties are attributd to the observation made in the data in its entirity and cannot be found in single elements or explained as a combination of components. Thereby, emergent properties are properties whose encoding requires combining outside general context information and data with respect to the observed scaling. Consequently, they require qualitative analysis of data relations and general context and cannot be captured by the local considerations analyzing single data items or local patterns under purely quantitative considerations. Beyond that, coherent structures in the data indicate the contingency of implications or dependencies. Reasoning about the data in order to infer complex information with respect to the general context goes beyond what is explicitly displayed or measured. Hence, information entropy or other statistical quantitative estimates do not capture actual contingency or emergent attributes.

1.5.9 The Need for an Additional Perspective

Despite its inherent limitations, the importance of quantitative data analysis should be no means be degraded. Its strengths render it suitable for a wide range of tasks, including, but not limited to, clustering and classification, direct representation of raw data or derived measures, or statistical information. Among the applications are the identification of structure and patterns or other interesting data subsets or the estimation of an observation's significance. In general, quantitative analysis performs well where the information can be read directly from the data. Where this is not the case, an additional perspective is needed to capture those properties of the data analysis result that are obtained

from analytical reasoning and interpretation. To install a holistic perspective on reasoning about visualization, the scope of discussion has to be extended by the reflection of the analysis process itself. Not only the presentation of data is important but also how well this presentation serves the actual analysis and how it supports the emergence of insight. Such an extension requires the explicit consideration of the qualitative aspects of data analysis.

Concerning the reasoning with visualizations, the sources of subjectivity in data analysis need to be identified and included into the discussion along with the predefined interpretations of displayed artifacts and structures determined by the visualization's designer. Towards a discussion what kind of conclusions can and should be drawn from reasoning about the visualization, a stronger focus on the semantics of graphical displays is required. Displays and interaction should be provided according to the intended purpose for reasoning rather than due to the shape of the data. Ideally, proper modelling of anticipated reasoning strategies should allow the exclusion of potentially misleading design decisions at an early stage during the design process.

To make analysis results comparable and reproducible, better documentation of the actual analysis process is needed. This kind of insight provenance needs to capture not only the data an insight has been derived from but also the outside knowledge applied to draw the conclusions leading to the insight. An explicit discussion of the reasoning process and the chains of consecutive conclusions followed by analysts also allows to evaluate the anticipated reasoning against the reasoning actually applied by analysts working with the visualization and to refine the visualization accordingly where the workflows differ. Provenance information can also be applied to develop novel workflows based on the knowledge about well-established reasoning strategies or validating and reusing insights already found by other analysis in collaborative setups.

The complexity of analysis should be assessed from a holistic perspectve including both reading the graphical display and reasoning about the recognized artifacts and structures. Towards higher efficiency for reasoning, mechanisms are needed to properly embed the visualization into the analysis context by adapting the workflows and the graphical representation to the reaoning strategies expected to be applied by the analysts working with the visualization. This requires an extension of the existing methods of predicting the complexity of visualizations by an assessment of the reasoning complexity as the qualitative component of the data interpretation and analysis proces.

1.6 Qualitative Visual Analysis

The requirements identified for an extension of quantitative data analysis by qualitative aspects aim at the installment of a holistic perspective on the visualization-based data analysis process. All of the limitations of quantitative analysis discussed above are concerned with aspects of reasoning about the data. To clearly distinguish them from the quantitative aspects related directly to the data, the following definition should be applied:

Definition: Qualitative Visual Analysis

Qualitative Visual Analysis captures all aspects of the visual data analysis process concerned with the interpretation of data, the provenance of insight including its derivation from knowledge outside the data, and the complexity of reasoning about the graphical data representation.

To enable the discussion of interpretations, qualitative visual analysis needs to explicitly model the semantics of artifacts and structures in the graphical display. The documentation of insight provenance requires a formal representation of the viewer's mental model of the visualization, including outside knowledge and the applied reasoning structure. Proper methods for the assessment of complexity of drawing conclusions within such a structure is required to enable the optimization of graphical data representations and provided interaction methods towards efficient reasoning about the data.

Qualitative visual analysis captures aspects of the analysis process that are usually not directly supported by the data but need to be derived from data interpretations. The explanation of phenomena ideally directly carries over to comparable data from another source. However, just like purely quantitative treatment of data is not sufficient to explain observations and phenomena due to the lack of interpretation, purely qualitative models only explain abstract phenomena without an obvious connection to evidence supporting this explanation. Qualitative visual analysis therefore is by no means a replacement of quantitative techniques. It is an addition, broadening the perspective towards a more holistic view on the analysis process, encouraging the consideration of a "big picture" when analyzing data or designing data analysis applications.

It is important to emphasize that being primarily concerned with those properties of data that cannot be read off or computed directly does not mean to introduce ambiguity or arbitrariness. Interpretations can be subjective, especially when made by human analysts. Yet, they are still bound to the context of the data, the task, and the domain. Drawing conclusions requires the application of logics. Where quantitative analysis considers the measurable and quantifiable properties of data, qualitative visual analysis is concerned with the deduction, derivation, induction, and inference of information. In this sense, qualitative visual analysis is an additional perspective on visualization. Focusing on the reasoning processes applied to obtain insight from data visualization, it asks how these processes can be optimally supported – by reflecting the domain-specific interpretation and reasoning techniques and proper documentation of conclusion provenance to foster the emergence of insight. Interpretation, provenance, and emergence all are valuable fields of study in their own right. While there is a growing interest in these topics, they need to be combined to reach their full potential. Qualitative Visual Analysis aims to provide the conceptual framework to add the explicit consideration of data relations and dependencies to the quantitative approach based on structure and values. Although it is neither a design paradigm nor an analysis technique, shifting the discussion of visualization towards a more holistic perspective has certain implications on the design and analysis processes that can be leveraged to derive powerful analysis workflows.

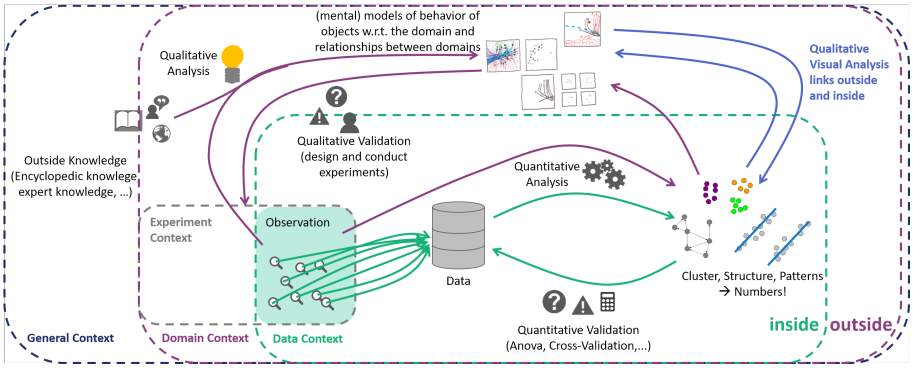


Figure 1.3: *Data analysis and context. The traditional evaluation of experiments (purple arrows) analyzes the observation either by quantitative means or by qualitative reasoning towards a model explaining the observation. Quantitative analysis yields descriptive models that are included into the qualitative consideration as the objects of discourse. Qualitative analysis may apply additional knowledge that might even be rooted outside the domain. It yields an explanatory model explaining the observation from a behavioral perspective. Consequently, its validation requires the design of new experiments against whose outcomes the model's prediction is to be compared. Data analysis (green arrows) instead is performed only on a subset of the actual observation, namely the sample obtained from data acquisition. Within the data context, quantitative analysis yields describing models similar to the classical analysis process. These results can be validated against the data by statistical techniques. However, being of purely descriptive nature, these models cannot predict the outcome of other experiments and thus can only reveal knowledge about the sample but not about the complete observation or even the experiment. Just like in the classical workflow, deduction of knowledge about the experiment – and hence the domain – necessarily requires an analyst interpreting the data and the models obtained from quantitative analysis from outside the data context. The central aim of qualitative visual analysis is to support this process by establishing an explicit link between the descriptive models inside and the behavioral model outside the data context.*

1.7 Qualitative Visual Analysis in the Reasoning Process

To understand the interplay between the different levels of context in data analysis, it is necessary to reflect the very process of analysis itself. The model illustrated in Figure 1.3, serves as a working hypothesis the remainder of this chapter is based on. For each level of context, its respective influence on the analysis result is discussed along the air-traffic surveillance example shown in Figure 1.4. Concentrating mainly on the practice aspects, some of the findings may appear rather trivial from a theoretical perspective. Yet, this impression is deceiving. The example has been chosen since the discussed factors influence its graphical appearance drastically. Although the visual effect will be much more subtle in many applications, the influence on the analysis results is still just as decisive.

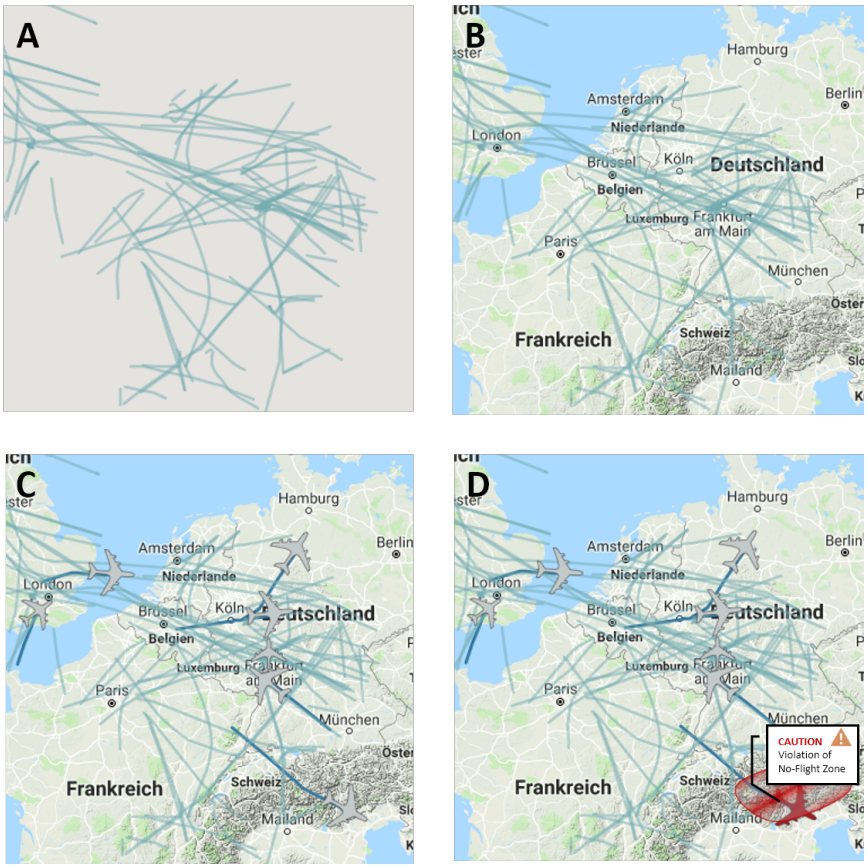


Figure 1.4: *Additional context changes the interpretation. Without any further context, any conjecture about the meaning of a set of curves would be purely speculative (A). Adding domain context in terms of geospatial (B) and semantic (C) information reveals that the paths are air-traffic routes over central Europe and the visualization shows airplanes traveling them. Let the task be to monitor the air traffic for critical deviations from the displayed routes. Without the task context, this involves active comparison of the airplanes' actual paths to the routes. Embedding this additional task context into the visualization, the insight about a violation is immediately emergent (D). Interpretations with respect to the general context influence the analysis performance decisively. These qualitative aspects should therefore be considered carefully and thoroughly and the general mechanisms underlying them should be discussed explicitly.*

Data is always sampled from an observation. Sampling is performed under certain conditions which for the purpose of this discussion are summarized as the context of an experiment. Experiment here refers to every process resulting in an observation from which data can be sampled. Data only samples parts of the observation and only reflects the observation itself. Sometimes it contains additional parameters but usually not the consideration behind the experiment's design. Hence, the data will typically not describe the system completely. In this sense, data is a snippet taken from the inside view of an observation, ignorant

of further knowledge of the world surrounding it.

For this discussion, the notion of context assigns every object of discourse to the most local applying context. For example, every result obtained from performing analysis or executing operations only on the data remains completely inside a data context. Hence quantitative analysis and validation are completely contained inside the data context as long as they do not incorporate any information that is not part of the data. This also applies if the algorithms and procedures take into account additional knowledge about the observation, since the observation is also completely inside the data context. To see the importance of this inclusion, consider the example shown in Figure 1.4 (A). Without further information, this data can literally be anything – given only the data one cannot distinguish a flock of birds from a set of fathoms composing a fabric. Adding the subset of the experiment context describing the actual observation it is revealed that these paths are actually trajectories over central Europe (B). Still, this does not reflect an analysis question. Adding the analysis context, it turns out that the visualization is about airplane trajectories to be monitored (C). This is the minimum of information required to perform analysis. Depending on education, cultural background, personal motivation, and other factors contributing to the viewer's background user context, a skilled analyst can now detect that the airplane in the southeastern corner of the displayed region deviates from its intended flight path. However, this information has been inferred from considerations that are not part of the data. The evaluation of trajectories with respect to the flight routes and the identification of a deviation as an air traffic anomaly requires nontrivial domain-knowledge. Visualization can highlight deviation and additional domain context like no-flight zones (D). Note that automatic classification of anomalies also involves outside knowledge that has been implemented by the programmer. The inference rules applied to interpret the data are not part of the data itself.

Indeed, the consideration of information outside the data context is necessary for reasoning. Cleveland and McGill define the nonmeasurable outside features as qualitative information and already account that data analysis concerns learning both quantitative and qualitative information [34]. Quantitative considerations can be performed completely inside the data context. User studies reveal that embedding the results of quantitative considerations into qualitative considerations to infer domain insight is substantially harder for viewers than reading information off the display [62]. Trafton et al. describe three different kinds of insight to be obtained from data visualization, requiring increasingly complex reasoning [139, 140]. One category is insight about the visualization which is basically understanding the data's visual representation. The second category reveals insight about the data such as relations, patterns, or trends.The third and most abstract category is concerned with insight about the analysis domain, answering the analysis question by inferring the inquired information from the depiction. In this formulation the three categories match the three-level model proposed by Ware [148]. Regarding the mapping of high-level tasks to these categories, Nazemi and Kohlhammer associate the first level with search tasks, the second with exploration, and the third with the actual analysis process [98]. Where the first and second category are concerned, Smuc et al. report results from a user study revealing that while insight about the data can only be obtained after insight about the visualization, viewers only need to under-

stand those parts of the visualization that are relevant for the insights they are interested to obtain [132]. This observation emphasizes Kosslyn's remark on the importance of pragmatics in visualization: visualization performance depends on its purpose [82]. Remarkably, all of these models have in common that reasoning about the data with respect to outside information is regarded as the most complex part of the analysis process. These considerations complete the model shown in Figure 1.3. The green arrows indicate information flows in data analysis and the purple ones indicate the classical observation-evaluation known from science and engineering. Note that combining these two workflows actually yields a structure similar to Daniel Keim's visual analytics pipeline [74]. Still, the link between the models obtained from quantitative analysis and the behavior-focused explanations resulting from qualitative considerations is only established by the viewer since it requires the extraction of information from inside the data context to its outside.

Since, being a subsample of the actual observation, data is typically ignorant of the conditions that lead to its creation, it is reasonable to consider two systems of reference when reasoning about data interpretation and analysis. These systems are the data context and the domain context it is embedded in – the central components of the analysis model shown in Figure 1.3. Note that this is actually common best practice: together, the purple and green arrows in Figure 1.3 are actually just a slight alteration of Daniel Keim's visual analytics pipeline [74]. Reflecting on the aforementioned limitations of the currently predominant analysis approaches focusing mostly on quantitative considerations, the key idea of qualitative visual analysis is to establish an explicit link between the inside and the outside of the data context by adding a qualitative component to the analysis workflow.

The result of qualitative analysis is a descriptive model of the observation, reflecting the data's behavior. It is based on rules and predictions rather than on measured facts and observations. The insights obtained in such a model define an adequate and accurate description of the analyzed data and its interpretation with respect to the data's inherent context, the analysis context determined by the task, the analyst's individual expertise, and the general domain background. Being derived from data, this model cannot be verified against the data itself. Hence, attempts to reflect on the fundamental truth underlying such a model from within the model are futile. As a consequence, the evaluation of a model obtained from qualitative analysis involves the development of a qualitative meta-model as a necessary step. In the analysis model shown in Figure 1.3, this is included as the design of new experiments. Note that this includes logical (and thus qualitative) reasoning about the soundness of conducted analysis steps. To support this verification, qualitative analysis is not only concerned with the interpretation of data and the emerging insights, but also with capturing information provenance and documenting the analysis process.

Summarizing these considerations, there are three substantial observations being made. First, the interpretation of data depends on the context. Data that is free of context is also free of meaning. To support reasoning, the representation should also consider the domain and the task. This requires the extraction of information from inside the data context to the domain context's qualitative behavioral models and thus emphasizes the importance of qualitative consider-

ations for the analysis process. The model building process also reveals that the essence of qualitative consideration is not perception but reasoning about the data. Since qualitative considerations take place in the domain context, the generation of knowledge or decision competence – and, hence, any data analysis – requires the consideration of outside information. The inside-outside principle specified in Section 1.5.1 hence is a central property of qualitative visual analysis.

1.8 Core Research Questions Inspired by Qualitative Visual Analysis

Summarizing the discussion, it is evident that some aspects influencing the result of data analysis are not well reflected in today's dominant data-centric quantitative approach to data analysis. Relying on only those aspects of data that are measureable or otherwise expressible in a quantifiable way, quantitative data analysis is inherently limited by data's structure and values. Although this kind of analysis is a purely empirical evaluation of data organized in a predefined fixed structure, the impression of objectivity is misleading. Without its context and an appropriate scale level, quantitative data is meaningless. However, context is not an intrinsic property of the data. The interpretation of data will thus differ between contexts. Similarly, the capability to detect and understand observations in the visualization is an individual skill and thus every analysis result subjective. Of course, due to conventions within the respective domains, analysis from the same domain will come to similar conclusions when investigating the same data. Still, the differences between their individual approaches to obtain this information and the differences in the actual analysis results need to be documented to enable comparison of results and explanation of deviations between obtained models. Being an emergent property, insight itself is inherently subjective. If visualization is meant to support insight, one aspect central to a theory of visualization is how insight in terms of domain understanding can be obtained from data analysis. In conclusion, it is apparent that quantitative analysis alone will not answer this question but needs to be extented by qualitative considerations. Qualitative visual analysis provides this extension.

Qualitative visual analysis asks for the interpretation of artifacts and structures in graphical displays, the reasoning strategies applied to draw conclusions and obtain insights from the graphical representation, and the complexity of those reasoning processes. Answering those central research questions requires systems that adequately capture the analysis domain's semantics, document the provenance of insight including its roots in knowledge external to the data, and foster the emergence of insight by optimally fitting the display and interaction to the reasoning mechanisms anticipated to be applied by viewers. This extension to the data-centric quantitative perspective on analysis finds three major fields of application: validation and theoretical evaluation of visualization systems, the specification of efficient workflows, and finding visualization designs that optimally support the reasoning process.

Concerning validation and evaluation, an explicit discussion of semantics mitigates many of the well-known problems with purely quantitative descriptions, including scale and unit problems. For example, exploring the semantics of a distance or similarity measure in order to assign proper meaning to the closeness of elements in the depiction provides a better assessment of visualization plausibility by preventing ambiguity in the interpretation of distance and hence increases the validity of analysis results. Such a model is introduced in chapter 2.

Workflows inspired by qualitative visual analysis focus on the reasoning process rather than on the data features. To enable the specification of such workflows, a decription of the reasoning process is needed that captures not only the data transformations applied by the visualization itself and the analyst's interaction but also the viewer's interpretations made in the context of the analysis question and the viewer's outside knowledge about the domain. Such models also allow capturing insight provenance incorporating the documentation of reasoning as an integral part into the analysis process. In chapter 3, analysis workflows derived from qualitative visual analysis are proposed.

The explicit consideration of reasoning in the assessment of visualization complexity results in increased accuracy in the tailoring of visualization designs towards the needs of the domain, the analysis application, and the individual analyst. Proper models allow to even predict the efficiency of visualizations with respect to an anticipated reasoning strategy. Design aspects of qualitative visual analysis are discussed in chapter 4.

Core References

- B. Karer, H. Hagen, and D. J. Lehmann: Insight Beyond Numbers. In *IEEE Transactions on Visualization and Computer Graphics*. IEEE, in preparation.

Chapter 2

Reasoning and Interpretations

Qualitative visual analysis requires a differentiation between what is perceived and interpreted and what is investigated and analyzed. Current models of visualization and visual data analysis workflows usually either focus on the phenomenon being investigated or on its graphical representation. Data is only a sample of the actual observation and studying visualizations means to study representations of the data rather than the data itself. Consequently, it cannot be expected that a mental model describing the analysis results automatically yields a correct description of the investigated domain. This chapter discusses the analysis of data utilizing visualizations as encodings of information is investigated. A model for the analysis process is proposed capturing the representation of data in terms of visual encodings and the process of interpreting those encodings in order to obtain information about the domain being investigated. Further formalization yields a theoretical framework allowing to study how different viewers and analysts read and interpret a given visualization.

To close the gap between reasoning about the visualization and inferring insight about the domain, semantic aggregation is introduced as a theoretical concept formalizing the understanding of visualizations as the inferece of semantics by the interpretation of structures displayed in the visualization. Those structures can be as simple as individual graphical elements or as complex as large contextual views. The consecutive execution of two formal automata enables to compute a prediction of the possible and applicable interpretations given a visualization and a reasoning system anticipated to be applied by the viewer. Visualizations and their interpretations can thereby be formalized in a graph-based representation being interpretable by both humans and machines. Extending the descriptive scope of the common cost-based approach to determine the complexity of visual data analysis, determining the complexity of those formal languages additionally considers the complexity of the reasoning process. The extended complexity model predicts findings reported in literature on user studies on the complexity of reasoning about visual data representations. The qualitative visual analysis cycle and the concept graph introduced in this chaper therefore are not only descriptive models to reason about the formation of mental models from visualizations but can also be applied as generative models allowing to tweak the design of visualizations towards optimal support of antic-

ipated reasoning chains.

2.1 On the Qualitative Visual Analysis Process

Every information to be found in graphical representations is subject to the viewer's ability to perceive, recognize, and interpret what is depicted. Although this question is of quite philosophical nature, it also has practical implications. Viewers with different interests and backgrounds will interpret the same data differently and draw different conclusions. Conceptual models of visualization and interaction usually concentrate on one of two foci: reading and understanding visualization for reasoning about the represented data or interaction to navigate the visualization and steer the content of a given view. Models taking a holistic perspective on both aspects typically do not provide formal descriptions for the respective model's individual elements or the transition between them. Another common observation among theoretical descriptions of working with visualization and visual analytics applications is that the difference between the visual data representation and the investigated domain is only considered implicitly. Notably this also applies to more formal models of visualization content and the encoding of messages to be communicated by the visualization and also to models for the complexity of working with visualizations. Consequently, every insight into the visualization is considered an insight into the data domain – a rather bold statement considering that the visualization is only a representation of data and even more so considering that visualization rarely shows raw data and and that the visualization in fact is a data transformation itself. Of course, there are certain requirements to this representation, for example that it should not distort the information to be expressed by the visualization or that it should attempt to avoid misunderstandings leading to false conclusions. However, such requirements are hard to prove formally, especially if the information to be obtained from the visualization is not to be found directly in the graphical representation but requires reasoning about what is to be seen. The more complex this reasoning, the harder it is to determine whether the visualization suffices those requirements to the quality of the data representation. Consequently, a thorough discussion of the complexity and correctness of visualization requires a detailed formal treatment of the qualitative aspects of visualization and visual analytics.

For a theoretical framework enabling this kind of reasoning about visualization, a model is needed that is capable of describing the whole process of generating the graphical representation of data, reading and understanding this representation, inferring an understanding of the investigated data domain from this understanding and possibly generating new questions to the domain from this understanding. Towards such a theoretical framework, this chapter introduces a model for the visualization and qualitative visual analysis process. The qualitative visual analysis cycle introduced in this chapter considers the whole cycle of sampling data from a domain of interest with regards to a certain analysis question, visualizing this data, reading and understanding the messages to be conveyed by the different structures in the visualization, interpreting those messages and reasoning about them, and generating a mental model of the

viewer's understanding of the visualization from which insight into the domain of interest is to be derived. For the transition between individual steps, the model relies on the established theory of generating visualizations from data by expressing them in terms of graphical languages and applying information theory for the description of structures to be found and interpreted when viewing the visualization and reasoning about its content.

The second central contribution of this chapter is semantic aggregation, a formalism refining the models developed for the qualitative visual analysis cycle enabling the computation of semantics applicable to the displayed artifacts and structures in a visualization with respect to a viewer's anticipated reasoning structures. Semantic aggregation assumes that the meaning of components in graphical displays is not static but assigned dynamically within the viewer's mental model of the visualization in order to satisfy the requirements of reasoning structures to be applied to the graphical display's interpretation. This flexible reinterpretation mechanism is combined with the idea of extracting interpretations applicable to the actually observed situation from a universe of possible interpretations for hypothetically possible observations. Combined with the workflows and ideas of the qualitative visual analysis cycle, semantic aggregation for example allows to asses whether a viewer will be capable of reading the depiction correctly, to predict the kind of conclusions an analyst will draw from the display, and the complexity of the qualitative reasoning process yielding those conclusions.

2.2 State of the Art

Graphical displays communicate information by encoding them into structures that can be perceived by the human visual system. To be recognized and understood by the viewer, these structures need to match some pattern in a repository of information on how to interpret visual stimuli. In the case of visualization, these visual stimuli encode data generated from some commensurable phenomenon, often but not necessarily a sample obtained from simulation or measurement in an experiment. The assignment of meaning in human visual information processing is performed in short-term memory by mapping the observation to learned structures in long-term memory [82]. In general, deriving global information such as trends and patterns from the data triggers different and more complex cognitive processes in the human brain and is harder than locating information that can simply be read off [62]. This is already reflected in Bertin's distinction between presentation and layout and Rensink's triadic architecture for the description of visual data representations [16, 117]. The three-level model is supported by experimental data gathered by Ratwani, Trafton, and their collegues in a series of experiments [116, 139, 140]. In a number of user studies on information visualization, they found three categories of insight to be obtained according to the difficulty users experienced in obtaining these insights. The first level is concerned with understanding the presentation itself. The second level reveals relations, patterns, and trends depicted in the visualization and – if the depiction reflects the data correctly – already provides insights into the phenomenon the data describes. The third category is

the inference of additional information based on reasoning about the obtained information with the help of user knowledge external to the visualization. In this formulation, the three categories match Ware's model [148]. From a task perspective, Nazemi and Kohlhammer associate the first category with search tasks, the second with exploration, and the third with the actual analysis process [99]. A particularly interesting finding in this direction has been reported by Smuc et al. who found in a user study regarding the first two categories that although insight about the data could only be generated after the viewers obtained insight about the visualization, they only needed to understand those parts of the visualization they actually applied for their reasoning [132].

In their analysis of visualization's purpose, Chen et al. qoute a collection of different characterizations of what actually defines visualization is towards finding an answer on what insight actually is [27]. The definition they develop agrees with their references in that visualization is used to infer information about some more or less abstract structure, either the data or the phenomenon it samples. Yet, the factors they derive as the variables governing the time to perform a visualization task also imply that this information is directly found. The observation of Smuc et al. contradicts this assumption. If users were capable of directly reasoning about the phenomenon the visualized data has been generated from, they would not need to understand the visualization before this reasoning could be performed. This is also supported by Petre and Green who report evidence that understanding visualization is unlikely a native ability of humans but can be learned [107]. Findings like this indicate that cognitive load and other human factors might actually be the expressions of the process underlying the understanding of graphical displays and the reasoning about them. Hence, there must be a collection of mappings between the graphical display, how it is understood by a viewer, the viewer's toolset for reasoning about the presentation, and – in the case of visualization for analysis purposes – the data and phenomenon being represented by the graphical display. Towards this direction, Vickers et al. propose a theoretical framework for the process of reasoning with visualizations based on category theory and semiotics [144]. Explicitly taking into account perceptive and cognitive abilities as well as knowledge, they are capable of describing a number of effects commonly observed in visualization applications. Among these effects are the possibilities that two viewers with different reasoning strategies will interpret the same visualization entirely differently and come to entirely different conclusions even if they read it exactly the same way and that - likewise, two viewers might read the same visualization entirely differently but still come to the same conclusion.

Information theory has been discussed as a model to mediate the transport of data and information between the graphical display and the viewer [29]. Investigating the definition of data, information, and knowledge in visualization, Chen et al. contribute a definition attempt motivated by perceptual and cognitive spaces by a model defining data, information, and knowledge to be different types of data in the computational domain [26]. Yet, information theory only covers the transport of information and is not directly concerned with its processing. As a consequence modeling the actual reasoning process requires a higher-level structure. Such a structure is provided by Keim's visual analytics models and the knowledge generation cycle proposed by Sacha et al. as a reasoning model for visual analytics [120]. However, these models typically as-

sume a unidirectional flow of information, accumulating the knowledge obtained from data analysis in the user's mental model. Yet, the experimental results referenced above actually contradict such an assumption.

Although this Thesis is mainly concerned with modeling reasoning processes, this requires a model of visualization structure and content to define the structures this reasoning is actually performed on. Theory on the design and formation of visualizations can roughly be categorized into two groups. Low-level approaches operate close to the data. Work in this direction often assesses the quality or performance of a visualization with respect to certain criteria. Typically they apply algebraic, information theoretic, or other formal constructions to describe visual encodings. Examples date back to Mackinlay's presentation tool and Wilkinson's grammar of graphics [92,150]. More recent representatives of this category are the visual embedding of Demiralp et al. [40], the algebraic design process by Kindlmann and Sheidegger [76], and Tominski's event-based approach [137]. The theory presented in this Thesis follows this direction by formalizing a model of information content and how this information is mapped to the graphical elements composing the visualization. Otto and Schumann propose a model similar to the one presented here in that it attempts to combine data wrapping them into information objects [104]. Doleisch et al. develop a feature description language to interactively define features of high-dimensional data based on the user's interest [43]. The combination of these two perspectives is a key aspect of the theory proposed below. A dualism is established between the data encoded by graphical elements and its interpretations. A strong focus on the semantics of graphical elements also allows weighting their presentation based on their relevance as proposed by Kosara et al. [81]. Similarly, graphical elements can be emphasized due to different semantic contexts. This emphasis can, for example, be achieved by the definition of relevance functions [70]. Yet, the theoretical framework introduced below follows a different approach: Rather than defining relevance functions, it leverages its graph-based nature to formalize relevance by reachability in an ontology-like structure.

The second direction of theory is the development of general frameworks of visualization design. Typical representatives of this direction are high-level models that either aim to support the visualization expert directly by providing feedback or guidance (e.g. [5,125,135]) or model the design process as a whole (e.g. [96,135]). The theory proposed in this Thesis is more low-level but can also be applied as a design tool. Examples for possible applications are the identification of misconceptions prior to implementation or a formalized discussion of design ideas with other designers and application partners.

2.3 Approach

The discussion follows three steps. First, an abstract model is outlined explaining the formation of a mental model from exploring visualizations and performing tasks. It also describes the switching between exploration and task formation and execution. The model draws directly from existing work, combining descriptions of different aspects of the analysis process into a common model. The second step provides a formal foundation for the different steps and

the transitions between them. The formal theoretical treatment focuses on the messages being conveyed by the visualization and those processed by the viewer's cognition. The result is a theoretical underpinning for the abstract model developed in the first part of the discussion, formalizing the interpretations of data being displayed in graphical representations. This theoretical underpinning is finally reviewed and extended to a formalism for the assessment of possible interpretations and conclusions to be drawn given a visualization of a specific data set. This is achieved by introducing the idea of semantic aggregation describing a dynamic assignment of meaning to elements in the graphical display such that the same elements can contribute to multiple different interpretations. Formalizing this idea into a theoretical model in which interpretations can be computed yields the concept graph, a versatile formalism for the specification of mental models of visulizations.

2.4 Related Work

The discussion of the qualitative visual analysis cycle primarily draws directly from results discussed in the introduction of qualitative visual analysis. An exception here is the consideration of information foraging theory for the determination of an analysis strategy inspiring the model.

The formulation of semantic aggregation and the concept graph extend work on a graphical representation of information from semantically heterogeneous environments in order to display its several different possible interpretations [73]. This work is loosely inspired on Situation Semantics and Situation Theory as introduced by Barwise and Perry [8, 10]. Situation theory additionally introduces logics to allow reasoning within the system [9, 11]. However, for the sake of generality, no specific restrictions are made to the logics and formalism to be applied within the framework discussed below other than that properties and relations are described in terms of predicates and functions bound to variables specifying entities or observations. The design of the concept graph's graphical representation is based on VOWL, a graphical representation of the Web Ontology Language which is used to model ontologies in the semantic web [7, 14, 90, 100]. The semantic web has been applied in combination with reasoning frameworks (e.g. [63, 78, 105]). Yet, although both formalisms are in principle compatible with the ideas intoduced in this chapter, the discussion reveals that certain extensions and adaptations are necessary to capture the actual reasoning process from within the model rather than requiring the addition of outside structure.

2.5 A High-Level View on the Qualitative Visual Analysis Workflow

Qualitative visual analysis explicitly takes into account the interpretation and reasoning processes involved with the analysis of data. Existing models for the description of working with visualizations tend to focus on the problem of obtaining predetermined information from the visualization. This is the question

commonly asked when discussing visual search, the ability to find information in visualization, cost-benefit approaches to assess the complexity of visualization, or the role of interaction to solve a given task. More complex models, especially those on explorative visual analysis, tend to consider a cyclic process in which new questions are generated from the findings made before. The classical visual analysis pipeline already expresses this idea by a feedback arch ranging back from the obtained knowledge to the original data. However, this arc is widely ignored and there is a strong tendency to consider visual analytics as a form of knowledge extraction implying a unidirectional flow from data to knowledge along the pipeline. Models specifically treating the topic of explorative visual analytics typically consider an evolutionary knowledge model and consequently contain one or multiple loops from the model generation back to the visualization or the data. Evolutionary models automatically incorporate the feedback loop of visual analytics.

Following the more general approach, the discussion in this chapter takes over the idea of an evolutionary knowledge model. However, it extends this perspective by adding a clear adding a clear distinction of what is shown in the visualization and how this information is identified. On the one side, there is the visualization which is a structured representation of data sampled from some observation, for example the result of an experiment or a survey. The formation of the visualization and its content is based on quantitative analysis and includes the results of data aggregation and processing, especially those results obtained from automatic data analysis steps like clustering or classification. From the qualitative perspective, structures in the visualization are identified, interpreted and processed on the conceptual and semantic level to obtain a mental model of the visualization. This includes local reasoning about individual structures and global reasoning considering several structures and taking into account outside knowledge that is not shown in the visualization but part of the analyst's context knowledge. This clear structural decomposition of the syntax-oriented structural processing of data on the one side and semantics-oriented content-analysis on the other side provide a suitable basis for the formalization of processes in quantitative and qualitative data analysis. Indeed, this separation is well present in collaborative data analysis workflows and even in visualization design studies, where data preparation, aggregation, and visualization is performed by another group of experts than the actual data analysis.

2.5.1 Information Foraging as Analysis Strategy

Qualitative visual analysis assigns an active role to the analyst. Actively searching for information and reasoning about observations rather than only consuming the presentation implies the existence of a processing pipeline similar to the processing of data for the generation of the visualization. To enable reasoning, visualization needs to transform data into sets of messages that can be interpreted by the analyst to obtain a mental model which is then mapped back to insights into the domain from which the data has been sampled. Towards a better idea of the actual transitions between the individual steps in the process, the analyst's role needs to be reviewed.

Information Foraging is a theory predicting human behavior during the process of gathering information from abstract information representations or during general investigative procedures such as internet recherché [108, 109]. The theory is applied to predict the behavior of a human consumer of information in static information displays and also to predict the interaction conducted by a user of an interactive system. The classical example to describe the behavior predicted by information foraging theory is the behavior of a bird searching for food in various bushes growing berries. Depending on its experience and taste the bird will scout for bushes that provide a large amount of its preferred berries and especially for those nourishing the bird at best. Once a suitable bush is found, the bird will start to feed from this bush. Note that this bush is not necessarily optimal in any way, it is usually the first bush found and likely the easiest to reach. Due to the bird feeding from the bush, the support of berries gradually descends, making it harder for the bird to feed. At some point, the bird will again scout for bushes in the surrounding. At this point, the bird will start to evaluate whether the expected cost of further exploiting the current bush will exceed the cost of moving to another bush. If so, the bird will move on to the next bush. Information foraging theory is based on the observation that humans show similar strategies when working with information sources. For example, the theory explains the surprising observation that instead of using the native search function provided by an internet shop, some users prefer to switch back to the original search engine that brought them to the shopping portal as they find it more convenient to refine their search there. This behavior becomes more frequent the harder it is to find the online shop's search function.

Information foraging theory predicts that an analyst performing a task will first scout for a suitable portion of easily accessible data and attempt to extract as much information from there as possible. The information is already further processed by local reasoning. If the information is found to be insufficient for the solution of the task, the analyst will attempt to access different information. Depending on the task, different kinds of information are interesting. The analyst will attempt to access detail information if the task requires to do so. If the task is to obtain an overview, the analyst's focus will move between different bits of information more frequently. However, if the information still is insufficient for the solution of the task or if the analyst is convinced to be unable to extract further information from the current view navigating interaction will be applied to move on to another portion of the data. For tasks involving global reasoning, at least two different portions of the provided information need to be focused at and to be reasoned about in conjunction. Tasks can change on the fly meaning that even if the task is to obtain an overview of the data, some detail information might be consumed if it is relevant for the kind of overview the user is interested in. For example, rather than in the general connections in a public transportation network, the analyst might be interested in an overview over the connections between certain points at certain times, for example to transition between a hotel, a conference venue, an airport or railway station, and a place where a social event being part of the conference program is located. Note that the actual strategy executed by the analyst depends on various factors, only one of which being the task to be solved. Others are personal experience of the analyst, the shape of the display and the complexity to extract certain information, as well as the interaction functionality provided by the system being used.

The strategy followed in the example is based on the search for single-direction transport options between two points. Consequently, accessing a source of information means to obtain the set of connections between two points and then finding the optimal connections best suiting the expected time frame for the actual transit. The obtained result is a transit schedule with an optimal option and some alternatives for each transit to be made. In this specific example, it is assumed that the transit routes are served by different companies with different pricing systems. The prices are provided together with each line. Hence, the user's task changes frequently between identifying connections and comparing prices. For simplicity, it is assumed that no company offers a suitable pricing model for multiple transits such as time-based tickets. Notably, the changing task does not necessarily induce a change of the view. It only changes the focus on the information provided within the same view. Even for connections where lines of multiple companies are used, only the total amount to be paid is relevant for the optimization task of finding the best transportation. The question being more relevant here is whether it appears more convenient to switch lines, which means to buy a new ticket at the intermediate station, or to take the probably more expensive or slower line that does not require to switch lines during transit. Just like the task being performed switches frequently over time, the relevance and importance of different parts of the information does. Hence, the criteria for the evaluation of the amount of further information to be gathered from a certain source change along with the task. As a direct consequence, the analyst might frequently return to some source of information after having left it if the reason to leave was a change of interest due to a change of the task rather than a result of being unable to extract further information for the solution of the different tasks.

Different levels of tasks and subtasks might be described simultaneously in information foraging theory, where the same strategies applying for the solution of the more general task also are applied for solving individual subtasks. From this perspective, explorative analysis can be described as the task to obtain an overview or an impression of the data by aggregating the relevant information into a descriptive mental model of the observation domain. Tasks directed towards the answering of more or less precisely formulated questions or regarding hypotheses to be tested against the data then can be described as attempts to confirm or disprove assumed structures in this mental model. Where visualization is concerned, exploration and task-solving both require the translation from structure in the visualization to cognitive structures in the mental model, representing the interpretation of the structures being shown in the visualization in terms of their assumed relations and respective meaning. This necessity can be leveraged to map the information foraging strategy to the high-level qualitative visual analysis workflow described before in order to obtain a more fine-grained model.

2.5.2 The Qualitative Visual Analysis Cycle

Bringing together the user behavior predicted by information foraging theory with the decomposition of the qualitative visual analysis process into a structural and a semantics part results in the model model shown in Figure 2.1. To

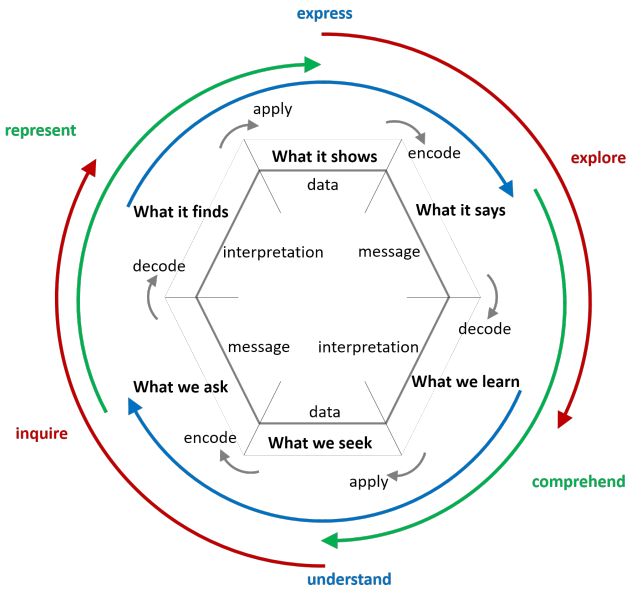


Figure 2.1: *An illustraion of qualitative ideas underlying the qualitative visual analysis cycle. The six different sections along the cycle denote the various states in the process. The arrows indicate the transition between the individual steps. Colored arrows categorize three pairs of antipodal transformations. The blue arrows indicate visualization and interpretation, the green arrows query and response, and the red arrows model explorative analysis. A closed circle can only be achieved by combining transitions from mulitple categories.*

enable the visual investigation of information for information foraging, a visual representation of the data has to be generated from data that in turn needs to be sampled from the domain of interest. Since the aim is to learn about the domain rather than the data or its representation, both data and visualization need to be considered as variables that can be changed if the task or the analysis require the analyst to do so. Although in many applications they will be fixed due to practical restrictions, especially the flexibility to change the visualization is sometimes important for the investigation of different aspects of the data. Since information foraging assumes that information is consumed in some way, the process of consummation also has to be modeled. Visualizations communicate information by forming structures and patterns in the graphical representation. Those structures and patterns can be read and interpreted by a viewer and generate a mental model of the graphical representation. Combining this mental model with outside knowledge about the general domain of interest, personal experience, or other forms of knowledge about information not contained in the visualization, an analyst is capable of enriching this mental model by further reasoning and ultimately enabled to map this mental model back to the original domain of interest in order to obtain insight about the domain rather than only about the data or its graphical representation.

Information foraging now implies different actions to be performed in order to work with visualizations along the identified steps. Interestingly, these actions form antipodal relations, each spanning three of the six steps and transforming the first to the third leveraging the intermediate step. The collection of transitions is indicated by the arrows in Figure 2.1. For example, the domain is represented by a visualization formed using the data sampled from it. Likewise, the mental model is formed by comprehending the messages to be conveyed by the visualization depending on which messages the viewer is able to perceive. This pair of relations describes the classical perspective on visualization generation and the analysis of a viewer's understanding of visualizations, including for example considerations on visual search. Interactive and explorative setups can be described by an antipodal pair of relations where the visualization is to be explored by perceiving the information being displayed in terms of the messages conveyed and the data being investigated is sampled from the domain based on the information inquired to express some part of the mental model, for example an hypothesis or a scenario. This kind of analysis is typical for what-if scenario analysis where different parameters affect the outcome of an experiment or simulation and hence need to be tested in order to find optimal results. Flood analysis for decision support is a typical example for this kind of application of visualization for decision support. The third pair of antipodal relations reflects exactly the high-level model of qualitative visual analysis proposed before. Data is expressed in terms of messages to be communicated to the user and the visualization is the medium enabling this communication. Understanding the messages perceived results in insight into the domain of interest if an analyst is capable to structure the perceived information in a mental model and to apply reasoning to map this mental model of the graphical representation back to the actual domain model.

Interestingly, each of the three antipodal transition pairs leave out an in-between transition between two of the steps in the model. For the strategy suggested by information foraging theory, however, a full cycle is implied: Scouting

for information means finding the right data to sample and the right visualization to represent it. The visual data representation needs to be consumed and processed further to evaluate whether or not the focus should be changed at a given point in time. This loop, combining the transitions "represent" – "explore" – "understand", models the process of explorative data analysis and other tasks aiming at the generation or extension of the mental model. The other possible combination of three consecutive transitions, "inquire" – "express" – "comprehend", models the execution of tasks that can be formalized as tests of structures in the mental model against the data. An example for the former type would be the task to obtain an overview over the domain of interest. Another example would be to form a hypothesis based on the observation and by applying reasoning involving contextual knowledge from outside the visualization. Validating this hypothesis would be an example for the second type of task. Qualitative visual analysis aims to take a holistic perspective on the visual data analysis process, including the reasoning performed on the data. Consequently, a formal model of qualitative visual analysis should follow the closed loops of transitions between steps.

2.6 Towards A Formal Representation of Qualitative Visual Analysis

The qualitative visual analysis cycle as introduced above is only a high-level description of the analysis process. Being derived from information foraging, it reflects the expected behavior of an analyst navigating the data to interpret it and draw conclusions from it. By the explicit inclusion of an interpretation process generating a mental model of the visualization, it implements the inside-outside principle introduced in Chapter 1. For a model for the process of obtaining insight about the domain from data sampled from the domain, a more sophisticated theoretical framework is needed. Such a model is developed in the following.

2.6.1 Formalizing the Qualitative Visual Analysis Cycle

Before the individual steps and transitions in the qualitative visual analysis cycle can be discussed in detail, the overall process has to be formalized. Figure 2.2 shows the cycle applying the formalization introduced below. Where it is not stated otherwise, the formalism discussed in this chapter is applied consistently throughout the remainder of this Thesis.

Data is assumed to be a finite set $\mathbb{D}$ of qualitative or quantitative entities obtained from descriptions or measurement sampling some potentially infinite set of information $\mathbb{I}$. $\mathbb{I}$ describes commensurable observations and entities. It is given in terms of predicates and functions in logics adequate for the description of the domain of interest. Unary predicates and functions denote intensional properties of entities and observations. Intensional properties are those properties that describe an entity or observation independently of any other entity or observations. Higher arity either denotes extensional relations describing the

influence an entity or observation has on another entity or observation or the states of affairs between observations and entities within an observation. In this sense, $\mathbb{I}$ can be understood as a semantics model of the domain, modeling what is known about the domain and how this knowledge is to be interpreted and understood. To structure $\mathbb{I}$, the functions and predicates are bound to higher-order predicates modeling commensurable entities and observations. The valuation of variables for an entity or observation determines its state. To better distinguish between stateless and stateful objects of discourse, valuated entities are called are called objects, valuated observations are called situations, and their unvaluated prototypes are called types, following the notion introduced by situation semantics.

Observations and entities are mapped to data by a representation relation $\rho : \mathbb{I} \to \mathbb{D}$ establishing a partial map from the space $\mathbb{I}$ of semantic information to the space $\mathbb{D}$ of sampled data. More precisely, $\rho : \mathbb{I} \to \mathbb{D}$ is a partial map expressing sets of known facts about an entity or observation $I \in \mathbb{I}$ that is sampled by data $D \in \mathbb{D}$. Higher order elements like sets of entities or sets of observations are captured by establishing a surrounding observation featuring the information about the contained observations and entities. Likewise, sets of data sets again form data sets in $\mathbb{D}$. There is no further restriction to the shape of $\mathbb{D}$ other than that every element is in some way related to some information in $\mathbb{I}$. Because data analysis attempts to learn about $\mathbb{I}$ from $\mathbb{D}$, the relation ρ is generally not known completely. This does not only apply to the entities and observations but also their respective predicates and function. Nevertheless, ρ plays an important role in hypothesis tests where the hypothesis can be formalized as an extension of ρ yielding some hypothetical map ρ'. Because ρ' is a hypothetical map from the information to data, it can be invalidated by the data or shown to hold at least for the data.

In most cases, the partial inverse of ρ, the sampling relation $\sigma : \mathbb{D} \to \mathbb{I}$ is known much better. Yet, σ is commonly also only known only up to a certain extent, making it only a partial inverse of ρ. After all, σ maps the data to the information it encodes. Finding this information is the actual aim of the analysis. One aim of data analysis hence is to complete the knowledge about σ as best as possible. Because σ is a partial inverse of ρ, this automatically means to obtain knowledge about the snippet of the domain information that is represented by the data. The reconstruction of σ, however, only provides insight about the data but not directly about the domain. Reasoning about the domain, especially about general principles expected to hold there, hence in most cases requires additional knowledge outside the data. Visualization provides an analyst with an interface to the data. This interface allows the analyst to connect the information sampled by the data and encoded by the visual display with the available outside knowledge in order to draw conclusions about the domain.

A necessary precondition for such an application of information to problem solution is the characterization of the problem by collecting data about it and the identification of a mapping between data and a repository of known information. In the context of visual data analysis, this repository is commonly referred to as a viewer's mental model. The viewer's knowledge about how to read a visualization correctly would then be the availability of an interpretation relation

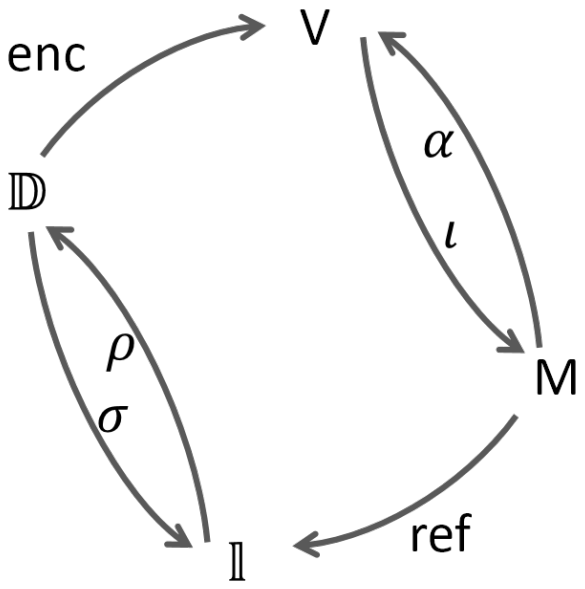


Figure 2.2: *A formalized version of the qualitative visual analysis cycle. The states have been replaced by sets of information entities and observations representing the domain information $\mathbb{I}$ and the mental model M and by sets of objects and situations representing the data $\mathbb{D}$ and the visualization V. The transitions are now given in terms of formal mappings.*

$\iota : V \rightarrow M$ matching the observed graphical elements in the visualization V to corresponding structures in the mental model M and its partial inverse, the anchoring relation $\alpha : M \rightarrow V$, formalizing the viewer's visualization literacy as the ability to externalize structures in the mental model by mentally mapping them onto structures displayed in the visualization.

In an ideal world, M and V behave exactly like $\mathbb{I}$ and $\mathbb{D}$. However, this is not to be expected in actual applications since the graphical representation is usually obtained by aggregating and reorganizing data. Likewise, the mental model is not just a reflection of information learned about specific data but rather a set of inference and derivation rules being applicable to a certain range of graphical data representations in order to infer information. In this sense, M can be said to reflect $\mathbb{I}$ if the elements in M are equipped with similar predicates and formulas as the entities and obsevations in $\mathbb{I}$. In this case, a mapping $ref : M \rightarrow \mathbb{I}$ can be established by matching corresponding predicates. The visualization V is an encoding of data $D \in \mathbb{D}$ with the aim to convey the information sampled by D. V hence is not a direct view on the data but only a representation of either the raw data directly sampled from $\mathbb{I}$ or the result of a series of transformations applied to this raw data. Note that – being a transformation itself – the visualization never provides a completely undistorted view on the data. The transformations applied to the data to generate the visualization are collected in an encoding-relation $enc : \mathbb{D} \rightarrow V$. Other than the representation of D, V offers additional functionality like, for example, interaction mechanisms. The actual

view on the data consists of a set of artifacts and structures being displayed in the visualization, encoding messages to be conveyed to the viewer.

The language determining the displayed messages is the graphical language $\mathcal{L}_V$, a concept well known from from literature on the composition of graphical displays such as Mackinlay's APT and Wilkinson's grammar of graphics [92,150]. The perception and correct interpretation of symbols arranged by the graphical language is neither guaranteed nor does it directly yield any sort of insight. The ability to perceive artifacts and structures depends on the viewer's visualization literacy – the knowledge about how to correctly read a visualization. This ability together with the cognitive ability to convert and interpret perceived artifacts and structures into the structures applied for reasoning defines a reading language $\mathcal{L}_R$. The conveyance of messages in visualization is governed by two factors: by the successfull transmission and receipt and by the correct translation from $\mathcal{L}_V$ to $\mathcal{L}_R$. The theoretical treatment of message conveyance in the visualization literature is dominated by approaches based on information theory. Information theory is commonly applied to discuss the transmission of messages and the likelihood that words of $\mathcal{L}_V$ appear and can be recognized by the viewer. Other approaches concentrate more on the translation between the two languages along the transmission. While the transmission is primarily a problem of appearance and visibility of symbols, the successful translation between $\mathcal{L}_V$ and $\mathcal{L}_R$ is not only a matter of literacy and recognition but also of perception. This work, however, is concerned with the general reasoning mechanisms involved with the cognitive processing of those messages and not with perception. Although those topics are highly interesting, the discussion of the actual transmission and translation process is left to the literature and this work instead concentrates on the identification and formation of $\mathcal{L}_V$ and $\mathcal{L}_R$. Throughout this Thesis it is therefore assumed that viewers will only reason about those words in $\mathcal{L}_R$ they are actually able to read from the visualization by perceiving them in the visualization and mapping them to a corresponding concept in their mental model of the visualization. In fact, the words in $\mathcal{L}_R$ deterimine substructures μ in the viewer's mental model M of the visualization. M is a representation of the viewer's understanding of the visualization. It reflects the reasoning structure along which artifacts and structures denoting words in $\mathcal{L}_V$ are mapped to entities and observations that are combined and reinterpreted to draw conclusions of different complexity. To this end, the mental model relates the concepts being read from the graphical display to the viewer's outside knowledge according to the inside-outside principle introduced in chapter 1. The combination of the outside knowledge with the information represented by the data and the messages conveyed by the visualization ideally forms structures in M that reflect the domain information $\mathbb{I}$. If so, domain information can be inferred from the construction of proper structures in M by the combination of the graphical display with the viewer's outside knowledge about how to interpret and what to conclude from the structures and artifacts found in the display. This kind of finding insight about the domain by establishing a map $ref : M \to \mathbb{I}$ back to the domain information closes the qualitative visual analysis loop. In summary, qualitative visual analysis is concerned with the cognitive process generating $\mathcal{L}_R$ from $\mathcal{L}_V$ by finding proper structures in the mental model M. The following discussion illuminates this process in more detail.

2.6.2 Transforming Data into Messages Conveyed by Visualization

Visualization does not show the data directly. It displays a representation allowing to draw conclusions about the data and – if combined with outside knowledge – about the domain of interest the data has been sampled from. Similarly, the data is only a snippet of the actual observation being made in the domain of interest which is why the domain cannot be understood by only reading the data without further reasoning – up to the exception of very simple domains. Reading a visualization, a viewer does not directly look at the data but rather at a representation attempting to aid the viewer in making sense of the data. Hence, there must be some form of mapping from what the viewer perceives in the visualization to the actual data being represented and to the facts about the domain the data witnesses.

For modeling the generation of visualizations from data, the idea of graphical languages has proven useful in the context of automatic visualization generation. This approach dates back to the pivotal work of Mackinlay's APT and Wilkinson's Grammar of Graphics [92, 150]. The authors explained models to translate formatted input data structures into visualizations. Depending on the input data format, an automatic choice of visualization type is possible. For example, such a system can map data being marked as parts of a common total to a pie chart and absolute values to a bar chart. Similarly, the bar chart can be translated to the pie chart by invoking the computation of percentages from the absolute values and feeding the new data back into the visualization generation process. In the theory of formal languages and their computation, this can be achieved by calling a so-called oracle, which is basically the invocation of an algorithm assumed or known to correctly apply the intended transformation. Assuming the data is formatted as a comma-separated list of formatting data determining which output state is being accessed, this can be modeled by a finite state automaton. This automaton's output states are interpreted as those states that actually draw something to the display. More precisely, output states are interpreted as oracles taking a list of key-value pairs and processing them to be displayed on the screen. If the automaton accepts the data string the visualization generated by the output states is considered a valid representation of the input data. Note that although the output elements already indicate the type of element, the data does not determine their actual shape or position. Shaping, scaling, positioning, as well as all other details about the appearance of a given portion of data on the display are up to the drawing mechanism being implemented in the oracle. Strictly speaking, this is already a deviation from the classical approach which is actually concerned with how to place marks and signs in a visualization and not only what marks and signs should be placed there. Yet, on the data-side, the focus of this Thesis is not how exactly the visualization is generated from the data but what the representation reveals about the data and about the domain of interest. Note that the type of automaton described above can only recognize regular languages, which means that the input string is necessarily formatted by a regular expression. An example of what this kind of automaton cannot do is the sequence of operations of drawing a number of items, then drawing something else, and only then drawing the same number of items as before. This would require remembering the number of items drawn

in the first part of the execution, something that cannot be expressed by regular expressions. A more sophisticated automaton model is discussed in a later section of this Thesis.

Generating the visualization is only one step. When designing visualizations, an important question to answer is how to map the data to artifacts and structures that will be recognized and understood by the viewer. Perhaps the most influential work on this question to this date is Bertin's Semiology of Graphics [16]. In his consideration of visual variables, visual signs and symbols, and their arrangement, Bertin established the foundation of modern theory on visualization design – although his work was focused on the design of graphical representations to be printed on paper rather than rendered on screens. The focus of qualitative visual analysis – and, hence, this Thesis – however, is not on the generation of recognizable and meaningful structures but much more on how those structures are being interpreted and processed once they are recognized. Consequently, the question how visual artifacts are structured and aligned and which visual variables are applied to determine their shape and general appearance in the presentation is only of minor importance for this work. Much more important to the scope of qualitative visual analysis is the question what information the viewer is expected to read and what an analyst is expected to conclude from a graphical representation of data. Depending on the information to be conveyed and on the complexity of the reasoning process being applied, different parts of the visualization will be considered. Especially global reasoning involves the simultaneous consideration of several artifacts and structures in the display. Indeed, although the elementary artefacts are the same symbols, the graphical structures being read and interpreted from visualizations differ from those being drawn.

Essentially, communication is always about understanding. Consequently, there can be no treatment of visually conveyed messages without considering the viewer's expected or intended interpretation. Qualitative visual analysis thus asks for the interpretation and cognitive processing of structures recognized in the visualization and hence is only indirectly concerned with semiology and semiotics. Instead, the focus is much more on syntax and semantics. Translating graphical representations into messages to be conveyed to the viewer involves more than the reproduction of the signs and visual variables being used to represent the data. Once the structures are perceived and recognized, the cognitive processes governing reasoning and interpretation have to be considered. Although the current state of the discussion does not enable the characterization of those processes yet, their structure and shape can already be determined – even though this determination only considers the perspective of the presentation.

Consider any given visualization. The artifacts and structures it displays are composed of marks and symbols whose appearance and alignment is determined by a set of rules. Those rules define how a program generates the visualization from a given set of properly formatted input data. For this part of the discussion, it is only of minor importance how exactly this transformation can be modeled – this question is discussed in more detail in a later section. The result of this transformation process – the visualization – can be described in terms of a graphical language $\mathcal{L}_V$. Every graphical language is generated as the words

being produced by a suitable formal grammar $\mathcal{G}(\mathcal{L}_V) = (N, T, P, S)$. The grammar generates the visualization by constructing a word in $\mathcal{L}_V$ by following a set of production rules P that, starting in some initial states S processes those states along sets of nonterminal symbols N until the artifact to be drawn on screen for the visualization is obtained as a terminal symbol T. The words of $\mathcal{L}_V$ are the individual marks and symbols together with the rendering information determining their appearance and position. Taking multiple words of $\mathcal{L}_V$ and following a set of syntax rules yields sentences and contexts as higher order structures on the graphical language. The composition of those structures is determined by the viewer reading the visualization. Understanding this mechanism is essential for the discussion of qualitative visual analysis as only the identification of structures in the visualization enables the inference of information from further processing. Therefore, the remainder of the discussion in this Section focuses on obtaining a technique for the construction of a model generating those structures over $\mathcal{L}_V$. Because this Thesis is concerned with the generation and processing of artifacts and structures being read from a visualization rather than their generation, the construction of $\mathcal{L}_V$ is not discussed in detail here. In fact, pocedures for this construction are actually well discussed in the literature. Perhaps the most widely known examples are Mackinlay's presentation toolkit and Wilkinson's grammar of graphics [92, 150].

2.6.3 Determining Structures over the Graphical Language

Although on the technical level the construction of sentences and contexts is arbitrary, a viewer will only be able and interested to interpret a subset of what is theoretically possible to construct from the words in $\mathcal{L}_V$. The rules determining the structures a viewer will perceive and process are determined by factors as diverse as the viewer's personal and professional experience, the analysis question, the context in which the analysis is performed, and even on sentiment and mood. The set of artifacts and structures recognizable by a viewer can be collected in another formal language, the reading language $\mathcal{L}_R$. $\mathcal{L}_R$ is constructed on top of the graphical language $\mathcal{L}_V$ determining the appearance of the visualization. It is the language denoting what a specific viewer will read in the visualization. It is important to note that $\mathcal{L}_R$ does not denote all structures that can be perceived by the user. Structures can indeed be recognized but remain unconsidered, for example, due to lack of relevance or interest. Even if a specific viewer is unable to recognize and interpret a certain artifact or structure in the display, it is still there and can be perceived. On the other hand, only those structures can be interpreted that the viewer is actually able to recognize and read.

Before the question how a structure readable by the viewer is formed in a visualization can be discussed, the question needs to be answered how those structures are actually being read. Not quite surprisingly, the structure of words in the reading language follows the levels of complexity discussed in Chapter 1. The first and most basic level is simple recognition or reading off of artifacts in the visualization. As an example, consider a bar chart. The viewer recognizes a bar as a bar, an axis as an axis, a mark as a mark, a label as a label, and so forth. Yet, to do so, the viewer must be capable of recognizing those elements as such,

meaning that the viewer must be able to read the visualization properly. This capability to read and understand visualizations correctly is commonly referred to as visualization literacy. Reading the chart means to form a sentence of words over the graphical language $\mathcal{L}_V$ determining the chart's appearance. For the bar char, this is achieved by combining an axis and a bar by some rule stating that the height of the bar along the axis is marked by the mark and that this height determines a value to be associated with the bar. A second rule then tells the viewer to associate the bar with an object being identified by the bar's label and yet another rule then makes the viewer associate the bar's value with this object. This kind of processing knowledge is not part of the visualization but part of the viewer's outside knowledge. An analyst might additionally attempt to draw conclusions from the depiction, for example by comparing the heights of multiple bars. To this end, a context must be formed in which this comparison can be executed. This again is achieved by applying rules determining that the heights of multiple bars now are compared to evaluate multiple objects while reading the value from each bar follows the same rule set as before. The resulting knowledge about the data is of the second level of complexity according to the three levels discussed in Chapter 1. The third level involves an interpretative transition of this knowledge to the answer of a question being phrased in the domain of interest and thus relies heavily on outside knowledge. For example, one bar being taller than the other might persuade some viewer without any additional outside knowledge about the visualization's general context to value the entity represented by the one bar higher than the one represented by the other. If the bars show profit to be made from selling different products, this interpretation would appear fairly obvious and certainly be correct. If it is the calories of different sorts of food to be combined to form a diet for different patients in a clinic, the interpretation will differ between a diet for an overweight person and a diet for an underweight person. It is thus clear that sentences and contexts in the reading language $\mathcal{L}_R$ contains parts not in the graphical language $\mathcal{L}_V$. How exactly $\mathcal{L}_R$ is formed from $\mathcal{L}_V$ thus depends on the viewer's or analyst's rule set serving as a reasoning system to be applied for reading the visualization. This reasoning system does not only determine the composition of structures from displayed artifacts but also determines the relations between those structures. In many – if not most – cases, sentences and contexts to be read from the visualization are constructed with the purpose to obtain quite specific information from reading the words, sentences, and contexts in a certain manner. Consequently, in order to determine production rules for a reading language $\mathcal{L}_R$, the viewer's ability to recognize, combine, and relate artifacts displayed in the visualization has to be modeled.

In the following, several relations need to be represented that further down in the discussion are combined into a graph-structure. This graph structure is later applied to infer the words of the reading language. The notation for those relations is inspired by Cypher, a database programming language used for the neo4j graph data base [53,101]. This notation is chosen for convenience, because it has been designed to reflect the graph-characteristics of the relations being represented while still being well-readable in text. In this notation, relations are binary and follow a simple subject-predicate-object logic where the subject and object are indicated by parentheses and the predicate by brackets. In a graph structure, the subject and object are nodes, and the predicate is the label of a

directed edge connecting the subject to the object.

Sentences over $\mathcal{L}_V$ require their words to be recognized by $\mathcal{L}_R$ and a relation combining them. Relations are transitive and can thus combine sentences to larger sentences. For example, $(mark) - [on] \rightarrow (axis)$ and $(height) - [of] \rightarrow (bar)$ determine two simple relations over words in $\mathcal{L}_V$, namely *mark*, *axis*, *height*, and *bar*. Strictly speaking, *height* is actually a property of *bar* and thus a subword. Since those sentences are also part of the visualization's generation, they are considered native to $\mathcal{L}_V$. Similarly, the sentence $(mark) - [determines] \rightarrow (height)$ is native to $\mathcal{L}_V$, because the visualization is constructed this way, including the sentence as part of its syntax. Inference from a legend instead denotes a graphical context over $\mathcal{L}_V$ as now two syntactically separate structures have to be combined based on some rule known to the viewer. The outside information about how to apply the legend with the local depiction and how to interpret the combination of both is part of the viewer's visualization literacy, which is either a skill acquired previous to viewing the visualization or to some extent learned from reading the visualization's legend, if provided. The viewer's literacy determines how sentences are formed and combined. Following a strict subject-predicate-object structure, the sentences described here can be mapped to the Resource Description Framework (RDF), the data exchange format underlying the semantic web [7, 14, 90, 100]. In this Chapter, the semantic web is applied to combine the different sentences being formed by the construction introduced here to a graph. The concatenation of simple sentences is one possible operation to obtain higher complexity. It allows to construct sentences like $(mark) - [determines] \rightarrow ((height) - [of] \rightarrow (bar))$. Concatenation is a special case of a more powerful formalism allowing the nesting of sentences by replacing the subject or object of a sentence by the subject of the sentence being nested. The nested sentence is then treated as the new subject or object of the sentence is it embedded in. Adding sentences like $(bar) - [represents] \rightarrow (object)$ to the above example, this additional outside knowledge allows constructions like

$$\begin{aligned}&(mark) - [on] \rightarrow (axis) - [determines] \rightarrow \\ &(height) - [of] \rightarrow ((bar) - [represents] \rightarrow (object))\end{aligned}$$

Outside knowledge can also include full sentences over $\mathcal{L}_R$ that determine a context combining two or more sentences of $\mathcal{L}_V$, for example

$$((height) - [of] \rightarrow (bar)) - [denotes] \rightarrow ((value) - [of] \rightarrow (object))$$

Sentences like this allow to model simple interpretations in the reading language $\mathcal{L}_R$, although those interpretations only consider the syntax component of the visualization and are ignorant of the semantics. As an example, consider the sentence

$$\begin{aligned}&(mark) - [on] \rightarrow (axis) - [determines] \rightarrow \\ &((height) - [of] \rightarrow ((bar) - [represents] \rightarrow (object)) - [denotes] \rightarrow \\ &(value) - [of] \rightarrow (object))\end{aligned}$$

It states that a mark on an axis denotes a value for an object being represented by a bar. However, in this syntax, there is no way to express that the two references to *object* in this sentence actually refer to the same entity. Notice that

this is not self-evident. Although violating the expectation that both instances of *object* refer to the same entity might not seem intuitive, it actually applies in cases where different properties of the object are represented by different visual variables determining the bar's appearance. Consider a bar whose height encodes a number of sales and whose width encodes the price of a product. By this encoding, the area determines the profit made from selling the product. Whether this is a good choice for the visual variables or not is not important here – the point is that in the above sentence, the height now refers to only the price while the bar refers to the whole product. Hence, in this example, the two instances of *object* actually refer to different entities. The consideration whether the instances refer to the same entity or not is of purely interpretative nature. The structures reviewed thus far constrain the recognition of combinations of words in the graphical language to sentences that can be read and processed by the viewer. Those sentences over $\mathcal{L}_V$ appear as words in $\mathcal{L}_R$ and the contexts over $\mathcal{L}_V$ often appear as sentences in $\mathcal{L}_R$ according to the combination rules determined by the outside knowledge being part of the nonterminals and production rules of the grammar generating $\mathcal{L}_R$. Note that the combination of elements in $\mathcal{L}_V$ into structures in $\mathcal{L}_R$ is an interpretation. Similarly, matching structures in the reading language to what is actually depicted and thus declared by the graphical language is an anchoring operation. This kind of rules forming structures in $\mathcal{L}_R$ based on combinations of structures in $\mathcal{L}_V$ thereby constraints the way the viewer is reading the visualization. Therefore, this kind of rule is called a *constraint* for the remainder of this Thesis.

There is a second type of relation forming structures in $\mathcal{L}_R$ which determines the formation of contexts over $\mathcal{L}_V$. This relation is called an *interpretation*. It summarizes displayed artifacts and structures and maps them to a semantic entity the viewer can process cognitively. Note that the equal terminology to the interpretation operation in the qualitative visual analysis cycle is on purpose. It is essentially the same operation since, strictly speaking, even the direct translation of a single word in $\mathcal{L}_V$ to a word in $\mathcal{L}_R$ is the mapping of a collection of artifacts to a semantic entity that can be processed by the viewer's cognition. In the semantic web, this can be modeled by a blank node collecting multiple incident relations and relating them to a single adjacent element. For example, the collection $(A) - [in\ the\ context\ of] \rightarrow (\_blank01)$, $(B) - [in\ the\ context\ of] \rightarrow (\_blank01)$, $(C) - [in\ the\ context\ of] \rightarrow (\_blank01)$, and $(\_blank01) - [involves] \rightarrow (D)$ maps the combined words in the sentences A, B, and C to the sentence D. D is a result of the viewer's interpretation and hence part of $\mathcal{L}_R$ but not of $\mathcal{L}_V$. It may also be processed differently than the individual sentences A, B, and C if the are processed out of context. An interpretation can, for example, mark the two instances of *object* in the above example as representatives of the same entity or the one instance as representation of a property of the other, depending on the viewer's understanding of the visual encoding.

2.6.4 A Qualitative Principle of Minimal Graphical Overhead

The totality of possible messages being conveyed by the visualization is the power set over the words in the graphical language,

$$\mathcal{L}_V^{tot} = \{w | w \in (\mathcal{L}_V \cup Sentence_V \cup Context_V)\} = \mathbb{P}(\mathcal{L}_V)$$

This total graphical language $\mathcal{L}_V^{tot}$ consisting of all possible words, sentences, and contexts that can be built by arbitrary combinations of words over $\mathcal{L}_V$ contains a large overhead of words that are not useful for the viewer. In an ideal world, every viewer's specific reading language only contains structures over $\mathcal{L}_V$ that are actually beneficial for the analysis, meaning that it only contains structures that can be processed in the viewer's cognitive reasoning model and that neither induce misunderstandings nor wrong interpretations. In such a setup, an ideal visualization would contain minimal possible overhead. The total graphical language would thus ideally be exactly the union of every possible viewer's individual reading language. Since this would require the consideration of the individual reading languages of theoretically infinitely many viewers and all viewers would need to be perfect in the sense that neither misunderstandings nor misinterpretations are made, this approach to characterize $\mathcal{L}_V^{tot}$ at a first glance appears rather pointless. However, keeping the overhead minimal for a selected set of viewers is an interesting approach to determine a rule for visualization optimization, the principle of minimal graphical overhead:

Principle: Minimal Graphical Overhead

For a given group X of viewers $x \in X$, the complexity for reading a visualization is minimized if $|\mathcal{L}_V^{tot} \setminus \bigcup_{x \in X} \alpha(\mathcal{L}_R^x)| \rightarrow \min$

In essence, this principle means that a visualization should try not to display more information than its viewers are able to read and process. Although this seems fairly obvious, it is important to point out the semantic component being considered by applying the anchoring of the reading languages to the graphical language. It it noteworthy that considerations in this direction usually do not address the semantic component but only the displayed content and thus suggest to minimize over $\mathcal{L}_V$. Typically, visualizations are engineered to address multiple different analysis questions. Yet, minimizing over $\mathcal{L}_V$ would restrict them to convey only the answers to specific questions. Characterizing graphical overhead in terms of $\mathcal{L}_R$ instead of $\mathcal{L}_V$ means to reduce the depiction to what different viewers can read, understand, and interpret in general rather than reducing $\mathcal{L}_V$ to what users need to answer only a specific question. This allows the visualization explicitly to include elements supporting the reasoning process on a qualitative level. Information reduction paradigms only considering the structural composition of visualizations and being ignorant of the viewer's actual or anticipated reasoning procedures instead bear the risk of destroying possibly valuable sentences and contexts.

Interestingly, minimal graphical overhead does not exclude additional graphics that are not part of the visualization and are thus often disregarded as chartjunk, because they usually do not serve informative purposes but rather distract

the viewer from actually important information. Yet, if this so-called chart-junk actually conveys relevant information, it might indeed be valuable for the user. For example, it could symbolize the content of a chart and thus allow the user to directly interpret the information without reading the image caption. Some findings and considerations in this direction are reported by Vickers et al. in their discussion on one of the relations in their approach to formalize visual analytics in terms of category theory [144].

2.6.5 The Reading Language's Descriptive Scope

In an ideal world, the total graphical language $\mathcal{L}_V^{tot}$ of a visualization V is the union of the anchoring of all viewers' individual reading languages in the graphical display. Although in practice the closest to this situation is a visualization with minimal overhead information in its interpretation, this ideal reveals the scope of the reading language $\mathcal{L}_R$. Central to this consideration is the idea are linked to each other by the interpretation and anchoring relations introduced above. Recall that the reading language is defined as the collection of words, sentences, and contexts over $\mathcal{L}_V$, where the construction follows a set of rules involving outside knowledge. Consequently, if $\mathcal{L}_V^x$ is the subset of the total graphical language a given viewer x can interpret, then $\mathcal{L}_R^x = \iota(\mathcal{L}_V^x)$. In an ideal world, the interpretation ι and anchoring α hence are the two directions of a bijective map since by definition $\mathcal{L}_V^x = \alpha(\mathcal{L}_R^x) \subseteq \mathcal{L}_V^{tot}$. In the real world, there can be misunderstandings and misinterpretations of artifacts and structures in the visualization. Even if those are not considered explicitly, they are still part of the total graphical language. Hence, the principle of minimal overhead remains valid. Yet, misunderstandings and misinterpretation lead to misconceptions and wrong conclusions and therefore are much to the detriment of the analysis result. Their consideration thus is an important component of qualitative analysis in general and qualitative visual analysis in particular. The reading language's descriptive scope and purpose is exactly to enable this consideration.

The reading language $\mathcal{L}_R^x$ of a given viewer x reading visualization V is composed of the part $\mathcal{L}_V^x$ of the total graphical language $\mathcal{L}_V^{tot}$ over V that the viewer understands and the viewer's outside knowledge $\mathcal{L}_{out}^x$. The outside knowledge formalizes the complete mental model and thereby the viewer's capabilities to read and interpret the visualization. Understanding here refers to the viewer's ability to read structures over $\mathcal{L}_V$. As explained above, this is modeled by the anchoring of the mental model $\mathcal{L}_{out}^x$ to the graphical language (which is essentially a projection). The reading language can thus be computed as:

$$\begin{aligned}\mathcal{L}_R^x &= \iota(\mathcal{L}_V^x) \\ &= \iota(\alpha(\mathcal{L}_{out}^x)) \\ &= (\mathcal{L}_V^x \cap \mathcal{L}_{out}^x) \cup \mathcal{L}_{out}^x\end{aligned}$$

The size difference between $\mathcal{L}_V^x$ and $\mathcal{L}_{out}^x$ indicates whether the majority of insight can be obtained from reading the visualization, whether the viewer is actually able to read off this information correctly, or whether interpreting the visualization involves large amounts of outside knowledge. Note that this does not make the difference between the viewer and the analyst: The question here

is not whether or not much reasoning is applied to obtain insight about the visualization. Comparing the sizes of $\mathcal{L}_V^x$ and $\mathcal{L}_V^{out}$ rather is an indicator whether the visualization plays a supportive or central role in the reasoning process. The former is characterized by much larger $\mathcal{L}_{out}^x$ while the latter case ranges from about equal size of both languages to much larger $\mathcal{L}_V^x$. In many if not most visualization applications, the latter situation will be the case. Note that if $\mathcal{L}_V^x \cap \mathcal{L}_{out}^x$ is small compared to $\mathcal{L}_V^x$, this indicates that although the visualization is important, the viewer lacks the visualization literacy to read off the entire information being offered. This does not mean that the results to be inferred are not part of the viewer's reading language $\mathcal{L}_R^x$ as they might still be derivable from what the viewer can read and the available outside knowledge. Nevertheless this is an indicator that the viewer will not be able to profit from the visualization's full potential. However, it can also mean that the principle of minimal graphical overhead is violated and the visualization is overly detailed or provides unneccessary information. Especially in applications of small multiples or when detail information is provided by an additional small visualizations, this addendum is commonly of supportive character and should thus not contain more information than is necessary to communicate and contextualize the additional information. If this indication does not agree with the visualization's intended role, the visualization has to be refined to either incorporate more outside knowledge or towards reducing the redundancy with other sources of information.

Visualizations are not random constructions of graphical elements but are designed to be read in a certain way. This is captured by defining an intended graphical language $\mathcal{L}_V^{poss} \subseteq \mathcal{L}_V^{tot}$ over $\mathcal{L}_V$. $\mathcal{L}_V^{poss}$ collects all structures over $\mathcal{L}_V$ that the designer expects the viewer to be literate enough to read and to be able to interpret within an anticipated reasoning structure. Every conclusion about the visualization and the data that cannot be drawn from within this reasoning structure cannot be guaranteed to be sound in the sense of not being based on a misinterpretation or misconception. Therefore, $\mathcal{L}_V^{poss}$ can be said to be the collection of all possible words, sentences, and contexts over the graphical language. Again, the language of all possible readings is $\mathcal{L}_R^{poss} = \iota(\mathcal{L}_V^{poss})$. Its outside knowledge component $\mathcal{L}_{out}^{poss}$ is the collection of all outside knowledge required to read and interpret the visualization completely and correctly in the sense of its design. Of course, not every deviation from $\mathcal{L}_R^{poss}$ is automatically an error. The viewer is equipped with a different set of outside knowledge than the visualization's designer and might thus be able to draw additional conclusions. However, the visualization is based on the designer's understanding of matters. Hence, the conclusions drawn from analysis based on any collection of outside knowledge can only be guaranteed to be sound if for the analyst $\mathcal{L}_V^x \subseteq \mathcal{L}_V^{poss}$.

The set of possible readings $\mathcal{L}_R^{poss}$ is useful to reason theoretically about visualization techniques. Given not only the technique but also a specific data set to be visualized, $\mathcal{L}_R^{poss}$ is instead too powerful because it states what information the visualization could show rather than what the visualization actually does convey about the data. As an example, consider a bar chart and let $\mathcal{L}_R^{poss}$ contain the sentence

$$((height) - [of] \rightarrow (bar)) - [greater\ than] \rightarrow ((height) - [of] \rightarrow (bar))$$

The bar entities are actually different and the $[greater\ than]$-relation only ap-

plies if two different bars are inserted. In the proposed formalization of the reading language's constraints and interpretations in terms of the semantic web and RDF, this condition is implicitly fulfilled since RDF is constructed over actual instances rather than abstract concepts of objects. Hence, the two bars are actually different instances of bar objects, each identified by its own unique identifier. To emphasize their belonging to the same class of object, extensions of RDF like RDF-Schema or OWL can be applied [7, 90, 100]. Those extensions allow to define classes of objects in terms of a so-called schema, essentially an additional label for the class type. The details are not important for the consideration here. For the remainder of this section, indexed variable names refer to instances to be distinguished. The visualization of actual data is a specific collection of words of the graphical language and hence a subset $\mathcal{L}_V^{\mathbb{D}} \subseteq \mathcal{L}_V$. Consequently, the structures over $\mathcal{L}_V^{\mathbb{D}}$ that are available for viewers to read are only a subset of $\mathcal{L}_V^{poss}$. The set $\mathcal{L}_V^{appl}$ of words, sentences, and contexts actually applicable to the data $\mathbb{D}$ being visualized by construction also reduces the possible reading language to a collection of actually possible readings, $\mathcal{L}_R^{appl}$. Just like before deviations in reading from $\mathcal{L}_V^{poss}$ indicated potential misconceptions and misunderstandings on the level of basic understanding of the visualization, deviations from $\mathcal{L}_V^{appl}$ now indicate wrong readings of actual data.

The major difference between the possible and the applied languages is that only the latter represent the actual data and thus the appearance of the visualization and the obtained information inferred in the mental model. For an analyst working with a visualization, it is hence enough to understand what is actually being shown in the graphical display to reason about the data representation. Following the principle of minimal graphial overhead, this implies, for example, that legends in visualization should not attempt to explain anything that is not shown. One interpretation of minimal graphical overhead is that no structures should be shown that do not map to corresponding structures in the reading language. For a legend, it only makes sense to interpret the relations between symbol and meaning shown in combination with corresponding elements to be found in the graphical display. If such an element is not present, there is nothing to be interpreted that is to convey the meaning indicated by the legend and hence there is neither a conclusion to be drawn nor domain knowledge to be found based on the particular interpretation rule specified by the legend. Consequently, showing entries in the legend that do not correspond to anything that can actually be seen in the visualization violates the principle of minimal graphical overhead. More formally, any sentence of the form $\lambda := (X) - [means] \rightarrow (Y)$ being part of the legend is automatically part of $\mathcal{L}_V^{poss}$ and thus also of $\mathcal{L}_V^{tot}$, because it is also part of $\mathcal{L}_R^{poss}$ along with every possible interpretation of the element if it actually appears in the visualization. If an instance of (X) is shown in the display, it is associated with λ in a graphical sentence Σ by an interpretation of the form

$$\begin{aligned}\Sigma &= (A) - [involves] \rightarrow (\lambda) \\ &= (A) - [involves] \rightarrow ((X) - [means] \rightarrow (Y))\end{aligned}$$

Of course, Σ is always part of the possible readings $\mathcal{L}_R^{poss}$. However, if there is no object (A) in the display, (A) is not part of $\mathcal{L}_V^{appl}$ and therefore Σ is not part

of $\mathcal{L}_R^{appl}$. Consequently, if λ is shown in the legend nevertheless, then

$$\begin{aligned}
&\left((A) \notin \mathcal{L}_V^{appl} \wedge \lambda \in \mathcal{L}_V^{appl}\right) \\
\Rightarrow &\Sigma \notin \mathcal{L}_R^{appl} \\
\Rightarrow &\lambda \in \left(\mathcal{L}_V^{tot} \setminus \alpha(\mathcal{L}_R^{appl})\right) \\
\Rightarrow &\neg \left(|\mathcal{L}_V^{tot} \setminus \alpha(\mathcal{L}_R^{appl})| \to \min\right)
\end{aligned}$$

Because the possible and applicable languages denote the intended graphical and reading languages this actually holds for every reader and by the formal proof the remark is promoted to a lemma derived from the principle of minimal graphical overhead:

Lemma: Concise Legends

A visualization's legend should never show symbolds that are not to be found in the display.

Indeed, this kind of over-explanation can be quite misleading guiding viewers towards looking for structures that cannot be found in the display. Note that although this remark appears quite obvious, it is actually a consequence formally deduced from the theoretical model of qualitative visual analysis discussed in this Thesis.

2.6.6 Constructing the Mental Model

In general, it will be hard to determine a specific viewer's exact reading language. Instead, it is much more convenient to formalize the intended reading language and then capture a viewer's readings and an analyst's considerations by installing proper provenance mechanisms allowing to evaluate the actual readings and considerations against those intended by the visualization's designer. If the structures do not match, especially if $\alpha(\mathcal{L}_R^x) = \mathcal{L}_V^x$ contains an element that is not in $\mathcal{L}_V^{appl}$ or even not in $\mathcal{L}_V^{poss}$, the viewer should at least be informed about this situation. Such a procedure helps preventing errors for at least two levels of the reasoning process, namely insight about the visualization and insight about the data. Concerning insight about the domain, there is no mechanism preventing an analyst from erring. After all, this would require to answer an analysis question before it has even been asked. To enable the efficient identification of potential errors, the mechanisms behind the formation of words in the reading, the grammar producing the reading language needs to be found. Interestingly, this grammar can be read off directly from the viewer's mental model of the visualization.

The reading language $\mathcal{L}_R$ essentially defines a set of rules determining the formation of sentences and contexts over the graphical language $\mathcal{L}_V$ of some visualization V. Constructing the reading language over $\mathcal{L}_V$ therefore means to reproduce a viewer's or an analyst's reasoning process. Two necessary relations have been identified for the construction of words in $\mathcal{L}_R$: constraints, and interpretations. Constraints combine multiple interpretable words or sentences over

$\mathcal{L}_V$ into a subject-predicate-object structure determining a relation between the subject and the object that constraints the total set of possible combinations of artifacts and structures in the display to those that can be processed by the viewer's cognition. Interpretations combine words, sentences, and contexts over $\mathcal{L}_V$ and bind them to a concept the viewer is able to reason about. The fact that this relation has the same name as the interpretation relation ι translating the graphical representation to the interpreted information is not a coincidence. Indeed, every word in $\mathcal{L}_V$ that the viewer can understand either has a one-by-one correspondence to a concept in the viewer's cognition or has an interpretation in combination with other words. For simplicity, one-by-one correspondence is treated implicitly and the corresponding elements simply appear as words in both languages $\mathcal{L}_V$ and $\mathcal{L}_R$. Applying the semantic web to model the structure in terms of a directed graph, reading the constraints and interpretations in the direction they point reveals the interpretations of artifacts and structures in the graphical language. More precisely, for every word $v \in \mathcal{L}_V$, the set of possible interpretations involving this word is determined by the upwards closure $\iota^{\uparrow}(v)$ of v's direct interpretation over the constraints and interpretations determining $\mathcal{L}_R$. The upwards closure of v's interpretation collects all words in $\mathcal{L}_R$ that are in the reflexive and transitive hull over the productions of $\iota(v)$ in the mental model. More intuitively, it collects all words in $\mathcal{L}_R$ that can be derved from $\iota(v)$ by the contraints and interpretations forming the mental model defining $\mathcal{L}_R$. Hence, the constraints and interpretations compose the mental model M representing the viewer's cognitive reasoning about the visualization. For every word v in the visualization, there is a substructure $\mu \subseteq M$ in the mental model for which $\mu = \iota^{\uparrow}(v)$. Consequently, for every substructure $\mu \subseteq M$, there must be a subset of words in the graphical language, $v = \alpha(\mu^{\downarrow})$ which is exactly the collection of words in the downwards projection of μ to its contained elements corresponding directly to artifacts and structures in the visualization. Let μ be a single node in the semantic web. To construct $\mu^{\downarrow}$, one needs to follow the possible chains of constraints and interpretations backwards. This means to replace the subject or object of a sentence by the yet to construct subtree such that the resulting word in $\mathcal{L}_R$ is constructed by a tree rooted in μ and whose leafs are either words of $\mathcal{L}_V$ or of the outside knowledge $\mathcal{L}_{out}$. If μ consists of multiple nodes, the construction follows a similar procedure. Where multiple choices are possible, all possible paths have to be followed to construct $\mu^{\downarrow}$ completely. The result of this construction are sentences over the reading language $\mathcal{L}_R$. Therefore, constructing the mental model means to construct the grammar producing the reading language. To construct the mental model, one needs to collect all constraints and interpretations and structure them in a semantic web.

Once the mental model is complete, the grammar generating the words in $\mathcal{L}_R$ can be read off as follows: Let $\mathcal{G}(\mathcal{L}_R = (N, T, P, S)$ be the grammar generating the reading language. Its terminal symbols $t \in T$ are those nodes in the semantic web that do not have any incident predicate. Every other node in the semantic web denotes a nonterminal symbol $n \in N$. The starting symbols $s \in S$ are all symbols that are related to some entity in the domain of interest $\mathcal{I}$. Note that this is a non-standard construction for a formal grammar since usually it is assumed that grammars only have a single starting symbol. However, this can easily be achieved if one allows a blank transition along an imaginary constraint connecting a single meta-node to each node representing a starting symbol. In

the semantic web, this can be achieved by adding a blank node as the meta-state and an empty relation connecting it to every node corresponding to a starting symbol. For the production rules, the constraints are considered to be of the form

$$c \Rightarrow (x) - [constraint\ name] \rightarrow (y)$$

where $x \in N$ is a nonterminal symbol being the subject or object of a constraint rule c and y can be a terminal or a nonterminal symbol. For interpretations, the construction is adapted by combining the interpreted elements using the constraint rules applying an imaginary *and*-constraint and connecting it to the actual interpretation applying a constraint named *involves*:

$$\begin{aligned} x \Rightarrow (x) - [involves] \rightarrow ((A) - [and] \rightarrow (B)) \quad & where \\ A, B \Rightarrow x \in N | x \in T | \, (A) - [and] \rightarrow (B) & \end{aligned}$$

A special case is the production scheme $x \Rightarrow y$ projecting each remaining nonterminal symbol x to a directly referring terminal symbol y.

Thus far, the discussion only considers the principle of constructing the mental model from a given sets of outside knowledge, constraints, and interpretations being applied to formulate the reasoning strategy. For a given set of data, the model additionally needs to be reduced from the possible interpretation to the ones applicable to the data. One approach to achieve this would be to instantiate the model by generating an entity in the semantic web for each occurrence of a specific node in the data. For a bar chart, this could mean to generate a *bar*-entity for each bar in the visualization. Although this is the way the data is supposed to be treated in the semantic web being a web of instances rather than schemas or concepts, this method has two significant drawbacks. First, it causes the network's complexity to explode. This would be bearable if there was a mechanism to determine the difference between the individual objects. However, without the ability to assign truth values that witness whether the relations and attributes actually hold for a specific instance or entity, such a distinction cannot be made and every instance is merely a copy of its prototype in the possible reading language. The second drawback is also a result of the semantic web's inability to assign truth values to relations but extends this to the problem that in the semantic web interpretations and constraints cannot be made conditional. Due to this fact, every relation that is syntactically correct is essentially a valid interpretation of the data. For example, consider the sentences

$$\begin{aligned} &((value) - [of] \rightarrow (bar_1)) - [greater\_than] \rightarrow ((value) - [of] \rightarrow (bar_2)) \\ &((value) - [of] \rightarrow (bar_2)) - [greater\_than] \rightarrow ((value) - [of] \rightarrow (bar_1)) \end{aligned}$$

Both sentences are absolutely correct in the construction as described before, even though this relation can actually only hold in one direction. The semantic web describes the state of instances and the relations between them but is not equipped with a reasoning mechanism. Reasoning about semantic web data requires to construct instances of objects from the web and to evaluate them in a logical structure completely external to the web. As a consequence, there is no way to associate truth values to relations and instance state information. In

the qualitative visual analysis cycle this is captured for the domain information by binding valuated entities and observations to higher order predicates representing the objects and situations instantiated by the valuation. However, the construction of the mental model thus far relies only on the semantic web and hence cannot reflect this complexity directly. Towards a convenient treatment of reasoning about domain information within the mental model, a more complex model than the semantic web is needed to adequately describe the mental model and the semantics being associated with data. Such a model is introduced in the next section.

Because the mental model determines the grammar generating the reading language, mental models for different viewers or analysts can be applied to compare their individual interaction and reasoning when working with the visualization. The structural composition of the mental models directly reveals deviations in the workflows and reasoning processes, possible short cuts, or potential reasons for the emergence of different opinions. Although the semantic web is somewhat limited for the discussion of the actually applicable readings for a given data set, it can indeed be applied to compare mental models at the level of possible readings. Note that the model for possible readings can always be obtained from a model of actual readings by documenting the interpretations and constraints applied by a viewer or an analyst during the interpretation and reasoning process and combining instances of the same type into a single node representing the type of object in the semantic web. Hence, the construction of the mental model described above can be applied to determine the mental models for different viewers or analysts and to compare them to each other as well as to the mental model determining the reading and reasoning structure as intended by the visualization's designer. For this comparison, one has to construct the corresponding mental models and to compare their topology. Assuming the semantic webs to be compared apply the same vocabulary for their nodes and predicates, this can be done automatically by parsing the RDF-triples listing the $(subject) - [predicate] \rightarrow (object)$-relations the semantic web is composed of. Note that by the construction using blank nodes, this also includes the interpretations. The different topology of the individual mental models reveals possible differences of the interpretation of a visualization. The explicit consideration of the reasoning process hence reveals whether the visualization will be interpreted similarly by all viewers and analysts. If this is not the case for a part of the visualization that is central to the reasoning process, the visualization probably needs to be refined as this might be a potential source for misconceptions and misunderstandings. In order to prevent such deviations between interpretations of different viewers or analysts, the mental model can be constructed for different anticipated reasoning strategies during the design process. For example, the different strategies can reflect the different professional backgrounds of viewers in a collaborative setup. Such an explicit consideration of the reasoning process as a qualitative property of visual data analysis hence allows to identify potential pitfalls in the visualization design early in the design process.

2.6.7 From the Mental Model to Domain Information

The mental model is only an image of the conclusions drawn from the visualization about the visualization itself and about the data. The highest level of reasoning, drawing conclusions about the information domain of interest $\mathbb{I}$ is not part of the mental model. Although it would be needed to close the circle of reasoning introduced in the qualitative visual analysis cycle, the reflectance map $ref : M \rightarrow \mathbb{I}$ from the mental model to the domain of interest cannot be provided at the current state of the discussion. Even if the domain is also described in terms of the semantic web, the formulation of connections only models associations, not actual mappings. Objects in $\mathbb{I}$ require a more complex description than the semantic web can provide, especially since $\mathbb{I}$ does not only contain entities and their relations but also knowledge about rules and principles being applied to them or characterizing their behavior. The next section therefore proposes a refined construction for the languages introduced above and a theoretical framework enabling the mapping of mental model information back to the domain of interest.

2.6.8 Structuring Visualizations

Descriptive theories of the composition of visualizations and other graphical displays often apply the concept of a graphical language to formalize the information to be conveyed. Mackinlay, Wilkinson, and others explained the creation and reading of visualizations as an encoding-decoding-mechanism where the encoding maps data to its graphical representation and the decoding is the capability of viewers to properly understand what they see in the graphical display (e.g. [92, 136, 151]). Casner presented similar ideas for task-efficient visualization, where an optimal encoding is found by translating cognitive tasks into perceptual tasks for which proper visualizations are known [23]. This Thesis adapts these ideas. The encoding is understood as a visualization pipeline mapping the data to a visualization that might be altered by interaction during the analysis process. The decoding is the interpretation of the visualization's components, matching them to the proper elements of the mental model.

Once again, note that the discussion in this Thesis is not directly concerned with the definition of proper encodings but still needs to consider the shape of a graphical representation as the basis of reasoning about semantic information. By the programming, the data has already been translated into a language. A formalism structuring the visualization's components with respect to the data being visualized hence only has to reflect how the visualization is generated given a specific set of data $\mathbb{D}$. However, reasoning about structures in the graphical display V requires to actually know those structures. Rather than only determining the depicted elements forming the words in the graphical language $\mathcal{L}_V$, those structures also construct graphical sentences and contexts over $\mathcal{L}_V$, yielding exactly the total graphical language $\mathcal{L}_V^{tot}$ as defined in Section 2.6.4. Note the difference between graphical sentences being part of $\mathcal{L}_V$ and the sentences and contexts being generated over $\mathcal{L}_V$ as part of the reading language $\mathcal{L}_R$. Where the former are a result of the visualization's construction

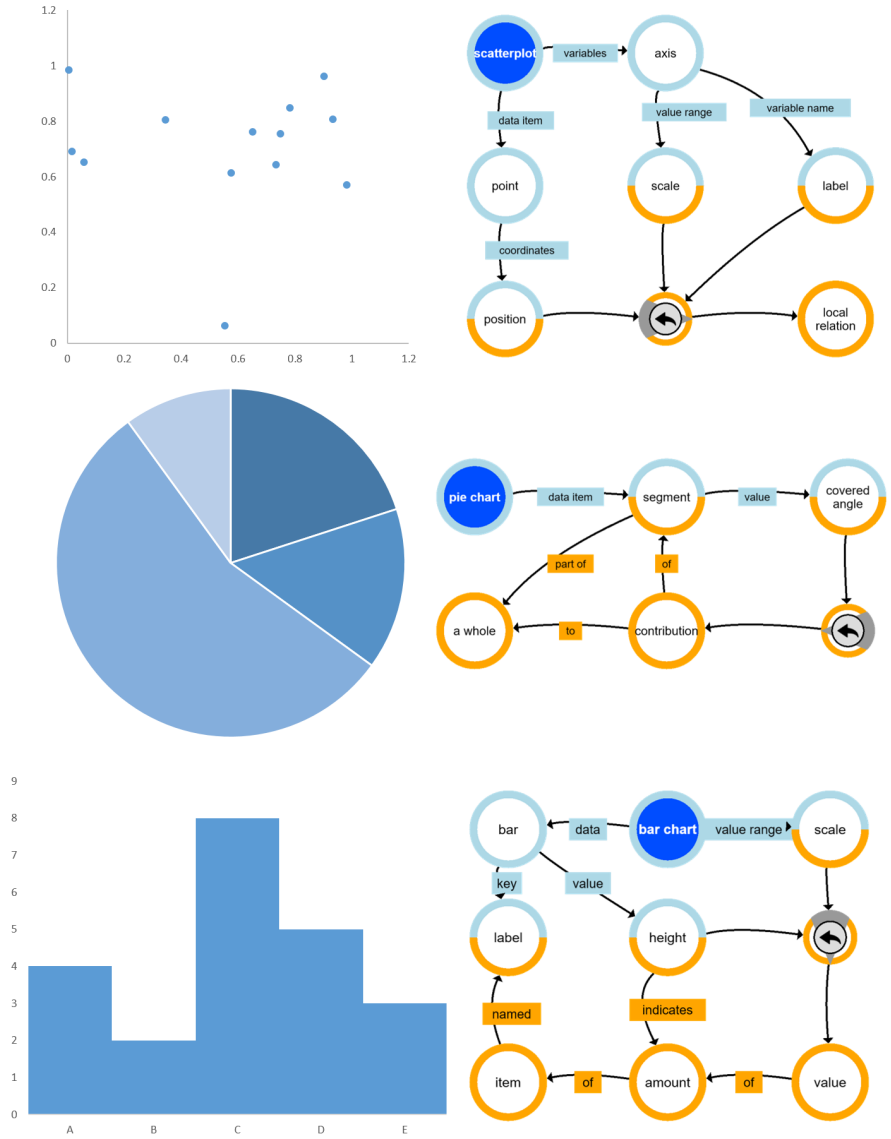


Figure 2.3: *Example concept graphs for a scatter plot, a pie chart, and a bar chart. The blue nodes and edges denote the syntax part modeling the automaton for the visualization V, the orange nodes and edges model the automaton describing the processing of information in the mental model M.*

from the data, the latter are determined by the viewer's understanding of the visualization in combination with available outside knowledge. Hence, graphical contexts over $\mathcal{L}_V$ determined by a specific viewer x's reading language $\mathcal{L}_R^x$ are likely to be combined from parts of $\mathcal{L}_V$ that are not combined syntactically during the visualization's encoding. For example, an axis in a scatterplot forms a graphical sentence whereas the association of a point in the plot with a mark on

the axis usually does not (if the mark and the point are drawn independently). Yet the combination of mark and point still forms a graphical context over $\mathcal{L}_V$ being part of the reading language $\mathcal{L}_R$ of a viewer who is able to interpret the scatterplot correctly. In $\mathcal{L}_R$ this is produced by combining the two sentences $(y-value)-[of]\rightarrow(point)$ and $(mark)-[on]\rightarrow(y-axis)$ in $\mathcal{L}_V$ by outside knowledge stating that the mark determines the value as formalized by the statement

$$((mark)-[on]\rightarrow(y-axis))-[determines]\rightarrow((y-value)-[of]\rightarrow(point))$$

An automaton reflecting the encoding $enc:\mathbb{D}\rightarrow V$ of data into structures in $\mathcal{L}_V^{tot}$ should hence not only be a test-system accepting correct candidates and rejecting structures not in $\mathcal{L}_V^{tot}$ but should rather reveal how the structures in $\mathcal{L}_V^{tot}$ emerge from the data such that the corresponding graphical sentences are available to be combined by the production rules for the reading language. In the scatterplot example, this would mean that not only individual axes and marks are recognized as being part of $\mathcal{L}_V$ but also the complete sentence formalizing the axis' composition containing several marks. This can be achieved by expressing the graphical encoding in a top-down approach by a transduction automaton. From here on, the visualization V is interpreted as a finite state transducer translating data into to structures in the graphical representation.

The visualization V is composed of sentences of a formal language $\mathcal{L}_V$. $\mathcal{L}_V$ is in turn is generated by a grammar $\mathcal{G}(\mathcal{L}_V)=(N,T_V,S_V,\xrightarrow{A})$ where the nonterminal symbols N are tuples $N=(A\times(\mathbb{D}\cup\Delta))$ of a type identifier $a\in A$ together with an element of the input data $\mathbb{D}$ or additional data Δ used to steer the visualization. The type identifier reflects an access relation in the data structure underlying $\mathbb{D}$. In code, the tuple $(position,(0,1))$ in the state "point" would appear as the variable "point.position" where the value stored in the variable is $(0,1)$. The single start symbol, S_V, is the visualization itself. The nonterminals N are processed by a production relation $\xrightarrow{A}$: $(N,T_V)\xrightarrow{A}(N,T_V)$ with the condition that any production can only be followed if the data identifier in the left-hand nonterminal matches the identifier of the production. Finally, the set T_V consists of terminal symbols that actually describe the visualization's compositional structure.

By this formulation, the grammar's start symbol, terminals, and productions directly define a graph that can be interpreted as an automaton consuming a word $w\in N^*$ over the input alphabet N. Transitions $(w_{-1},t_0)\xrightarrow{a}(w_{-2},t_1)$ can only be executed if w is not empty and for the last letter $w_{-1}=(a,x)\in N$ in w the variable identifier a matches the transition. Executing the transition consumes the last letter of w. Where necessary, transitions back to a parent state can be executed without consuming a letter by adding an ϵ-edge that takes the empty word ϵ as the data. This feature is useful if multiple elements of the same type are to be added to a higher-order element like the points in a scatterplot or multiple bars in a bar chart as shown in Figure 4.10. For simplicity, the notation assumes that there is an implicit ϵ-edge going in the opposite direction of any edge in the automaton.

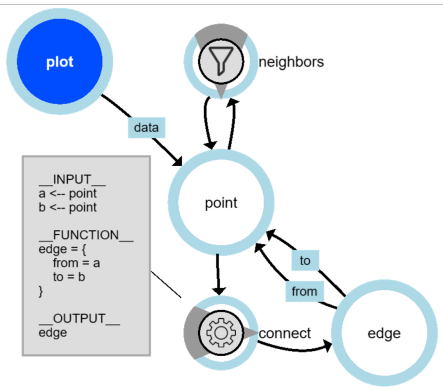
One can already determine the elementary components of the visualization as nodes without outgoing edges other than ϵ-edges leading back to their parent

elements to install siblings of a given representation. These graphical elements are the symbols of lowest complexity, directly shown in the graphical representation, for example as bars in a bar-chart or segments of a pie-chart. Combinations of graphical elements being held by higher order elements form graphical sentences. For example the segments of a pie chart denote graphical elements and the chart itself is a sentence consisting of a set of arranged words. Yet, visualization also needs to consider context.

Being based on formal languages and grammars, it makes sense to apply Chomsky's complexity hierarchy to describe the complexity of the grammars discussed [31]. It divides the formal languages into four categories of complexity of which the model thus far can only emulate the two lowest ones – the regular and the context-free languages. Brushing+Linking is a genuine example for a context-sensitive visualization (cf. Figure 2.5). The appearance of one part of the visualization depends on the user's interaction with another part. An example for Chomsky's Type-0 languages is the change of appearance of some volume rendering if a change is made to a transfer function steering the rendering. This can be achieved by adding two additional special types of transitions – operations modeling transformations and filters selecting data subsets. Both transition types are assigned some code for the computation performed within. The functions they execute can take multiple input parameters from all over the graph – given these states have already been parsed by the automaton. Operations are executed like any other transition, by providing a letter in the input word calling the transition and containing the input parameters as its data. The output of a filter or transformation is an additional word to be concatenated to the current input word. An example run of an automaton featuring filters and transformations is shown in Figure 2.4. The context-sensitive and type-0 constructions model the basis for global reasoning where the interpretation depends on the combination of graphical elements or sentences into graphical contexts acoording to a literate viewer's reading language $\mathcal{L}_R$. The total graphical language $\mathcal{L}_V^{tot}$ of a visualization V hence is the collection of graphical elements, sentences, and contexts. Again, note difference between the graphical sentences and contexts and the sentences and contexts formed by the reading languae $\mathcal{L}_R$ over the words or elements of the graphical language $\mathcal{L}_V$. The reading language formalizes the structures a viewer is capable of reading in the visualization utilizing outside knowledge. In contrast, the total graphical language $\mathcal{L}_V^{tot}$ extends the graphical language $\mathcal{L}_V$ by combining the words $w \in \mathcal{L}_V$ into sentences and contexts based on the syntax defined by the visualization automaton V. Still, the total graphical language and the reading language are defined on top of the graphical language and are likely to share a large intersection. Also note that, as a power set, the total graphical language $\mathcal{L}_V^{tot} = \mathbb{P}(\mathcal{L}_V)$ necessarily contains all of $\mathcal{L}_V$, $\mathcal{L}_V^{tot}$, and $\alpha\mathcal{L}_R$ but is yet another language.

2.6.9 Processing the Graphical Representation

In summary, the automaton V representing the visualization's syntax structure allows to read off graphical elements, sentences, and contexts as the substructures containing certain patterns in the automaton that have been parsed by the chain of state transitions when processing a valid input data word. These



input word	state	transition	remarks
(data, C) (neighbors, [A, B]) (data, B), (data, A)	plot	data	Transition name fits identifier in input word letter
(data, C) (neighbors, [A, B]) (data, B)	point	ϵ	Implicit empty word transition one step back does not consume a letter
(data, C) (neighbors, [A, B]) (data, B)	plot	data	
(data, C) (neighbors, [A, B])	point	neighbors	execution of neighbors-filter finds A and B to be neighbors and extends the input
(data, C) (connect, [A, B])	point	connect	Connection creates new edge and extends input accordingly
(data, C) (to, B) (from,A)	edge	from	
(data, C) (to, B)	point	ϵ	Back to edge, since edge is not yet complete (back to plot would cause failure)
(data, C) (to, B)	edge	to	
(data, C)	point	ϵ	Go back to plot, adding the last point
(data, C)	plot	data	
	point		Input empty → done

Figure 2.4: *An example run of the visualization automaton V for a simple plot of data points adding an edge between points if they meet a certain neighborhood criterion. The input word needs to guarantee that all data for a filter or transformation has been provided prior to its execution. The graph reveals right away that the viewer will be able to read graphical elements and sentences like points and edges reaching from one point to another in the depiction. After running the automaton on the example input, it is clear that there is such an edge ranging from a point A to a point B.*

elements are the foundation for reasoning about meaning. Their interpretation is the input for the processing of semantic information in the mental model M which is described in the following. Prior to this discussion, it is necessary to analyze what actually are the smallest carriers of meaning in graphical descriptions. This Thesis discusses the visualization and the mental model as structures reflecting the relation between information about some commensurable object of discourse and the data sampled from it to generate the visualization. As a consequence, all data must be linked to some information. Because the mental model should reflect this for the visualization, every syntactic structure in the visualization must also represent some semantic information in the mental model. Hence, the smallest graphical element is an element in the visualization that cannot be subdivided into further units that still can be mapped to elements of the mental model. Typically, these are the elements reflecting single data items like representations of points in a scatter plot or the bars in a bar chart. Attributes like the color or size of such objects are not graphical elements as long as they do not specialize the meaning in the mental model, for example by making an explicit difference between red and green points. If the color only indicates the points belong to different clusters, this will not map them to states

in the mental model and hence the color is no graphical element on its own. If, instead, the fact of being a red point or a green point can be mapped to states in the mental model independently of the object's property of representing a point, the point's color is a graphical element. However, a minor adjustment makes the graphical language explicitly distinguish red points and green points in terms of different states in the visualization automaton and thus again reserves the property of being a graphical element to objects rather than their attributes. Hence, despite the small number of exceptions to this rule of thumb, the smallest carriers of meaning commonly are those objects in the graphical display that directly reflect single data items.

Similar to the visualization, the mental model M is also identified as a transduction automaton. Its input is a word of the reading language $\mathcal{L}_R$ over the graphical language $\mathcal{L}_V$ of the visualization V and its output is any semantic information that can be derived from the interpretation $\iota^{\uparrow}(\alpha(v))$ of some word $v \in \mathcal{L}_V$ by following the derivation and inference rules specifying the production rules of the mental model automaton M. Similar to the automaton V reflecting the visualization's syntax structure, there are two kinds of transitions in M. The constraints C model direct semantic relations between situation and object types, just like in situation theory. (Re-)Interpretations R are the semantic counterpart of transformations on data and thus follow the same application principle. Both constraints and reinterpretations can be subject to additional background conditions. Essentially, the mental model is constructed as discussed above in Section 2.6.6. However, instead of being based on the semantic web, the states in the model are now associated with entities and observations being described by logical predicates and functions just like the domain knowledge $\mathbb{I}$. Interpretations in this structure thereby link the information about entities and observations directly to the intensional and extensional semantic information characterizing them. This establishes a dualism between syntactic and semantic information. Just like for the grahical language, the annotation for the semantics is again adopted from a model for the description of semantics in heterogeneous data structures [73].

The possible interpretations $\iota^{\uparrow}(v)$ involving some word $v \subseteq \mathcal{L}_V$ do not necessarily follow a linear structure. As a consequence, multiple starting states will be initiated simultaneously in M. The transitions are being executed nondeterministically and asynchronously and the computation might stop at any time. Note that if multiple graphical elements of the same kind are considered, for example the points in a scatterplot, this also induces multiple interpretations of differently valuated objects of the same semantic type. To model this parallel processing of multiple objects, each state in M is assigned a token holding the semantics for each element under consideration, much like in a Petri net. The map matching the tokens to he automaton's state then denotes the automaton's current configuration. Every transition within M may be executed if there is a token for which its potential background condition holds. Yet, the transition does not have to be executed and the token might just remain in its current state. For reinterpretations, the multiple inputs accumulate until a set of input parameter objects supports the background condition. As soon as this is the case, the reinterpretation can be executed for the aggregated input. The computation might stop at any point in time or if no further transitions are executable. The result then is the situation supporting the semantics of the

automaton's current configuration. To capture the momentary semantics for reasoning, especially where data or appearance attributes matter, the interpretation maps states within the automaton's current configuration to a semantic structure as follows:

Definition: Direct Semantics

The direct semantics associated with a set of states $S \in \mu$ in a subsructure $\mu \subseteq M$ of the mental model M is given as the tuple $\mathcal{D}(\mu) = (S, C, R)$ containing the types represented by the states S, the constraints C, and the reinterpretations R directly connected to them by either an incoming or outgoing edge.

If the mental model M with substruture μ is aligned with a visualization V and the graphical element, sentence, or context aligned to μ is $w = \alpha(S) \in \mathcal{L}_V$, the attributes of w also contain semantic information about the immediate structure. Likewise, potential filters on w add semantics by assigning w to a category based on the filters' respective truth values. Those data relations and filters incident or adjacent to any state $v \in \alpha(S) \subseteq V$ are encoded by the valuation of the extensional and intensional predicates and functions associated with the state S in μ and are thus part of the direct semantics. By this formalism, the graph can be applied to reason about the generally possible interpretations and the actually applicable relations alike. However, this does not apply to transformations. Excluding transformations from the definition of direct semantics is a consequence of modeling transformations as actual processes to be executed on data. Although they thereby establish a map between data items of possibly different semantics, the ability to be processed is not an actual attribute of data. Instead, the ability to process data is a property of the transformation. The semantics of transformations is hence properly captured by adding an interpretation mapping the situation containing all of the transformation's input but not the output data to the situation explicitly containing only the output data. This procedure can, for example, be applied to model user interaction triggering a change of state of the visualization, altering the displayed content. Explicitly reflecting interaction in the mental model is useful for modeling interaction-based reasoning like the provision of details on demand where demand is indicated by a certain zoom level.

2.6.10 The Concept Graph

Thus far, the discussion focused on the output of M. While the output is characterized by now, the input requires some further consideration. The combinations of states in M to be selected as starting points for processing are not arbitrary if the model is meant to be applied for reasoning about a specific visualization. Although, technically speaking, the viewer can combine arbitrary graphical elements to contexts, the visualizations structure and its resulting demands to the viewer's visual literacy constraint this choice in certain ways. For example, a viewer of a scatterplot will not attempt to relate the position of the points to the axes' labels rather than their scales – at least this will not

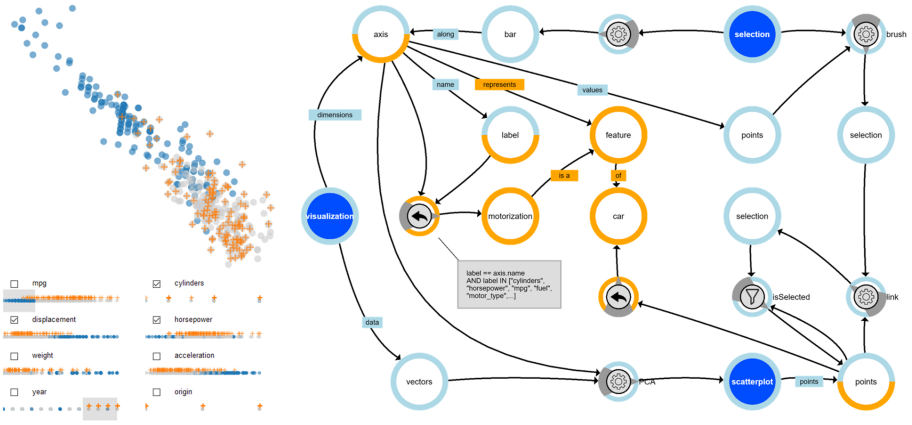


Figure 2.5: *Brushing+Linking is a genuine example for context-sensitive visualization. In the example, brushing ranges along axes marks points in a projection of multidimensional data. The data shows features of a series of cars – which is correctly interpreted by the mental model applied. Some of the features denoted by the axes are associated with motorization. Note that this does not include the displacement. Another viewer might apply a different definition, interpreting the displacement as a motorization feature. If it cannot be guaranteed that two viewers will apply the same definition for their interpretations, this is a potential source for misunderstanding between viewers because of their different iterpretation of the exact same data. (Data: auto-mpg [88])*

provide information useful for the analysis. On the other hand, the assumption that the viewer would implicitly run the visualization on some data and correctly reconstruct the structures representing these data does not reflect reality. Instead, the viewer will construct the sentences or structures over the graphical language as connected subgraphs of the visualization automaton following the transitions between states forwards and backwards alike. Going forwards indicates the interpretations of artifacts and structures being read in the visualization, whereas going backwards organizes elements in sentences and contexts defining the reading language $\mathcal{L}_R$ over the graphical language $\mathcal{L}_V$. Sometimes the viewer will also be aware of a need for some information to be read off the visualization. In these cases, the mental model's reasoning structure will be followed backwards until a mapping to some structure in the visualization can be established. This is exactly the construction of the reading language $\mathcal{L}_R$ originally introduced in Section 2.6.3. Figure 2.5 shows an application of brushing+linking. In this visualization of car data, a viewer can highlight points in the displayed scatter plot by specifying range intervals in the individual data dimensions using a brushing interaction. Such an interaction can, for example, determine all the axes related to motorization. In the example, this results in

the following sentence over $\mathcal{L}_V$:

$$
\begin{aligned}
&\big(motorization\big) - [involves] \rightarrow \\
&\Big(\big(\,(axis) - [named] \rightarrow (cyliner)\,\big) - [and] \rightarrow \\
&\big(\,(axis) - [named] \rightarrow (hp)\,\big) - [and] \rightarrow \\
&\big(\,(axis) - [named] \rightarrow (displacement)\,\big)\Big)
\end{aligned}
$$

Although the condition associated with the interpretation lists more possible names for axes contributing to motorization, the data only provides those three and hence only those three axes are found to be applicable to the description of motorization in the example data set.

Towards a more simple treatment of such an alignment, the automata modeling the visualization V's syntactic structure and the mental model M's semantic information are merged into a new structure, the *concept graph* $G(V, M)$ by aligning the nodes representing the words of the graphical language and those of the reading language following the anchoring and interpretation relations $\alpha : \mathcal{L}_R \rightarrow \mathcal{L}_V$ and $\iota : \mathcal{L}_V \rightarrow \mathcal{L}_R$. The resulting graph contains V and M completely and both of them can be evaluated as before. The concept graph contains three kinds of nodes: Syntax nodes represent the states of V that do not have a counterpart in M. Semantics nodes represent the states of M that do not have a corresponding state in V. The nodes for which such a correspondence can be established, are called concept nodes or simply concepts. The edges of V and M are copied into the new graph. The graph has initially been proposed to describe the different interpretations of data in semantically heterogeneous environments [73]. Figure 2.6 is an overview over the symbols used in the concept graph.

In addition to modeling the structure of the visualization and the mental model, the concept graph has another interesting property: Interpreting the complete structure as the collection of all possible assignments and derivations of data to graphical elements and semantic information as a universe of possible semantics, the concept graph allows to assess the applicable semantics actually supported by the data by eliminating nodes and edges in the graph that are never traversed by either the data processing in the visualization or the processing of graphical elements, sentences, or structures in the mental model. In essence, this means to determine the words of the reading language that can actually be produced by the words of the graphical language obtained from the actual input data. Especially in very complex visualization systems and reasoning structures or, if only subsets of the data are investigated, computing the applicable semantics prior to the study of paths in the reasoning structure can reduce the complexity of further computations on the remaining parts of the visualization and mental model automata.

2.6.11 Semantic Aggregation and Meaning

In philosophy, semiology, and linguistics, meaning is the relation between a sign, symbol or other commensurable entity and the information it represents.

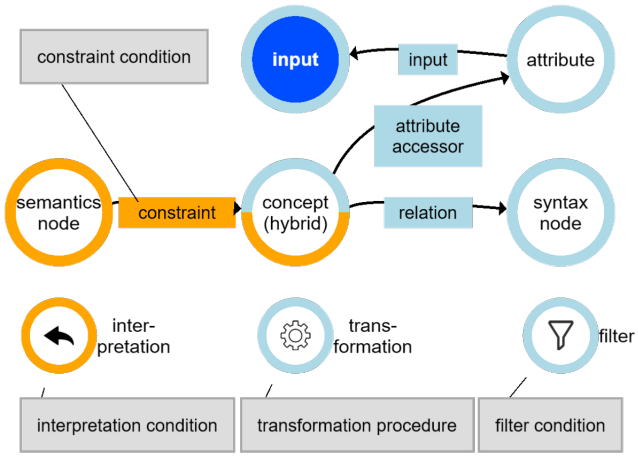


Figure 2.6: *Overview over the conept graph's notation. The concept nodes establish the mapping between the visualization and the mental model. They thus represent the interpretation and anchoring relations. The blue subgraph actually is the automaton V modeling the visualization. The graphical elements, sentences, and contexts that can be read as collections of components in V serve as the input to be processed in the mental model M represented by the orange subgraph of the concept graph. By merging V and M, the concept graph reflects both syntax and semantics of the visualization with respect to the viewer's mental model.*

Other than the interpretation relation, the meaning does not only relate data to some directly associated semantics but instead covers all semantics inferable and derivable from the given data following the constraints and reinterpretation operations. For a visualization, this means that, depending on the mental model provided, different aspects of meaning can be assessed. By the inside-outside principle discussed in Chapter 1, these aspects include the user's personal and general context. The former captures human factors such as perceptual and cognitive capabilities, cultural and individual background, personal preferences and education. The general context provides encyclopedic knowledge, domain knowledge and other context not directly related to the individual. As described above, in a concept graph $G(V, M)$ of a visualization V and a mental model M, the interpretation $\iota^{\uparrow}(\alpha(r)) \subseteq M$ maps a word r of the reading language $\mathcal{L}_R$ to a substructure μ in M. Because the reading language considers structures rather than just elements in the visualization, it would be insufficient, to consider only elements $v \in \mathcal{L}_V$ in order to determine the interpretation of a structure recognized by viewer in a visualization. However, by its construction, the total graphical language $\mathcal{L}_V^{tot} = \mathbb{P}(\mathcal{L}_V)$ neccessarily contains all structures a viewer can recognize in the visualization anf thereby contains the complete part of the graphical language that can be mapped back to the visualization by the anchoring relation. As $\alpha(\mathcal{L}_R) \subseteq \mathcal{L}_V^{tot}$, the anchoring $w = \alpha(r)$ of any word $r \in \mathcal{R}_L$ must also be inside $\mathcal{L}_V^{tot}$. Now, applying the interpretation directly to the graphical element, sentence, or context given in terms of $w = \alpha(r)$, the meaning within the interpretation system determined by the mental model M can be assessed for any word w in the total graphical language $\mathcal{L}_V^{tot}$ by computing the transitive

hull $\iota^{\uparrow}(w)$ over the constraints and reinterpretations in M:

Definition: The Meaning of Graphics

The meaning of any word w in the total graphical language $\mathcal{L}_V^{tot}$ over the graphical language $\mathcal{L}_V$ of some visualization V with associated mental model M is exactly the language

$$\mathcal{L}_{\iota^{\uparrow}(w)} = \{x | x \in \mathcal{L}_R \wedge \iota(w) \Rightarrow_{C,R}^{*} x\} \subseteq \mathcal{L}_R$$

collecting all words in the reading language $\mathcal{L}_R$ that can be derived in M from w's interpretation $\iota^{\uparrow}(w)$.

Expressing this idea in the concept graph $G(V, M)$, the language $\mathcal{L}_{\iota^{\uparrow}(w)}$ for some $w \in \mathcal{L}_V^{tot}$ can be computed by aggregating the direct semantics of all nodes reachable from any component of w's immediate interpretation $\iota(w)$ following the constraints and reinterpretations in G. The equivalence is established by recognizing that following the constraints and reinterpretations in the applicable semantics of G induced by w by a parallel breadth-first search with multiple seed nodes and aggregating the direct semantics of each semantic node or concept reached along the computation into a surrounding situation. This situation necessarily supports all situations being words of $\mathcal{L}_R$ that can be derived from the input w. Its result models an aggregation of semantics, $\mathcal{A}(w)$ of $w \subseteq V$:

Definition: Aggregate Semantics

The aggregate semantics $\mathcal{A}(w)$ of some word w of the total graphical language $\mathcal{L}_V^{tot}$ over some graphical language $\mathcal{L}_V$ for a visualization V aligned with the mental model M by the concept graph $G(V, M)$ is obtained as the set $\mathcal{A}(w) = (S^{\uparrow}, C^{\uparrow}, R^{\uparrow}) = BFS_G^{C \cup R}(w)$ of all semantic nodes S in G traversed by the parallel breadth-first search $BFS_G^{C \cup R}(w)$ over the constraints C and interpretations R in $G(V, M)$, starting in the elements $v \in \mathcal{L}_V$ that w is composed of.

Again, note that $\alpha(\mathcal{L}_R) \subseteq \mathcal{L}_V^{tot}$. The computation of the aggregate semantics can therefore also be applied to every structure a viewer or analyst is capable of recognizing and interpreting in the visualization V given the mental model M. Being the situation under which every direct semantics of every reachable combination of nodes in $BFS_G^{C \cup R}(w)$ can become a fact, the following theorem holds for the aggregate semantics:

Theorem: Aggregate Semantics and the Meaning of Graphics

Given a visualization V, a mental model M, and the concept graph $G(V, M)$ combining them, the aggregate semantics supports all possible interpretations of any word $w \in \mathcal{L}_V^{tot}$ – and hence its meaning. Thereby, the following holds for the aggregate semantics $\mathcal{A}(\iota(w))$ of some word, sentence, or context $w \subseteq \mathcal{L}_V^{tot}$:

$$\forall w.\ w \in \mathcal{L}_{\iota\uparrow(v)} \rightarrow w \in \mathcal{A}(v)$$

where $\mathcal{L}_{\iota\uparrow(w)} = \{x | x \in \mathcal{L}_M \wedge \iota(w) \Rightarrow^* x\}$.

For the proof, recall that by the definition applied for the construction of the mental model, M is the collection of all direct semantics derivable from all words of the reading language of $\mathcal{L}_R$. The reading language in turn determines all words, sentences, and concepts over the graphical language $\mathcal{L}_V$ for which an interpretation actually exists. The direct semantics are contained in the breadth-first-search result by construction and thus must be part of the aggregate semantics. □

Unfortunately, the converse does not hold, because $\mathcal{A}(\iota(w))$ aggregates direct semantics along whole paths whereas the meaning $\mathcal{L}_{\iota\uparrow(w)}$ only considers direct semantics of reachable node sets in the mental model. A simple example of this discrepancy is that $\mathcal{A}(\iota(w))$ theoretically allows multiple reinterpretations to hold in parallel, whereas $\mathcal{L}_{\iota\uparrow(w)}$ forces a decision. Note that this is actually an effect oberved in optical illusions where sometimes two possible interpretations exist simultaneously and the human brain – while being totally aware of the simultaneous coexistance of both sructures – can only switch between either one for interpretation. In arts, this effect is alled bistability and is only one example for a larger class of effects leading to semantic instability in graphical depictions [97]. Nevertheless, the aggregate semantics serve as an efficient way to test whether some combination of direct semantics is a candidate of being part of the meaning. The problem is that due to the necessary distinctions of what can and cannot be combined in $\mathcal{L}_{\iota\uparrow(w)}$, the computation of the language determining the meaning of w is extremely costly. The problem can be reduced to finding the set of reachable states given some start configuration in a petri net – which can be shown to be at least NP-hard for all relevant problems (cf. [49]). The computation of $\mathcal{A}(w)$ instead is bound by the size of M as the largest possible direct semantics of any node set. The aggregate semantics can hence be computed in polynomial time and is bound by $\mathcal{O}(|N|^2 + |C|^2 + |R|^2)$ if the mental model M consists of the nodes N, constraints C, and reinterpretations R. The squares are a consequence of the requirement that the sets in $\mathcal{A}(w)$ are defined as disjoint unions of the sets contained in the direct semantics of nodes passed while executing the breadth-first search.

2.6.12 An Application Example

Towards an example applying the concept graph in conjunction with the qualitative visual analysis cycle, consider a relatively simple example from flow visualization where the visualization is to be used for the investigation of 2d vector fields in order to find vortices in those fields. The information $\mathbb{I}$ about the investigated phenomenon hence is known to contain information about a vector field and its behavior. From an interview with a domain expert, a visualization designer learns that vortices are the result of vortical motion of particles over time. Even though it is not entirely clear what this vortical motion might be, the information that vortices are identified if vortical motion is found is definetely part of $\mathbb{I}$. The available data $\mathbb{D}$, however, only contains a set of 2d vectors sampling particle positions X, masses M, and impulses P over a time interval T as the result of a simulation run.

The mental model M shown in Figure 2.7 reveals that the viewer will be able to identify a vortex in a field of particle traces by investigating their rotation. Note that this is a qualitative consideration so the viewer will be able to assign this interpretation to traces without exact computations on the data. Indeed, this amount of detail is sufficient to assess whether the viewer will in principle be able to make a correct inference this amount of detail is sufficient. However, the information about the phenomenon only contains state information. The visualization V thus has to close the gap between momentary state and trace over time. This is achieved by computing the particle trace information by integrating the particle state over time. Aligning the mental model M with V now shows that the viewer is indeed capable of concluding the existence of vortices from within the visualization. For the proof, assume that $\mathbb{D}$ actually contains a vortex which is correctly depicted by the path traces generated in V. The aggregate semantics $\mathcal{A}(w)$ of any trace being part of the vortex now contains a state where $w \in \mathcal{L}_V^{tot}$ is identified as being part of the vortex. Since the pathline has a corresponding structure $pline \in \mathcal{L}_R$, the viewer can recognize the it as $w = \alpha(pline)$ and apply those interpreations inside $\mathcal{A}(\alpha(pline))$ that yield the interpretation as being part of the vortex within the mental model. Therefore, the viewer is able to detect the vortex.

The second application is to reason about what else the viewer might find in the visualization. Note that M also contains the information that the viewer can determine whether a vortex has a source or sink by additionally reasoning about the parallelism of lines shown in V – which is also part of the aggregate semantics $\mathcal{A}(w)$ for the corresponding structure $w \in \mathcal{L}_V^{tot}$. If such an interpretation is applicable to any part of the data, this is valuable insight into the data.

However, it is not yet guaranteed that the viewer's predictions about the occurence of a vortex are correct. Assume that an interview with a domain expert reveals that vortices are characterized by nonzero vorticity in the field. The computation of vorticity can be added to the visualization V. However, the mental model M does not yet contain any structure capable of processing this additional information. Adding a legend to V that explains how the vorticity is related to finding vortices in the field allows the viewer to adjust the mental model M to process this additional information – that is, to add additional states with the proper direct semantics. Such an improved understanding of

key features of the visualization and added methods for their processing allow the viewer to draw more accurate conclusions.

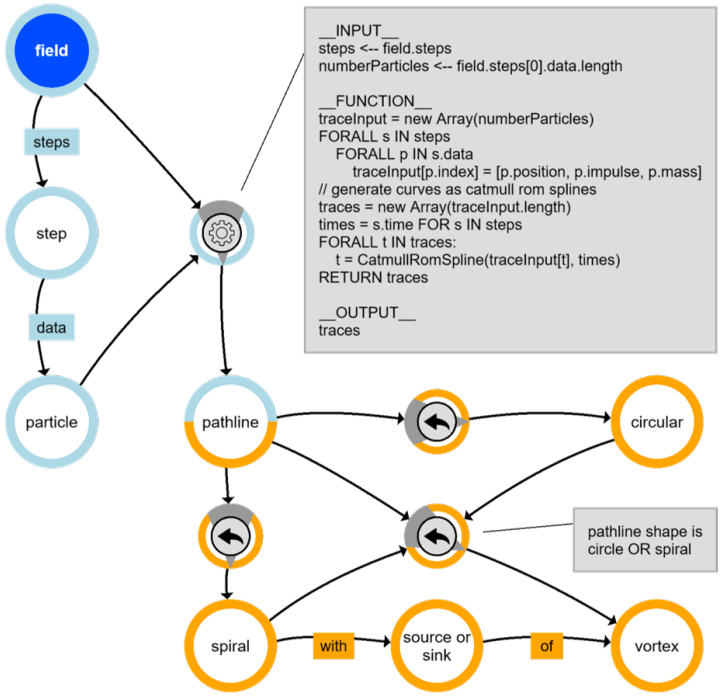


Figure 2.7: *Example concept graph for a visualization designed for the identifaction of vortices in vector fields. The data is a set of particle configurations listing the mass, impulse vector, and current position for a fixed set of particles moving in the flow field over some time. The mental model represents domain knowledge and implicit user experience and indicates that the property of being a vortex can be derived for path lines rather than pont clouds. Hence, a visualization like a series or animation of hedgehog plots would not be compatible with the mental model. The designer thus needs to add a transformation computing the path lines from the simulation steps in order to provide the viewer with a data representation from which the inquired information on vortices can be found by processing the graphical display in the viewer's mental model.*

2.7 On the Complexity of Reasoning with Visualizations

Determining the complexity of the different transformations in the qualitative visual analysis cycle reveals insight into the complexity of the reasoning process. In the following, the theoretical models introduced above are discussed with respect to aspects that contribute to the complexity of reasoning about and with visualization. To this end, the quantitative and qualitative factors contributing the the complexity of the computation of the different transitions between steps

in the qualitative visual analysis cycle are identified and mapped to the three categories of insight derived from the literature in Chapter 1. The complexity model reflects the findings reported there and extends them by identifying the actual factors contributing to the complexity of the individual insight category. This discussion results in a third fundamental principle of qualitative visual analysis serving as a general design guideline.

The qualitative visual analysis cycle features two transitional relations encoding the data into the visualization and reflecting the information encoded in the mental model in the observed information domain. In addition to that, two transitional states consist of pairs of relations translating between the domain information and the data as well as between the the mental model and the visualization. Insights are obtained by constructing those mappings. Hence, determining the complexity of the reasoning process requires to assess the complexity of the generation of the four mappings. Each complexity has a quantitative and a qualitative component, the former of which is concerned with the number of elements to be transformed, the latter with the complexity of the translation between the formal languages or data formats involved. As it turns out, for each of the mappings one component dominates the complexity while the other either cancels out or contributes only little to the overall complexity. Based on the function of the respective mapping in the qualitative visual analysis cycle, the following four complexities are identified:

The sampling complexity is the complexity of the construction of the pair σ/ρ linking the domain information $\mathbb{I}$ with the data $\mathbb{D}$. Since ρ is the partial inverse of σ and every addition to σ is automatically reflected by ρ, the complexity reduces to that of the reconstruction of σ. σ is reconstructed from the mental model by assigning information resulting from reasoning about the visualization to the data. The assignment itself is a trivial operation assigning a number of logical formulas to a number of data elements. The actual reconstruction is thus dominated by the number of the objects and situations providing the additional information and the size of the data contributing those objects and observations. The sampling complexity is thus governed by quantitative considerations.

The graphical complexity is the complexity of the encoding relation $enc : \mathbb{D} \to V$ generating the visualization from the data. Its qualitative component is the complexity of the concept graph's syntax part. Although this computation can be arbitrarily complex, the complexity is dominated by the quantitative part which is determined with the number of data items to be processed.

The reasoning complexity determines the complexity of the relation pair α/ι translating between the graphical language and the reading language. As long as the visualization remains unchanged, the number of graphical elements remains the same. Hence, the quantitative component of this complexity is constant for every moment in which reading the momentary state of the visualization triggers the reasoning process. Its complexity is thus dominated by qualitative considerations, namely the complexity of the mental model and the

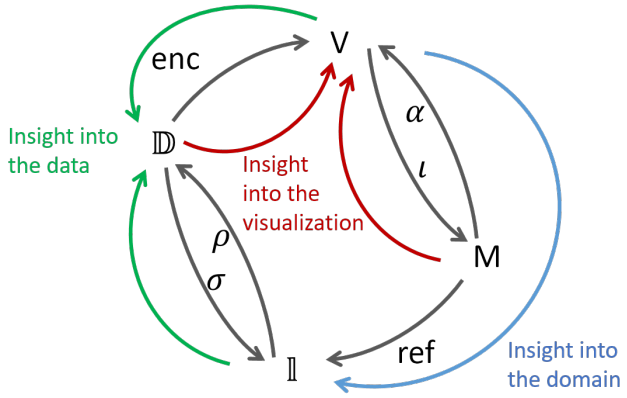


Figure 2.8: *Information flow in the qualitative visual analysis cycle for the three different categories of insight. Remarkably, insight into the domain is the only type of insight for which the flow is unidirectional.*

computation of the applicable semantics in the formalism of semantic aggregation.

The information complexity is the complexity of the reflection relation $ref : M \to \mathbb{I}$ mapping the mental model back to the domain information. Like for the reasoning complexity, the number of entities and observations in the mental model to be mapped back to corresponding entities and relations in the domain information remains constant during each execution of the relation. The qualitative part of this complexity is determined by the complexity of the predicates and functions to be matched between entities and observations in M and $\mathbb{I}$. Although proper naming conventions can simplify this process significantly, its complexity in essence depends on the logics applied to describe the information to be mapped between M and $\mathbb{I}$. Therefore, this complexity measure is also dominated by qualitative considerations.

Figure 2.8 provides an overview over how the different kinds of complexity identified above contribute to the three categories of insight discussed in Chapter 1. Insight about the visualization involves the mental model and the data. Its complexity is hence determined by the graphical and the reasoning complexity. Insight about the data is obtained from the reconstruction of σ. Since this requires to map the artifacts and structures in the visualization back to the data that created them, the contributing complexities are the graphical and the sampling complexity. Insight about the domain is obtained from reasoning about the graphical display and mapping the interpretations back to the domain information. Hence, the contributing complexities are the reasoning and the information complexity. In the following, the three types of insight are reviewed in further detail regarding their respective complexity.

2.7.1 The Complexity of Insight into the Visualization

Despite being the most fundamental form of insight according to the literature cited in the discussion of the three categories in Chapter 1, insight about the visualization can be surprisingly complex to obtain due to its mixed character featuring quantitative and qualitative contributions. The quantitiy of artifacts and structures in a graphical display can be quite overwhelming. Too much detail can hinder the understanding of the visualization by making the viewer process too complex definitions of objects, especially if this requires to consider a large number of conditions on the constraints and interpretations. Shneiderman's mantra of overview and detail is a very popular strategy of coping with this problem [129]. However, there are scenarios where this strategy is not applicable, because an overview is not directly available but rather has to be constructed by carefully collecting and combining available hints. An example is digital forensics where the evidence yields the understanding of the case and not vice versa. Of course, overview and detail is again a good strategy to present the results of the investigation. Applications where overview and detail is an appropriate choice of the presentation strategy are typically relatively fixed in their structure. In those applications, interaction is mainly offered for navigation and filtering. Understanding the visualization in this kind of application thus primarily depends on the graphical complexity and is hence dominated by quantitative concerns. The bottom-up strategy is instead required in cases where visualization applications are characterized by the aggregation of data from various previously unconnected data sources, probably across separate views. They commonly offer powerful interaction mechanisms to change the depiction and the data alike. In such a setup, analysis usually suffers from the fact that at least in the beginning understanding is dominated by outside knowledge rather than by what can be read off in the visualization. This kind of visualization application is hence dominated by qualitative complexity considerations in the beginning although this gradually shifts to quantitative considerations during the analysis process as the visualization becomes more complex due to the interaction extending the data and the display.

2.7.2 The Complexity of Insight into the Data

Both components of the complexity of finding insight into the data are dominated by quantitative considerations. It is thus not surprising that this is the complexity best reviewed in the state of the art given the literature's focus on quantitative considerations. The complexity of this kind of insight is commonly assessed by cost-based approaches (e.g. [28,143]). In the qualitative visual analysis cycle, the complexity of insight into the data is governed by the number of objects and situations found in the visualization about which the additional information is to be mapped to the data in order to reconstruct σ and by the number of data items affected by this mapping. The actual insight about the data is a consequence of establishing the connection between the visualization and the mental model which allows to associate knowledge from outside the data with the data by associating it with words, sentences, and contexts over the graphical language and mapping it back to the data used to generate those

parts of the grahical display. The actual mapping is rather trivial since it only assigns the functions and predicates describing the outside knowledge to the corresponding data. Even if this requires the creation of an additional set or class inside the data, the complexity is still governed by the collection of the data in the corresponding class rather than by adding the description. Since ρ is the inverse of σ its reconstruction does not add additional complexity to the finding of insight into the data. In the next chapter, an explicit augmentation of the data by insight about the data is discussed as a workflow amplifying the obtainment of new insight by making the results of previous analysis steps accessible to further analysis.

2.7.3 The Complexity of Insight into the Domain

Insight about the domain is the only one of the three categories that follows a unidirectional path along the transformation arcs in the qualitative visual analysis cycle. This actually matches the current standard model of visual analytics which also implies a unidirectional flow of information from data to knowledge [74]. Interestingly, it consists only of the complexities dominated by qualitative considerations. This brings up the question whether cost-based approaches as they are applied for the complexity of finding insight into the data are actually capable of appropriately assessing the complexity of finding domain insights. In fact, when reasoning about components of visualizations, the components are fixed during the actual reasoning process. Large numbers of elements are abstracted into observations simplifying the reasoning. This is either achieved explicitly with the help of interaction or implicitly by interpretations inside the mental model. The implicit variant can actually be observed when analysts comment on their reasoning strategy. Rather than reflecting vague or ambiguous definitions, characterizations like "this number of points" or "that structure over here" actually refer to an observation that is clearly defined within the analyst's mental model. What makes this description appear unclear or ambiguous is the fact that mental models and hence the definitions differ between different analysts. Further research into the direction of this kind of qualitative reasoning is duely needed to make the potential of reasoning with and about qualitatively defined objects and situations accessible to visualization applications. It should be pointed out once more that the notion of being qualitative in qualitative visual analysis does not refer to qualitative data but is to be understood as the explicit consideration and discussion of the qualitative aspects of data analysis contributing to the analysis results. In particular, those aspects are the interpetation of data, the complexity of reasoning, and the provenance of insights. Qualitative characterizations of objects and situations in the visualization are obtained from reasoning within the mental model M associated with a visualization V to obtain the reading language $\mathcal{L}_R$ over the graphical language $\mathcal{L}_V$ that in turn determines the depiction of artifacts and structures in the display. The qualitative descriptions are a result of the combination of the visual data representation with outside knowledge constructing the words in $\mathcal{L}_R$. Furthe research should not try to avoid such a scenario but instead explore techniques to allow analysis to embed their outside knowledge into the visualization, for example by proper annotations of the anchoring $\alpha(v)$

of a word $v \in \mathcal{L}_R$ in the visualization. Enabling analysts to share their outside knowledge when working with visualizations is especially important for collaborative setups where the exchange of knowledge between viewers with different backgrounds is likely to yield better analysis results or more informed decisions.

2.7.4 A Qualitative Design Principle

The above discussion of the three categories of insight and how they map to the different kinds of complexity associated with the transitions in the qualitative visual analysis cycle reveals some remarkable observations. Recall that the literature referenced in the discussion of the three categories of insight in Chapter 1 reports findings about an ordering of the insight by ascending complexity, namely insights about the visualization, about the data, and about the domain. Combined with those findings, the discussion in this section indicates that at least by the differenct kinds of complexity identified in the discussion of the qualitative visual analysis cycle, qualitative reasoning in general seems to be harder than quantitative reasoning. Interestingly, the studies and models referenced in Chapter 1 report insight into the visualization to be the least complex kind of insight. Yet, the theoretical model developed in this chapter partly contradicts those findings because insight about the visualization combines qualitative and quantitative complexity. Indeed, it is not hard to imagine a visualization that is extremely hard to decipher but might be quite effective in supporting insights about the data or even about the domain once the viewer learned to read it correctly and fluently. Still, this should rather be prevented by visualization design since learning how to work with an overly complex visualization is likely going to provoke more frustration than the benefit of finding insight thereafter can compensate.

In any case, it appears that the visualization should attempt to support reasoning by representing the data in a way fitting the mental model and by displaying only what is necessary and relevant for the analyst to obtain the inquired knowledge. Note that this does not mean to focus on solving individual tasks without paying attention to any other information probably relevant for further analysis and potentially yielding interesting insight. However, the question is whether the analyst is actually interested in this additional information, whether it is relevant for the question at hand. For example, when analyzing email traffic to identify the message exchange about finance transactions in some sort of fraud analysis, it might be an interesting insight that one of the communicating persons received also quite a lot of advertisement, for example for medication. Yet, this insight is not immediately relevant for the analysis and indeed distracts the analysis from the actual task at hand which is to find the emails regarding the finance transactions. It therefore can safely be ommitted from the display. Note, however, that this information should never be deleted completely as it can indeed become relevant in a later step of the analysis process. For example, the analysis of the finance transactions could reveal that the bank account information for the transactions was hidden in the advertisement emails. Even if so, this is another part of the analysis and the advertisements are still not relevant for the initial task. Although it should be carefully considered whether or not data should be ommitted from

the generation of a visualization, asking this question is necessary to follow the principle of minimal graphical overhead. Since the information complexity and the sampling complexity cannot be influenced by the visualization, the graphical and the reasoning complexity should be minimized. In fact, minimizing those two complexities actually reduces the complexity of finding insight for all three categories. The graphical and the reasoning complexity are the complexities associated with the execution of the syntax and the semantics parts of the concept graph. The concept graph therefore is not only a descriptive model to reason about the formation of mental models from visualizations but can also be applied as a generative model allowing to tweak the design of visualizations towards optimal performance with respect to anticipated reasoning chains. Yet a complete treatment of the complexity of executing the syntax and semantics parts of the concept graph requires a deeper understanding of the mental models applied by analysis and therefore has to be left to future work. The rationale behind this is that even though the complexity of the mental model can be minimized theoretically, there is no reason to assume that an analyst would automatically apply this optimal mental model. Therefore, studies about the formation of mental models should be preferred over a purely theoretical consideration and the design of visualizations should be adapted to the reasoning strategies actually applied by viewers rather than attempting to form those strategies. In fact, the designer of a visualization cannot shape the way an analyst is going to reason about the graphical representation. Even if the anticipated mental model is theoretically optimal, the visualization will be harder to use the more the analyst's actual mental model differs from the anticipated reasoning structure. This conclusion is more than just a guideline but actually installs a third fundamental principle of qualitative visual analysis:

Principle: Design for Reasonig

Visualization should support the way analysts think, not attempt to shape it. The visualization design thus must be adapted to the mental model, not the mental model to the visualization.

2.8 Discussion and Prospect

Towards a theoretical framework for qualitative visual analysis respecting the inside-outside principle, this chapter proposes a formal model for explorative and task-based reasoning with visualizations. Further formalization of the model's individual steps and the transitions between them results in a generally applicable theoretical model for the structures actually being read in a visualization given a predetermined set of rules to be applied to understand and interpret the artifacts and structures displayed in the visualization. Applying the qualitative visual analysis cycle to describe the visualization process guarantees that the qualitative aspects of the analysis process are explicitly considered in the discussion. Other than most existing theoretical descriptions for the analysis process, the qualitative visual analysis cycle does not take the obtainment of insight for granted once the corresponding structure is found in the visualization. Instead,

it asks explicitly for the exact composition of structures to be perceived and how they are meant to be processed by the viewer's cognition. This description is achieved by the more detailed model describing the individual steps along the analysis cycle. Applying formal languages and grammars for the construction, the theoretical framework fits well into the currently dominant approach to theoretical reasoning about visualization being based on modeling the perception of artifacts and structures in terms of messages to be evaluated by information theory. For example, information can be applied to determine the likelihood of a word of the reading language to occur in the visualization or to be perceived by the viewer. This of course requires to apply the methodology described above to determine the graphical language and the reading language defined on top of it and thereby define the messages to be exchanged between the visualization and a viewer or analyst. Although this is an interesting direction for further work, it exceeds the scope of this Thesis which is the investigation of the reasoning and cognitive processing rather than the chance to identify a given structure. However, it should not go unnoticed that the perspective presented here is in some way as incomplete as the work focusing only on perception. Where other work assumes insight to be found as soon as the corresponding structures are perceived, this work assumes structures to be perceived as soon as they appear. In the future, both perspective will have to be combined towards a holistic perspective of the reading and reasoning process. However, this Thesis is concerned with the reasoning process as this is an actual gap in the liteature whereas issues of reading and perception are covered more widely, especially in literature on the human factor in visual data analysis. For this reason, perception issues are widely omitted in the discussion. The model introduced in this chapter provides the basis for the discussion throughout the remainder of this Thesis.

Formalizing the qualitative visual analysis cycle aims at establishing a theoretical framework for the description of the different steps along the workflow modeled by the cycle and the transitions between them. Such a formalization allows to reason about an analyst's interaction with visualization applications and about the reasoning behind the conclusions drawn from working the visualization applying formal logics to define predicates and formulas describing the procedures followed by the analyst. This includes proving the ability of the analyst to solve certain tasks using the visualization in conjunction with a certain mental model defining the inference rules that will be applied by the analyst to reason about the graphical representation of data and hence enables researchers to investigate and prove hypotheses and theorems on the correctness and completeness of the results obtained if the visualization is studied by a specific analyst or group of analysts. Applying general reasoning and perception principles like the well-known Gestalt laws as the rules underlying the reasoning process and combining them with abstract representations of visualization components, universal theorems can be formulated, proven, and applied to derive further theory about the readability of visualizations and the visualization literacy of viewers as well as the inferences an analyst is able to make.

Applications of the model require at least two of $\mathbb{I}$, V, and M to be known in order to obtain results about the respective third one. The most common case is that the analysis asks for yet unknown domain infomation being part of $\mathbb{I}$. This scenario is the initial situation of exploratory analysis and other analysis setups asking an open analysis question. In such a setup, every addition

to $\mathbb{I}$ can be considered an insight. If M is the unknown model, the focus is not on drawing conclusions but on obtaining an understanding of a scenario. Typical examples are situational awareness and other applications centering on the communication of overviews and detail information about the object of discourse to the user – often as the preparation for subsequent analysis. If V is unknown, the available domain knowledge determines the known parts of $\mathbb{I}$ and $\mathbb{D}$ that in turn allows tailoring the visualization to a specific user or, for example if M actually represents encyclopedic domain knowledge, even a class of users. This is actually the common setup in visualizaion design studies and describes the derivation of optimal graphical representations from knowledge about the general domain ($\mathbb{I}$) and the specific application (M). In general, a mixed form of these problems will be encountered and the focus will gradually change over time. Starting with the design, the focus is on determining a good visualization. Getting an overview over the situation then puts the focus on extending M to match the visualization and subsequent analysis reveals insights extending $\mathbb{I}$. Depending on the type of insight to be predicted, different parts of the qualitative visual analysis cycle need to be considered. According to the three levels of insight complexity discussed in Chapter 1, those levels are insights about the visualization, insights about the data, and insights about the domain in ascending complexity. Insights about the visualization require to identify the concepts determining the entry points of the reasoning process by establishing the direct link between the graphical language $\mathcal{L}_V$ and the reading language $\mathcal{L}_R$. This requires knowledge of M and V. Insights about the data are formalized by the reconstruction of the representation relation η from extending the sampling relation $\sigma : \mathbb{D} \rightarrow \mathbb{I}$ by knowledge obtained from interpeting the visualization. This essentially means to map the available outside knowledge to the artifacts and structures in the visualization and to additionally associate this information with the data encoded into those structures and artifacts by following the *enc*-relation mapping the data to the visualization backwards. Again, this requires the knowledge of M and V, but also some information about the raw data and the information it is meant to sample from the domain. The most advanced type of insights, insights about the domain require establishing the reflectance-relation $ref : M \rightarrow \mathbb{I}$ linking the mental model to the domain information. To achieve this, the mental model needs to be equipped with a more sophisticated definition of entities and observations than in the other cases. Because this requires knowledge about the predicates and functions formalizing the domain knowledge, establishing this kind of relation requires the knowledge of both M and $\mathbb{I}$. Interestingly, the formation of this kind of insight can be described completely without knowledge about V. This opens interesting opportunities for visualization design enabling the tailoring of visualizations towards a mental model M that is constructed exactly towards answering a question about the domain by reflecting a respective hypothesis. Of course, this kind of hypothesis has to be tested against the data. However, data does not form entities and observation but rather instantiates them as a collection of objects and situations. Validating the hypothesis against the data hence means to predict the applicable interpretations of the visualization based on the mental model and the available data. This is achieved by the notion of semantic aggregation.

The proposed formalism of semantic aggregation has been shown to be applicable to the description of interpretations of a wide range of graphical repre-

sentations and to be especially useful to assess the possible conclusions viewers are able to draw from analyzing data using visualizations. Its modularity and extensibility allow to tailor the model to a given problem, such that – while maintaining its general applicability – its predictions remain accurate even for extremely problem-specific questions. Together with the concept graph as a model for the structural composition of graphical displays and the closure capabilities and strategies applied in the mental model used to understand, interpret, and analyze the depicted information, semantic aggregation is capable of describing insight as the gain of information previously unassociated with the graphical representation both qualitatively as the reconstruction of a mapping from the data to the new information that has been derived from it and quantitatively by the number of graph elements contained in the new information's direct and aggregate semantics as indicators for the new information's direct complexity and total influence on further information inference.

The concept graph provides only one way to establish a formalism allowing to reason within the framework of semantic aggregation. Although it is replaceable as the underlying model, it comes with an appealing set of features. First, it allows to process syntax and semantics in a single model and to directly link semantics to complex interpretations. Second, its graphical representation makes it comparably easy to apply. The visualization part reflects the visualization's design while the semantics part reflects association chains and the possibility to combine semantic ideas into other ideas. In this form, it appears much like a mental map with an additional classification feature. However, the model's descriptive power depends on the logics applied to specify the semantics.

Although the concept graph's original form [73] does not provide the formalism added in this work, it already introduces another appealing feature: The idea that the data carve out a set of actually applicable semantics from a universe of possible semantics allows to formalize the formation of observations made by experiment within the realm of a theory and to dynamically adapt the interpretation and conclusions that can be made within the theory to the experimental observation. The third major property of the concept graph as it has been formalized in this Thesis is the formalism's scalability. While modeling the structure of graphics rather than the elements contained, it does not scale in the data but in the complexity of the graphical display. Likewise, the semantics model scales in the complexity of the theory applied for the analysis. As a consequence, the representation remains compact even if large amounts of data are being displayed and processed. Traversing the concept graph yields the possible or applicable semantics that can in principle be derived from the data. By the redefinition of the concept graph's syntax and semantics components to automata, actual data and structures observable in the visualization can still be processed explicitly by running the respective automaton.

In the current form, semantic aggregation and the concept graph still have some limitations. A quite obvious issue is that even though the model directly reflects the structure of graphical representations or closure strategies of the mental model, strategies are needed for efficient modeling of these procedures, especially for very complex structures. The heuristics applied to the generation of this chapter's content were to start with lower complexity and to add the filter, transformation, and reinterpretation operations only later. This allows to

quickly get an idea of how the final model will look like but the late addition of complex structures sometimes requires adaptations. Although identifying the need for such adaptations during modeling rather than during programming is actually quite useful, there must be strategies how to efficiently generate the model – especially regarding the formulation of complex domain knowledge and adapting the visualization to this context as it is a task often encountered in visualization design. Moreover, as of now, quantitative reasoning can only be modeled by the installation of filters and prefiltering the data. A simple fix to this problem is to slightly relax the definitions and allow the mental model to process the results of filters as data attributes. After all, these results are already included in the direct semantics associated with a concept. However it is still a challenge to model relative definitions like a set being much larger than another. This becomes more challenging the more the perception of such a feature is specific to a specific person's interpretation. Currently, this can be achieved by exploiting the nondeterminism of the automaton representing the mental model and simply adding interpretations modeling these states. Which interpretation is applied depends on the user. In the future it might be necessary to add some additional determinism to the mental model to solve this problem. Such an additional determinism is also relevant for the constraints since as of now they merely model associations than conclusions and background conditions on constraints – other than evaluating filters – are more of theoretical relevance than of actual use in the model.

Towards more detailed descriptions of the mental model, future work on the theory will investigate the dynamic addition of logical reasoning by installing a mechanism for the invocation of oracles to determine semantic properties of the situations being processed in the mental model. Akman and Surav [2] successfully applied such a model to information retrieval. An advanced version of their technique, resolving some limitations of their approach, should be applicable to the theoretical framework of semantic aggregation. Although the model scales well in its descriptive capabilities and the execution of the automata – after all the visualization and mental model can be parallelized to a large extend – computational efficiency is likely to be an issue for very large data sets. To this end, suitable data structures and computation methods have to be identified to allow quick computation of situations even for large data sets. Interactive computation times will enable new forms of interaction based on the semantics and on the information to be found in data rather than on the data or its graphical representation. Demiralp et al. have proven such attempts to be useful on a limited set of information to be found proposing what they call insight queries [39]. It will certainly be interesting to investigate an extension of their findings to general information.

2.9 Summary and Conclusion

In this chapter, a model of the qualitative visual analysis process is introduced along with a formal treatment of the individual steps in the model. The result is a theoretical description of the structures viewers are expected to read in visualizations based on the required input for the reasoning processes determining how

viewers interpret the displayed information. Being based on the notion of formal languages and the exchange of messages between the visualization and the viewer or analyst interpreting it, the formalism is well compatible with the currently dominant direction in visualization theory describing the reading process in terms of information theory. However, where the focus of the information theoretic approach is on the occurrence and perception of messages, the discussion here focuses on the interpretation of messages after they have been perceived and on the construction of the messages a viewer or analyst actively tries to perceive in the visualization. The theoretical considerations yield four central results. The qualitative visual analysis cycle provides a scheme for a description of visual information analysis explicitly considering a major qualitative aspect of visual information analysis, namely the reasoning to be applied to the data. The principle of minimal graphical overhead is a direct consequence of the discussion about which messages can theoretically be read from the visualization and which messages actually make sense to be read. The third major result is the relation between the mental model and the reading language allowing to formalize the interpretation of the graphical display in a structure enabling the comparison between the models of different viewers. This comparison is not only important for the understanding of how different viewers and analysts work with visualizations but also reveals potential sources of misunderstanding or ambiguous interpretation by comparing the reasoning and interpretations of users applying different sets of outside knowledge to understand the visualization. The formalism of semantic aggregation denotes the fourth major result, providing a computable prediction of the domain information an analyst is able to conclude from the graphical presentation based on the mental model. It generates a language of the possible semantic information being associated with graphical representations of data as the semantic information the graphics are intended to convey. Together, the qualitative visual analysis cycle, semantic aggregation and the concept graph close a gap in visualization theory, combining theory on the structure and organization of graphics with reasoning strategies, knowledge, and other human factors determining the information an analyst can obtain from a graphical depiction.

Core References

- B. Karer: A Language-Based Framework for the Interpretation of Graphical Displays. In *IEEE Transactions on Visualization and Computer Graphics*. IEEE, (in preparation).
- B. Karer, D. Fernández-Prieto, and H. Hagen: The Situation Universe: Visualizing the Semantics of Integrated Data Structures. In B. Kozlikova, T. Schreck, and T. Wischgoll (Eds.), *EuroVis Short*. The Eurographics Association, 2017
- B. Karer, I. Scheler, H. Hagen, and H. Leitte: ConceptGraph: A Formal Model for Interpretation and Reasoning During Visual Analysis. In *Computer Graphics Forum*. The Eurographics Association, 2020.

Chapter 3

Workflows Inspired by Qualitative Visual Analysis

The second question in focus of this work is the finding of efficient workflows for qualitative visual analysis processes. In the first part of this chapter, a structured workflow streamlining the flow of information along a cycle of data mining, visual analytics, and machine learning steps is presented. Insights found during any step of the analysis are captured as additional information that can be processed during further analysis, resulting in an amplification of insight towards an increasingly rich source of semantically meaningful views on the data being aggregated during analysis. An explicit augmentation of the data with insights about the data makes the interpretations of artifacts and structures in the visulization accessible for further analysis. Thereby, semantic information is included into the data analysis process which is why the proposed method stands apart from similar approaches. Interacting not only directly with the data but also with meaningful substructures featuring clear and documented semantics allows a more intuitive support for reasoning than abstract interaction on the data level would. Towards maximizing the efficiency of such an analysis workflow, the second part of this chapter discusses the feasibility of automated generation of visual analytics pipelines based on qualitative considerations. Where insight provenance reports the reasoning strategies being applied by viewers and analysts, automatic visualization generation relies on finding representations enabling to draw the conclusions needed to answer the analysis question. The tight binding of visual data encodings and their semantics discussed in this Thesis inspires the idea to determine paths of consecutive transformations for the generation of visualizations by the information to be obtained from those visualizations rather than only by the structure of the available data. The discussion yields an algorithm that allows to ask for visualizations supporting complex reasoning processes in order to foster the finding of inquired insight. Those visualizations are to be composed from a set of available data transformations and visualization components.

3.1 Related Work

The integration of automated data analysis and visualization is the fundamental idea behind visual analytics. The classical visual analytics pipeline as proposed by Daniel Keim [74] is well reflected in existing systems for interactive data analysis. A survey conducted in 2016 reveals that most visual analytics pipelines follow this principal scheme and specialize certain aspects [147]. Here, an alteration of the pipeline is proposed, merging the knowledge and data models and thereby augmenting the data being analyzed by the insights found during analysis.

There is a variety of tools offering to combine Data Mining and Visualization. Some of them also include Machine Learning algorithms. KNIME and Orange are only two of the more well-known examples [15, 41]. These tools typically offer a graphical interface for the specification of data processing pipelines and visualization to study the results of pipeline executions. However, they do not feature the direct reintegration of obtained insights into the data that is proposed here. A recent survey reviewed 19 open source tools for data mining with respect to their quality and their features [3]. While most of the tools provide a visualization of the resulting model, less than half of them offer to visualize the data. Only about half of the tools (10/19) allow saving and reloading the results and only five can export the obtained results to common exchange formats like XML. Since the workflows in these tools are typically implemented as unidirectional linear or tree-like structures, saving and reusing obtained models is a necessity to implement an iterative approach like the one proposed in this chapter. Most of the reviewed tools are focused on the construction of data processing pipelines. This work instead focuses on the data itself, especially on the metainformation obtained by the user who interprets the visualization. Putting the focus on reintegrating obtained insights into the data increases the resource requirements. For this kind of scaling problems, Starič et al. recommend to work with light-weight visualizations supporting the parallel and asynchronous execution of algorithms [133].

The most relevant related work to the proposed workflow is the human-centered Machine Learning framework proposed by Sacha et al. [119]. Similarly to the approach proposed here, an iterative workflow based on Keim's Visual Analytics model is discussed, where the analyst applies domain knowledge to steer machine learning algorithms to support the analysis process. The paper also provides a good overview over existing approaches implementing parts of such a pipeline along with an in-depth discussion of tasks and analysis steps to be performed in such a setup. In their discussion, Sacha et al. focus on interaction for model building and parameter refinement to improve the performance of machine learning algorithms by leveraging the user's domain knowledge. In contrast, the approach proposed here is focused on restructuring and augmenting the data. Altering the analysis pipeline as illustrated in Figure 3.1 directly integrates the analyst's mental model with the available data. Taking into account insights obtained from previous analysis steps effectively extends the capabilities of the framework proposed by Sacha et al.

Taking a closer look at the automatic generation of visualizations, of course Jock Mackinlay's seminal work on a presentation toolkit has to be mentioned

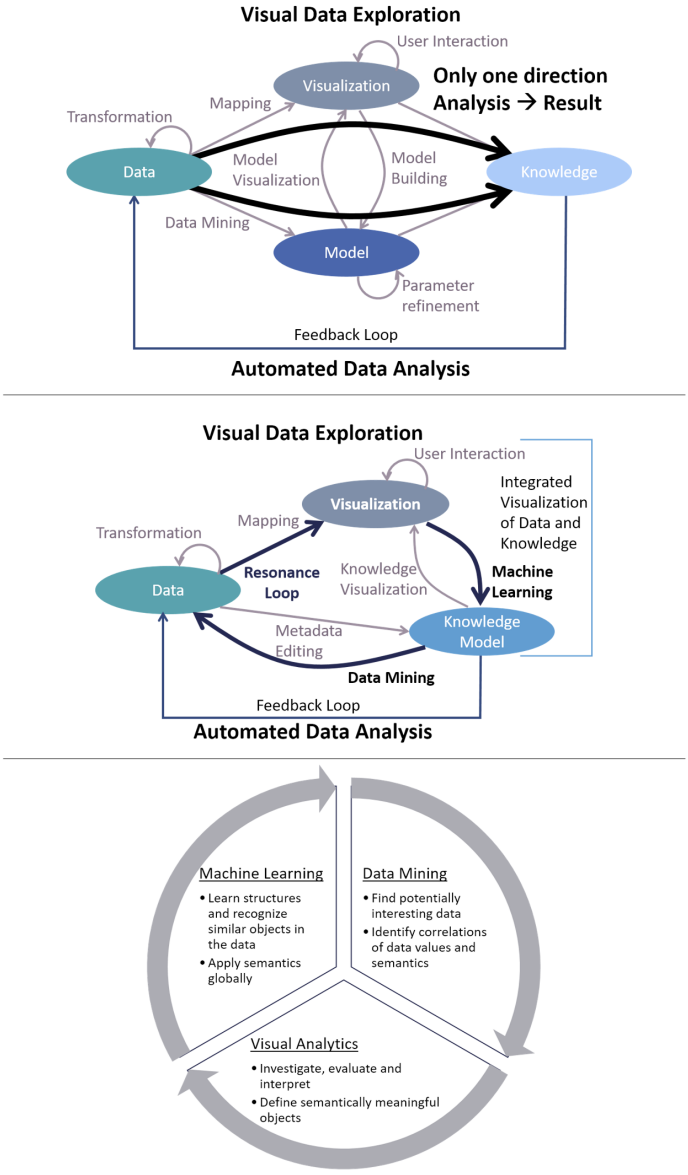


Figure 3.1: *Daniel Keim's model of visual analytics (top), the modification proposed here (center), and the proposed analysis workflow (bottom). Most existing visual analytics applications follow the classically assumed unidirectional flow of knowledge from the visualization and a data model to the viewer. Merging the knowledge and the data model implements an augmentation of the data being analyzed by the insights obtained from prior analysis steps. Leveraging this augmentation mechanism, an iterative workflow combining methods from data mining, visualization, and machine learning, implements a resonance loop fostering the obtainment of new insights from previously obtained results.*

[92]. Mackinlay was among the first researchers to interpret visualization systems as formal languages generated by grammars. Other authors followed this idea, although focusing on various different aspects, like covering different types of data [151], a strong focus on tasks [23, 126], a focus on the content of linked data [136], or perception-oriented embedding [40], to mention only a few examples. The aim of this discussion is not to reinvent the wheel. The focus is rather on how to combine models for automatic data analysis and visualization design to deliver optimal data representations to the viewer. All of the models listed here are of a constructive nature, describing the structure of visualization. In principle, all of them can be combined with the methods discussed in this chapter to cover different aspects of the properties the viewer needs the representation to feature. The necessary condition is that a set of visualization techniques is available that has been evaluated with respect to the qualitative aspects of their representations of data so the necessary predictions about the qualitative properties of the resulting visualization can be inferred.

Indeed such systems exist. VisIRR, for example, is a semiautomatic visualization toolkit for information retrieval [32]. Being heavily data-centric, it lets the user query for data and proposes visualizations hat are predicted to provide presentations of sufficient quality. A similar approach, although more on the construction side, is followed by the idea of a form-semantics-function coming from the field of visual data mining [130]. Here, the composition of visualization systems is derived from the semantics of the data to be visualized. This approach is quite appealing as it attempts to steer the visualizations shape by the information it is meant to convey. Again, the intention here is not to automatically define only the visualization but rather the whole data preparation and processing pipeline. However, the information model of the qualitative visual analysis cycle introduced in Chapter 2 is applied to steer this process.

From the information-oriented perspective, interesting work has been done composing visualizations by the semantics of the displayed artifacts and structures rather than the data to be visualized [93]. Very appealing is the idea to let the analyst define search queries based on examples. By the focus on semantics, the viewer can specify a query based on content based similarity. Thereby, an analyst can literally ask for "something like this" and point at some data. Perhaps the most popular technology in this domain is the semantic web [14] along with its data exchange format RDF [85], providing a simple, graph based representation of basic facts about the relations between data. An example for a more sophisticated formalisms for ontologies based on the semantic web is the web ontology language (OWL) [90]. From the perspective of this work, a better candidate to represent the information associated with data and algorithms would, however, be the concept graph in conjunction with the qualitative visual analysis cycle. After all, the focus here is not on querying for existing data and semantics but for information supported by data that can be algorithmically derived from the available data. In essence, the data supporting the answer to the information query is not directly available. What is available is a number of tools that probably allow transforming the available data into a form supporting the information sought. The intention of the discussion of the possibility of automatic visual analytics pipeline generation hence is to assess whether the algorithmic construction of such a chain of transformations is possible and computationally feasible. Because this ultimately asks for semantic information

rather than only data transformation, inference tools from the semantic web or other domains are not directly applicable to this task.

Being interested in the design of visual analysis pipelines, it is necessary to investigate their shape. A well-written survey on the development of visual analytics pipelines found that up to specializations, Daniel Keim's original model of the visual analytics pipeline is well established and adaptations only deviate slightly from it [147]. The idea is thus adapted here for the principal construction, assuming that data is visualized either directly or after being transformed by data mining or machine learning procedures. Note that this essentially determines the encoding-relation in the qualitative visual analysis cycle and hence is compatible with the model. Machine learning methods are deliberately allowed to be incorporated due to their great potential and increasing influence in the field which is well described in a recent survey [48]. Concerning the definition of data transformation pipelines to be executed prior to analysis, graphical programming languages for the arrangement of transformations like the one offered by KNIME have been developed to control the process [15]. Such languages can be applied by the user to steer or correct the proposed automatic approach if necessary. An example workflow using some of the tools mentioned here could be to define the transformation paths and load the result into AutoVis [151]. Once set up, the pipeline is executed automatically. Yet, there still is some space left for further automatization. An algorithm for automatic pipeline generation would attempt to derive a complete cover of the information inquired by the user as some formal representation of the analysis goals.

3.2 Towards Tight Integration of Quantitative and Qualitative Visual Analysis

Driven by advances in data mining, machine learning, and visualization, decentralized data collection, aggregation, integration, and analysis became almost ubiquitous. The advantage of making the results of automated data analysis accessible for human interpretation is well documented by the success of visual analytics. Yet, the obtained insights commonly remain entirely with the human analyst. Therefore, the mental model of the visualization necessary to apply the qualitative visual analysis cycle is typically not directly available. To exploit the implicit knowledge that is with the analysts, it needs to be made explicit. One possible approch to obtain an unkown mental model is to extract it from the documentation of the analysis process in the form of insight provenance information. Properly encoded, this enables the reintegration of obtained insight into the further analysis process. The mental model defined this way differs from the concept graph and from the one introduced for the qualitative visual analysis cycle in Chapter 2 in that it is based on individual objects and instances rather than on abstract entities and observations. Yet, this is necessary as rather than defining an abstract model top-down, provenance necessarily constructs a mental model bottom-up starting with the initial graphical data representation. To achieve this, an improved visual analytics pipeline is developed.

The focus of this chapter is an interative workflow merging the data model

and a model of the insights about the visualization and about the data aggregated during analysis. This allows performing analysis directly in this model rather than only on the data's graphical representation. Insight can thus not only be obtained from analyzing the data directly but also from the interpretation and inferred information obtained from previous analysis steps since by the unification of the data and the knowledge model insights found in previous steps become available for automatic analysis. Machine learning is applied to make the analysis feasible for large data sets by quickly rolling out insights found locally for some structure to similar structures in the data.

3.2.1 Searching for Insight and Panning for Gold

If data is too large to be processed at once or analysis is to be performed in an online setup constantly generating new data, the analysis is often limited to comparably small subsets of data being streamed through the system. Having to decide on which subset of the data is to be evaluated, one could claim that insight is only worth as much as the added value it generates. For the search of valuable information in streaming data, the metaphor of searching for a needle in a haystack often applied in big data contexts in some sense translates to panning for gold in a river.

Data Mining, Visualization, and Machine Learning each have their own approaches to finding information. Using explorative visualization, the user would attempt to find a spot along the river where the yield of panning for gold is maximized. This could very well require to explore the whole river. Data mining would try to analyze and cluster particle patterns in the stream. The interpretation where to find the gold and how to extract it from the stream is left to the user. Sophisticated machine learning algorithms would find an efficient strategy to extract large amounts of gold – if they were trained properly. If the training data is not of sufficient quality, the algorithm might as well just extract tons of sand.

A combination of the three approaches could for example proceed as follows: The data mining's clustering is interpreted by the user by means of visual analytics. The most promising streams are bundled by canals and led into a cycle to increase the potential yield even further. The gold to be extracted has a specific shape and floating behavior. Panning for some gold and labeling particles accordingly yields training data for machine learning. In some sense, learning to keep the gold and let the other particles pass in an optimal manner can be thought of as optimizing the pan. The result is an optimized gold extraction procedure to be applied to the water stream. Feeding the gold obtained from panning back into the system thus results in an accumulation of more gold and better yield. A larger gold yield means an improved return on invest. For data analysis, this means an improved efficiency of insight obtainment.

3.2.2 A Resonance Loop Amplifying Insight

In the classical model of visual analytics proposed by Daniel Keim (cf. Figure 3.1), the user applies interactive visualization and data mining to build, verify and refine a data model. The additional information offered by the model generates an added value for the interpretation of the visualization providing insight into possibly hidden relationships and dependencies in the data. Like in the thought experiment outlined above, this process can be seen as a flow of data (particles) along a stream where different means of analysis (the pans) are applied to extract valuable insight (gold). Although Keim's model includes the notion of a feedback loop from knowledge to data, this loop is only of conceptual nature and indicates the idea that users may choose to concentrate on different data based on the knowledge obtained from previous analysis [75]. The obtained insight remains out of system, rendering the extraction of knowledge essentially unidirectional.

It is not uncommon that the information cannot be read off directly but has to be inferred by reasoning about multiple data elements. Being aware of this problem, visual analytics applies data mining to obtain data models in which the information can be found more easily than from studying only the raw data. Finding the information might induce a new analysis question. An unsuccessful viewer instead could apply interaction to edit parameters steering the preprocessing, or decide to investigate different portions of the data or an entirely different data set. This is the feedback loop in Keim's model. Patterns, redundancies, or other interesting observations might not only be hidden in the data's values but also in the interpretation. Sometimes, the information to be found within the data but is hard to detect. This is the case, for instance, if the information is to be derived from transformed data, for example from the derivatives of a scalar field rather than from the field itself. If the derivatives are not part of the data, the feedback loop proposed here instead allows to evaluate the derivatives in local neighborhoods and to label the resulting new data accordingly. Investigating the derivatives and identifying the interesting information, the viewer can now select and label the respective derivative data. Machine learning can be applied to roll out these findings to the rest of the derivative data and the parameters steering this process can be optimized by a workflow similar to the one proposed by Sacha et al. [119]. Each derivative value can be mapped back to the original data points which can now be evaluated with respect to the insights found in the derivatives. Rather than only considering different data, the analyst thus concentrates on different qualitative information associated with the data which is made possible by including the knowledge obtained about the derivatives into the data and aligning it with the original data.

The visual analytics pipeline is thereby transformed into a resonance loop, amplifying the generation of new insight. An illustration of this model compared to Keim's model is shown in Figure 3.1. In the analogy of panning for gold, the feedback loop maps to the application of data mining to finding promising data sources (rivers with high yield) and integrating them properly in a preprocessing step (channeling the flow). The analyst infers insights from interpreting a visualization (the gold obtained from manual panning) and feeds

the results back into the system as metadata. Machine learning is applied to iteratively refine the data and knowledge model (optimizing the pans). With the assistance of automated procedures to mine and analyze previously obtained knowledge and apply it to the data, new insight can be derived from previous results (amplifying insight).

Optimal results require a tight integration between the three domains in a workflow leveraging each field's specific strengths and alleviate each other's weaknesses. The purpose of data mining is the detection of previously unknown patterns in the data. Their interpretation is left to a human analyst. Visualization is an interface for humans to make sense of data. Yet, finding and interpreting structure requires a skilled user and often also considerable amount of time. In direct comparison to data mining, the focus of machine learning is more on the identification of patterns already known. Its results, however, rely heavily on the proper choice of training data. Assigning roles to the three domains according to these strengths and weaknesses implements the workflow illustrated at the bottom of Figure 3.1. While on the global scale the proposed workflow implements a loop of applications of data mining, visualization, and machine learning, each executed procedure is based on an individual linear transformation pipeline. There is a variety of open source tools available for the creation of such pipelines [3].

The metadata to be edited can take multiple forms. Perhaps the simplest method to map analysis results back to the data is the assignment of labels. There are no strict restrictions to the data's shape other than that it needs to be compatible with the applied data mining and machine learning algorithms. Since the metadata is meant to formalize insights found during analysis and these insights will typically be of a qualitative, descriptive nature, it makes sense to apply a data structure explicitly mapping sets of data items to semantic information. The transformations applied during data processing and user interaction organize this structure in a graph allowing the navigation of analysis results obtained thus far. If in such a setup the applicable data mining and machine learning procedures offered by the system are known for every set of data items, pipelines processing the data to serve complex information queries can be generated automatically. This is discussed in further detail below. In its most simple form, the metainformation is simply a set of labels applied to the respective set of data items. However, more complex structures like a semantic web or other kind of ontology defined on top of the data are feasible and allow more sophisticated analysis and inference structures operating directly on the knowledge model. From such a model, the reading language can be constructed as described in Chapter 2. Once the reading language is known, a concept graph generating this reading language for the given visualization can be constructed.

To make the assignment of labels or other metainformation feasible for large data sets, a semiautomatic distribution of metainformation can be achieved by searching for data patterns to be labelled rather than for individual data items. Depending on the shape of the data patterns, suitable machine learning algorithms can be trained using the labeled data to roll out the labels to corresponding structures in the remaining data. As an example, consider a point cloud obtained from scanning, for example, an asteroid's surface. Due to measurement errors, there is some noise in the data and the surface is not smooth.

While the analyst would be interested in studying craters, the measurement errors induce false local critical values. Simply smoothing or averaging the surface could, however, result in the loss of important detail. Data mining can be applied to categorize local neighborhoods of points with respect to the points' position relative to an averaging surface. The clusters will reveal bumps, dents, ridges, and other structures. For the analysis of craters, too small neighborhoods result in a large number of erroneously found crater-structures whereas the cluster criterion does not yield reliable results for too large neighborhoods. A simple application of machine learning would be to find craters by searching for the largest structures whose similarity to a local bump or dent does not fall below a certain threshold.

3.2.3 Example Use Case

Irregular influences on air-traffic patterns like thunderstorms do not follow spatial patterns. Their influence on air-traffic routes can thus not be accurately predicted based on historical data. Nevertheless, historical data can be considered to identify possible evasion routes. The following discussion shows how the proposed workflow could be applied to solve this problem by mapping each analysis steps to the domains of data mining (DM), visualization and visual analytics (VA), and machine learning (ML).

If a storm warning is announced, historical data is mined for past storms in the same region (pattern recognition, DM, channeling streams). The analyst assigns grades to the trajectories of representative planes evading the storm in order to assign them to equivalence classes reflecting their quality (find and evaluate structures, VA, manual panning for gold). Those grades are now rolled out to the other evasion routes by a classification algorithm (classification, ML, optimizing pans). Quality measures determine how well each path fits into its class (cluster quality assessment, DM, determine yield quality). Where necessary, the identified classes are subdivided into two or more subclasses by assigning proper labels (evaluate quality and detect subclasses, VA, increase the gold yield). These adjustments to the classifier's definitions improve the results during reclassification (reinforcement learning, ML, optimizing the pans). The controller identifies the best-graded routes for every relevant direction and reevaluates their embedding into the actual surveillance data (VA, panning for gold). The planes can then be assigned to the evasion routes according to the classifier trained before (ML, increase yield). Storing the routes for future reference, candidates can be obtained directly from the collection rather than having to be extracted them from historical data (amplify insight).

3.2.4 Avoiding Credibility and Reliability Issues

Feeding back the results of visual analytics to into the data to make it accessible for machine learning and data mining enables the derivation of new insight from previously obtained results. With the benefits, there also come pitfalls and risks. In visual analytics workflows, uncertainty usually only propagates between the data and the obtained model from the data and the model to the visualization

[20]. Feeding back analysis results into the data and the model introduces two additional types uncertainty: a quantitative uncertainty in the classification obtained from machine learning and a qualitative uncertainty regarding the credibility and reliability of the results obtained from human data analysis.

Other than the human analyst, the computer does not reflect on the data it receives as input. Thus, errors in the analysis will not be detected by the computer and propagate through further computation. When attempting to roll out analysis results to the whole data set, misclassification errors can be corrected by refining the classification schemes. Still, there is a risk of an "analyst-induced oscillation" where continued optimization attempts eventually result in an overfitting detrimental to the classifier's performance.

To assess the credibility of metadata defined in previous analysis, provenance information must be stored along with the metadata. Without such information, errors made in previous steps or assumptions inapplicable to the current investigation might yield false analysis results. Note that, being part of the metainformation added to the original data, the provenance information can be accessed and processed like any other data.

To test the model's reliability, it can be tested against the addition of new (artificial) data and against assertions. The metadata and definitions together define a model for the observation. If the model is accurate, it should predict the metadata of newly added data points correctly by applying the definitions obtained from previous analysis. Assertion checks can be performed by specifying a condition that has to hold under the model. This assertion is then evaluated on each relevant data item generating a label with the evaluation's result. The labels can then be used for further analysis to check whether the assertion holds on the correct data elements.

3.3 Motivating Automatic Visual Analysis Pipeline Generation

Combining automatic data analysis with human reasoning based on visualization, visual analytics has become an integral component of modern data analysis applications. While individual advances in visualization, data mining, and machine learning contribute to this success, the key element of visual analytics is the efficient combination of the different techniques to obtain solutions fostering the derivation of new insight. The workflow introduced above supports this kind of analysis by structuring the process and explicitly leveraging the ability to reintegrate insights obtained back into the data. Yet, this integration can still be quite challenging. While tools have been developed to support the efficient generation of data preparation and processing pipelines, finding a combination of algorithms that reveals the insight an analyst is aiming to obtain still depends on the analyst's understanding of the algorithms' effects on the data and experience in their application.

Towards a more efficient process of generating data preparation and processing pipelines for visual analytics, propose a partial automatization of the process

is proposed in the following. The key idea is to let the computer reproduce information inquired by the analyst. To achieve this aim, the computer needs to find a sequence of transformations that derive data supporting the inquired information from the available raw data. Yet, there is a twist. The algorithm would necessarily decide whether it is actually possible to derive this information as the interpretation of the result of a sequence of transformations applied to the raw data. This is an instance of the halting problem – and therefore impossible to solve. Fortunately, there are special cases, in which a restricted version of the problem is decidable. A second problem is that information refers not to the data itself but rather to its interpretation with respect to the analysis question. An example would be treating distance as an indicator for neighborhood. Therefore, an analyst can only query for information that is already part of the mental model – and thus already known. As it turns out, the trick is to specify the characteristics of the results of data analysis and visualization algorithms. The analyst then asks for a view on the data that has specific properties and for interpretations that are explicitly associated with the results of data transformations as properties of the transformation's algorithm. Ideally, this view allows an efficient evaluation of conjectures and hypotheses against data but can still be explored easily.

The following discussion introduces an approach to the automatic design of visual analytics pipelines driven by the information being part of the interpretation of data after a number of transformations. To this end, it is investigated under which conditions it is decidable whether some information can be derived from the raw data by sequences of data transformations. The actual algorithm treats the transformation procedures as building blocks in a directed graph of possible transformation sequences. It is shown how this graph can be generated from a schematic description of the data transformation and how the resulting change of information associated with the data taking place in each transformation procedure can be obtained. The discussion proceeds as follows:

1. It is proven that the problem whether it is possible to derive information from raw data is in general undecidable but can be decided for special restrictions which are typically the case for real world applications.
2. The discussed model is extended to visualization and the conditions for the ability of a visualization system to present inquired information are derived.
3. An algorithm scheme for the automatic generation of visual analytics pipelines covering the whole span from raw data to visualization is outlined.

3.3.1 Information and Data Transformations

This chapter is concerned with details of the encodes-relation $enc : \mathbb{D} \rightarrow V$ discussed as the transformation of data $\mathbb{D}$ sampling the domain information $\mathbb{I}$ into a graphical representation V. Therefore, a more fine-grained notation than in Chapter 2 is required for the discussion. Like before, data refers to to qualitative or quantitative variables. Now, there is a distinction between

raw and refined data, the former being the data provided "as is", the latter the result of data transformations. For information, the definitions apply as before. Yet, an additional demand is that facts are comparable within the same set of information, meaning that it is decidable whether two entities or observations are equal. The representation of information is often based on logics, predicate logic to be precise. However, the details are domain- and application specific. For example, modal logic can be applied to model systems with several alternatives and (linear) temporal logic allows to model processes. Another way to represent information is the semantic web [14], where data semantics are modeled in terms of a graph given as a set of triples encoding edges between graph nodes. The relationship between data and information is still defined by the partial functions $\rho : \mathbb{I} \to \mathbb{D}$ and $\sigma : \mathbb{D} \to \mathbb{I}$. For the remainder of this chapter, $\mathbb{D}$ and $\mathbb{I}$ are considered global objects demanding that they contain all data and all information that is either directly available or can be computed or otherwise obtained from other data and information. In contrast, $D \subseteq \mathbb{D}$ and $I \subseteq \mathbb{I}$ refer to sets of known data and information. Note the slight difference that D and I are now subsets rather than elements.

The question of containment is a natural consequence of organizing information in sets. Recall that in Chapter 2, the domain information has been introduced as predicates and functions bound to higher-order predicates modeling entities and observations. Similarly, the objects and situations observed in actual data consist of valuated predicates and functions. For finite information sets $I \subseteq \mathbb{I}$, decidability of set containment thus follows trivially from the comparison of the contained predicates and functions based on the valuation of the contained variables. Concerning the possibility to derive certain information from the raw data, the situation is, however, quite a bit more complex: A map between sets of information has not been introduced thus far and for the sake of generality it remains so. Instead, an indirect approach is applied. Let $\tau : d_0 \to d_1$ be a transformation transforming some data $d_0 \in D$ into some data $d_1 \in D$. The change of data resulting from the application of τ can result in a change of the associated information – although this is not necessarily the case. An indirect mechanism for information transformation is thus given by following the data: Let i_0 and i_1 be the information associated with d_0 and d_1 respectively. Since in this case the relation between data and information is explicit, it is evident that the representation and sampling operations are defined in both directions and cover the respective sets completely as inverses. Therefore, the transformation $\theta : i_0 \to i_1$ of information $i_0 \in I$ into $i_1 \in I$ can be expressed indirectly by the transformation of data as $\theta(i_0) = \sigma(\tau(\rho(i_0)))$. Because data transformations (if applicable to the data) can be chained, the derivability of information can now be defined as the existence of a sequence of data transformations from the source data to some data whose interpretation covers the target information. That is, information $i \in I \subseteq \mathbb{I}$ is derivable from some data $d \in D \subseteq \mathbb{D}$ if and only if there is some transformation $\theta^* := i_0 \to^* i$, such that there is a sequence of data transformations $\tau^* := \rho(i_0) \to^* \rho(i)$. In particular, this means $\exists \tau^* . i = \sigma(\tau^*(\rho(i_0)))$. Extending this notion to the derivability of information I from some data D, yields that I is derivable from D if and only if for the elements $d_k \in D$, one obtains

$$\exists \theta^* . \big(\forall d_k \in D . \big(\exists \theta_k^* . \big(\theta_k^*(i_0) = \sigma(d_k) \big) \big) \big)$$

For single elements, data transformations and θ^* are transitive, the latter being a direct consequence of the former. As before, the upward closure of contained information $I^\uparrow$ is thus all information that is directly assigned with or can be derived from some data D. In Chapter 2 it has indirectly been mentioned that for some information $I = \sigma(D)$, $I^\uparrow$ is precisely the transitive hull of the information associated with the data along every transformation path starting in $D = \rho(I)$. The difference here is that in Chapter 2 this has been considered for the interpretation in the mental model whereas here the reflection-function $ref : M \to \mathbb{I}$ mapping the mental model back to the domain information is assumed to be applied implicitly. By this construction, some information i is derivable from I if and only if $i \in I^\uparrow$. Of course, the question remains whether containment in $I^\uparrow$ is actually decidable. The answer to this question determines under which conditions one can prove information to be derivable from raw data and thus available for visualization.

3.4 Information Derivability

The idea of a visualization system in which the containment of information can be proven is quite appealing for multiple reasons. Not only does it allow inferring the completeness and correctness of depicted information for solving a given task but it also can enable the user to specify information patterns that would be derived automatically by the system. Unfortunately, it turns out that information derivability is, in general, undecidable. However, there are certain special cases where it is indeed possible to prove information derivability – and even to do so automatically. In the following, the details of those observations are discussed.

3.4.1 Foundational Considerations on Decidability

In principle, the proof that information can be derived from some data is simple. One may just stop searching if a solution is found. Unfortunately, it is in general not possible to prove the opposite, namely that some information cannot be derived from a data set. Intuitively speaking, if the inquired information cannot be derived from the data, a data analyst blessed with infinite creativity could literally spend eternity trying to figure out a solution with no way to prove that the effort is ultimately futile.

Theorem: Undecidability of Information Derivability

The information derivability problem is in general undecidable.

Information I is derivable from some data D if there is a continuous path of transformations linking the data of the source with some data whose interpretation function supports I. If the intention is to check algorithmically whether this is the case, allowing arbitrary transformations to be applied would allow to apply a potentially infinite set of operations to assess whether I can

be derived directly from the raw data. Hence, the general problem of deciding whether information is derivable is an instance of the halting problem and thus undecidable [142]. □

Still, at least partial success can be achieved. If an algorithm finds a transformation path generating data that supports the inquired information, it terminates and returns a correct result.

Lemma: Positive Semi-Decidability of Derivability

The information derivability problem is positively semi-decidable.

For the proof, consider an algorithm that tries every possible transformation and checks whether the result supports the inquired information, for example by evaluating semantic aggregation in the mental model given in terms of the concept graph like it is discussed in Chapter 2. Because information containment is decidable, a positive result will be identified correctly once it is found. If the information is derivable, such a result exists and the algorithm will eventually find and correctly return it. □

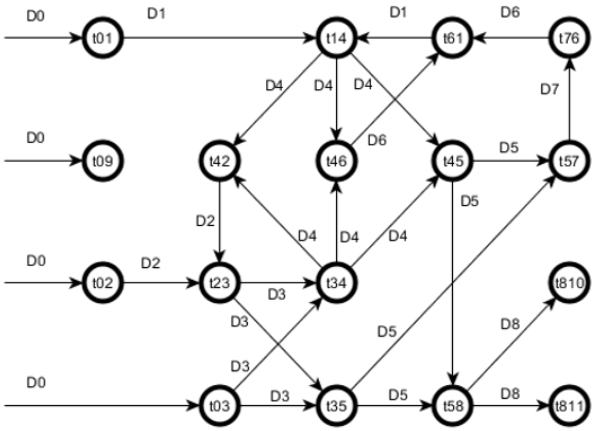


Figure 3.2: *Automaton of transformation sequences. This finite state automaton links the applicable transformations t by the data formats they exchange when being executed sequentially. Every state is an accepting state. Thus, given any sequence of data formats recognized by the automaton, the sequence of transformations generating the last data entry can be read from the nodes along the path.*

Restricting the problem to a limited set of finitely many transformations, things become a lot easier. With only finitely many decisions which transformation apply to each set of data, the tree of all transformation sequences can only branch finitely in any node. By König's Lemma [79], this reveals that if the tree should be infinite, there must be a branch of infinite depth. However, along such an infinite path, at least one transformation has to occur twice because the number of available transformations is less than the path length. In fact, by the same argument, at least one transformation has to occur infinitely often.

The execution chains of transformations define a regular language. Assuming the possible combinations are known, an automaton linking each transformation to its possible successors can easily be constructed. Every state in this automaton is accepting. An example of such an automaton is shown in Figure 3.2. In automata theory, there is a construction extending regular languages to words of infinite length called a Büchi Automaton [22]. It recognizes words that pass a certain state in the automaton infinitely often. For the transformation sequences, this means that some transformation occurs infinitely often which is the case if and only if an infinite transformation chain is found. Hence, Büchi automata decide whether infinite paths occur. From here, one obtains:

Lemma: Properties of the Restricted Derivability Problem

The following properties hold for the restricted problem of information derivability with only a limited number of applicable transformations:

1. It is decidable whether transformation chains of infinite length can occur.
2. If no transformation chains of infinite length are possible, information derivability is decidable.
3. If such chains can occur, the problem is still semi-decidable.

The third proposition is probably the easiest to prove since it follows trivially from the semi-decidability of the general problem. Where infinite sequences are concerned, the possibility of such a chain does not even require the full power of the Büchi Automaton. Because the interest is only in the existence, it suffices to check the finite state automaton of transformation sequences for the presence of loops. If no loops occur, the automaton is a directed acyclic graph which can be turned into a tree by separating joins of paths into different branches. Since there are only chains of finite length and only a finite number of transformations to branch in each node, by König's Lemma, the tree must be finite. If so, an algorithm simply needs to follow all transition chains and check whether the data corresponding to the respective nodes supports the inquired information which is decidable by definition. □

As it turns out, not allowing any cycles at all is restricting the problem a little too far. Recall that in the definition transformations are bound directly to the data they are applicable to. Let this restriction be integrated into the language of possible transformation chains. The resulting automaton is sketched in Figure 3.3. Loops can now only occur if the data admits it. At a first glance, this renders the problem harder since some of the infinite transformation chains that before necessarily were loops are now open paths. However, these open paths can still be detected in the non-constrained automaton. Considering that the description of a loop is a finite sequence of transformations, it is even possible to distinguish them from the ones in the constrained construction by comparing the transformation sequences. From here, it is only needed to require that cycles must be compatible with the data.

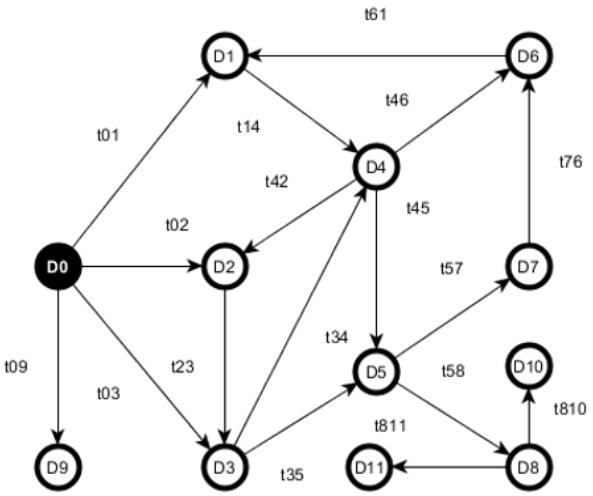


Figure 3.3: *The transformation graph. Black nodes indicate starting states. Every state is accepting. This automaton provides the data and therefore the information passed when executing a sequence of transformations. The data transformation or information derivation graph is the dual of the automaton of transformation sequences with the additional condition the state's data format must now be compatible with the transformation's for the edge to be allowed in the graph. Data D and, by the interpretation function, information $\Delta(D)$ are derivable from the raw data $D0$ if and only if a path in this graph connects the raw data node $D0$ with the node for D.*

Theorem: Decidability of the Restricted Problem

If, in a graph of transformation sequences, the applicability of a transformation is determined by compatibility with the data in the transformation's source and target states, infinite sequences are either open paths or closed loops. For graphs with no infinite open paths, the restricted information derivability problem is decidable.

For the proof, it suffices to show that cycles can be contracted into a single state merging all the cycle's states into a single one that is also the combined source for all transformations leaving the original cycle and the target for all transformations reaching any node in the original cycle. To see this, consider an arbitrary node in the cycle, let D be its data and I its associated information. Whether some Information J is derivable from D can be assessed by attempting to find J in $I^{\uparrow}$, the upward closure of I, containing all information derivable from D. Obviously, every node in the cycle is reachable via transformations starting at D. Therefore, $I^{\uparrow}$ contains the information of every data node in the cycle. Indeed, it even contains every upward closure of these information sets. Since this applies to every node in the cycle, $I^{\uparrow}$ is identical for all of them. Therefore, the cycle can be contracted into a single node that serves as a unified source and target for all transformations entering or leaving the original cycle and represents the contained information by either computing the upward

closure of the information contained in any of the cycle nodes or by computing the union $I^C \subseteq I^{\uparrow}$ of the information sets of the nodes along the original cycle. From the latter, the complete upward closure can be computed by following the original cycle's outgoing edges. □

The above derivability theorem is indeed quite remarkable because these cases are still covered if the length of transformation sequences after contraction of every cycle in the graph is limited. The idea behind contracting cycles is illustrated in Figure 3.4. Within an upper bound to the length of transformation sequences, the derivability problem is decidable with respect to the bound. Thereby, the further consideration can be restricted accordingly. Next, an automatic solution to assess derivability of information and return proper transformation chains to compute the data supporting this information is developed.

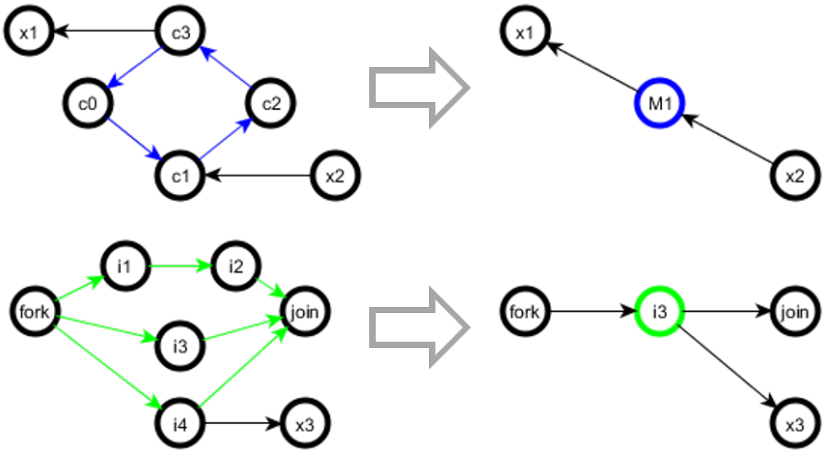


Figure 3.4: *Contraction of cycles (upper) and fork-chain-structures (lower). Cycles are contracted into a single node holding their internal structure and their individual knowledge and access to transformations internally. The same applies to fork-join-structures with the essential difference that the actual fork and join nodes remain unchanged. Figure 3.5 shows a complete run of the contraction procedure on an artificial example.*

3.4.2 Automatic Extraction of Information

The aim here is to find an algorithm enabling the automatic extraction of information from available raw data. To this end, it has to be assessed whether the inquired information can be derived from the data and whether the data can be transformed into a form supporting the information. It has been observed that this is decidable if a maximum number of consecutive transformations is given and if there is only a finite number of transformations available for application. For real-world applications, both requirements can safely be assumed to be met. Applying too many transformations to the data is infeasible, especially for large data sets and the number of applicable transformations is limited by the functionality offered by the analysis software. Even if additional custom algorithms

can be implemented on demand, the number of applicable transformations is still finite for any given point in time. However, it is actually intended to include iterative procedures converging towards asymptotic results, for example as a solution towards optimization problems. The procedures executed along an iteration define a loop. If the information to be represented by the data is chosen properly, the change of data along the iteration does not affect the information. In this case, the upward closure $I^{\uparrow}$ of the information associated with any state of the data along such an iteration cycle is finite. Recall that I denotes a subset of known domain information which in real-world applications would have to be reflected by some sort of mental model and typically have to be mapped to the data a priori.

In the following, it is assumed that the partial map $\sigma : \mathbb{D} \to \mathbb{I}$ describing the information sampled by any data found during the process to some subset in the possibly infinite set $\mathbb{I}$ of information hypothetically derivable from $\mathbb{D}$, is already defined for the known data states. It is further assumed that a set of transformations $T \subseteq \{\tau | \tau := \mathbb{D} \to \mathbb{D}\}$ linking differently formatted sets of data is known beforehand. T is a finite subset of the set of all hypothetically possible data transformations over $\mathbb{D}$. T induces a graph $G = (D^{\uparrow} \subseteq \mathbb{D}, T)$, where $D^{\uparrow}$ is the set of all results of transformation sequences $\tau^* \subseteq T$ applied to a set D_0 of raw data that is to be analyzed. If the length of the sequences τ^* is restricted to some $k \in \mathbb{N}$ and the cycles in G are contracted as described in the proof of the decidability of the restricted derivability problem, it is decidable whether some information can be derived from the raw data within an upper bound of k steps. In the following, a simplification procedure is introduced that enables efficient algorithms to assess derivability and infer proper transformation sequences.

3.4.2.1 Simplification by Contraction

In the proof for the decidability of the restricted derivability problem, a simplification procedure is found that contracts cycles in the graph into a single node representing all the contained states and associated information. Of course, when applying this type of contraction, it is necessary to keep track of the transformation paths within the cycle. However, if each transformation is labeled with a unique identifier, this is trivial to achieve. The procedure itself is illustrated in Figure 3.5. Once all cycles have been removed from the graph, there are two other useful contraction mechanisms. The second step is to contract forks and joins of paths into a single node given that all transformation paths branching from a given node (the fork) meet in the same target node (the join). Fork-join-structures can easily be assessed by applying cycle detection interpreting the graph's edges as undirected after all cycles in the directed graph have been contracted. Any cycle detected this way is a candidate for being part of a fork-join-structure. Of course, one still has to check whether all branches actually meet the join node. Therefore, it makes sense to start with contracting small fork-join-structures and to gradually increase the size during contraction. By reducing the branching within nested fork-join-structures, this also reduces the task's overall complexity. The result of applying these steps is a tree rooted in a single starting node representing the raw data. Note that a simple union of data structures into a single object can create this node in the case the raw

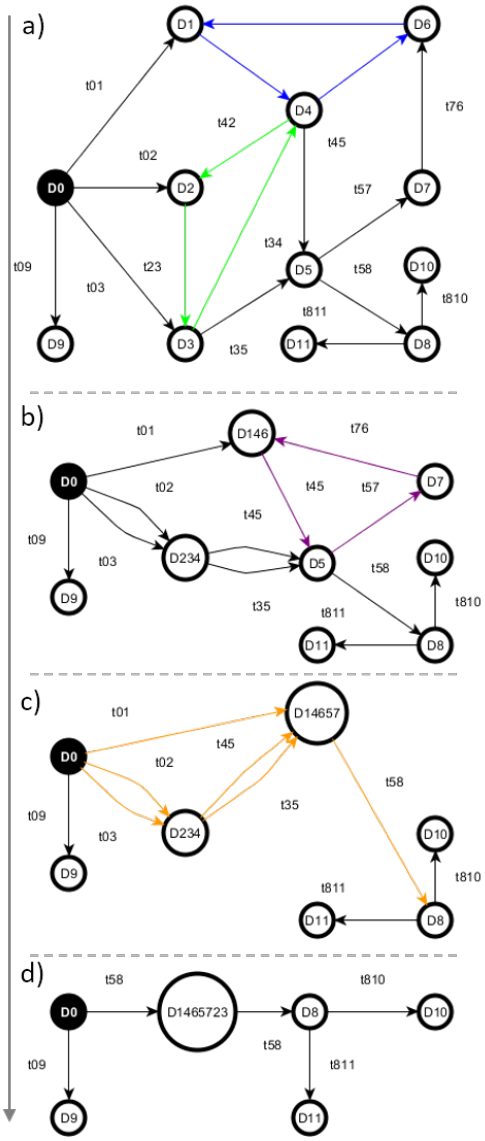


Figure 3.5: *Example of a contraction procedure applied to a transformation graph. Starting with the smaller cycles the algorithm first contracts two cycles in the initial graph (a), followed by a third cycle containing one of the previously contracted nodes (b). Now that no cycles are found anymore, a fork-join-structure is identified and merged (c). The result of the contraction procedure (d) is a directed acyclic graph of nodes that cannot be further contracted by either of the definitions provided in Figure 3.4.*

data stems from multiple sources. Therefore, it can be assumed that the raw data can always be represented by a single data node. The last step, although optional, is to contract the simple paths between branching nodes. Since the sampling map σ has been restricted to map the data $d_k \in D$ represented by any node to finite sets I_k of information, the unions $I_k^C \subseteq I_k^{\uparrow}$ of information

sets computed as part of the contraction procedures are also finite sets. Therefore, information containment is decidable and derivability can be assessed by traversing the resulting tree and checking containment for the information sets associated with each individual node. Performing the assessment for each element of a set of inquired information also directly provides the percentage of how much inquired information can indeed be derived from the data.

3.4.2.2 Assessing Derivability

In principle, information derivability can be directly assessed from the upwards closure $I_0^{\uparrow} = \bigcup_{D_k \in D_0^{\uparrow}} \sigma(D_k)$ of the information contained in the raw data. However, in order to obtain an efficient procedure to obtain transformation sequences to be applied to transform the data into a form supporting the inquired information, one should keep track of the applicable transformations for each node in the original graph $G = (D, T)$ of data and transformations. This is exactly what is done during contraction.

When contracting data and computing the union I^C of information assigned to the corresponding nodes in the original graph, each subset I that has been merged into I^C is associated with the transformation path needed to derive it from some starting point in the data. For cycles, this set is the same for every node and since for every node D_k in the cycle, all reachable nodes are in $D_k^{\uparrow}$, any node qualifies as the starting node. For fork-join-structures, the starting point is the fork, and for simple paths, the starting point is the path's first point. The contraction of simple paths yields a hierarchy of derived information. The edges are labelled with transformation sequences and the data is associated with tuples of information as well as the further transformations that need to be applied to obtain this information. Instead of collapsing this tree, it is directly applied as the data structure in which information containment is inferred. If the sets of information are implemented as lists, the lists of each remaining data node can be connected, resulting in a single large information set which is exactly $I_0^{\uparrow}$, the set of all information derivable (within k steps) from the raw data. Assessing the derivability of information from the raw data with respect to the available algorithms therefore reduces to checking whether the inquired information is contained in $I_0^{\uparrow}$. From the tree structure, the transformation paths can be inferred by following the tree's edges backwards until the root is reached once the information has been found. The actual transformation sequence is then a concatenation of the tree's edge labels. The remaining transformations to be applied to obtain the actual shape as it was prior to contraction are stored in each node along with the corresponding set of data.

The algorithm terminates and returns the correct information and transformation if a solution is found. If not, the algorithm also terminates since the length of transformation paths to be checked is restricted. However, the result obtained in general is only whether it was possible to derive the information from the raw data in up to k transformation steps. A negative answer is reliably correct if and only if the original structure does not permit infinite chains of consecutively applied transformations. Fortunately, this is decidable since as has been shown in the discussion of the lemma on the properties of the restricted

derivability problem, such infinite paths map to infinite words in a regular language and can thus be recognized by a Büchi Automaton. Therefore, if such a situation is encountered, the user has to be informed about the unreliability of a negative result.

3.4.2.3 Considerations on Runtime Complexity

Where runtime complexity is concerned, three categories of algorithms need to be distinguished. The first one is the computation of the actual transformations. This is highly specific to the data and the applied algorithm and thus not in the scope of this work. It should be noted, however, that the procedure can be rather time consuming, especially if translations between data structures have to be applied before the subsequent algorithm can be executed.

The second category is the assessment of derivability and the inference of the corresponding transformation path. Finding the correct path actually depends on the length of the longest branch in the tree obtained from the contraction procedures. Considering that before contraction the maximum path length was bounded either by a number $k \in \mathbb{N}$ or, in presence of infinite open paths, limited to k, this is an upper bound for the tree depth. Since the format of information and therefore the equals-relation depends on the application, no assumptions can be made on the runtime needed to compare two sets of information. Therefore, it is just assigned the function $\mathcal{R}$. In the worst case, every transformation is applicable to every data set and yields data with globally unique information. In this case, the derivation graph with respect to n applicable transformations becomes an n-ary tree of depth k. Observe that even for comparably small chains of applicable transformations the runtime complexity skyrockets if powerful toolsets are provided for the analysis. The worst case runtime complexity for assessing information derivability is $\mathcal{O}(k^n \cdot \mathcal{R})$. Even for only 10 algorithms and an upper bound of five consecutive transformations, even if only one information item is searched and the largest set contains only five items, this amounts to the comparison of 500,000 individual information elements. As it seems, the worst case upper bound quickly skyrockets to ridiculous amounts of runtime complexity even for comparably simple examples. However, this is unlikely to happen, especially in visualization. While there are algorithms like sorting and searching that may be almost universally applicable, the majority of the algorithms is not. For example, in a visualization system, a wide variety of visualization techniques may be offered. However, these options are applied only once, drastically reducing the complexity. Assume that the hypothetical example is a classification task offering two filters for outlier detection as preprocessing, two distance measures, two classification techniques, one color map showing the ground truth, and three different visualization techniques. To solve the classification task, data must be preprocessed, compared, classified, colored, and rendered to the screen – a total of five steps. Counting the possibilities, the five consecutive computations yield 24 different states for the worst case runtime. Note that there still is a maximum path length of five and there are still ten applicable transformations. If it is again assumed that the analyst compares one information item to sets of five items, only 100 comparisons need to be performed.

From the considerations on the hypothetical visualization task, it is conjectured that in real world applications, the derivation graph can safely be assumed to be rather sparse and the upper bound of runtime complexity is far from being met in actual applications. Note that this claim is more bold than it may appear at a first glance. While, for example, a human programmer will usually not invoke a sequence of several sorting algorithms immediately after each other, a computer only looking at the compatibility of data formats will definitely consider this a valid option. Therefore, the information about what kind of transformations should be combined with each other should be added to the system, probably as part of the information associated with the transformation itself. The hypothetical example reveals that grouping of the algorithms into steps and implementing rules for their proper combination can already reduce the theoretically possible complexity tremendously.

The last category are the algorithms for contraction. Contraction consists of three steps: contracting cycles in the directed graph, contracting fork-join-structures, and contracting the remaining simple paths. The runtime complexity of connecting the linked storing of the remaining information sets is negligible since connecting to linked lists can be performed in $\mathcal{O}(1)$ and the other steps' complexity is more than linear in the number of graph nodes. Towards the detection of cycles in the graph G, recall that the graph actually models $D_0^{\uparrow}$, i.e. all data sets D that can be computed from the raw data D_0 by transformations $\tau \in T$. Cycles can therefore be detected by computing the topological ordering of the nodes starting in the node representing the raw data D_0. If the derivation graph is not acyclic, the algorithm will eventually detect a cycle to be contracted. Since the topological order can be established using depth-first search (DFS) , it can be established in $\mathcal{O}(|D_0^{\uparrow}|+|T|)$. This means the computation is linear in the number of graph nodes and edges. Note that if the graph contains cycles, the number of edges can be significantly larger than for an acyclic graph. In general, $|D_0^{\uparrow}| + |T| \ll k^n$. Because the actual sparsity of the derivation graph depends on the domain and application, no restricting assumptions can be made without sacrificing the generality of the discussion. Since DFS logs visited nodes as part of its execution, the transformation paths describing cycles are obtained together with the cycle. Hence, contracting a cycle is linear in the number of its nodes since the union of their information sets can be computed by concatenating the linked lists storing the sets of information. If a cycle is found, it is immediately contracted and the DFS is continued from the new node, dropping the results of the former search. The procedure is continued until no cycles are left. In the worst case, the data is aligned along a long line where D_0 is a starting node and DFS identifies the first cycle as being the cycle containing the tree's single leaf and its immediate parents. If the new node is again part of a cycle of two nodes, this cycle is also merged. Continuing this procedure up to the root requires as many steps to go down until the leaf is reached as it requires to contract the cycles on its way back up. The worst case is therefore linear in the number of nodes and the upper bound obtained is $\mathcal{O}(|D_0^{\uparrow}| + |T|)$. For the detection of fork-join-structures, the procedure is similar but one needs to either apply breadth first search (BFS) or interpret G to be an undirected graph. While the latter method is again based on cycles and can apply the same algorithm as before, it requires an additional reconstruction step to obtain the actual paths contributing to the structure. BFS not only reveals those paths directly but

also covers complex branching situations. The runtime complexity is the same as for the contraction of cycles.

After contraction, a tree structure is reached again and the remaining upper runtime bound for assessing derivability in the graph is, again overestimated, $\mathcal{O}(k^n \cdot \mathcal{R})$. Contraction thus allows to asses derivability of information more efficiently, speeding up the process especially in applications like business intelligence or process control monitoring.

3.5 Automatic Pipelines for Visual Analytics

In the qualitative visual analysis cycle, visualizations are generated by the graphical language $\mathcal{L}_V$ which in turn is the language generated by applying an automaton implementing a corresponding formal grammar to the input data. The information to be found from reasoning about the visualization depends on the outside knowledge and can be inferred from evaluating a mental model that also determines a grammar generating the reading language $\mathcal{L}_R$, the viewer's counterpart to the graphical language. Specifying the automaton for the graphical language, a hierarchy can be defined, defining structures like the points in a scatterplot or the arrangement of scatterplots in a matrix. Connecting those structures to the corresponding elements in the reading language links them to the information they convey to the viewer.

The capability of a visualization to depict certain information depends on whether this information is derivable from the raw data and that it is contained in the upward closure of the information directly associated with at least one visualization element. Since only visualization systems offering finitely many options to represent data are considered and number of consecutive transformations is limited, this setup satisfies the preconditions for decidability of the restricted derivability problem. A direct consequence of this is:

Lemma: Decidability of Information Conveyance

Let V be a visualization system with k algorithms for data representation or transformation. The ability of V to convey certain information after a sequence of not more than n consecutive data transformations is decidable.

Derivability is a necessary condition for the ability of a visualization system to convey inquired information. However, it is not sufficient since not all derivable information is necessarily encoded by some visualization element that is actually rendered to the screen. There are two cases of such non-explicit information derivability in visualization systems which here is referred to as **intrinsic** and **extrinsic inferability**. Intrinsic inferability does not expose the information to the user. Typically the viewer is shown a consequence of the actual information. A trivial example for this case is a progress bar. Although the system does no expose to the viewer what action it is currently performing, the consequence, namely the progress made so far, is visualized and provides

the user with useful information. Likewise, extrinsic inferability involves the viewer into the process. This information is indeed encoded by visualization elements but not by the basic ones. Hence, it is not presented explicitly and the viewer has to infer it from the data context. This is typically the case when the information is encoded by a history of states or by relations between explicitly depicted items. An example is the trajectory of a driving car or tasks where the viewer has to infer abstract properties from Gestalt principles.

3.5.1 Interchangeable Pipeline Blocks

Now that it is known that the one can seamlessly integrate visualization into the theory developed thus far in this section, building blocks of dynamically adjustable visual analytics pipelines can be defined. Defining transformation graphs for each technique to be applied during data preparation and for data visualization, one can construct pipelines by highlighting not only the raw data nodes but also distinguished output nodes marking data that is exposed to the outside. Connecting the output of one graph to the raw data input of another, this effectively concatenates the applicable transformation paths and thus merges the information stored in both graphs. For example, a visualization technique can be attached to a data preparation step like Figure 3.6 demonstrates it for the visualization of a projection obtained from principle components analysis (PCA) as points in a scatterplot.

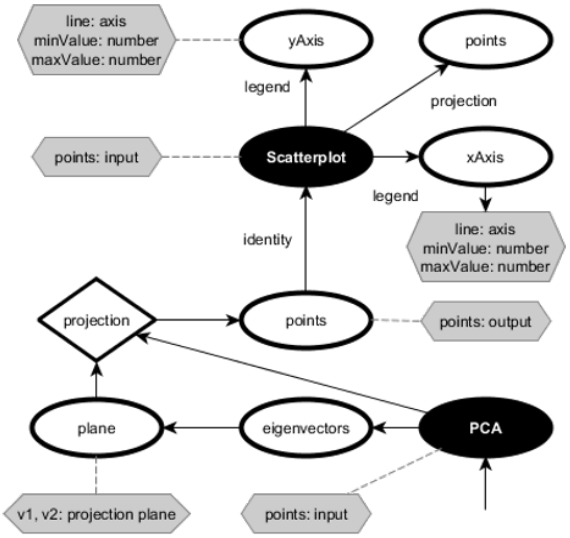


Figure 3.6: *Results of Principal Components Analysis (PCA), visualized as a scatterplot. Grey fields indicate an informal representation of the interpretation-function where entries are of the format (data:information). The output of PCA is a set of points in a plane which are passed to the scatterplot visualization module. This determines the scatterplot's raw data and thereby applies the visualization to the computed data. The diamond shape is a blank node used as a shortcut for functions taking multiple input parameters.*

Towards more sophisticated analysis, consider an example where an analyst tries to identify and highlight clusters in some multivariate data set. Using parallel coordinates, the analysis expert infers discriminating features based on the distribution of lines along the different axes and projects the data to these dimensions. Using some distance measure, the analyst runs a clustering algorithm, subdividing the data into k subsets in an iterative procedure. Feeding the cluster distribution back into the original data as an additional dimension, a new axis indicates the clusters which proves that in this setup, a parallel coordinate plot is capable of visualizing the inquired information which data item belongs to which cluster. However, the addition of an additional axis makes it hard to infer the clusters when scrolling sideward through the diagram. Therefore, the analyst chooses to extend the visualization by a color coding for the classes. The whole interaction and the corresponding transformation graphs are shown in Figure 3.7.

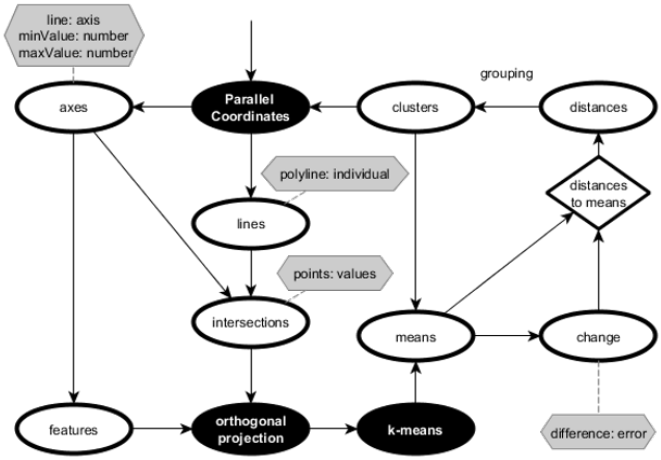


Figure 3.7: *Parallel Coordinates plots can visualize class membership. To prove that a visualization is possible, it does not suffice to provide the example – one has to prove the information is actually there. Here, the plot and an arbitrary clustering method are needed, for example k-means. For the proof, the clusters are fed back into the visualization system as part of the raw data. This is equivalent to adding another table entry to the original data and thus interpreted by the plot as an additional axis. With the new axis the visualization supports the inquired information about (computed) class compositions.*

In the example, the algorithms behind the different operations are simplified and much information is hidden in the transformations. However, it would be unsound to model every detail of the procedure, especially the sequential parts. Apart from the fact that this would not provide any further information, the sequences would be consumed by contraction anyway. In the example, apply contraction has not been applied yet so the the algorithms could be shown in detail. It is observed that the whole graph actually consists of four blocks, namely the parallel coordinates as the applied visualization technique, the projection, the clustering procedure, and the color map. These blocks are essentially independent of each other with the only restriction that blocks can only be connected

if the output data format of the one block fits the raw data format of the other. It is assumed that this is the case for real world applications of visual analytics suites – or achievable through the invocation of translation procedures. For example, there are tools to define data preprocessing pipelines by graphical programming like KNIME [15] in which this obviously has to be the case as otherwise, the constructed pipeline would not be executable. The major difference between the approach discussed here and these tools is that here, it is actually not intended to ask the analyst to define the paths manually. Instead, the analyst asks for nontrivial information patterns which the system extracts and visualizes automatically. Figure 3.8 shows an example of a more powerful collection of analysis and visualization techniques. Although the collection is still rather small, it can already serve nontrivial requests like the demand for a visualization technique which optimally preserves some feature that is only present in derived data, for example cluster membership or the alignment of scatterplots in a scatterplot matrix. This is possible because the visualization blocks hold information not only about their general properties but also about their applicability and specific advantages.

3.5.2 Towards Automatic Pipeline Generation

Prior to the definition of the algorithm, the relationships between the previously obtained results should be summarized and it should be discussed how they enable an automatic generation of visual analytics pipelines which always terminates and provides correct results. A result is defined as being correct if it either returns that the inquired information is not derivable or it derives the information completely and correctly and presents it in a suitable visualization with respect to the information to be conveyed and the analyst's additional requirements.

Since each block is a set with only finitely many data transformations and only a finite number of blocks is available, the conditions for decidability of the restricted derivability problem apply if it is also required that a finite upper bound for the length of sequences of connected blocks exists. Alternatively, one can demand each sequence of blocks to eventually contain a visualization and the computation to stop there until further processing is manually invoked by the user. Since this also guarantees termination of the computation, the conditions for the decidability of the restricted derivability problem are still met and derivability of information is decidable. Since this implies that the lemma on the decidability of information conveyance also holds, the system's ability to visualize information – be it explicitly or by intrinsic inference – is also decidable. Note that extrinsic inference is never decidable since it depends on the viewer's understanding of the data. The fact that it is possible to infer the correct information does not necessarily prevent misinterpretation. Since functionality can be wrapped into blocks which can be aligned in transformation sequences, a variety of different views on the data can be computed. The theorem on the decidability of the restricted derivability problem and the lemma on the decidability of information conveyance trivially hold across execution blocks if one chooses to connect the blocks by sets of identity transformations. Since the amount of data is finite, the transformations also terminate. Because every

visualization block supports information about its own applicability and quality, the algorithm terminates, is capable of deriving the visualization correctly, and to visualize it with the most proper available method. The final piece in the puzzle is the contraction procedure. Being applied to each block, it ultimately reduces the representation to the form of the system shown in the upper left corner of Figure 3.8, representing each process by a single node after abstracting away the implementation. Therefore, existing software suites can be applied to the actual computation of the transformation sequences obtained by the algorithm – as long correct descriptions of the information being derivable in the respective modules are provided.

For a visualization system with a set of data transformation blocks and a set of visualization blocks, the algorithm first computes the contraction for each of the transformation graphs constituting the several blocks. In a second step, it establishes the links between the blocks by checking for each pair of blocks whether the information in the output nodes is contained in the information of the other block's input node. The algorithm assumes that data formats are compatible if the information matches. The resulting graph is exposed to the viewer as a means to communicate the transformation paths applied towards visualizing the data. This way, the procedure is transparent to an analyst who can manually readjust transformation paths if needed. This preprocessing is only needed while setting up the program. As long as no new blocks are added to the system, its results can actually be saved in a configuration file. The last preparation step is to execute contraction on the graph of blocks as illustrated in Figure 3.8. Recall that contraction gathers all information in a linked list under a tree of transformation paths connecting contracted data nodes. To infer the correct visualization for some set of information, the algorithm first verifies the derivability of each item in the set using the algorithm outlined in the section on automatic extraction of information. If the item is derivable, the corresponding path is also available. For each of the inferable elements, the corresponding subset of the linked list is scanned for contained information about visualizations applicable to this information. Again, the transformation paths are known since they have been preserved by the contraction procedure. Traversing the tree, the algorithm finds the sections in the linked list of information items that correspond to the respective visualization techniques and traverses them for information to which kind of data they apply best. With this information, a ranking is computed which part of the derived data should be visualized with which technique. Note that if additional criteria on the visualization have been inquired by the analyst, the corresponding techniques have already been detected as part of the first steps and are ranked higher than the remaining techniques. Also note that visualizations themselves can also be treated as data allowing for aligned and nested visualizations if the analyst requires them. Actual implementations of the models for data and information as well as the ranking procedure are application specific. One solution for the ranking is to apply the effectiveness criterion Mackinlay proposes for APT [92]. Some other applicable data and information models are mentioned in this chapter's related work section.

3.5.3 An Example

For a more intuitive access to how the system works, an example is discussed in the following. Assume an analysis of the well-known Iris flower data set containing measurements of the petals of 150 Iris flowers of three different species in four dimensions. Further assume an application of the analysis system shown in Figure 3.8. As a preprocessing step, the system's graph would be collapsed just like it is shown in the figure. A first query to the analysis system could now be to ask for *a single-projection overview that minimizes the projection error*. The system recognizes that it can load data, does not need to apply any preprocessing and that among the information associated with PCA as a projection techniques, there is are predicates formalizing that PCA optimizes variance and minimizes the projection error. Since the output of PCA is data points, it would then opt for the scatterplot as the visualization technique of choice. Because the collapsing collects the transformation paths and stores them for each collapsed cell, the computer can retrieve the transformation algorithms directly. The path obtained by connecting the blocks would this be exactly the graph depicted in Figure 3.6. Starting from this overview, the analyst would be interested in *investigating local neighborhoods of similar items*. Going through the collapsed graph immediately reveals that the system is capable of providing this information because neighborhood graphs are included in the toolset and node-link diagrams have the information associated that they can show neighborhoods. The computer selects to compute the k nearest neighbors for each node since the analyst asked for local neighborhoods and applies the scatterplot from before to position the graph's nodes. From this image, the analyst gets the impression that there must be three groups among the flowers. The next question thus is to *somehow group the flowers by their similarity*. Clustering algorithms are associated with the information that they group elements, so the computer attempts to find a suitable path through the derivation graph containing a clustering algorithm. k-means is chosen since it is based on local similarities. As a result, the flowers are tagged with their class names and the tags are passed to the scatterplot together with a colormap for the groups. The three groups of flowers in the dataset have thus been identified. With a more sophisticated system than the one applied in this szenario, the analyst could for example ask to *refine the clustering based on local cluster outliers*. Given the necessary information in some nodes, the system could derive strategies like filtering outliers in each single cluster and attempt to rematch them to the cluster they best in the best.

3.6 Summary and Discussion

This chapter proposes an extension of the visual analytics pipeline achieving a tight integration of data mining, visualization, and machine learning and a feasibility proof for the information-driven automatic construction of visual analytics pipelines along with an algorithm for its realization. Where in the original workflow the information obtained from automatic analysis is available for the human analyst but the insights obtained are not meant to be processed by the computer, the new workflow applies machine learning to close this gap.

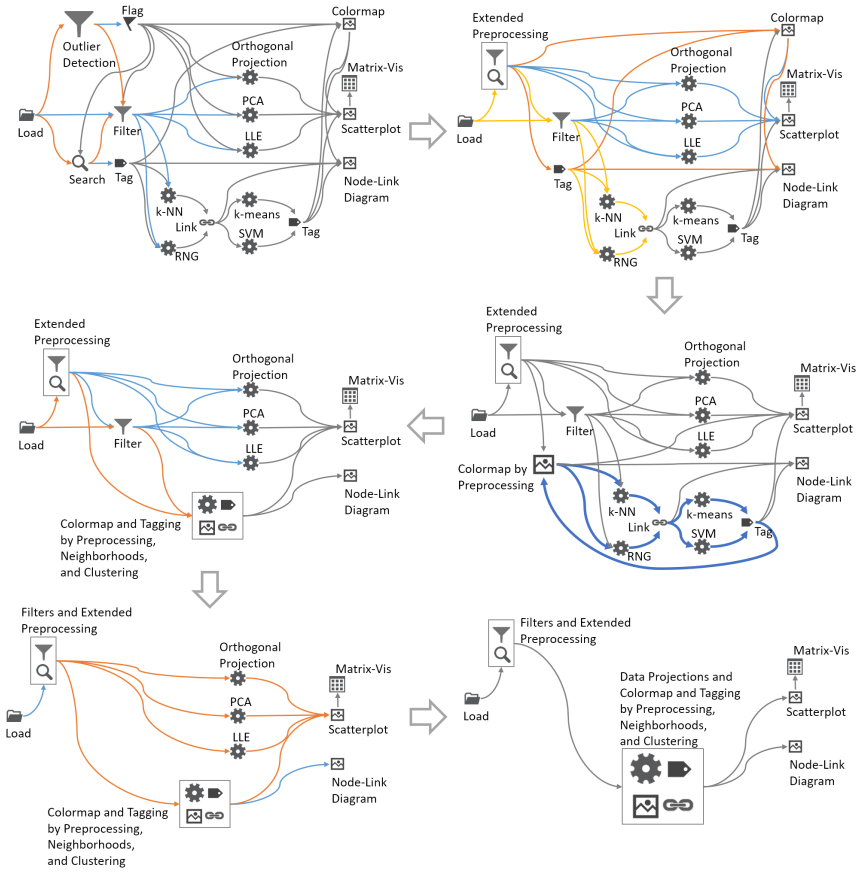


Figure 3.8: *Collapsing applied to a set of transformation and visualization algorithms. To indicate that the algorithms have already been contracted and to emphasize their role in the visualization system, different icons are applied for their visualization. The diagram is to be read clockwise; the entries to the left are the original state and the final state when no further contraction is possible. Determination of fork-join-structures is based on a breadth first search starting in a given center. Blue edges denote states already covered but without success, yellow denotes paths of the graph that are candidates to which contraction has not yet been applied due to the orange structures containing less nodes or being closer to the original data. Whenever something changes, the algorithm starts to check the corresponding entries.*

The resulting iterative workflow leverages the three domains' respective specific strengths to foster the obtainment of new insights from previous results. Implementations of this workflow can be expected to increase the efficiency of data analysis, yielding more sophisticated insight in less time.

Focusing on insight provenance means to emphasize the documentation of the analysis process and its results. A detailed provenance structure requires a mechanism to protocol analysis results in the visualization – for example highlighting structure in the data and tagging this structure with semantically

meaningful labels.

To support this kind of workflow, the automatic generation of visual analysis pipelines is considered. To this end, the problem of deriving information from raw data is investigated. The encountered reachability problem can be solved for almost all real world applications. A schematic for an algorithm for the automatic generation of visual analytics pipelines is outlined based on the findings discussed. The obtained decidability results motivate the outline of an algorithm for the automatic derivation and visualization of data with respect to inquired information. Being independent from specific data and information models, the technique is generally applicable. However, this comes at the cost of a certain vagueness in the descriptions of the algorithms whenever the data structure or the organization of information becomes important for the discussion. The results are of a general nature, leaving the definitions of the applied models for information and data to the implementation. Yet, the models discussed in this Thesis are perfectly compatible with the mechanism.

The execution of a sequence of data transformations is organized in combinable building blocks. Each of them is associated with its own information, allowing the system to leverage this information to derive insights about the data. Again, keeping the system general means that this information is not data-specific. Therefore, in a plug-and-play scenario where some data is passed to an analyst who then intends to use the system for data analysis, only general information about the respective algorithm's effect on the data can be inquired. In contrast, data specific information can be added in analysis environments where the information associated with data does not change frequently, allowing for more sophisticated information inquiries tailored to the specific application. During the discussion of the framework, the necessity to add additional information directly to the data has been avoided since this requires an information model and thus limits the discussion's generality. However, the potential of interactive feedback of analysis results into the data and transformation model should not be underestimated. For example, reintegration of information can help to iteratively optimize information derivation procedures or to quickly reproduce analysis results on other data. The structured workflow introduced in the first part of this chapter can be applied to achieve this.

One weak point of automatic visualization is the ranking of the different methods to be applied for different tasks. It could be expected this to become even worse if the automatization is extended to the whole pipeline, for example when multiple algorithms for clustering or classification are applicable to the same data. Fortunately, this is only partially true. Indeed, problems occur wherever a ranking of the performance of methods is involved. However, the mathematical toolset applied for automatic analysis and data preparation can be measured or at least compared relative to the individual algorithms' performance on representative test data whereas the performance of visualization has to be evaluated based on human perception and cognition The latter can be achieved applying the concept graph to construct an expected domain-specific mental model whose concepts can be determined for each visualization. Based on this construction, the decision whether the inquired knowledge is supported by the visualization is obtained by the computation of the applicable semantics in the semantic aggregation formalism proposed in Chapter 2 by computing the

semantic aggregation of the data under the assumption a specific visualization was applied to the data. It is concluded that in most applications the uncertainty bottleneck in finding optimal solutions will remain with the identification of the proper visualization technique. Combined with a system implementing the workflow described in the first part of this chapter, insights found during analysis can be reintegrated into the system. This includes general insights about visualization obtained from evaluating visualization techniques. Based on this information associated with the algorithms instead of having to inquire the optimization of statistical or numerical properties like "the least amount of clutter" or "best preservation of distances", an analyst could literally ask for "the best view on my data".

Making the final transformation procedure transparent to the user is crucial for the reproducibility of results, especially because it explains the visualization. In this section, this documentation of the program's performance is established by presenting the graph of linked blocks to the user. The graph also serves as an interface for interaction and exploration, allowing to recursively unfold or collapse the nodes to investigate the transformation procedures they represent and the transformation sequences chosen by the algorithm. Interaction for editing or extending transformation sequences allow the user to improve and extend the presentation if needed.

There are some open questions left where the implementation of the algorithm for automatic information derivation is concerned. However, one could also replace the automatic approach by a semiautomatic one, where the computer only proposes viable transformation paths rather than deciding between them. Because the optimal solution depends partly on the viewer, a semiautomatic implementation letting the viewer decide between the alternatives avoids the necessity to provide measures and rankings for optimal data representation. On the other side, the fully automated approach is easier to use, especially for novice users. The main concern to propose the fully automatic algorithm was to show that it is actually possible to define a structure of data and its associated information in which it is – even if further restrictions have to apply – derivability and the ability to visualize certain facts are decidable. This possibility yields that there is a formalism in which every visualization can be expressed (by its pipeline) and in which it is decidable or at least positively semi-decidable for visualizations whether the information they are intended to convey is derivable and can be visualized. Indeed, this is achieved by the computation of applicable semantics in the concept graph based on semantic aggregation.

Core References

- B. Karer, I. Scheler, and H. Hagen: Panning for Insight: Amplifying Insight through Tight Integration of Machine Learning, Data Mining, and Visualization. In I. Nabney, J. Peltonen, and D. Archambault (Eds.) *Machine Learning Methods in Visualisation for Big Data 2018*. The Eurographics Association, 2018
- B. Karer, I. Scheler, and H. Hagen: A Step Towards Automatic Visual Analytics Pipeline Generation. *Electronic Imaging 2018*, IS&T, 2018 **Kostas Pantazos Memorial Award for Outstanding Paper in Visualization and Data Analysis**

Chapter 4

Qualitative Considerations for Visualization Design

This chapter is concerned with the third major research question addressed in this Thesis, the influence of qualitative considerations on visualization design. The discussion starts with an excursion on lessons learned during several years of cooperating with the German police in a project to design a visual interactive system for the support of manhunts in the vicinity of crime scenes as part of the early response to a crime incident. As the discussion reveals, qualitative considerations indeed influence the design workflow positively. To further investigate this, the chapter continues with two case studies. The first case study is concerned with the influence of qualitative considerations on the general design process. To this end, it discusses the solution approach to the problem of limited domain information in the project with the German police. In this project, limited access to classified information on the intention behind necessary interaction steps hindered the design of an efficient interaction and overview visualization supporting the task. The solution is based on the explicit discussion of qualitative information implicitly formalizing the reading language without explicit construction of the mental model. Compared to information visualization, the correlation between the data's structure and its graphical representation is much stronger in scientific visualization. Typically, this binding determines the geometry of the resulting representation. For this reason, the geometric representation ideally reflects the viewer's anticipated reasoning. If this reasoning is based on the shape, this means that the geometry should properly reflect this shape. The second case study therefore discusses the development of a geometric representation of thin inextensible elastic surface strips that reflects local reasoning based on the bending and twisting behavior to encode important properties of the strip into semantically meaningful parameters with an unambiguous interpretation.

4.1 Practice Lessons – Learned the Hard Way

The design process of visualization applications typically follows a linear workflow of establishing an understanding of the problem and application domain, designing and implementing the solution, and verifying it with feedback from domain experts. High-quality solutions to application problems typically require a considerable effort to tailor the visualization to the application domain's special demands. This requires designers of visualizations to acquire an in-depth understanding of the domain, the data, and the analysis process to be applied. However, even though visualization experts are well used to working in interdisciplinary projects, there are cases where it is hard or even impossible for the visualization expert to acquire the necessary amount of domain insight to deliver solutions tailored to the application's domain-specific needs. Perhaps among the hardest examples are projects where major aspects of the domain knowledge are confidential and therefore inaccessible to the visualization expert. Establishing a fruitful cooperation in such a constrained scenario is challenging since it demands visualization experts to design a solution to a problem they can hardly develop an in-depth understanding for. Still, they need to provide high-quality solutions while respecting concerns for privacy and security, especially in the public sector.

This section is a report of some experiences made in a long-term project with the German police. It presents how the issues emerging from limited communication and domain understanding have been resolved and how the project has project became a success. Therefore, the focus is not on the project's actual results but how they were achieved despite not knowing the details of the ongoing processes. The project's aim was to improve the performance of manhunts in the close vicinity of a crime scene as an immediate response to the crime incident. Since the details behind the organization of manhunts are confidential, the designers were forced to optimize the solution towards a problem they would never be able to understand thoroughly. It should be emphasized that the tactics themselves are secret. User studies with sanitized data, for example on fictional manhunts, would still have revealed the tactics to the scientists conducting the experiment. As a replacement for such studies, close contact with an experienced practitioner was established. His role was to evaluate the approaches from the domain experts' perspective and to steer the development into a direction generating an added value for the practitioners without revealing classified information. What began as a pragmatic decision to provide at least some means to incorporate domain knowledge into the development process eventually became a success model. The occurrences reported here influenced the project's fate decisively: If those issues had not been overcome, the project would have failed.

4.1.1 Dealing with Confidential Domain Information

A manhunt in an area surrounding a crime scene can roughly be described as a number of patrol cars searching a number of sectors around the crime scene or place of an accident for a person or an object. Problems occur from sub-

optimal search patterns resulting from the fact that the search paths of the individual units are not coordinated. Coordinating these paths, however, is also not possible since it would likely bind too much attention to following the route rather than observing the surrounding. However, the decision which sectors to apply or how many units to send there is up to the police officer in charge, the dispatcher. The dispatcher's role is to coordinate the measures taken to counter a crime. Being in charge of the operation, the dispatcher must maintain an overview over a possibly highly complex situation while coordinating numerous forces and making quick decisions considering a maximum of possible alternatives for the fugitive's behavior.

The initial project description asked to improve the performance of manhunts in an area surrounding a crime scene by visual communication of a somehow optimized spatial distribution of sectors on a map. The first question of the designers was, of course, how this process actually worked. The idea: If the process is understood, tasks can be identified and the visualization can be optimized towards those tasks. However, the tactical details were confidential. Hence the only thing known to the designers was that a few patrol cars were driving in sectors close to a crime scene.

In the first attempt, Voronoi-Diagrams were mapped around a number of seed points in the street network such that the resulting sectors were bounded by a cycle in the graph given by the street network. This first solution was not dynamic enough and needed to better adapt to the situation, especially if additional units became available. As it turned out, the requirements had changed in the meantime due to the complexity of new potential situations: A system was needed that could dynamically adapt to the development of the situation. Eventually the obvious had to be admitted: one cannot optimize an unknown process.

The conclusion: If the application workflows are secret but the visualization and interaction need to support them, they should be designed in close cooperation with the application domain. More precisely: They should be *designed by practitioners* from the application domain. With the police's help, tasks have been defined independently of their context within the manhunt. Those tasks were the definition of an area of action to be subdivided, the number of sectors, the dispatch of available units, and the subdivision or deletion of existing sectors to react to the dynamics of the situation. The close cooperation with a particular practitioner (a dispatcher) as a direct contact provided further insight into how the sectors would be communicated – without the need to reveal confidential information. This information was applied to define a better refinement criterion for the algorithm identifying the sectors on the map and soon report a substantial gain in the quality of the developed prototype was achieved.

4.1.2 Clear Requirements despite Communication Limitations

When the project started, only little information was available but a few functional requirements, a very rough idea of the processes going on and the requirement that the solution would have to be intuitive. The task to decipher what "intuitive" meant was – for the moment – left to the designers. For some

reason, the requirement for an intuitive design is still a common task visualization experts are expected to easily achieve. In the first follow-up meeting, the application partners already asked to put more emphasis on intuition. The construction was reviewed to achieve an even more simple description of the search spaces and a graphical user interface was implemented based on localized context menus that would work on touch-interfaces and desktop systems alike. The devastating remark in the next meeting came directly from a police officer who would apply the software as dispatcher: "That's too much clicking for me".

Yet, the progress made in the meantime in optimizing the subdivision of the search space fortunately convinced the police to continue the project. A refined list of requirements for the user interface was promised to be provided. As a direct reaction to the meeting, it was also decided to change the development paradigm to prototype-driven tests. From now on, mockups and demonstrators would be integrated into the development of further interaction methods in close collaboration with the police officer who would eventually be the user of the final application.

This incident made the domain experts aware that the requirement for intuitive software was by far not as clear as they had been convinced it was. To counter the problem, a close collaboration with this expert was established in order to provide the necessary information of how the dispatcher would work with the developed software. Of course, confidential details could still not be revealed but by using conceptual sketches to discuss and plan the interaction process, mockups to demonstrate new ideas early, and prototypes to iteratively approach "intuitive" solutions, the expert's process and domain knowledge could be exploited to design an efficient workflow. On the other side, this close collaboration enabled a quick extension of the prototype to meet the demands of the police for additional features. With the demonstrators being easy to adapt and extend, entire system for unit dispatch was mocked within a few hours (that is, within a single night shift at the police department) after reviewing and refining a sketch provided by the police officers and adapting it to the application. The system is actually working within the framework. It only lacks access to a service providing the actual police data – again confidential information.

4.1.3 Reducing the Gravity of Inevitable Changes

Throughout the course of the project, the designers were frequently faced with changing requirements and the need to incorporate additional use cases. Almost every time a new iteration was presented to the application partners, new potential was found that would cause the requirements to change so drastically at least one module of the software would have to be redeveloped from scratch. Had the designers known and understood the details of the problems they were to solve, they could possibly have foreseen this. Yet, in applications where these details need to be kept secret, such events seem inevitable and should be expected when designing the software's architecture. In particular, the software needs to be quickly adjustable to drastic changes. This requires a highly modular structure of independent software packages. Fortunately, an early design

decision was to implement a service oriented architecture where the visualization would be deployed as a web service that could access a number of services providing data and functionality via a service broker. Different modules could therefore be developed in parallel and independent of each other. Within the structure of the final development workflow, this also allowed to focus on single components without having to worry about the "big picture" of the whole software system.

Especially in the first minutes after a crime has been committed, available validated information is sparse. There are many alternatives and possibilities but only few reliable facts to base a decision on. Still, the dispatcher has to act quickly. Yet, given the sparsity of reliable information, statistical predictions or extrapolation of ongoing movement would most likely deliver wrong results. While technically it is possible to provide such simulations, it would be unreliable and therefore unsound to do so. The designers hence proposed to instead provide context-sensitive additional information for decision support. For example, the positions of bus stops or parks could be highlighted or added as a map overlay if the dispatcher considers them relevant for the definition of sectors or for unit dispatch. The decision, however, should be left with the dispatcher rather than the computer since being in charge of the operation, the dispatcher is also accountable for failure.

4.1.4 Breaking the Ice

Having given a number of invited talks for numerous audiences, from potential users to decision makers close to politics, the reactions were just as diverse as the audiences. Probably the most critical question has been asked in a talk for practitioners: "Is your tool meant to replace the experienced police officer at the local police station?". Of course, it is not. However, this question is symptomatic for an acceptance problem that is not depending on the domain of application but simply the result of an understandable reluctance of practitioners to accept the ideas of a non-domain-expert without further evidence of the proposed solution's quality. One frequent comment among the audiences of the invited talks was that practitioners would suffer from poor usability of software products not developed towards efficient interaction. Especially in the time-critical situation of a manhunt, efficient interaction with the dispatch system is a self-explaining necessity. Therefore it is not very surprising that in some talks there were members of the audience who only were convinced of the developed tool's quality after a live-demonstration of the software was presented. Eventually, the live-demonstration became an integral part of the invited talks – with all the risks coming with a potentially not completely bugfree software. Despite an initial fear that bugs in the software would ruin the presentation, the surprising experience was made that even if an error occurred the trust built up by the demonstration exceeded the doubt emerging from the occurrence of the bug.

Since the police officer the designers directly collaborated with was so deeply involved in the design of the user interface, it was only natural to invite him to accompany the talks and discuss the aspects more relevant to the application

domain. The rationale behind that is simple: If the domain experts are the ones who can define best what they need to solve their problems, they are also the ones who are best at explaining the identified solutions to their fellow practitioners. The designers' role in the talks was reduced to the discussion of relevant technical aspects and operating the demonstration scenario which the domain expert would comment from the police's point of view. Indeed, audiences were much less reluctant to accept the solution if the application aspects were explained by the domain expert rather than the developers of the system.

4.1.5 Iterative Development Workflows for Design Projects

During the course of the project, not only the requirements changed but also the focus of the project itself. Additionally, the fact that important domain information was confidential had to be compensated throughout the project. As a consequence, traditional linear software development models were inapplicable and a dynamic and agile development workflow had to be implemented that would incorporate the expertise of the application domain into the design and refinement process.

Because the developed visualization was meant to improve an existing process the help of a domain expert was needed to define the visualization's features. While it was clear that practitioners' perspective and knowledge had to be incorporated into the design process, it soon became clear that a feature-oriented development paradigm like Scrum [127] would be hard to implement. The problem was that, not being a visualization expert, the practitioner would be forced to define features during the design phase of which he did not know they would be useful later. What the practitioner could provide was information about the shortcomings of existing workflows and where improvement was necessary – as long as this would not reveal too much about the tactics. Therefore, the development-workflow that eventually emerged is following a process-oriented paradigm rather than focusing on software features.

In retrospect, while – as intended – the actual implementation phases became much shorter the further the development paradigm approached the iterative workflow, getting the required feedback from the domain experts took considerably more time than initially expected. The explanation is that initially not enough time had been scheduled to compensate for the limited availability of the stakeholders. Their great dedication to the project has to be emphasized. Still, even making meetings possible on short notice when it was necessary, the higher in rank the participating stakeholders are, the harder it is to arrange an appointment, especially since due to security restrictions not all of them are allowed to receive phone calls or emails from outside the police. Fortunately, the cooperation partners offered to coordinate the meetings themselves. The iteration cycles were shortened by decomposing the project into smaller parts which could be developed asynchronously. This way, the direct contact among the practitioners could take over the meetings on the low level alone and larger stakeholder meetings only had to be arranged when milestones had been reached or directional decisions were to be made. Due to the better availability and the

practitioners' dedication, development time of individual work packages reduced from about a year to roughly four months, including testing, fixing bugs, and necessary changes demanded by the stakeholders.

4.1.6 Summary: Qualitative Considerations Influence Design

All the issues reported above are of qualitative nature and although the report concentrates on a specific project, the findings are not specific to the project's restrictions and hence can be generalized to other design projects. Thereby, the report already points out that qualitative considerations are useful for the design of applications. It is remarkable that these qualitative considerations are rarely pointed out explicitly as such. Instead they implicitly become part of the design in the form of design choices or design requirements. A closer look into the literature also reveals that the three aspects of interpretation, provenance, and design are commonly not considered in combination. Like any other design workflow, a workflow focusing on qualitative visual analysis would attempt to first identify and implement the domain perspective. Deviating from the purely data-centric perspective, the second step would be the identification of the reasoning and inference mechanisms typically applied in the domain. This would prepare the design of proper analysis processes to be implemented along with the visualization. The visualization would then be designed towards reflecting the interpretation on the domain's reasoning and inference techniques.

The benefit in development efficiency can be assessed from the timeline shown in Figure 4.1. The further the development workflow evolved into the final form, that is the further it developed from a data-driven approach to a workflow guided by qualitative considerations, the shorter the development cycles became due to an increasing influence of the practitioners on the visualization's design.

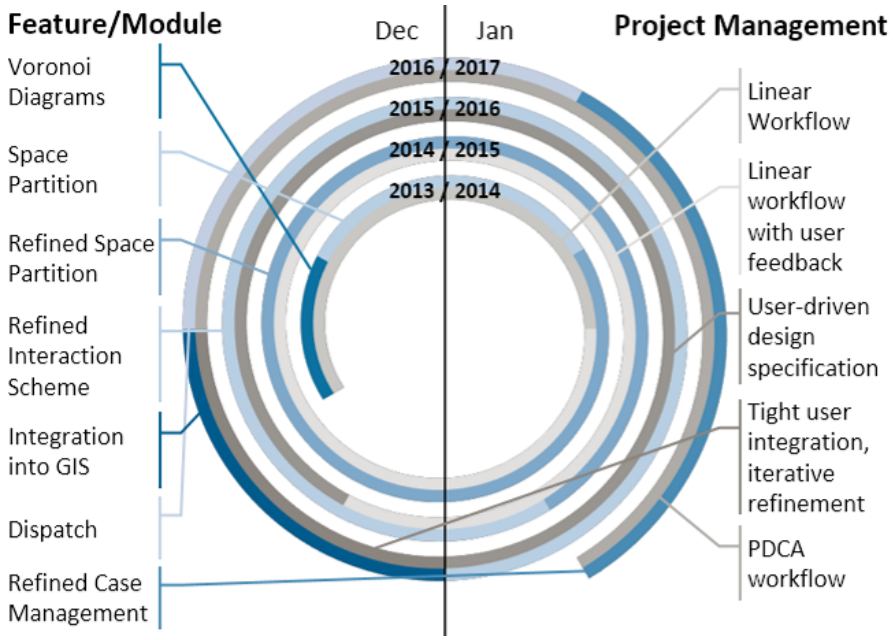


Figure 4.1: *Timeline illustrating the benefit in development speed due to the design workflow's evolution over time. Without access to important domain information, early versions of the space partition and its visualization required time-consuming readjustments. Involving a domain expert into the conceptualization of the user interface reduced the necessity for subsequent changes and therefore the development time significantly.*

4.2 Case Study: Designing Interactive Visualizations Despite Sparse Availability of Domain Information

Confidential information is an important issue when developing visualization applications for the security sector. The visualization and user interface often need to be optimized towards the task to be performed. Such an optimization requires an understanding of the application domain's processes and procedures. In policing, an example for such a processes is a manhunt initiated as part of the first response to a crime reported to the police. Information about the police's tactics is not publicly available but would be needed to optimize a visual interactive system for managing manhunts.

Especially if relevant and important domain information is confidential, the development of an optimal solution needs to consider this information without explicitly communicating it. The following discussion introduces such an approach. In particular:

1. A process-oriented approach to the development of visual interactive systems is explained. The problems of sparse domain information are circumvented by deriving implicitly communicated design goals from discussions with practitioners.
2. Two implementations of this approach applying different software development project management paradigms are described. Both examples are presented together with an example from a project with the German police. Sparsity of domain information results from confidential tactics and unclear design goals.
3. The approach's performance for the two examples and its applicability to other projects suffering from sparse availability of domain information is discussed.

The development approach discussed here is the result of a series of adaptations to challenges encountered during an ongoing project with the German police that started in 2013. The discussion is primarily concerned with the derivation of design goals in environments where access to domain information important for design considerations is severely limited, for example because it is confidential. For the project, this means that police officers were able to express and specify important properties of the final solution's design without the need to reveal tactical considerations behind the design decisions.

4.2.1 Context: Manhunts in the Vicinity of Crime Scenes

This section discusses a case study based on a project with the german police. Its initial aim was to improve a sector-based technique for manhunts in the vicinity of crime scenes. This specific kind of manhunts is typically executed immediately after a crime incident has been reported. Hunting a fugitive criminal in this early phase of a situation is hard since due to its immediate nature, typically only few information about the situation is available. Hence, sparsity of domain information in this case results from two problems: First, the exact tactics are

confidential information so the interaction system has to be tailored towards processes not revealed to the visualization experts. Second, there is no complete description of the situation to be handled and therefore no complete definition of the information to be visualized.

Speed is the key to a successful response to the situation. Therefore, the dispatcher has to be enabled to quickly identify the situation and the known facts, identify the available units, and to dispatch them according to the tactics related to the kind of manhunt the dispatcher decides for. Additionally, this decision has to be easy to communicate to ensure efficient unit dispatch. In the following discussion, two different approaches are considered, namely sector- and ring-based manhunts in the vicinity of a crime scene.

4.2.1.1 Sector-based Manhunts

The sector-based approach to manhunts in the vicinity of a crime scene is based on a subdivision of the area to be searched for a fugitive criminal into a collection of sectors used to coordinate the dispatched units. The sectors' boundaries are known to the police officers and additionally made available aboard the patrol cars. An ideal sector is convex with respect to the street map's topology, which means that every street inside the sector is reachable following a shortest path without leaving the sector. Note that convexity in the street map's topology does not necessarily geometric covexity on the street map. Additionally, a sector boundary should be closed and simple to describe. The latter requirement can be achieved by minimizing the number of street names involved in the description. Figure 4.2 shows the evolution of the sector shapes applied in the developed project prototype over the years and reveals how they came closer to reach these design requirements. Note that these requirements are not the requirements the police imposes to actually define the sectors but serve as a good functional basis to develop a prototype with. The actual specifications are confidential.

4.2.1.2 Ring-based Manhunts

The idea behind the ring-based approach is to intercept a fugitive by covering so-called neuralgic spots with patrols. The neuralgic spots are selected by a certain definition of centrality and positioned relative to a ring defined around the crime scene – hence the method's name. In the developed prototype solution, the defining ring is intersected with the paths of higher-level roads and crossings closest to these intersections are found along these roads. Like for the sector-based approach, the actual set of properties defining the neuralgic spots is confidential and only present a "working model" that helps designing the prototype.

4.2.2 Related Work

In general, visualization design frameworks offer high-level guidance or process steps to support visualization design. The most widely used standardized ap-

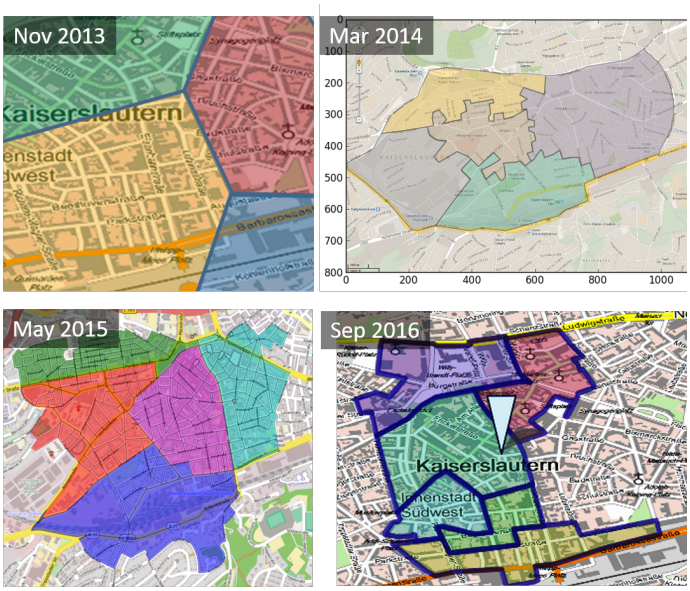


Figure 4.2: *Evolution of the subdivision over the years. Starting with Voronoi diagrams it quickly became obvious that the sectors would better be bound to the street network. This would simplify communication via radio and better relate the sector boundaries to the actual operation space. The details behind the system currently in action are confidential and thus not accessible to the visualization designers. Therefore, new criteria had to be derived indirectly from discussions with practitioners.*

proach to developing visualizations is the visualization pipeline [64]. It roughly describes four separate stages: data analysis of the given raw data, filtering of its result, the mapping to graphical primitives best suited to convey the data's information, and rendering the result to the display. Many models for visualization design follow the pipeline approach and its several extensions and alterations.

Developing linearly along the pipeline can result in a failure of the whole project as a consequence of potentially fatal misconceptions in the design of the data processing or rendering steps. To alleviate this risk, newer model specifications attempt to better manage the requirement for changes and to frequently update the design during the course of development, for example by following an incremental approach to development [125]. Another approach is to add verification steps to the pipeline, similar to the V-model's extension of the Waterfall paradigm. Tamara Munzner's Nested Model for Visualization, for example, introduces four layers to be implemented incrementally and proposes proper evaluation techniques for each development layer, [96]. The Five Design-Sheets methodology (FdS) is a relatively recent addition to the visualization design literature [118]. It outlines an efficient design workflow applying sketches to foster reflection on ideas and reconsideration of decisions during the design phase. Although the method discussed here is based on implementing prototypes, sketches are applied during discussions with domain experts. To be applicable, FdS requires the designer to understand the task and – even more important – to access real-world data. The authors explicitly mention privacy reasons as a factor preventing the application of FdS. Indeed, a major challenge specific to confidential domain information is that in some projects designers are permanently confronted with the need to develop a solution without ever being granted access to real-world data. While more work is needed in this direction, the results discussed below indicate that, in combination with the presented process-oriented iterative approach, there are projects where FdS can actually be applied even if domain information is sparse due to access restrictions.

Concerning the general feasibility of visualization application development, Sedlmair et al. identify a list of 32 pitfalls posing a risk of failure to visualization projects [128]. Some of these pitfalls are contradicted by the findings reported below. A more detailed discussion follows in Section 4.2.7.

The methodology presented here is related to participatory design, letting potential end users take part in the development process [19]. Yet, the end users are no experts for interaction or visualization and, due to confidentiality, are not allowed to openly reveal their domain knowledge or allowing the desigers to study the tasks and processes in their actual application contexts. Without these limitations, derived the requirements could have been derived by field research methods, such as contextual inquiry [47, 67]. This particular method has been reported to be effective in geovisualization design [89]. Since the application is based upon a geospatial visualization of situations in crime fighting, this approach would certainly have been useful. The common factor between these methods is that they rely heavily on an open exchange of domain knowledge and real-world data, a requirement that cannot be met if the availability of domain information is limited by confidentiality.

To take advantage of the practitioners' domain knowledge, an idea related

to Schön's reflective practitioner [124] is followed, attempting to let designers learn from an experienced practitioners' reflections on existing applications and processes. The limitations imposed by confidentiality, however, restrict the applicability of this approach to the extraction of domain information from the implicit communication between practitioners. To this end, a dialog setup is established where practitioners comment and share ideas proposed and refined by designers. For such discussion setups, it has been proposed to involve at least one participant in the role of a translator or liaison to bridge the communication gap between domain and visualization experts [128, 131]. If available domain information is sparse but important for the design, this is in general a good idea. Since the special restrictions of confidentiality essentially prohibit the communication gap from ever being bridged, the only option would be to install a so-called interdisciplinary liaison, a person with knowledge in visualization as well as in the domain [131]. Not being allowed to pass domain knowledge to the designers, the liaison's role would require expert knowledge in both domains. It can be expected that such an expert is rarely available, especially if general availability of domain information is sparse.

Although only single prototypes have been implemented in the project, the proposed discussion setup implements aspects of parallel prototyping [44]. Multiple visualization and interaction designers propose their ideas independently but work on the same design and consider the other designers' changes to the design in their ideas. Towards a more efficient design process, the discussions are, however, based on sketches. In some sense, this particular component of the approach to the derivation of design goals can be interpreted as a combination of parallel prototyping and the FdS methodology. However, both techniques actually require domain knowledge to yield high quality results. Embedding the discussion in an iterative workflow aiming at continuous improvement of the obtained solution rather than the instant obtainment of perfect results partly alleviates the sparsity of available domain information.

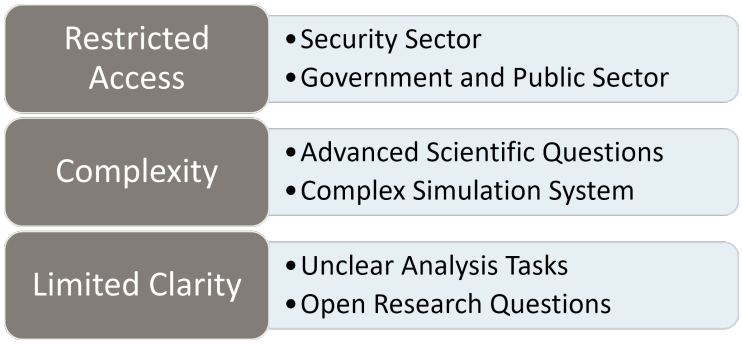


Figure 4.3: *The reasons for projects to suffer from sparse availability of domain information can be split into three major non-disjoint categories.*

4.2.3 Sparse Domain Information

Sparsity of available domain information hinders the designer of a visualization and interaction system in tailoring the developed solution towards the specific needs of the application and the system's eventual users. Without access to detailed domain knowledge, the challenge to create a system well-supporting its users in their tasks becomes significantly harder.

There are many reasons for domain information to be sparse. For this work's considerations, they can roughly be categorized into three classes, as listed in Figure 4.3. The focus here is on limited availability of domain information in the public sector. To be precise, the access to background information concerning the procedures and tactics for manhunts in the vicinity of crime scenes is restricted, the corresponding documents are confidential.

In a linear development workflow, the software development process would start with the identification of functional requirements which would then be translated into technical specifications of the software to be implemented. Of course, the police knows their processes and therefore could specify functionality aspects to be met by a software solution. However, defining such requirements becomes harder for the police where aspects are concerned that are specific to the software itself rather than the requirements directly reflecting policing tasks. Visualization and interaction are two important representatives of such requirements. Being no visualization experts, the police initially only required the interface to be "intuitive". The police acknowledged that deriving tasks from the procedures executed during a manhunt would be a reasonable approach to identify what factors rendered an interaction intuitive for them. Yet, they had to deny access to this information because it is confidential. Specifying functional requirements to the visualization and interaction based on tasks and processes was therefore not an option.

4.2.4 Deriving Design Goals from Implicit Information

Instead of formalizing design goals for the interaction as functional software requirements, a process-oriented perspective was eventually taken over. This choice might seem counter-intuitive at a first glance considering that the information about the tactics underlying the processes is confidential. However, the incorporation of domain knowledge into the interaction design only requires the participation of at least one person having this knowledge. This person does not need to be a visualization designer but can also be from the application domain.

Visualization and interaction techniques can be openly discussed with practitioners. If potential designs are discussed and analyzed by experienced application experts, emphasis on certain elements of the design can be derived from their valuation of certain alternatives. An open discussion allowing domain experts to exchange opinions integrates their experience as implicit knowledge into the design process. The role of the designers of a visual interactive system is to propose solution alternatives and to tailor these solutions to the application

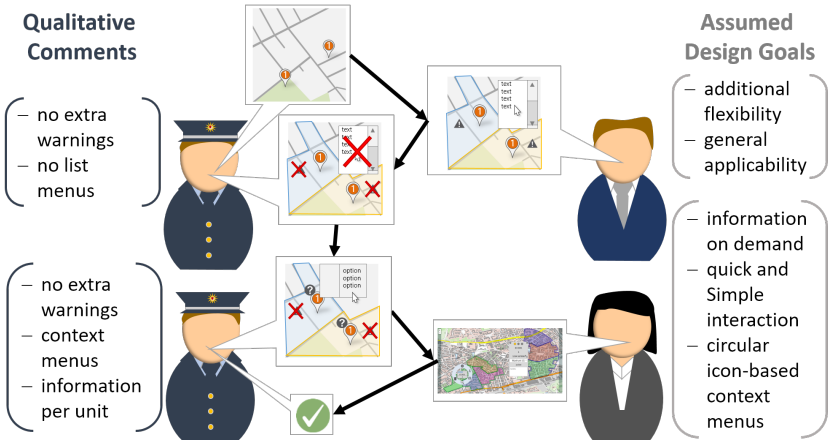


Figure 4.4: *Illustration of the process-centric dialog for the derivation of design goals from implicit information. Based on an initial concept for the interaction developed by experienced practitioners (left), designers (right) propose changes to the visualization and interaction. The practitioners discuss these propositions. Focusing on the interaction process, the practitioners do not need to reveal background information about the considerations and tactics underlying processes and tasks. Implicit information like a focus on certain features indicate importance. Qualitative comments, especially "liking" or "disliking" ideas, reveal optimization criteria. Interpreting this information, the designers derive design goals and refine their proposition-for the solution.*

experts' qualitative remarks.

Discussing interaction scenarios for different aspects of situation management, the police can comment on positive and negative aspects of different propositions. These discussions can be understood as an exploration of the design space, attempting to optimize a number of design aspects. The principle is illustrated in Figure 4.4. The discussion starts with an example workflow defined by an experienced practitioner, ideally a potential end user. This draft also defines the functional requirements related to policing. Care has to be taken to focus the discussion on individual tasks to prevent tactics from being communicated accidentally. For example, rather than discussing the process "manhunt", a possible topic for the discussion could be "dispatch units into sectors". The software designers then propose different ideas for the refinement of the initial draft under different aspects. These propositions combine the explanations of ideas with sketches of potential information displays, user interfaces, and interachtion techniques. The police comments on these propositions based on their experience. Rather than actively thinking about the potential workflow being applied to their tasks, they just comment on the quality. The visualization designers note observations, paying special attention to the implicit valuations of the proposed ideas. Formulations like "for me, this needs too many mouse clicks" (overly complex), "that would mean I could act much faster" (speed is an issue), or "I actually use to place these widgets next to each other" (these aspects are related) translate to optimization criteria and therefore to design objectives. Ideally, the ideas proposed for the solution are directly applied to

iterate the discussion towards a potentially optimal solution. This solution is implemented and the resulting prototype is the basis for a discussion of further improvements.

4.2.5 Process-Orientation in the V-model XT

The V-model XT is the standard model for software development in projects in the German public sector [80]. It is less focused on the tools applied for the actual development process than on the result to be obtained. The benefit of this approach is an increased flexibility, allowing to tailor the design process towards closer involvement of stakeholders or practitioners and to incorporate incremental and agile development techniques. When domain information is sparse, misconceptions and changing requirements are likely to occur. Proper implementation of the V-model XT allows to adapt to such events.

Due to these properties, the model was chosen as the initial project management paradigm for the project with the German police. The adaptation of the V-model XT is depicted on top of Figure 4.5. The idea is to loop the implementation step until the solution reaches sufficient quality. To assess this quality, practitioner interviews would be conducted to extract the implicit information about design aspects of the visual interactive system to be developed.

However, the V-model XT in its essence is still oriented towards functionality. It should be noted hat in the beginning of the project, the user-interviews were intended to identify functional requirements. As a result, the project management approach had been tailored towards user interaction but the developed solution failed to fulfill demands of the processes executed by the user. Only later, the quality of the solution could be improved, due to a change of the style of the interviews were conducted. In particular, the focus of the discussion was shifted from asking what kind of interaction the practitioners needed to hypothetical discussions of potential benefits and problems resulting from different means of interaction. Later, the use of sketches and mockups was added to improve the discussion of the interaction workflow. This gradual improvement of the interviewing technique eventually resulted in the open discussion setup described above.

4.2.5.1 Application: Sector-Based Manhunts

Sector-Based manhunts apply space partitions to distribute the units taking part in a manhunt over the search area. Sectors are available on maps accessible to a dispatcher. When initiating a sector-based manhunt, the dispatcher picks the sectors from the map and assigns each sector a number of available units, typically patrol cars. The task is to use visualization to communicate the assignment more easily, displaying the sectors on a map, and to design a proper interaction system that allows the dynamic definition of sectors depending on the situation. This system has to be flexible enough to cover different movement directions and speeds of the fugitive but still be intuitive to use for the dispatcher to allow for an efficient – and quick – assignment of sectors.

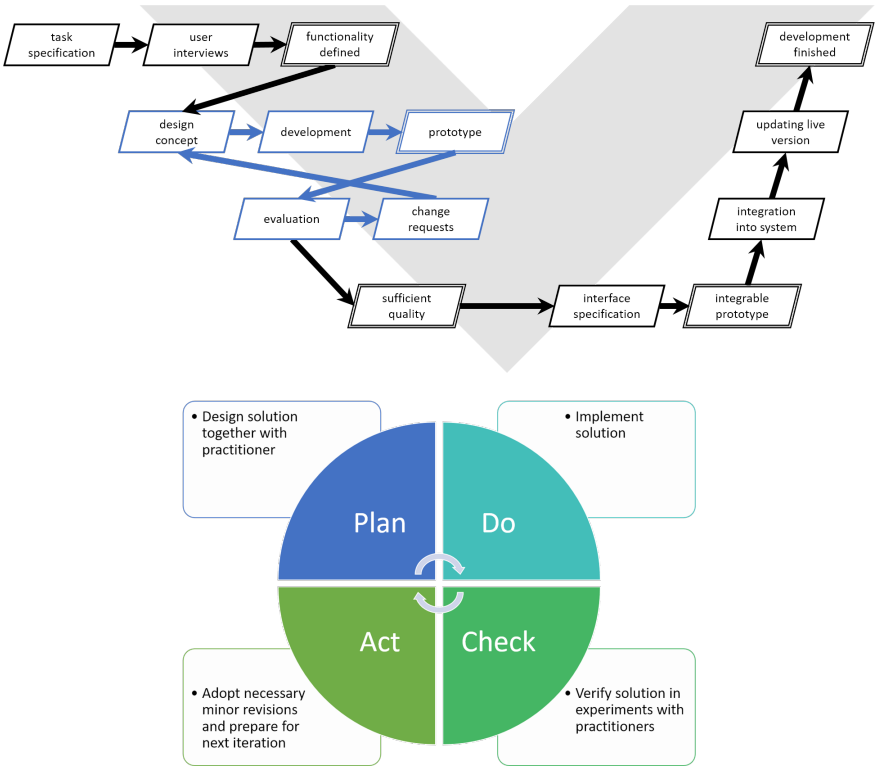


Figure 4.5: Top: *The implementation of the V-model XT applied in this work is tailored towards close collaboration with practitioners. Boxes with double-line boundaries indicate development milestones. The edges marked in blue on the left form a loop modeling the iterative design approach followed in the implementation of this model. This segment also integrates the process-centric approach deriving design goals from implicit information obtained from dialogues with practitioners.* Bottom: *PDCA for visualization applications. Each iteration starts with the discussions with practitioners, preferably potential users of the software to be developed. This discussion implements the focus on the interaction process and the derivation of design goals from implicit information (Plan). A prototype is developed based on the derived design goals (Do). This prototype is evaluated in another discussion round together with practitioners (C). If corrections are necessary, the system is adapted accordingly (Act).*

4.2.5.2 Development History

One major problem in designing the software was that the actual tactics and processes going on during the selection of sectors could not be revealed to the designers. Not only are those processes secret but also was the dynamic definition of sectors a novel tweak to the technique so the application partners could not tell immediately what would be the functionality of a proper interaction technique. The expectation was that a design towards maximum flexibility would allow for adjustment at a later point, adding any missing functionality. This approach was derived from an early discussion between project stakeholders who exchanged their opinions about different application scenarios and their respective requirements to the software's functionality. As it turned out, the assumption that the provided flexibility was needed was a crucial error. Without any domain knowledge, the user interface had been over-engineered offering settings for things of only minor importance for the definition of sectors, such as the means of transportation used by the fugitive. Of course, they had to be considered in the dispatcher's decision but it would have been enough to just size the area accordingly. This excessive detail in controlling the initiation of a manhunt caused an experienced practitioner to comment that the interaction would be "too much clicking for me". Simplicity and speed were the key factors identified in the further discussion, rather than flexibility and general applicability as it had been derived from the first discussion.

As a consequence of the false design decisions derived from the stakeholder discussion, the designers asked to get in contact directly with practitioners who would be potential users for the system to be developed. Unfortunately, in the initial attempts to do so, the designers unintentionally asked for confidential information, which of course could not be made available for them. The workaround was to reduce the interview to a description of potential workflows for the sector definition which a police officer could comment on as a practitioner. This way, experienced dispatchers could apply their knowledge without having to reveal it. Once a suitable workflow had been defined, it was further simplified and then implemented. The final workflow for the definition of sectors is outlined in Figure 4.6. Reporting is cut to a minimum and can be performed after the actual unit dispatch step. The interaction aims at a quick definition of sectors to which available units can be deployed. The new interaction scheme allows to define a set of sectors tailored to the situation within seconds, covering the consideration of the position of the crime scene, the time since the incident occurred, the means of transportation used by the fugitive (if any), and the movement direction (if known).

4.2.6 PDCA for the Design of Visualization Applications

The visual interactive system designed for sector-based manhunts did not only simplify the management of situations. Its final implementation showed potential to optimize the whole process by mitigating some limitations inherent to the original process. This motivated a more holistic approach to the development of further software modules. To this end, it was decided to lean from the Con-

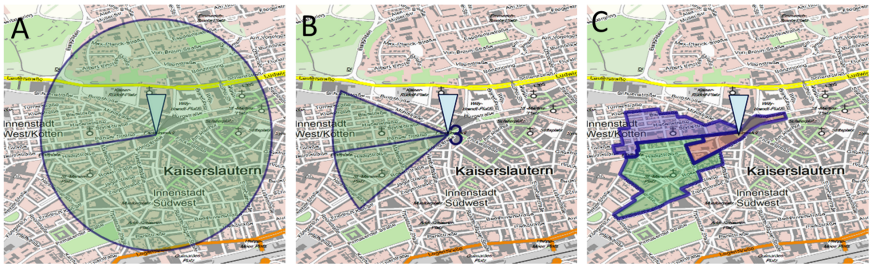


Figure 4.6: *Interaction for the definition of sectors for the sector-based approach to manhunts in the vicinity of a crime scene. The dispatcher determines a center, usually the crime scene, and a circle determining the search radius (A). If it is known, the fugitive's direction of movement can also be specified and the circle can be reduced to a cone for a more fine-grained positioning of sectors (B). It can also be turned into a ring segment if the dispatcher wishes to exclude an area in the direct vicinity of the crime scene. The sectors are then computed and mapped to the street network (C). With only a few mouse clicks, the dispatcher can tailor he manhunt specifically to the situation within seconds.*

tinual Improvement Process, a concept well-known in industry. This technique considers all aspects of the process, not only the software. However, since the project aimed for an IT-solution, the discussion focuses only on the aspects of interacting with the system. Other than before, the designers would not only attempt to optimize the interaction towards the needs of currently implemented procedures. Instead, they would also discuss potential optimizations of the dispatch process itself to optimize the workflows leveraging the full potential of the developed visual interactive system. To this end, the "Plan, Do, Check, Act"-paradigm (PDCA) was adapted, an iterative method for managing process optimization projects well known in industry [38].

The traditional PDCA-model aims to constantly improve industrial processes by iterating a four-phase loop: *Plan*, *Do*, *Check*, *Act*. In the *Plan*-phase, the process is analyzed and formalized, the parameters steering the process are identified, and their influence on the process is determined. Once these factors are known, problems or potential improvements are identified and a concept is designed how an improvement can be achieved. This improvement is implemented in a small-scale experimental environment in the *Do*-phase and verified by experimental evaluation during the *Check*-phase. Necessary adjustments to the model are made during the *Act*-phase, where it is also decided whether the process needs another iteration through the improvement loop.

Installing PDCA as a short-cycled iterative workflow, rapid prototyping was combined with mockups of several features. The advantage is an early available demonstrator whose design is driven by the application expert and that for each iteration cycle is refined in two steps: a functional step driven by the application domain and a technical refinement driven by the designers' expertise in visualization and user interface design. The identified refinements are passed to the next iteration. After the last iteration of a four-step loop, the results are presented to a wide audience to obtain additional comments and to advertise

the software in order to establish general agreement among the police. The aim was to approach a finalizing standardization step to provide the basis to hand the developed demonstrator over to the police as a basis for the public request for proposals for the implementation into their software systems. The bottom half of Figure 4.5 illustrates the implemented adaptation of PDCA formalizing the implemented development-workflow.

The cooperation between the designers and practitioners is closest during the Plan- and Check-phases. The Plan-phase serves to identify processes, required visualization features, and proper interaction techniques. The software meeting these requirements is developed during the Do-phase. The Check-phase involves the practitioners to assess whether the implemented visualization and interaction techniques suit the practitioners' needs or adaptations have to be made. Like in the standard PDCA model, these adaptations are made in the Act-Phase. For each module to be developed, the loop is iterated until a direct contact from the application domain – ideally an experienced user – agrees that the solution is sufficient.

4.2.6.1 Application: Ring-Based Manhunts

The ring definition is available to the dispatcher on a map, along with a set of predefined neuralgic spots. The task is to design an interaction system allowing to tailor the ring to the situation that triggers the manhunt, especially if a fugitive's movement direction is known.

4.2.6.2 Development History

During the development of this module, the designers were able to draw from the experiences made in the design of the interaction scheme for the sector-based approach. Unfortunately, the problems also carried over: Since the exact procedure to be executed is confidential information, it was hard to identify a design goal. Additionally, the dynamic and interactive definition was new, even to the dispatcher. The dynamics allow additional tactical considerations, especially concerning the combination of ring- and sector-based methods. Of course, these considerations could not be the topic of a discussion between practitioners and visualization experts. Yet, they somehow had to be included into the design of the developed solution.

Applying PDCA, the dialog between an experienced dispatcher and the designers of the visualization was established early during the development of this module. In the Plan-phase, one of the first decisions made was to integrate the interaction for the circle definition into the existing interaction scheme for defining the sectors. This design decision was agreed upon because practitioners felt that the familiar workflow for the similar task would render the tool easier to handle while the two different actions were still well distinguishable. However, rings and sectors would be treated separately, even if this would require to execute the interaction workflow twice if both methods were to be applied simultaneously. This has the advantage that mixed forms are only implicit,

Figure 4.7: *Interaction for the ring-based approach to manhunts in the vicinity of a crime scene. The ring-based approach to manhunts aims to identify points on the map where due to the traffic flow it is likely to intercept a fugitive. The user interaction is integrated into the system for sector-based manhunts. Switching to the ring definition is triggered by sizing the inner radius equal to the outer radius (A, B). The ring on the map can be reduced to a segment to concentrate available forces in the movement direction of a fugitive criminal (C). The system automatically computes optimal interception points (D).*

simplifying the log files for reporting. In the discussions, practitioners explicitly expressed that mixed forms would likely be too complicated to define in short time and repeatedly remarked that simple actions would be easier to include into reports, especially as part of the live-report that has to be prepared during the actual course of events. The workflow for initiating a ring-based manhunt is illustrated in Figure 4.7. Basically, the dispatcher uses the resizing feature covering the consideration of the time since the occurrence of an incident to determine the ring's radius. The ring-shaped area for the sector definition is reduced to a circle – or circle segment if the fugitive's movement direction is known. According to the dispatcher's setting, a number of neuralgic spots is identified automatically by weighting all available neuralgic spots along the defined circle segment and picking the ones with the highest score.

This interaction is not yet the perfect solution. The Check-phase revealed that the dispatcher would need the means to readjust the selection of neuralgic spots in case the heuristic provided suboptimal results. This requirement was derived from the observations that practitioners participating in the discussion asked many "What if"-questions, for example "What if I do not have sufficiently many units in range?" or "What if I suspect the fugitive to avoid large street crossings?". Still, without access to the actual definition of neuralgic spots, it will be hard to optimize the automatic procedure. To allow the police to keep this exact definition secret, the solution agreed upon is that they will eventually be provided with the means to define their own weight functions for the heuristics. The Act-phase of this module thus consisted of the addition of an interaction step allowing to either accept the proposed spots or to move them to other available spots by some sort of drag and drop interaction. Since there can be a large number of candidate positions along a circle segment, the exact visualization and interaction for this step was chosen to be the subject of an additional iteration of the PDCA cycle, once the initial solution is implemented as a basis for further discussion.

4.2.7 Comments on Design Study Methodolgy

The paper of Sedlmair et al. on design study methodology is one of the few works summarizing the observations of several experienced scientists collected over a number of projects, including success but also failures [128]. They identify 32 pitfalls posing a threat to a visualization project's success. The circumstances forced us to find solutions to some of these problems that might also be beneficial for other projects.

Confidential domain information is an instance of not being provided with data or getting access to domain information, a situation that Sedlmair et al. recommend "should be considered as a red flag for design studies". The results reported here contradict such a strict judgement – at least for the cases where data actually exists but cannot be made available to the designers due to access restrictions. If practitioners participate in the discussion of potential solutions with respect to their expected performance, their experience is considered in the design.

Letting practitioners discuss solutions rather than problems is also considered a pitfall by Sedlmair et al. The results reported above suggest that this can be partly allevieated by careful moderation. The designs discussed by the practitioners were proposed by designers. While practitioners were not allowed to explicitly reveal the problems with the existing solution, they could openly talk about problems with the proposed designs.

A third pitfall suggested by Sedlmair et al. is the relevance of an engineering project to a researcher. Technically, the project's interaction component is an engineering problem. Viewing the added value to science as a return on invest by solving a project's central research question, an engineering project will likely not be a profitable endeavour. Yet, this kind of project can indeed have an added value for science. First, reports about such projects provide the community with evidence on whether the theoretical design tools actually work. Such reports are needed to verify the success of design methodology in a number of projects. This case study contributes the observation that embedding iterative, prototype driven development implementing aspects of participatory design into classical project management frameworks can be successful even if access to domain information is limited. Second, the methodology to solve visualization problems is a research question on its own. Independent of a solution to a research question, engineering problems can be considered a sandbox to test new design methodologies or to evaluate the result against solutions found using established methodology. Considering that domain information in the reported project was confidential, the developed prototype could not be compared to an approach where practitioners could openly communicate their problems. Still, a method for the development of an engineering solution for an interaction problem is contributed, even if the engineering question itself remains open due to sparse availability of domain information.

4.2.8 Comments on Confidential Domain Information

In the development of the module for sector-based manhunts, the derivation of implicit design goals yielded good results well accepted among practitioners. However, care has to be taken to find the right discussion partners. Stakeholders who are not potential end users of the system may focus on components less relevant for the actual workflow than they suspect. Experienced practitioners will potentially focus on other aspects. Ideally, an experienced practitioner who is also a potential user of the system to be developed should be part of every discussion round.

If the tasks and processes are confidential, a holistic perspective designing workflows without consideration of existing techniques is unavoidable. The PDCA-based approach formalizes such a holistic perspective. An interesting feature of this approach is that it scales well with different complexity. It can be applied to the design of individual modules as well as the design of larger components consisting of multiple modules. The scope depends on the expertise of the practitioners participating in the discussion. However, this approach also requires a careful choice of those practitioners. An important mechanism is PDCA's check-phase as its application on the module level allows an early correction of misconceptions.

Discussing the design with an experienced practitioner, domain knowledge can still be considered during the design process but is combined with the methodological knowledge of visualization and interaction designers to find an optimal solution. Focusing on implicit information being communicated by practitioners, the actual tasks and workflows do not need to be communicated. For example, the workflow for defining sectors for manhunts is only one alternative of a set of possible interactions and thus can be openly discussed. In contrast, the workflows and processes to define the number of sectors and dispatch of available units are confidential. This tactical information is not required to be communicated explicitly if practitioners can instead comment on how they would like the interaction to be designed.

4.2.9 Comments on General Sparse Domain Information

Some observations applying to the case of confidential domain information are transferable to other projects suffering from sparse availability of domain information. For sparsity of domain information due to access restrictions, the observations should apply almost seamlessly. The case of confidential tactics and considerations behind workflows is an example of this category.

If access to domain information is not restricted, the necessary information can likely be obtained from domain experts. However, sometimes this information is not explicitly available. One example is the implicit knowledge practitioners obtain about processes as part of their experience. Based on their experience, practitioners tend to interact with systems in different ways. Tailoring interaction to the users' workflows means to optimize it based on this experience. In domains where information is not confidential, observations in

user studies or experiments with simulations can help to understand the tasks and to extract the implicit knowledge. On the other hand, education is based on explanation. An adaptation of the informal discussion setup for confidential information would let practitioners exchange and discuss their experiences.

If the domain is too complex for a visualization's designer to obtain a thorough understanding, the discussions can be applied with a focus on aspects the designer might not understand properly. However, if the domain experts know what is important to them, they can just tell the designer directly. If the emphasis is unclear, for example in advanced scientific experiments, the interviews are not transferable since in these cases it might not even be clear whether the domain experts' focus is the correct one to apply.

If domain information important to the design of a visual interactive system is not clearly defined, it is also unlikely to be retrievable as implicit information. In such a setup, emphasis on different aspects of the interaction workflow is likely to change frequently. The interviews might, however, hint the direction of such a change. Embedded into PDCA, they can potentially be applied to navigate the design space from a holistic perspective on the whole workflow.

4.2.10 Conclusion: Qualitative Considerations Support the Design Process

The above case study reports observations on the performance of a technique to derive design goals for visual interactive systems in projects that suffer from sparse availability of domain information. A process-centered perspective involves stakeholders and potential users in the discussion about alternative interaction workflows. In these discussions, the participants emphasize certain aspects of the visualization and interaction. Design goals can be derived from the implicit information provided by this emphasis and the participants' reactions to different propositions made by the visualization and interaction designers. The approach is embedded into two different project management paradigms being applied to the development of modules of a system for the management of manhunts in a policing application. The discussions enable the police to communicate the emphasis on different aspects of the interaction design without the need to reveal confidential information about tactics or chains of actions. Deriving implicitly defined design goals enables the development of high-quality visualization and interaction solutions tailored to the practitioner's domain-specific requirements even if available domain information is sparse.

In the beginning of the project, the design aimed towards a maximum of flexibility. Yet, the original data driven approach aiming for providing a maximum of simultaneously conveyed information and offered functionality resulted in an overly complex design. Redesigning the system following the principle of minimal graphical overhead resulted in a simplified information display and a much more focused interacion. Interaction has been reduced to a necessary minimum of tools. Its design is primarily determined by the purpose of the interaction gesture rather than by the functionality to be offered. For example, the interaction sequences for subdividing the area into sectors and for defining neuralgic points on an approximate ring around a crime scene are triggered by the same

button indicating the purpose to start a manhunt. The motivation to start a ring-based manhunt is indicated by setting the inner radius to the same size as the outer radius of the area to be searched, clearly showing that the purpose is not to highlight an area on the map but rather a ring. Designing interaction sequences by their purpose rather than only by the result they are meant to achieve also makes the workflow more intutive for the user as it implements the principe of design for reasoning. Design for reasoning was also the central idea behind the installation of PDCA as a development workflow. However, due to the limited access to domain information, this principle was particularly hard to follow. Describing the project setup in terms of the inside-outside principle, the outside knowledge was almost completely left with the practitioners and the designers only had access to a minimum of necessary concepts provided to them as some kind of interface to adapt the visualization and interaction workflow to. Yet, the installed close cooperation enabled the practitioners to guide the designers towards providing a system whose conveyed messages optimally fit the available outside knowledge without having to reveal it.

In conclusion, a purely data-driven design approach did not solve the challenges posed by the project. In the beginning, the display was not of sufficient quality and the interaction was not efficient enough. Only the explicit consideration of qualitative aspects such as designing efficient purpose-driven workflows turned the tables. The qualitative principles of visual information encodings inspired a development workflow that eventually led the project to success despite the limited access to valuable domain information. In this poject, the qualitative considerations did not only contribute to the design itself but also to the implementation of an adequate development process. Qualitative considerations hence affect not only the actual design but also the general collaboration with practitioners.

4.2.11 Addendum: Opinions of a Collaborating Police Officer

The following interview was conducted with a police officer participating in the project this case study is based on. The questions were asked in an informal setting and the answers were noted by the interviewer who later translated the questions and answers from German to English. The statements are not commented and no further explanation is added.

1. **Q:** Please describe your role in the project

   A: User, consultant for the police's requirements to the management system for manhunts.

2. **Q:** Please describe your professional background, including your role in the organization and its relation to the project.

   A: Role: Polizeiführer vom Dienst (PvD), within the operations center, responsible for the first response to every kind of crime, ranging from small to major incidents. Relation to the project: Experienced user of the software currently in use, profound experience in managing different kinds of situations. Multiple ideas for the improvement of the state of the art application.

3. **Q:** Where did/does communication...

   a ...function well in the beginning?

   A: Quickly agreed on principal direction for the solution. Issues with different domain-specific language and expectations have been solved quickly.

   b ...need to be improved in the beginning?

   A: In the beginning, it was unclear which information could be passed to the designers. Difficulties were encountered in the attempt to articulate requirements without revealing confidential information. Finding simple examples for explanations while holding back confidential information could be challenging.

   c ...function well as of now?

   A: Over the years, a common ground with the designers has been established. Communication thus became much easier. Close contact to the designers improved communication and mutual understanding.

   d ...need to be improved as of now?

   A: Sometimes, there are still misunderstandings due to lacking knowledge of the respective other domain on both sides.

4. **Q:** If applying: Please provide reasons for:

   a 3.b)

   A: The police was not aware of the technology's full potential. It was not clear which ideas could be implemented. The requirements to the design were also not completely clear.

   b 3.d)

   A: Because the necessity to reveal confidential information has been circumvented, the designers do not know this information. On the other hand, the application side did not acquire additional competence in the design of visualization and interaction systems.

5. **Q:** On a scale from 1 (unimportant) via 3 (neutral) to 5 (very important): How important do you consider secrecy in the context of this project?

   A: 5

6. **Q:** Please comment: How difficult do you find it to specify functional requirements to the interaction with the software (workflows, processes, chains of actions, chains of events, reaction to events, ...) without the explicit consideration of (confidential) tactical aspects?

   A: Not very difficult. Examples can be taken from television. Still, explanations need to be fragmentary to prevent revealing tactics and other information that is indeed confidential and not known from movies or other popular culture.

7. **Q:** Please comment: How much experience do you have with the process-centric, discussion-based approach to the derivation of design goals? How many discussions did you attend?

   A: Almost every discussion since May 2015. More than 20.

8. Please comment: What are, from your personal perspective, the (dis-)advantages of the discussions' focus on the interaction process? Please focus your consideration primarily on secrecy of confidential information and quality of achieved results.

   A: Getting to know each other better, the quality of results improves and they tend to be tailored better to the police's requirements. Discussions are more time-consuming than providing lists with requirements. However, they allow to immediately resolve potential questions or misunderstandings. Discussions also allow both sides to identify the potential and limitations from the perspectives of policing and computer science.
   In the discussions, the practitioners' focus on policing aspects sometimes prevents the explicit discussion of aspects considered obvious. The developers' attention in following the discussion captures this information and considers it for the solution's development.

9. **Q:** In retrospect: What are the advances you observed in the collaboration with designers during the project, especially where communication is concerned? Can you specify events that could be considered a turning point in the project's history? If so, can you specify the changes?

   A: Two turning points:
   May 2015: Presentation of a first mockup. Misunderstandings have been identified and resolved. As a consequence, close contact between designers and practitioners has been established. This contact primarily served the purpose of providing feedback to the designers. From here on, development proceeded in smaller steps with frequent iterations.
   June 2016: Prototype. From now on, the prototype was demonstrated to a wider range of practitioners and domain experts. This broad audience introduced new ideas and impulses for further development.

10. **Q:** On a scale from 1 (unimportant) via 3 (neutral) to 5 (very important): How important do you consider the close collaboration between the police and experts for IT solutions in general for police IT applications?

    A: 5

4.3 Case Study: Surface Strip Geometry Design

Continuum mechanics studies the kinematics of materials based on continuous mass models. Applications of this theory are often concerned with rod- or shell-like elastic structures subject to bending and twisting. While models for these shapes are well established, the special case of thin elastic strips remains a challenge since neither the models for rods nor the models for plates and shells directly apply to these surfaces. Being defined as the envelope of a set of straight lines smoothly transitioning along a space curve, ruled surfaces directly reflect the shape of thin elastic strips. Among these, the developable surfaces are of special interest since many applications are concerned with the bending of planar sheets of material, for example plywood or metal. Unfortunately, in most surface models applied in this context, developability cannot be guaranteed trivially. A rather rigorous but still common approach to this is to explicitly model the surface as the isometric deformation of a planar reference configuration. For continuum mechanics in general and engineering purposes especially, it is more convenient to work with a model for arbitrary ruled surfaces that covers isometric deformation and developability as special cases with simple constraints.

Models for thin and narrow inextensible elastic surface strips answer questions like whether the bending behavior of a ribbon cable will cause friction by contact with surfaces in the cable's way if the object the cable is plugged into is moved. The description of such properties is based on the surface strip's local behavior. Following the principle of design for reasoning it is thus important to provide a local formulation of the geometry that allows to infer important local properties of the surface and its deformation directly from a set of easily accessible features with clearly defined semantics.

Motivated by this kind of local perspective a new method for modeling ruled surfaces is introduced that is more tailored to engineering applications than comparable techniques, especially where reasoning is based on the local shape of the surface. The surface is described by two coupled moving frames of reference that are chosen deliberately such that their transition equations are governed by a minimal and complete system of invariants for ruled surfaces. This system of invariants has the following properties particularly appealing to modeling and design applications:

1. It is completely defined in the arc length of the centerline as the single parameter.
2. It supports arbitrary shapes of ruled surfaces.
3. It guarantees developability by a trivial constraint.
4. It guarantees isometry of deformations by a trivial constraint.
5. It provides a concise formulation of the bending energy for thin inextensible elastic ribbons of arbitrary but finite and noninfinitesimal width.

The method directly reproduces results from the literature. Perhaps the most remarkable contribution of this work is a simple functional for the bending

energy of arbitrarily shaped ruled surfaces based on an analytically exact yet easily discretizable surface definition.

The remainder of this section is structured as follows: After a brief discussion of related work, the system of coupled frames is introduced and a bending energy functional for arbitrary ruled surfaces is developed. The discussion then reflects on the model's descriptive power and limitations focusing on the physical interpretation of the model and the constraints imposed by actual real materials.

4.3.1 State of the Art

In recent years, new applications like the simulation of molecules in (bio-)chemistry or ribbon-shaped nanostructures in material science increased the interest in mathematical models for the mechanics of thin, inextensible elastic surface strips. Since for these applications one is typically interested in the effect of applied forces and equilibrium states, a substantial part of this recent work is originated in continuum mechanics. For example, certain polymers can be described by a generalization of the Frenet-Serret equations for space curves to the description of surface strips [114]. Being the surfaces that can be obtained by bending a flat piece of some material, the developable surfaces are of special interest to continuum mechanics. These surfaces are also relevant in contemporary architecture [54,86]. Nondevelopable surfaces, on the other hand, are of their own interest due to their higher stability [91]. Closing the loop back to material science, a recently proposed model for the interatomic bonds in carbon structures applies nondevelopable surfaces and their elastic behavior to describe the elastic properties of graphene and diamond [13].

For the description of the mechanical properties of surface strips, rod- or shell-based models are commonly applied. Yet, the approaches differ in the definition of the underlying geometry. One can roughly classify these attempts into three groups. The first one applies implicit surfaces, usually expressed in terms of B-Splines or NURBS [25,59,113]. More direct approaches try to exploit the vast theory of differential geometry [46]. The third group applies theory dating back to the 1920's, combining line geometry with Plücker coordinates or similar formalisms to describe ruled surfaces, [18].

In continuum mechanics, it is common to model ruled surfaces by constructions similar to a material frame directly following the structure of the thin sheet of material along a dedicated centerline and the surface normal along this line [42]. While these models are capable of describing arbitrary ruled surfaces, properties like developability have to be enforced by nontrivial constraints. Constructive approaches attempt to guarantee this property by explicitly modeling the desired shape as an isometric deformation of a planar reference geometry. Any ruled surface obtained this way is developable but the flat reference configuration for a given deformed surface is not arbitrary. This point is emphasized here as it appears that it is an actually quite often encountered misconception [30].

Modeling ruled surfaces by surface invariants provides simpler constraints for properties like developability or freedom of twist. For example, line geometry

models ruled surfaces by transporting the generators along the line of striction [18, 68, 106, 111]. While this surface definition is intuitive, strictly binding the surface to the line of striction reduces the degrees of freedom for modeling significantly. The reason is that while for a given line of striction the surface is not unique, there is a unique line of striction for each ruled surface. Hence, changes to the surface either require the identification of a new line of striction or impose constraints on the correct deformation with respect to the line of striction's bending. Introducing the striction as a design parameter, arbitrary centerlines can be defined. This approach is known as the "natural parameterization" of ruled surfaces [84]. However, there are certain degenerate cases which cannot be modeled in this framework, including certain developable surfaces or segments of these, especially cylinders and conoids.

From the perspective of continuum mechanics, it is especially important to know the deformation energy applied to a piece of material. For a thin strip, this reduces to the bending energy. A common approach to guarantee the modeled shapes to be physically sound is to minimize the bending energy. An equation for the bending energy of an infinitesimally narrow ribbon has been introduced in 1930 [121]. It has later been applied to the bending energy of a Möbius strip [152]. For narrow strips, generalizations of these functionals to nondevelopable surfaces have been proposed [45]. The problem of computing the bending energy of an arbitrarily wide Möbius strip has only recently been solved [33, 42, 134]. However, the energy functionals proposed in this context again depend on the comparison of the investigated shape to a planar reference geometry that is not necessarily known in an actual application. For modeling and design purposes, it is much more convenient to be provided with a functional independent of the knowledge of a reference configuration. The bending energy does not involve any process quantity and thus is a function of state.

4.3.2 Ruled Surfaces

A ruled surface $S = L(t) + s \cdot E(t)$ is defined as a family of straight lines $E(t)$ transitioning smoothly along a space curve L, the *centerline*, such that L intersects every *generator* $E(t)$ exactly once. The tangent planes along a generator $E(t)$ rooted in some point $L(t)$ on the centerline are either constant along $E(t)$ or rotate smoothly around the generator such that for infinite distance s to the centerline, the surface normal covers all possible directions except the one of the shortest connection between $E(t)$ and its direct neighbors. This behavior distinguishes the *developable* from the *nondevelopable* ruled surfaces. For the remainder of this discussion, L is assumed to be parameterized by its arc length t and the angle $\sigma = \sphericalangle(\dot{L}(t), E(t))$ is called the *striction*.

4.3.3 Drall-Based Modeling

Consider a straight centerline L and fix the striction to $\sigma = \frac{\pi}{2}$, which means the generators are orthogonal to the centerline. Letting the generators rotate around L with unit speed and fixing a distance s to the centerline, the curve $C_s(t) = L(t) + s \cdot E(t)$ is a helix. The direction of the tangent $\dot{C}_s(t)$ to this

curve along $E(t)$ for fixed t varies with the distance s. For the angle $\varphi = \sphericalangle(\dot{C}_s(t), \dot{L}(t))$, one obtains:

$$\tan\varphi = -sd \tag{4.1}$$

for some constant d. Exploiting this proportionality and applying d to parameterize the screwing of the surface's tangent plane around the generators $E(t)$ in distance s relative to its orientation along the centerline, this screwing defines a curve $C_s(t)$. Due to the resemblance of the traces of the curves $C_s(t)$ to the rifling of a gun barrel and in appreciation of the pivotal work on the theory of ruled surfaces developed during the first half of the 20th century, d is from here on referred to as the *drall*. The discussion of the drall follows closely the explanation in Blaschke's Lectures on Differential Geometry [18]. Yet, other than in Blaschke's work, Cartesian rather than Plücker coordinates are applied. The description of the tangent plane's rotation around the generator requires knowledge about its angular velocity:

$$\delta = \frac{d}{ds}\varphi = \frac{d}{ds}\arctan -ds = \frac{-d}{1+s^2d^2} \tag{4.2}$$

To apply the drall for modeling the surface, it is defined by the angle φ at distance 1 to the centerline, which means that $d = -\tan\varphi|_{s=1}$. The angular velocity δ at each point along a generator is computed by integrating the angular acceleration

$$\delta' = \frac{d}{ds}\delta = \frac{2sd^3}{(1+s^2d^2)^2} \tag{4.3}$$

provided that an initial value for δ is known at some point. From (4.2), it directly follows that in the centerline $\delta_0 = -d$. Therefore, the surface can be modelled applying

$$\delta = \tan\varphi|_{s=1} \tag{4.4}$$

The importance of the drall for the model lies in its close connection to developability:

Theorem: Developability and Drall

A ruled surface is developable if and only if δ vanishes identically along the centerline.

Proof Towards a proof, it has to be shown that the drall – and therefore δ – is invariant under reparametrization. A change of drall necessarily causes a change of the surface metric. To see this, consider the trapezoid formed by two infinitesimally close neighboring generators, their direct connection in distance s to the centerline, and the infinitesimally short section of the centerline between the two generators. The drall rotates the tangent plane around one of the generators. Therefore, any change of drall induces a change of the direction of the connection between the two generators. Because the surface is smooth, at

least one of its generators also has to change its direction. Since neighboring generators do not intersect, this necessarily induces a change of the length of the direct connection between the generators at some distance s to the centerline. Therefore, a reparameterization either does not affect the drall or it changes the surface's metric. Since reparameterizations do not deform the surface, the drall has to remain constant under reparameterization. Because of the connection $\delta = -d_0$, for the drall d_0 at the centerline, the same applies to δ $\square$.

The proof contains another result:

Corollary: Deformations and Drall

A deformation is path-isometric if and only if it does not change δ at any point along the centerline.

Path-isometric deformations only preserve the arc lengths of curves on the surface. The positions of points on the surfaces given by the surface parameters s and t may change. This corollary is especially relevant for a more thorough discussion of deformations following below.

Swapping the roles of the centerline and the generators in the above construction yields a similar, drall-like, parameter for the rotation of the tangent surface around the centerline. Since the centerline is not straight, the construction is considerably harder. However, for modeling purposes it is desirable to have direct control over the shape of the centerline and therefore expect it can be expected to be known a priori. In this case, the angular velocity ω can be computed as the rate of change of the direction of the surface normal along the centerline. Since ω determines the twisting of the normal around the centerline, it is a measure for the surface's *geodesic torsion*. The invariance of ω under surface reparameterizations can be proven analogously to the invariane of δ. In the literature on the kinematics of elastic strips, strictly isometric deformations are sometimes referred to as "pure bending" since they also do not allow a change of the twist. In combination with the above corollary, one obtains:

Theorem: Parameter Invariance of Pure Bending

A transformation is a "pure bending transformation", meaning that it is strictly isometric, if and only if neither ω nor δ change in any point along the centerline.

4.3.4 The System of Coupled Frames

In this work, the angular velocities δ and ω are applied to model the ruled surface by a system of two moving frames of reference that share the surface normal N along the centerline as their common direction. Given the centerline L together with its tangent $\dot{L}$ and a generator E, one can locally compute N by taking the cross product $N = [\dot{L}, E]$. To complete the frames, two auxiliary directions are introduced: $K = [\dot{L}, N]$ and $T = [E, N]$. The frame $\{\dot{L}, N, K\}$

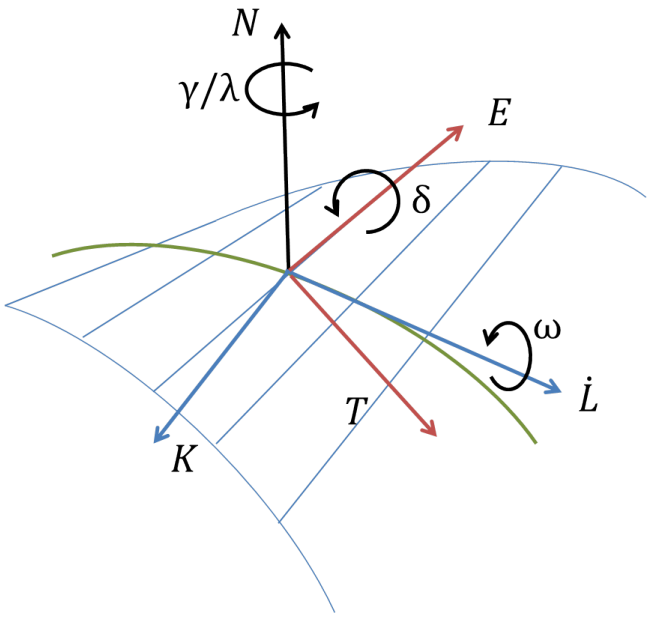


Figure 4.8: *The system of coupled frames. The centerline (green) is given by any surface curve that does not intersect any generator (grey) more than once. The $\{E, N, T\}$-frame (red) describes the generators' movement along the centerline's tangent and therefore determines the surface's geometry. The $\{\dot{L}, N, K\}$-frame (blue) capture the physical properties of an actual material. The two frames are coupled by sharing the one of the directions, the surface normal $N = [L, E]$, and both contribute to the surface's parameterization which is given in the centerline's arc length t and the distance s to the centerline at the respective generator $E(t)$. Their transition along the surface is described by the angular velocities ω, δ, γ, and λ, although the latter two can be expressed in terms of the formers' derivatives applying the striction, i.e. angle between $\dot{L}$ and E.*

is then applied to model the surface transformation along the centerline. Being a material frame, it describes the physical properties of an actual thin strip made of elastic material whereas the $\{E, N, T\}$ determines the geometry into which the surface is embedded. If the centerline is the line of striction, the $\{E, N, T\}$-frame is in some literature also referred to as the Sannia frame [103]. Note, however, that allowing the centerline to be an arbitrary surface curve that intersects each generator exactly once it is explicitly not required to be the line of striction. The system is illustrated in Figure 4.8. The distinction between a frame modeling the physical properties of some actual material and the frame defining the geometry the material is embedded in already indicates that the frames are not necessarily aligned. Indeed, the striction, the angle between $\dot{L}$ and E may change during deformation. However, the focus in this discussion is on the effects this induces to an actual physical surface strip. For the discussion here, it is enough to keep in mind that there is a difference between an actual physical surface and the geometry it is embedded in. The only remark to be made regarding the mechanical perspective is that this setup allows a geometrical treatment of the deformation problem and therefore to impose geometrical

constraints rather than mechanical ones on aspects important for modeling such as guaranteeing developability or the isometry of deformations – resulting in a treatment much easier than the optimization-based approach common in continuum mechanics.

Being coupled by the surface normal, the transition of the one frame directly affects the other. In what follows, this connection is used to derive the transformation of the one frame from the transformation of the respective other frame. Linking the frames by the surface normal and applying the striction angle $\sigma = \sphericalangle(\dot{L}, E)$, the following connections are obtained:

$$\cos\sigma = \langle E, \dot{L}\rangle = \langle K, T\rangle = \sin(\tfrac{\pi}{2} - \sigma) \tag{4.5}$$

$$\sin\sigma = \langle \dot{L}, T\rangle = -\langle K, E\rangle = \cos(\tfrac{\pi}{2} - \sigma) \tag{4.6}$$

$$\dot{L} = E\cos\sigma + T\sin\sigma = \langle E, \dot{L}\rangle E + \langle \dot{L}, T\rangle T \tag{4.7}$$

$$K = T\cos\sigma - E\sin\sigma = \langle E, \dot{L}\rangle T - \langle \dot{L}, T\rangle E \tag{4.8}$$

$$E = \dot{L}\cos\sigma - K\sin\sigma = \langle E, \dot{L}\rangle \dot{L} - \langle \dot{L}, T, \rangle K \tag{4.9}$$

$$T = K\cos\sigma + \dot{L}\sin\sigma = \langle E, \dot{L}\rangle K + \langle \dot{L}, T\rangle \dot{L} \tag{4.10}$$

The connections between the surface normal, K, and the centerline's principal normal and binormal $\mathcal{N}$ and $\mathcal{B}$ are useful for simplification. They are given by

$$N = \langle \mathcal{N}, N\rangle \mathcal{N} + \langle \mathcal{B}, N\rangle \mathcal{B}$$
$$K = \langle \mathcal{N}, N\rangle \mathcal{B} - \langle \mathcal{B}, N\rangle \mathcal{N}$$

With these equations, transformation equations in t-direction along the centerline and in s-direction along the generators can be derived. Figure 4.9 shows the principle behind the procedure for both directions. Because the considerations are almost identical, only the transformation in t-direction is discussed in detail and only a brief overview is provided for the s-direction.

Let $L(t)$ be a space curve parameterized by its arc length. In addition to the curvature κ and torsion τ of L, define four functions are defined in t:

- the angular velocity ω of N rotating around L
- the angular velocity δ of N rotating around E
- the rotation γ of E around N while moving along L
- the rotation λ of the geodesic parallels to L around N while moving along E

Moving along L, $\{\dot{L}, N, K\}$ is adjusted first. After that, the $\{E, N, T\}$-frame is rotated around the new centerline tangent $\dot{L}$ according to ω and around the surface normal N according to the value of γ (cf. Figure 4.9). If the surface normal does not rotate around the centerline in t, N and K transform like the centerline's principal normal $\mathcal{N}$ and binormal $\mathcal{B}$. The actual rotation may thus be decomposed into two steps, namely applying the transformation of the centerline's Frenet-Serret-frame to $\{\dot{L}, N, K\}$ and rotating both frames with angular velocity ω around the transformed centerline tangent $\bar{\dot{L}}$. Equations (4.7) and (4.8) are applied to express E, N, and T in terms of $\dot{L}$ and K and the

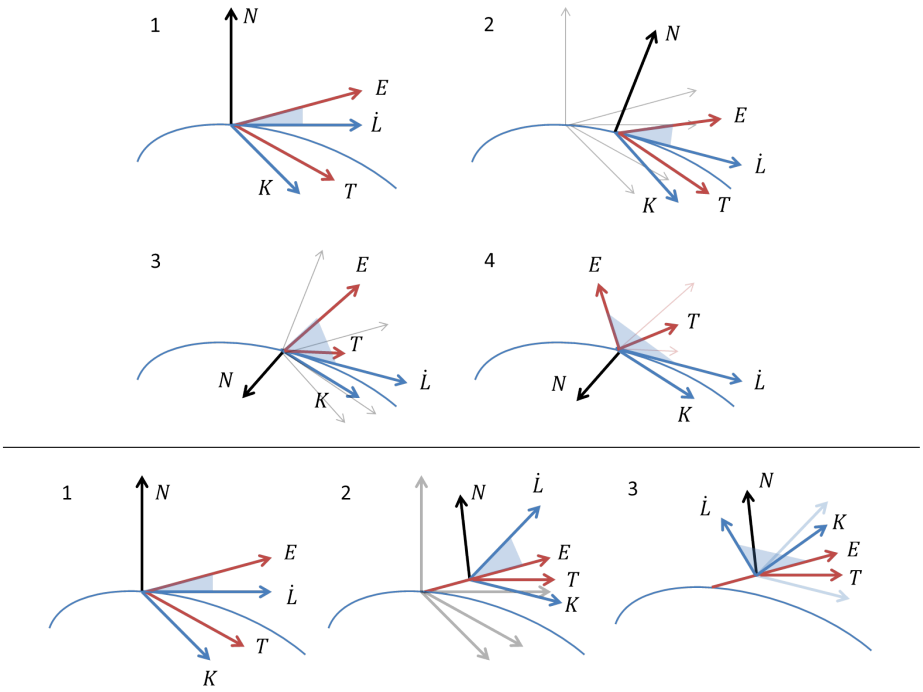


Figure 4.9: *Illustration of the frames' transformation along the centerline in t-direction (top), and along a generator in s-direction (bottom). The surface normal is colored in black, $\dot{L}$ and K in blue, and E and T in red. The triangle between $\dot{L}$ and E shows the orientation of the surface. Faded arrows denote the directions of the last step and are shown for comparison. From upper left to lower right for t: Initial configuration (1), transformation of $\{\dot{L}, N, T\}$ along with the centerline's Frenet frame (not shown) and adjustment of E and T (2), rotation of E, N, T, and K around $\dot{L}$ with velocity ω (3), and the rotation of E and T around N with velocity γ (4). From left to right for s: initial configuration (1), rotation of N and T around E with angular velocity δ and adjustment of $\dot{L}$ and K (2), and rotation of $\dot{L}$ and K around N with angular velocity λ (3).*

respective new values $\hat{\dot{L}}$ and $\hat{K}$ are inserted into the equation. Particularly, one computes

$$\begin{aligned}
\hat{\dot{L}} &= \overline{\dot{L}} = \dot{L} + \kappa \mathcal{N} dt \\
\hat{N} &= \overline{N} + K d\Omega \\
\hat{K} &= \overline{K} - \overline{N} d\Omega \\
\hat{E} &= \langle E, \dot{L} \rangle \hat{\dot{L}} - \langle \dot{L}, T \rangle \hat{K} \\
\hat{T} &= \langle E, \dot{L} \rangle \hat{K} + \langle \dot{L}, T \rangle \hat{\dot{L}}
\end{aligned}$$

where $d\Omega$ denotes the integral of the rotation around L with velocity ω along the infinitesimal distance dt.

In the second part of the transformation, the $\{E, N, T\}$-frame is rotated

around N with velocity λ, setting

$$\begin{aligned}\tilde{E} &= \hat{E} - \hat{T}d\Gamma \\ \tilde{N} &= \hat{N} \\ \tilde{T} &= \hat{T} + \hat{E}d\Gamma\end{aligned}$$

where $d\Gamma$ is defined analogously to $d\Omega$ before. This rotation does not affect $\dot{L}$ and K. After this transformation, the original vectors are subtracted from each term and both sides are divided by the infinitesimal interval dt. Letting dt approach zero, this yields the frame axes' derivatives in t-direction. Applying $\frac{d\Omega}{dt} = \omega$ and $\frac{d\Gamma}{dt} = \gamma$, one obtains:

$$\ddot{L} = \kappa\mathcal{N} \tag{4.11}$$

$$\dot{\mathcal{N}} = -\kappa\dot{L} + \tau\mathcal{B} \tag{4.12}$$

$$\dot{\mathcal{B}} = -\tau\mathcal{N} \tag{4.13}$$

$$\dot{N} = \tau K - \langle\mathcal{N}, N\rangle\kappa\dot{L} + \omega K \tag{4.14}$$

$$\dot{K} = -\tau N + \langle\mathcal{B}, N\rangle\kappa\dot{L} - \omega N \tag{4.15}$$

$$\begin{aligned}\dot{E} = &-\gamma T + \langle E, \dot{L}\rangle\kappa\mathcal{N} + \langle\dot{L}, T\rangle N(\tau + \omega) \\ &- \langle\mathcal{B}, N\rangle\langle\dot{L}, T\rangle\kappa\dot{L}\end{aligned} \tag{4.16}$$

$$\begin{aligned}\dot{T} = &\gamma E + \langle\dot{L}, T\rangle\kappa\mathcal{N} - \langle E, \dot{L}\rangle N(\tau + \omega) \\ &+ \langle\mathcal{B}, N\rangle\langle E, \dot{L}\rangle\kappa\dot{L}\end{aligned} \tag{4.17}$$

For the generators, the transformation equations can be derived analogously. First, the $\{E, N, T\}$-frame is rotated around E in order to obtain the transformed $\{\dot{L}, N, K\}$-frame by applying equations (4.9) and (4.10) inserting the new directions $\{\hat{E}, \hat{N}, \hat{T}\}$. After that, the $\{\dot{L}, N, K\}$-frame is rotated around N with velocity λ. Similar to before, these two steps introduce angles $d\Delta$ and $d\Lambda$ as the integrals of the angular velocities δ and λ over a distance s along the generator. Again, the derivatives are computed, applying $\frac{d\Delta}{dt} = \delta$ and $\frac{d\Lambda}{dt} = \lambda$ and the terms with the remaining differentials that vanish for $dt \to 0$ are canceled out. The resulting equations for the transformation in generator direction are:

$$\dot{L}' = -\langle\dot{L}, T\rangle\delta N - \lambda K \tag{4.18}$$

$$N' = \delta T \tag{4.19}$$

$$K' = -\langle E, \dot{L}\rangle\delta N + \lambda\dot{L} \tag{4.20}$$

$$E' = 0 \tag{4.21}$$

$$T' = -\delta N \tag{4.22}$$

4.3.5 The Fundamental Forms

Having introduced the coupled frames along the centerline, the discussion now turns to the description of the transverse characteristics of the surface. Given two generators $E(t_0)$ and $E(t_1)$, the geodesic parallel to the centerline in distance s along the generators will become longer for larger values of s. Since the frame

axes should be normalized, this behavior needs to be captured as part of the metric.

Let $X_s(t_0)$ and $X_s(t_1)$ be two points on the surface in equal distance s along the generators $E(t_0)$ and $E(t_1)$ to the centerline L at parameter values $t_0 < t_1$. The transition from $X_s(t_0)$ to $X_s(t_1)$ along the connecting longitudinal line with constant s is then given as:

$$\begin{aligned} X_s(t_1) - X_s(t_0) =& (L_0(t_1) + s \cdot E(t_1)) \\ & - (L_0(t_0) + s \cdot E(t_0)) \end{aligned} \tag{4.23}$$

where the index denotes the distance of a point to the centerline. This equation can be expressed entirely in terms of the derivatives of $L_0(t_0)$ and $E(t_0)$ in t-direction. To this end, one sets $t_1 = t_0 + dt$ for some infinitesimal interval dt and for each variable at t_1, one applies the ansatz $\circ(t_1) = \circ(t_0) + \dot{\circ} dt$, where $\circ$ is a placeholder for the different variables. This allows inserting the derivative equations (4.11) – (4.17) and applying the connections (4.5) – (4.10) for simplification. The equations are rather bulky but their computation is straight forward. Hence, the details are skipped here for the sake of brevity. After simplification, the derivative with respect to t can be taken. In this step, almost all remaining terms cancel out. Letting $dt \to 0$, what remains is

$$\frac{d(X_s(t_1) - X_s(t_0))}{dt} = \dot{L}_s(t) = \dot{L}_0(t) + s\dot{E}_0(t) \tag{4.24}$$

For the distortion in t-direction in distance s along the generators, this yields:

$$l = \|\dot{L}_0 + s\dot{E}_0\| \tag{4.25}$$

With this equation, the coefficients of the first fundamental form become:

$$g_{11} = 1; \qquad g_{12} = g_{21} = l\langle E, \dot{L}\rangle; \qquad g_{22} = l^2 \tag{4.26}$$

The second derivatives $X_{st} = \dot{E}$ and $X_{ts} = \dot{L}'$, are given by equations (4.16) and (4.18). Since the generators are straight lines, one has $X_{ss} = E' = 0$. Employing the Frenet-Serret-equations for $\dot{L}$ yields $X_{tt} = \ddot{L} = \kappa\dot{L}$. Note that the derivative equations only hold for normalized vectors. The vectors thus have to be multiplied by their lengths, which in the case of $l = \|\dot{L}\|$ (cf. eqn. (4.25)) depends on both parameters, s, and t. The coefficients of the second fundamental form then become:

$$h_{11} = \langle X_{ss}, N\rangle = 0 \tag{4.27}$$

$$\begin{aligned} h_{12} &= \langle X_{st}, N\rangle = \langle 1 \cdot \dot{E}, N\rangle \\ &= \kappa\langle E, \dot{L}\rangle\langle \mathcal{N}, N\rangle + \langle \dot{L}, T\rangle(\tau + \omega) \end{aligned} \tag{4.28}$$

$$\begin{aligned} h_{21} &= \langle X_{ts}, N\rangle = \langle \tfrac{d}{ds} l \cdot \dot{L}, N\rangle = 0 + \langle \dot{L}', N\rangle \\ &= -l\delta\langle \dot{L}, T\rangle \end{aligned} \tag{4.29}$$

$$\begin{aligned} h_{22} &= \langle X_{tt}, N\rangle = \langle \tfrac{d}{dt} l\dot{L}, N\rangle = 0 + \langle l\ddot{L}, N\rangle \\ &= \kappa l\langle \mathcal{N}, N\rangle \end{aligned} \tag{4.30}$$

By the symmetry of the second fundamental form, $h_{12} = h_{21}$ provides a connection between the striction angle σ and the invariants κ, τ, ω, and δ:

$$\begin{aligned} h_{12} &= h_{21} \\ -l\delta\langle \dot{L}, T\rangle &= \kappa\langle E, \dot{L}\rangle\langle \mathcal{N}, N\rangle + \langle \dot{L}, T\rangle(\tau + \omega) \\ \frac{1}{\tan\sigma} &= -\frac{\tau + \omega + l\delta}{\kappa\langle \mathcal{N}, N\rangle} \end{aligned} \tag{4.31}$$

As it turns out, equation (4.31) is central to the system of coupled frames proposed here. Therefore, before turning to the derivation of the bending energy equations, this result is discussed in more detail.

4.3.6 A Minimal and Complete System of Invariants for Arbitrary Ruled Surfaces

First, the following generality theorem should be proven:

Theorem: Generality

Given the surface invariants κ, τ, ω, and δ, the system of coupled frames described above completely defines all possible ruled surfaces.

For the proof, note that the system of coupled frames completely contains the skew frame of E. Kruppa's natural parameterization [84]. Thus, the model is at least as powerful as Kruppa's frame. Since this only excludes a few examples, a direct proof is attempted, explicitly modeling the missing classes of surfaces in the framework. Those are the cylinders, the cones, and the tangent plane surfaces. Examples of these surfaces can be found in Figure 4.10. All of these surfaces are developable and thus constrained by $\delta = 0$ identically along the centerline. For the cones and the cylinders, a closed centerline with $\omega = 0$ is applied and the angle between the centerline's principal normal and the surface normal is chosen such that the generators constitute the required shape, which means they are parallel for the cylinders and intersect in a single point at a certain distance to the centerline for the cones. For the tangent plane surfaces, the solution is a little tricky, since the striction angle σ must be guaranteed to vanish identically to let the generators fall together with the centerline tangents. Since σ is not among the modeling parameters, equation (4.31) needs to be solved for $\sigma = 0$ to obtain the space of possible solutions. Towards an example, require $\omega = 0$ and define the surface normal to be orthogonal to the centerline's principle normal. Equation (4.31) now reduces to $\sigma = \arctan\frac{-\kappa}{\tau} \cdot 0$, and the centerline is almost arbitrary. Almost, because for a straight centerline, forcing $\sigma = 0$ results in the surface's degeneration to a straight line. However, the system remains stable in this case if the centerline's torsion τ is set to 0, allowing to pass through such degenerate points or sections along the surface. However, this comes at the price that one has to take care not to run into such physically unsound cases when applying the technique to model real surface strips. Being able to model the cones, the cylinders, and the tangent planes, it is straight forward to

also model their combinations by simply connecting corresponding ruled surface sections with properly aligned first and last generators and smoothly connected centerlines. Since ω and δ are angular velocities and therefore derivatives, their smooth transition along the centerline suffices to establish a smooth connection of two surface segments along the common generators. Therefore, one can not only construct all combinations of developable segments but generally all sorts of combinations between developable and nondevelopable surface segments. In other words, arbitrary ruled surfaces can be defined. □

Linking the striction angle to four of the six parameters used to determine the surface thus far, equation (4.31) gives rise to a number of interesting observations:
The invariance of δ and ω has already been proven and it is clear that being the invariants of the centerline κ and τ also need to be surface invariants. Since for γ and λ, one actually has $\gamma = \frac{d\sigma}{dt}$ and $\lambda = \frac{d\sigma}{ds}$, equation (4.31) can be applied to express γ and λ in terms of (the derivatives of) the other four parameters used to define the surface. By this connection, the surface can be described completely in the four invariants κ, τ, ω, and δ.
It can easily be seen that omitting κ and τ in the invariant system would limit the descriptive scope of the model by restricting the shapes of possible centerlines. Similar applies for δ, because it distinguishes the developable from the nondevelopable ruled surfaces. Without ω, only model twist-free surfaces could be modelled. An example for a surface for which neither δ nor ω vanish is shown in Figure 4.11. It is therefore concluded that the system of four invariants is also minimal:

Theorem: Minimality of the Invariant Set

$\{\kappa, \tau, \omega, \delta\}$ is a minimal and complete system of invariants for arbitrary ruled surfaces.

4.3.7 Curvature and Bending Energy

The system of coupled frames is now applied to the derivation of the bending energy for arbitrary ruled surfaces of arbitrary width. To this end, one needs to express the Gaussian and Mean Curvatures in the model.

The expressions for the surface's principal curvatures are obtained by computing the eigenvalues of the shape operator. Inverting the matrix of the first and applying the second fundamental form, the shape operator $\mathcal{L}$ can be ex-

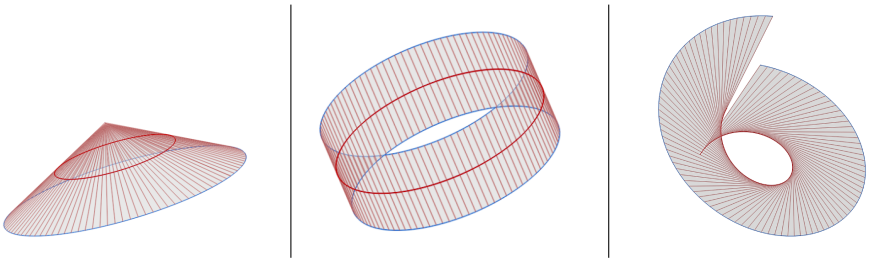


Figure 4.10: *A cone (left), a cylinder (center), and a tangent plane surface (right) rendered using the presented model. These surfaces and their combinations span the class of developable surfaces. Therefore, $\delta = 0$ identically along the centerline for all of them. For the examples, one also sets $\omega = 0$ although this is not necessary for a tangent plane surface. Other than developable surfaces, the system can describe nondevelopables applying the natural parameterization and is thus complete in the sense that arbitrary ruled surfaces can be described. For the tangent plane surface, the centerline is chosen to be one of the boundary curves. Note that this choice does not require any adjustment to the formalism describing the surface since the model allows choosing an arbitrary centerline on the surface. While the cylinder and the cone can be modeled directly, the tangent plane surface requires to solve equation 4.31 for $\sigma = 0$. For the example, the surface normal is fixed to be orthogonal to the centerline's principle normal at all points along the centerline, resulting in a free choice of the centerline's shape.*

pressed in terms of the following coefficients:

$$h_1^1 = -\frac{1}{l^2\langle \dot{L}, T\rangle^2}\left(\kappa l\langle E, \dot{L}\rangle^2\langle \mathcal{N}, N\rangle + l\langle E, \dot{L}\rangle\langle \dot{L}, T\rangle(\tau + \omega)\right) \tag{4.32}$$

$$h_1^2 = \frac{1}{l^2\langle \dot{L}, T\rangle^2}\left(\kappa\langle E, \dot{L}\rangle\langle \mathcal{N}, N\rangle + \langle \dot{L}, T\rangle(\tau + \omega)\right) \tag{4.33}$$

$$h_2^1 = -l\delta\langle \dot{L}, T\rangle - \frac{l^3\delta\langle \dot{L}, T\rangle\langle E, \dot{L}\rangle}{l^2\langle \dot{L}, T\rangle^2} - \frac{\kappa l^2\langle \mathcal{N}, N\rangle\langle E, \dot{L}\rangle^2}{l^2\langle \dot{L}, T\rangle^2} \tag{4.34}$$

$$h_2^2 = \frac{l^2\delta\langle \dot{L}, T\rangle\langle E, \dot{L}\rangle}{l^2\langle \dot{L}, T\rangle^2} + \frac{\kappa l\langle \mathcal{N}, N\rangle}{l^2\langle \dot{L}, T\rangle^2} \tag{4.35}$$

The principal curvatures κ_1 and κ_2 are then given by the eigenvalues of $\mathcal{L}$:

$$\kappa_1, \kappa_2 = \frac{1}{2}\left(\operatorname{Tr}\mathcal{L} \pm \sqrt{\operatorname{Tr}^2\mathcal{L} - 4\det\mathcal{L}}\right) \tag{4.36}$$

For the determinant, one obtains:

$$\det\mathcal{L} = -\delta \tag{4.37}$$

Note that by the definition of δ, along the centerline the trace of $\mathcal{L}$ is exactly

the drall. For the trace, three equivalent representations can be derived:

$$\begin{aligned}\operatorname{Tr}\mathcal{L} &= \frac{\delta}{\tan\sigma} + \frac{\kappa\langle\mathcal{N},N\rangle}{l\langle\dot{L},T\rangle^2} - \frac{\kappa\langle\mathcal{N},N\rangle\langle E,\dot{L}\rangle^2}{l\langle\dot{L},T\rangle^2} - \frac{\tau+\omega}{l\tan\sigma} \\ &= \frac{\kappa\langle\mathcal{N},N\rangle}{l}\left(1+\frac{1}{\tan^2\sigma}\right) && (4.38) \\ &= \frac{\kappa^2\langle\mathcal{N},N\rangle^2 + (\tau+\omega+l\delta)^2}{l\kappa\langle\mathcal{N},N\rangle} && (4.39) \\ &= \frac{2\delta}{\tan\sigma} + \frac{\kappa\langle\mathcal{N},N\rangle}{l\sin^2\sigma} && (4.40)\end{aligned}$$

In this discussion, $\operatorname{Tr}\mathcal{L}$ generally refers to (4.38). The representations (4.39) and (4.40) are listed here for completeness since they are closer to the results typically found in the literature.

Using the determinant and trace of the shape operator, equations for the Gaussian curvature K and the mean curvature H can be provided:

$$K = \kappa_1 \cdot \kappa_2 = -(\det\mathcal{L})^2 = -\delta^2 \tag{4.41}$$

$$H = \frac{1}{2}(\kappa_1+\kappa_2) = \frac{1}{2}\operatorname{Tr}\mathcal{L} = \frac{\kappa\langle\mathcal{N},N\rangle}{2l}\left(1+\frac{1}{\tan^2\sigma}\right) \tag{4.42}$$

Prior to the construction of the bending energy integral, it is important to mention that this is the first time the discussion's generality is limited. While the model can describe the stretching and shearing of ruled surfaces under deformation, for the bending energy this discussion is only concerned with the special case of unshearable inextensible elastic strips. Unshearability and inextensibility require that only deformations are allowed that do not change the length of any arbitrarily chosen surface curve, which means, the deformations need to be path-isometric.

Given a surface element da and the principal curvatures, a common ansatz to the derivation of a bending energy function is $E = \int_S \kappa_1^2 + \kappa_2^2 da$. For the developable surfaces, this energy functional becomes $\int_S H^2 da$ with H being the mean curvature. However, this functional does not cover the additional longitudinal bending induced by the drall on nondevelopable surfaces. To achieve generality, an energy functional motivated by mathematical physics is proposed [35]. It describes the energy of a surface-like body that is planar at rest and whose potential energy is determined by the integral over the quadratic form of the principal curvatures of the surface obtained from bending:

$$E = \int_S A(H^2 - 2K) + 2BK\,da \tag{4.43}$$

for the mean curvature H, the Gaussian curvature K and material constants A and B.

The mean curvature and Gaussian curvature in the model do not depend on a reference geometry. Therefore, the bending energy may be computed directly using equation (4.43). For developable surfaces, this additionally yields a trivial constraint for energy minimization. Typically this problem is solved leveraging

the elasticity by applying variational calculus to derive the equilibrium equations [42, 134]. For a developable surface with fixed centerline, inserting (4.39) as the expression for the mean curvature into (4.43), the only degree of freedom remaining is ω, since its value also determined $\langle \mathcal{N}, N \rangle$. It directly follows that for a developable surface the bending energy is minimal if ω vanishes identically along the centerline and the surface normal is equal to the centerline's principle normal[1].

Letting the surface strip become infinitesimally narrow, one observes that the angle between the centerline normal $\mathcal{N}$ and the surface normal N becomes less significant until it vanishes completely (That is, for nonvanishing σ) Since along the centerline, one always has $l = 1$ because of the assumption of arc length parameterization, as the width approaches 0, l approaches 1 globally along the surface's generators. At the same time, the δ and ω become more and more insignificant since in an infinitesimally narrow setting, the surface appears locally flat. The only invariants remaining are κ and τ. Inserting equation (4.39) into H in the bending energy then results in:

$$E = \frac{A}{4} \int_S \frac{(\kappa^2 + \tau^2)^2}{\kappa^2} da \tag{4.44}$$

which is exactly the functional Sadowsky proposed in his investigation of the Möbius strip in 1930 [121]. In some literature, the functional can be found in another formulation more similar to the usage of equation (4.38) for the definition of the mean curvature H [33]. In this case, one obtains:

$$E = \frac{A}{4} \int_S \kappa^2 \left(1 + \frac{1}{\tan^2 \sigma}\right)^2 da \tag{4.45}$$

4.3.8 Computing the surface

Considering the implementation, there are essentially two approaches to defining the surface. One way is to directly integrate the invariants and transport the frames accordingly similar to the computation of a space curve using the Frenet frame. Another approach allows to specify the drall and the geodesic curvature explicitly while the centerline is provided directly from outside the system, for example as a Bézier curve.

Direct integration requires the definition of all four invariants in the centerline's arc length parameter t, a starting point, an initial tangent, and an initial surface normal. Note that for the computation, the definitions of the invariants need to be at least C^1-continuous. The whole surface can then be computed by solving the initial value problem using the transition equations for the frames of reference. Where needed, the centerline's normal and binormal can be obtained from its Frenet frame using the invariants κ and τ. While more sophisticated methods can be implemented, experiments show that integration by the Euler method already yields good results. Note that computing the bending energy density along the centerline on the fly produces almost no overhead since the

[1] for $\langle \mathcal{N}, N \rangle \to 0$, only $\frac{\tau}{\kappa \langle \mathcal{N}, N \rangle}$ remains and approaches infinity.

components of the mean curvature are also needed to obtain the striction angle and therefore need to be computed anyway.

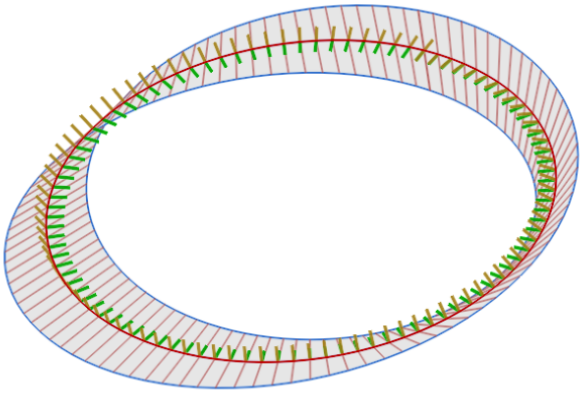


Figure 4.11: *A Möbius strip. Since its centerline is a circle, the strip is not developable. In the model discussed here, developability is characterized by the vanishing of the invariant δ.*

If the centerline (and thus κ and τ) is determined by other means, e.g. as a Bézier curve, and the invariants ω and δ are again given as C^1-continuous function in the centerline's parameter t, the frame directions can be computed directly for arbitrary values of t. For the transverse direction, the respective direction of the centerline's geodesic parallel is obtained from equation (4.24) and frame can therefore be computed directly given that $N = [L, E]$. Since $\dot{E}$ is given by equation (4.16) and thus computed for any generator constructed, this allows the computation of an arbitrarily dense net of surface points in two passes on the GPU, where the first one computes the generators along the centerline and the second one samples the frames long the generators.

In case of high centerline curvatures and geodesic torsion, the density of generators available to tessellate the surface can quickly decrease with further distance to the centerline. Rather than computing more generators, equation (4.24) can be applied to obtain geodesic parallels to the centerline.

4.3.9 Physical Interpretation of the Generators

Pure bending describes the idea of deforming a surface based only on bending, which means that deformations may neither change the geodesic torsion nor the drall since both parameters together determine the surface's twist. For the special case of inextensible materials, one additionally needs to guarantee that the centerline does not change its length. Techniques to describe the length-preserving bending of space curves exist and can be applied to define the centerline and introduce this behavior. While restrictions on the shape of the

surface can be enforced trivially, the effort of guaranteeing conditions to hold on the transformations depends on the application.

The transformations that can be described by pure bending are exactly the ones that preserve the distances between two fixed points on the surface as well as their positions in the surface parameters s along the generators and t along the centerline. Since these transformations are obviously isometric, they are of special interest to modeling surface strips from planar sheets of material: Isometric transformations preserve developability. While in the model discussed here this is equivalent to the invariant δ vanishing identically along the centerline, other models commonly need additional side conditions to achieve developability. Constructive approaches that explicitly model a given shape as an isometric deformation of a planar strip of material therefore often restrict the set of allowed deformations to those defined by pure bending. However, while pure bending fixes the drall and the geodesic torsion ω, the latter does not influence developability, indicating that this restriction might be too strict when judging it from a purely geometrical perspective. Indeed, by equation (4.41), the Gaussian curvature does not depend on ω and therefore δ is the only variable whose value needs to be preserved in an isometric deformation. It can be observed that although two points marked on the surface may change their s- and t-coordinates during deformation, the metric is preserved as long as δ does not change along the centerline. This notion of isometry, the preservation of the arc lengths of any curve on the surface, is sometimes referred to as *path isometry*.

An experiment shows that indeed path isometry adequately describes the bending behavior of ruled surfaces. Consider a long and narrow rectangular strip of paper where several lines parallel to the short edge are marked as generators of the surface. Bending this strip into a looping as shown in Figure 4.12 and pulling apart the edges carefully so the strip does not buckle, one can observe that the lines that originally denoted the generators are now curved. However, this deformation was actually achieved by pure bending: At no time has a twist been introduced. The apparent torsion is only due to the curvature and torsion of the centerline and is a consequence of the fact that the angle between the centerline's principal normal and the surface normal is preserved during the deformation.

If the paper strip is stretched after forming the looping, $\int_L \omega dt$ approaches $\frac{\pi}{2}$ along the centerline. Since δ cannot change its value as otherwise the paper would tear, the striction σ has to compensate the deformation induced by the potential change of δ. Since $\sigma = \arccos\langle \dot{L}, E\rangle$, points aligned along the material frame's K-axis not change their positions in the surface's parameters s and t, since s is determined by the generators E and by the coupling of the frames $\{\dot{L}, N, K\}$ and $\{E, N, T\}$, every change of the striction also changes the angle between K and E. As a result, the lines drawn on the paper strip seen in Figure 4.12 do not share the same generator anymore. Still, their geodesic distance to the centerline must have been preserved by the isometry of the transformation since otherwise the paper would have torn apart. For the same reason, the applied deformation must have preserved the metric. This yields the conclusion that the transformation must be path isometric; even if individual points on

Figure 4.12: *A bent paper strip. One can observe that the formerly parallel straight black lines are now curved. Since the transformation is built without changing the parameterδ the surface remains torsal. Hence, the striction has changed and strict isometry does not provide a correct transformation for this phenomenon.*

the surface do not maintain their coordinates given in the surface parameters, the preservation of ω and δ during the deformation of the centerline has been achieved by a change of the striction angle σ.
Generalizing this experiment yields four important observations:

1. The developability-preserving deformation between two physically sound shapes of inextensible elastic strips is adequately described by path isometric transformations of a ruled surface.
2. The generators do not directly represent physical properties.
3. The path isometric deformations that can be applied to a developable ruled surface are neither necessarily free of twist nor restricted to pure bending.
4. The converse to (1.) does *not* hold: While path isometry preserves vanishing Gaussian curvature and thus developability, additional constraints are needed to guarantee the resulting shape to be physically sound.

It is important to point out that path isometry is a much weaker notion than strict isometry since it does not necessarily preserve the positions of points in the coordinates induced by the surface's parameters s and t. One also has to be cautious of the fact that such a deformation may introduce self-intersections. As another remark, note that while one could expect that for a planar sheet

the direction of the generators and therefore the striction σ would be arbitrary, the proposed model indeed requires the generators of a flat rectangle to be orthogonal to the centerline if the centerline is straight and parallel to the long surface edge. The reason is that since if all invariants vanish, by equation (4.31) $\frac{1}{\tan\sigma}$ also needs to vanish, which in the limit holds for $\sigma = \frac{\pi}{2}$.

4.3.10 Deformations of Actual Materials

For a real, noninfinitesimal surface, thickness restricts the possible deformations. While for thin surfaces like plywood or paper the degrees of freedom remain the same, the boundary surfaces now have to be considered because the transformations can induce self-intersections and buckling in these surfaces, even if this does not occur in the surface in the center. Additionally, transformations can induce a shear of the boundary surface relative to the one in the center. In its current form, the model does not consider those restrictions. With a focus on the geometry of ruled surfaces, the model does not yet include a description of the forces acting on a material during deformation. Thus, although the model is able to accurately describe arbitrary shapes of ruled surfaces, the transformation between two surfaces is currently only restricted by the geometry. Yet, since the surface's geometry is completely described by the invariant system, physical correctness can be achieved by introducing additional side conditions restricting the values of the invariants. Equipping the model with equations capturing the forces acting on the surface during deformation is one of several extensions to the model that seem promising directions for further work.

4.3.11 Conclusion: Qualitative Considerations Help to Adapt the Design to the Reasoning Structure

In this case study, an analytically exact model for ruled surfaces of arbitrary shape is developed to enable reasoning about thin and narrow surface strips from the perspective of the local bending behavior. A minimal and complete system of invariants determines the surface in a single parameter – the centerline's arc length. Other than its simple formulation fostering intuitive surface modeling, a particularly appealing property of the model is its independence of planar reference geometries typically needed to guarantee important properties important to engineering applications. This is achieved by coupling a moving frame of reference describing the mechanical properties of a surface strip with a frame describing the space into which this material strip can be embedded, allowing to treat the deformation problem purely geometrically. Although the exact embedding of the physical surface into the geometrical description is yet open, the presented model describes arbitrarily shaped ruled surfaces and their bending behavior and yields a bending energy integral for arbitrarily shaped ruled surfces of arbitrary width.

The construction implements the principle of design for reasoning. Important local features identified are the drall and the geodesic torsion determining the surface's twising and the centerline's curvature and torsion characterizing

the its bending. The parameterization has been chosen accordingly. Examples for important drived properties are the developability of the surface and the isometry of surface deformations. The local surface description based on the identified features and their interpretation can be described by the concept graph. In the developed surface model, important local surface properties can be concluded directly from qualitative considerations rather than having to be derived from complex optimization procedures. For the visualization, the concept graph indicates that developabity can either be inferred by the local depiction of the drall or of the gaussian curvature map. For the deformation of a strip, the difference of the drall should instead be shown explicitly and it should especially be highlighted where the drall's value does not change as this indicates local isometry of the deformation.

In conclusion, the concept graph can be utilized along with the developed description of the surface's geometry to determine visualizations following the principle of design for reasoning. Designing visualizations this way yields graphical representations of the strip allowing to conclude important local properties directly from the depiction without the requirement for complex reasoning combining multiple variables or relating the displaye surface to a planar reference configuration as it is necessary in the related work defining the current state of the art of representing inextensible surface strips.

4.3.12 Implications of Qualitative Visual Analysis for the Design Process

Qualitative considerations not only improve the design itself but also support the design process. In this chapter, two case studies show the influence of qualitative considerations on design decisions. The qualitative principles of visual information encodings found in Chapters 1 and 2 motivate the design decisions that led the respective project to success. The principle of design for reasoning has the most significant influence on both case studies. In the project with the police it motivated the introduction of PDCA as the development workflow and for the surface strip model it motivated the design of a local description of the geometry. The principle of minimal graphical overhead is of general relevance for the design and is well reflected in the different visualizations developed for the project with the police. The same holds for the inside-outside principle that affects both projects at least implicitly describing the data space and the space of possible interpretations. Thereby it establishes the basis necessary for the design to follow the principle of design for reasoning. In conclusion, qualitative considerations significantly influence the design and the development of visualization applications. Therefore, they should always be explicitly considered when discussing the rationale behind design decisions.

Core References

1. B. Karer, A. Freund, M. Horst, I. Scheler, and H. Hagen: Dealing with Sparse Domain Information – Visualization Practice Lessons. *IEEE VIS – Visualization in Practice Workshop*. IEEE, 2017.
2. B. Karer, A. Freund, M. Horst, I. Scheler, T. Kossurok, and F.-J. Brandt: Designing Interactive Visualization Despite Sparse Availability of Domain Information. *IEEE computer graphics and applications*, volume 38(5), pages 54-69. IEEE, 2018
3. B. Karer, and H. Hagen: Drall-Based Ruled Surface Modeling. In *Graphical Models*. Elsevier, (under revision).

Chapter 5

Summary and Conclusion

As core topics of qualitative visual analysis, interpretation mechanisms, insight provenance, and analysis complexity inspire the three fundamental research questions treated in this Thesis: The question for qualitative visual analysis models for validation and analysis of visualization techniques, the question for qualitative visual analysis workflows, and the question for the implications of qualitative visual analysis on visualization and interaction design. This concluding chapter summarizes the achievements reached and discusses their relation to each other.

5.1 Achievements

The discussion of qualitative visual analysis as an additional perspective to the data-centric quantitative view on data analysis reveals three central qualitative principles of visual information encodings. As a direct consequence of the emphasis on the reasoning process, the inside-outside principle promotes the idea to discuss visualization and interaction with respect to the reasoning structure anticipated to be applied by the viewer to a general principle rather than a rule of thumb. The development of a visualization and visual data analysis model implementing this principle yields the principle of minimal qualitative graphical overhead and the principle of design for reasoning. The former inspires efficient analysis workflows and the latter allows to evaluate design choices against the anticipated reasoning models.

In this Thesis, three fundamental questions inspired by the idea of qualitative visual analysis are treated. A model for the qualitative visual analysis process is developed, allowing to formally express the reasoning structures anticpiated to be applied by analysts working with the visualization. A sophisticated model of visualization content and the artifcats and structures being read from visualization is developed, allowing a formal specification of a viewer's mental model of the visualization, including the outside knowledge determining the interpretation of structures recognized in the display. Based on those considerations, workflows are defined, extending the classical approach of visualizing the data

by an insight provenance mechanism allowing to reintegrate insight found during analysis into the data and thus making it accessible for further analysis steps, effectively implementing an insight amplification mechanism. Towards workflows optimally supporting reasoning structures, a feasibility proof for the automatic information-driven design of visual analytics pipelines extends the idea of insight amplification by a technique to determine data transformation and visualization sequences by mapping available algorithms to the anticipated reasoning structure. A review of a long-term project with the German police motivates questions for the influence of qualitative visual analysis on the visualization design process. Two case studies demonstrate how qualitative considerations can be incorporated into the design process of visual interactive systems and how the choice of the representation can be adapted to the anticipated reasoning structure.

Combining the models and techniques discussed in this Thesis, a qualitative discussion of the design and application of visualization techniques becomes feasible. The formalism for the specification of the mental model enables the comparison of vsualization techniques based on their mapping to an anticipated reasoning structure. Applying this idea during the design phase thus allows to choose an optimal visualization design based on the available information about the reasoning mechanisms to be applied during analysis. The formal treatment of analysis strategies enabled by the qualitative visual analysis cycle and the concept graph allows to evaluate visualization techniques from a theoretical perspective, validating the design choices against formal requirements posed by the reasoning process. Analysis workflows can be optimized towards the reasoning process by aligning sets of available transformation and visualization techniques with the mental model. Combining the structured workflow for insight amplification discussed in this Thesis with the formslism for the specification of the mental model allows to continuously extend the mental model during analysis. Since the concept graph is interpretable by machines and humans alike, this implements a mechanism for insight provenance that can also be leveraged for the continuous adaptation and extension of the visualization towards optimal support of reasoning during further analysis steps. Formalizing the design process based on the qualitative visual analysis cycle allows to tailor visualization and interaction towards anticipated interpretation and reasoning structures. The concept graph can be applied to validate design decisions on a theoretical base and to infer predictions about the performance of visualizations that can serve as theoretically motivated hypotheses to be tested in user studies when evaluating the visualization. Applying the concept graph to document insight provenance in experimental user studies reveals hints for the adaptation and improvement of visualization based on the difference between the anticipated and the oberserved reasoning structures. Combined with a development workflow based on an iterative improvement process like the ones discussed in this Thesis, this kind of formal validation enables to prove the effectiveness of design choices.

In summary, the techniques and models discussed in this Thesis enable the formal and provable assessment of visualization performance. Being a direct consequence of the explicit consideration of the reasoning process into the discussion of visualizations, this kind of theoretical study and evaluation of visualization is not possible with the purely quantitative data-centric perspective on

visualization being the state of the art prior to this Thesis.

5.2 Prospect

Thinking further in the direction of the existing and potential application of qualitative visual analysis reveals some promising directions for future work. Models and mechanisms for the embedding of data visualization into its general context will support the abstraction of context-specific interpretations by decomposing the requirements into the kinds of context they reflect. Combined with efforts to formalize task-specific and domain-specific interpretation conventions, explicit treatment of qualitative design aspects will provide new workflows extending the existing methodology for the assessment of design requirements. For example, multi-stage workflows could attempt to tailor a visualization design to the domain context in an early stage and to apply fine-tuning to capture the analysis and user contexts later. The formal decoupling of different contexts in the formal model will allow cross-domain applications dynamically tailoring the output to the corresponding viewers' individual backgrounds. Observing the ongoing trend to integrate data across domains in semantically heterogeneous environments, such techniques will become increasingly important. Based on these techniques, dynamic assignment of interpretations adds qualitative features to existing methods for analysis based on dynamic feature definition. Navigating the qualitative feature space and adjusting the visual presentation accordingly will empower domain experts to fine-tune the visualization dynamically by interacting with prototypes pre-tailored to the domain context. Making the implicit bonds between visual encodings and their context-specific interpretation explicit and studying the mechanisms underlying this connection will reveal insight into the very process of visualization-based data analysis itself. Models determining data interpretation with respect to the given domain will enable the characterization of visualization techniques by the kind of information they convey rather than the data they represent. More generally, the ability to collect all information necessary to solve a task is important for user guidance, automatic visualization, and visualization propositions. Especially generic applications where the visualization is created by the analyst rather than a visualization expert will benefit from endeavors in this direction. The qualitative mechanisms and principles underlying visualization-based data analysis will also provide a new perspective on evaluation. Insights into information emergence and provenance will allow to reflect on the complexity of the reasoning process and how well it is supported by the visual representation. Endeavors in this direction will yield insight-driven design paradigms, oriented directly towards the analysis result.

5.3 Concluding Remarks

The numbers do not lie. The numbers do not err. Yet, the numbers also do not tell. Whatever conclusions are drawn from data analysis, at some point the data has to be interpreted with respect to its general context. Hence, in order to obtain knowledge or to generate decision competence by data analysis, it is

necessary to go beyond the limited perspective of quantitative analysis operating only inside the data context and to take into account the data's surrounding context. Analyzing data concentrating only on measurable, quantitative properties is limited by the very nature of data and the restrictions of the domain and general context. Qualitative visual analysis overcomes those limitations by focusing on the reasoning rather than on the data. The qualitative principles of visual information encodings derived from qualitative visual analysis inspire a powerful set of formal technqies to be added to the available toolset for the design and discussion of visualization methodology, interaction, and analysis workflows. The major contribution of this Thesis is to be seen in the development of a formal theoretical framework applicable throughout the whole life cycle of a visualization application, from the tailoring of the design towards the domain to the documentation of insight provenance by analysts working with the visualization. By its focus on the reasoning process, rather than on the data and features, this formalization allows to provably predict and validate the effectiveness and efficiency of visualization. Properly applied, the models and techniques proposed in this Thesis render qualitative visual analysis a powerful extension to purely data-related considerations towards a holistic perspective on the design and application of visualization for data analysis.

Bibliography

[1] J.-W. Ahn and P. Brusilovsky. Adaptive visualization of search results: Bringing user models to visual analytics. *Information Visualization*, 8(3):167–179, 2009.

[2] V. Akman and M. Surav. Contexts, oracles, and relevance. pages 23–30, 1995.

[3] A. H. Altalhi, J. M. Luna, M. A. Vallejo, and S. Ventura. Evaluation and comparison of open source software suites for data mining and knowledge discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(3), 6 2017.

[4] R. Amar and J. Stasko. A knowledge task-based framework for design and evaluation of information visualizations. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 143–150. IEEE.

[5] M. Angelini and G. Santucci. Modeling Incremental Visualizations. In M. Pohl and H. Schumann, editors, *EuroVis Workshop on Visual Analytics*. The Eurographics Association, 2013.

[6] F. J. Anscombe. Graphs in statistical analysis. *The American Statistician*, 27(1):17–21, 1973.

[7] G. Antoniou, , G. Antoniou, G. Antoniou, F. V. Harmelen, and F. V. Harmelen. Web ontology language: Owl. In *Handbook on Ontologies in Information Systems*, pages 67–92. Springer, 2003.

[8] J. Barwise. The situation in logic. In G. J. D. Ruth Barcan Marcus and P. Weingartner, editors, *Logic, Methodology and Philosophy of Science VIIProceedings of the Seventh International Congress of Logic, Methodology and Philosophy of Science*, volume 114 of *Studies in Logic and the Foundations of Mathematics*, pages 183 – 203. Elsevier, 1986.

[9] J. Barwise. Constraints, channels and the flow of information. In P. Aczel, D. Israel, Y. Katagiri, and S. Peters, editors, *Situation Theory and its Applications Vol. 3*. CSLI Publications, 1993.

[10] J. Barwise and J. Perry. *Situations and Attitudes*. MIT Press, Cambridge, MA, 1983.

[11] J. Barwise and J. Seligman. *Information flow: the logic of distributed systems*, volume 44. Cambridge University Press, 1997.

[12] L. Bavoil, S. P. Callahan, P. J. Crossno, J. Freire, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Vistrails: Enabling interactive multiple-view visualizations. In *Visualization, 2005. VIS 05. IEEE*, pages 135–142. IEEE, 2005.

[13] I. Berinskii and A. Krivtsov. A hyperboloid structure as a mechanical model of the carbon bond. *International Journal of Solids and Structures*, 96:145 – 152, 2016.

[14] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.

[15] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel. *KNIME: The Konstanz Information Miner*, pages 319–326. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[16] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983.

[17] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *ICDT '99: Proceedings of the 7th International Conference on Database Theory*, pages 217–235, London, UK, 1999. Springer-Verlag.

[18] W. Blaschke. *Vorlesungen über Differential-Geometrie I. Zweite Auflage*. Verlag von Julius Springer in Berlin, 1924. (GEN) Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen Band 1.

[19] S. Bødker. Creating conditions for participation: Conflicts and resources in systems development. *Hum.-Comput. Interact.*, 11(3):215–236, Sept. 1996.

[20] G.-P. Bonneau, H.-C. Hege, C. R. Johnson, M. M. Oliveira, K. Potter, P. Rheingans, and T. Schultz. Overview and state-of-the-art of uncertainty visualization. In *Scientific Visualization*, pages 3–27. Springer, 2014.

[21] S. Bresciani and M. J. Eppler. The pitfalls of visual representations: A review and classification of common errors made while designing and interpreting visualizations. *Sage Open*, 5(4), 2015.

[22] J. R. Büchi. On a Decision Method in Restricted Second-Order Arithmetic. In *International Congress on Logic, Methodology, and Philosophy of Science*, pages 1–11. Stanford University Press, 1962.

[23] S. M. Casner. Task-analytic approach to the automated design of graphic presentations. *ACM Trans. Graph.*, 10(2):111–151, 1991.

[24] S. M. Casner et al. Cognitive efficiency considerations for good graphic design, technical report. *Online] http://handle. dtic. mil/100.2/ADA218976*.

[25] H.-Y. Chen, I.-K. Lee, S. Leopoldseder, H. Pottmann, T. Randrup, and J. Wallner. On Surface Approximation Using Developable Surfaces. *Graphical Models and Image Processing*, 61(2):110 – 124, 1999.

[26] M. Chen, D. Ebert, H. Hagen, R. S. Laramee, R. van Liere, K. L. Ma, W. Ribarsky, G. Scheuermann, and D. Silver. Data, information, and knowledge in visualization. *IEEE Computer Graphics and Applications*, 29(1):12–19, Jan 2009.

[27] M. Chen, L. Floridi, and R. Borgo. What is visualization really for? *CoRR*, 2013.

[28] M. Chen and A. Golan. What may visualization processes optimize? *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2619–2632, Dec. 2016.

[29] M. Chen and H. Jaenicke. An information-theoretic framework for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1206–1215, Nov 2010.

[30] Y.-c. Chen and E. Fried. Möbius bands, unstretchable material sheets and developable surfaces. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 472(2192), 2016.

[31] N. Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, September 1956.

[32] J. Choo, C. Lee, H. Kim, H. Lee, Z. Liu, R. Kannan, C. D. Stolper, J. Stasko, B. L. Drake, and H. Park. Visirr: Visual analytics for information retrieval and recommendation with large-scale document data. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 243–244, Oct 2014.

[33] D. Chubelaschwili and U. Pinkall. Elastic strips. *manuscripta mathematica*, 133(3-4):307–326, 2010.

[34] W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 1985.

[35] R. Courant and D. Hilbert. *Methoden der Mathematischen Physik I.* Heidelberger Taschenbücher. 2013.

[36] P. P. da Silva, S. Deborah, D. L. McGuinness, and R. Mccool. Knowledge provenance infrastructure. *Data Engineering Bulletin*, 26(4):26 – 32, 2003.

[37] P. P. da Silva, D. L. McGuinness, and R. Fikes. A proof markup language for semantic web services. *Inf. Syst.*, 31(4-5):381–395, June 2006.

[38] W. Deming. *Out of the Crisis.* Cambridge, Mass. : Massachusetts Institute of Technology, Center for Advanced Engineering Study, 1986.

[39] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Rapid data exploration through guideposts. *CoRR*, abs/1709.10513, 2017.

[40] C. Demiralp, C. E. Scheidegger, G. L. Kindlmann, D. H. Laidlaw, and J. Heer. Visual embedding: A model for visualization. *IEEE Computer Graphics and Applications*, 34(1):10–15, Jan 2014.

[41] J. Demšar, T. Curk, A. Erjavec, Črt Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, and B. Zupan. Orange: Data mining

toolbox in python. *Journal of Machine Learning Research*, 14:2349–2353, 2013.

[42] M. A. Dias and B. Audoly. "Wunderlich, meet Kirchhoff": A general and unified description of elastic ribbons and thin rods. *Journal of Elasticity*, 119(1-2):49–66, 2014.

[43] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the Symposium on Data Visualisation 2003*, VISSYM '03, pages 239–248, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[44] S. P. Dow, A. Glassco, J. Kass, M. Schwarz, D. L. Schwartz, and S. R. Klemmer. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. In *Design Thinking Research*, pages 127–153. Springer, 2012.

[45] E. Efrati. Non-Euclidean Ribbons. *Journal of Elasticity*, 119(1-2):251–261, 2015.

[46] E. Efrati, E. Sharon, and R. Kupferman. Hyperbolic non-Euclidean elastic strips and almost minimal surfaces. *Physical Review E*, 83(4), 2011.

[47] M. Elizabeth Raven and A. Flanders. Using contextual inquiry to learn about your audiences. *ACM SIGDOC Asterisk Journal of Computer Documentation*, 20:1–13, 02 1996.

[48] A. Endert, W. Ribarsky, C. Turkay, B. W. Wong, I. Nabney, I. D. Blanco, and F. Rossi. The state of the art in integrating machine learning into visual analytics. *Computer Graphics Forum*, 2017.

[49] J. Esparza and M. Nielsen. Decidability issues for petri nets. *Petri nets newsletter*, 94:5–23, 1994.

[50] O. J. Espinosa, C. Hendrickson, and J. Garrett. Domain analysis: a technique to design a user-centered visualization framework. In *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, pages 44–52. IEEE, 1999.

[51] J.-D. Fekete, J. J. Van Wijk, J. T. Stasko, and C. North. The value of information visualization. In *Information visualization*, pages 1–18. Springer, 2008.

[52] M. Fink, J.-H. Haunert, J. Spoerhase, and A. Wolff. Selecting the aspect ratio of a scatter plot based on its delaunay triangulation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2326–2335, 2013.

[53] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, and A. Taylor. Cypher: An evolving query language for property graphs. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, pages 1433–1445, New York, NY, USA, 2018. ACM.

[54] G. Glaeser and F. Gruber. Developable surfaces in contemporary architecture. *Journal of Mathematics and the Arts*, 1(1):59–71, 2007.

[55] M. Golemati, C. Halatsis, C. Vassilakis, A. Katifori, and U. o. Peloponnese. A context-based adaptive visualization environment. In *Tenth International Conference on Information Visualisation (IV'06)*, pages 62–67, July 2006.

[56] D. Gotz, J. Lu, P. Kissa, N. Cao, W. H. Qian, S. X. Liu, and M. X. Zhou. Harvest: an intelligent visual analytic tool for the masses. In *Proceedings of the first international workshop on Intelligent visual interfaces for text analysis*, pages 1–4. ACM, 2010.

[57] D. Gotz and M. X. Zhou. An empirical study of user interaction behavior during visual analysis. *Technical Report RC24525*, 2008.

[58] D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. *Information Visualization*, 8(1):42–55, 2009.

[59] A. Goyal, M. R. Dörfel, B. Simeon, and A.-V. Vuong. Isogeometric shell discretizations for flexible multibody dynamics. *Multibody System Dynamics*, 30(2):139–151, 2013.

[60] D. P. Groth. Information provenance and the knowledge rediscovery problem. In *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004.*, pages 345–351, July 2004.

[61] D. P. Groth and K. Streefkerk. Provenance and annotation for visual exploration systems. *IEEE transactions on visualization and computer graphics*, 12(6):1500–1510, 2006.

[62] J. T. Guthrie, S. Weber, and N. Kimmerly. Searching documents: Cognitive processes and deficits in understanding graphs, tables, and illustrations. *Contemporary Educational Psychology*, 18(2):186 – 221, 1993.

[63] V. Haarslev and R. Möller. Racer: An owl reasoning agent for the semantic web. In *Proceedings of the International Workshop on Applications, Products and Services of Web-based Support Systems, in conjunction with the*, pages 91–95, 2003.

[64] R. Haber and D. McNabb. Visualization idioms: A conceptual model for scientific visualization systems. In L. R. G.M. Nielson, B. Shriver, editor, *Visualization in Scientific Computing*. IEEE Computer Society, 1990.

[65] A. Hall, P. Ahonen-Rainio, and K. Virrantaus. Insight provenance for spatiotemporal visual analytics: Theory, review, and guidelines. *Journal of Spatial Information Science*, 2017(15):65–88, 2017.

[66] A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? In *VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases*, pages 506–515, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[67] K. Holtzblatt and S. Jones. Contextual inquiry: A participatory technique for system design. *Participatory Design: Principles and Practices*, 01 1993.

[68] J. Hoschek. Globale Geometrie der Regelflächen. In J. Tölke and J. Wills, editors, *Contributions to Geometry*, pages 363–370. Birkhäuser Basel, 1979.

[69] D. Huff and I. Geis. *How to Lie with Statistics*. W. W. Norton, 2010.

[70] P. Janecek and P. Pu. A framework for designing fisheye views to support multiple semantic contexts. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '02, pages 51–58, New York, NY, USA, 2002. ACM.

[71] H. Janicke and M. Chen. A Salience-based Quality Metric for Visualization. *Computer Graphics Forum*, 29(3):1183–1192, 2009.

[72] H. Jänicke, T. Weidner, D. Chung, R. S. Laramee, P. Townsend, and M. Chen. Visual reconstructability as a quality metric for flow visualization. *Computer Graphics Forum*, 30(3):781–790, 2011.

[73] B. Karer, D. Fernández-Prieto, and H. Hagen. The Situation Universe: Visualizing the Semantics of Integrated Data Structures. In B. Kozlikova, T. Schreck, and T. Wischgoll, editors, *EuroVis 2017 - Short Papers*. The Eurographics Association, 2017.

[74] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Information visualization. chapter Visual Analytics: Definition, Process, and Challenges, pages 154–175. Springer-Verlag, Berlin, Heidelberg, 2008.

[75] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual data mining. chapter Visual Analytics: Scope and Challenges, pages 76–90. Springer-Verlag, Berlin, Heidelberg, 2008.

[76] G. Kindlmann and C. Scheidegger. An algebraic process for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2181–2190, Dec 2014.

[77] J. Kleinberg. An impossibility theorem for clustering. *Adv Neural Inform Process Syst (NIPS)*, 15, 01 2003.

[78] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen. The protégé owl plugin: An open development environment for semantic web applications. In *International Semantic Web Conference*, pages 229–243, 2004.

[79] D. König. Über eine Schlussweise aus dem Endlichen ins Unendliche. *Acta Litt. ac. sci. Szeged*, 3:121–130, 1927.

[80] Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung. V-modell xt. Technical report, 2004.

[81] R. Kosara, S. Miksch, and H. Hauser. Semantic depth of field. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, INFOVIS '01, Washington, DC, USA, 2001. IEEE Computer Society.

[82] S. M. Kosslyn. Understanding charts and graphs. *Applied Cognitive Psychology*, 3(3):185–225, 1989.

[83] M. Kreuseler, T. Nocke, and H. Schumann. A history mechanism for visual data mining. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 49–56. IEEE, 2004.

[84] E. Kruppa. Zur Differentialgeometrie der Strahlflächen und Raumkurven. *Sitzungsberichte d. Österr. Akad. d. Wiss., Math.-naturwiss. Kl., Abt. 2a, Bd. 157. 1949, H. 6/10*, pages 143–176, 1949.

[85] O. Lassila, R. R. Swick, W. Wide, and W. Consortium. Resource description framework (rdf) model and syntax specification, 1998.

[86] S. Lawrence. Developable Surfaces: Their History and Application. *Nexus Network Journal*, 13(3):701–714, 2011.

[87] D. J. Lehmann and H. Theisel. The lloydrelaxer: An approach to minimize scaling effects for multivariate projections. *IEEE Transactions on Visualization and Computer Graphics*, to appear, 2017.

[88] M. Lichman. Uci machine learning repository. chapter auto-mpg. University of California, School of Information and Computer Science., Irvine, CA, 2013.

[89] D. Lloyd and J. Dykes. Human-centered approaches in geovisualization design: Investigating multiple methods through a long-term case study. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2498–2507, Dec 2011.

[90] S. Lohmann, S. Negru, F. Haag, and T. Ertl. Visualizing ontologies with VOWL. *Semantic Web*, 7(4):399–419, 2016.

[91] D. Lordick. Intuitive Design and Meshing of Non-Developable Ruled Surfaces. In *Proceedings of the Design Modelling Symposium*, pages 5–7, 2009.

[92] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, Apr. 1986.

[93] R. Möller, V. Haarslev, and B. Neumann. Semantics-based information retrieval. In *In Int. Conf. on Information Technology and Knowledge Systems*. Springer Verlag, 1998.

[94] C. M. Mooney. Age in the development of closure ability in children. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 11(4):219, 1957.

[95] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. V. den Bussche. The open provenance model core specification (v1.1). *Future Generation Computer Systems*, 27(6):743 – 756, 2011.

[96] T. Munzner. A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics*, 15(6), 2009.

[97] C. Muth and C.-C. Carbon. Seins: Semantic instability in art. *Art & Perception*, 4(1-2):145–184, 2016.

[98] K. Nazemi and J. Kohlhammer. Visual variables in adaptive visualizations. In *UMAP Workshops*, 2013.

[99] K. Nazemi, A. Kuijper, M. Hutter, J. Kohlhammer, and D. W. Fellner. Measuring context relevance for adaptive semantics visualizations. In *Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business*, i-KNOW '14, pages 14:1–14:8, New York, NY, USA, 2014. ACM.

[100] S. Negru, F. Haag, and S. Lohmann. Towards a unified visual notation for OWL ontologies. *Proceedings of the 9th International Conference on Semantic Systems - I-SEMANTICS '13*, page 73, 2013.

[101] Neo4j. Neo4j - the world's leading graph database, 2012.

[102] C. North. Toward measuring visualization insight. *IEEE computer graphics and applications*, 26(3):6–9, 2006.

[103] B. Odehnal. Subdivision Algorithms for Ruled Surfaces. 2008.

[104] K. Otto and H. Schumann. An information-model for presentation generation. 08 1998.

[105] B. Parsia and E. Sirin. Pellet: An owl dl reasoner. In *Third international semantic web conference-poster*, volume 18, page 2. Publishing, 2004.

[106] M. Peternell, H. Pottmann, and B. Ravani. On the computational geometry of ruled surfaces. *Computer-Aided Design*, 31(1):17 – 32, 1999.

[107] M. Petre and T. Green. Learning to read graphics: Some evidence that 'seeing' an information display is an acquired skill. *Journal of Visual Languages & Computing*, 4(1):55 – 70, 1993.

[108] P. Pirolli and S. Card. Information foraging in information access environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 51–58. ACM Press/Addison-Wesley Publishing Co., 1995.

[109] P. Pirolli and S. Card. Information foraging. *Psychological review*, 106(4):643, 1999.

[110] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the working conference on Advanced visual interfaces*, pages 109–116. ACM, 2004.

[111] H. Pottmann, M. Peternell, and B. Ravani. An introduction to line geometry with applications. *Computer-Aided Design*, 31(1):3–16, 1999.

[112] A. J. Pretorius and J. J. Van Wijk. What does the user want to see? what do the data want to be? *Information Visualization*, 8(3):153–166, 2009.

[113] F. Pérez and J. Suárez. Quasi-developable B-spline surfaces in ship hull design. *Computer-Aided Design*, 39(10):853–862, 2007.

[114] S. M. Rappaport and Y. Rabin. Differential geometry of polymer models: worm-like chains, ribbons and Fourier knots. *Journal of Physics A: Mathematical and Theoretical*, 40(17):4455, 2007.

[115] R. M. Ratwani, J. G. Trafton, and D. A. Boehm-Davis. Thinking graphically: Extracting local and global information. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 25, 2003.

[116] R. M. Ratwani, J. G. Trafton, and D. A. Boehm-Davis. From specific information extraction to inferences: A hierarchical framework of graph comprehension. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 48(16):1808–1812, 2004.

[117] R. A. Rensink. The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42, 2000.

[118] J. C. Roberts, C. Headleand, and P. D. Ritsos. Sketching designs using the five design-sheet methodology. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):419–428, Jan 2016.

[119] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. What you see is what you can change : Human-centered machine learning by interactive visualization. *Neurocomputing*, 268:164–175, 2017.

[120] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE transactions on visualization and computer graphics*, 20(12):1604–1613, 2014.

[121] M. Sadowsky. Ein elementarer Beweis für die Existenz eines abwickelbaren Möbiusschen Bandes und Zurückführung des geometrischen Problems auf ein Variationsproblem. *Sitzungsberichte Akad. Berlin 1930*, pages 412–415, 1930.

[122] B. Saket, A. Endert, and Ç. Demiralp. Data and task based effectiveness of basic visualizations. *CoRR*, abs/1709.08546, 2017.

[123] M. Scaife and Y. Rogers. External cognition: how do graphical representations work? *International journal of human-computer studies*, 45(2):185–213, 1996.

[124] D. A. Schön. *The Reflective Practitioner How Professionals Think in Action*. 1983.

[125] H. J. Schulz, M. Angelini, G. Santucci, and H. Schumann. An enhanced visualization process model for incremental visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(7):1830–1842, July 2016.

[126] H.-J. Schulz, T. Nocke, M. Heitzler, and H. Schumann. A design space of visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2366–2375, 2013.

[127] K. Schwaber and J. Sutherland. The Scrum guide, 2001.

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

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

[130] S. J. Simoff. *Form-Semantics-Function – A Framework for Designing Visual Data Representations for Visual Data Mining*, pages 30–45. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[131] S. Simon, S. Mittelstädt, D. A. Keim, and M. Sedlmair. Bridging the Gap of Domain and Visualization Experts with a Liaison. In E. Bertini, J. Kennedy, and E. Puppo, editors, *Eurographics Conference on Visualization (EuroVis) - Short Papers*. The Eurographics Association, 2015.

[132] M. Smuc, E. Mayr, T. Lammarsch, A. Bertone, W. Aigner, H. Risku, and S. Miksch. Visualizations at first sight: Do insights require training? In A. Holzinger, editor, *HCI and Usability for Education and Work*, pages 261–280, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[133] A. Starič, J. Demšar, and B. Zupan. Concurrent software architectures for exploratory data analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(4):165–180, 7 2015.

[134] E. Starostin and G. van der Heijden. The shape of a Möbius strip. *Nature Materials*, 6:563–567, 2007.

[135] M. Streit, H.-J. Schulz, A. Lex, D. Schmalstieg, and H. Schumann. Model-driven design for the visual analysis of heterogeneous data. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):998–1010, June 2012.

[136] K. Thellmann, M. Galkin, F. Orlandi, and S. Auer. *LinkDaViz – Automatic Binding of Linked Data to Visualizations*. 2015.

[137] C. Tominski. Event-based concepts for user-driven visualization. *Information Visualization*, 10(1):65–81, 2011.

[138] M. Tory and T. Möller. Evaluating visualizations: do expert reviews work? *IEEE Computer Graphics and Applications*, 25(5):8–11, Sept 2005.

[139] J. G. Trafton, S. Marshall, F. Mintz, and S. B. Trickett. Extracting explicit and implict information from complex visualizations. In *International Conference on Theory and Application of Diagrams*, pages 206–220. Springer, 2002.

[140] J. G. Trafton and S. B. Trickett. A new model of graph and visualization usage. Technical report, NAVAL RESEARCH LAB WASHINGTON DC, 2001.

[141] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, second edition, 2001.

[142] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1936.

[143] J. J. Van Wijk. Views on visualization. *IEEE transactions on visualization and computer graphics*, 12(4):421–432, 2006.

[144] P. Vickers, J. Faith, and B. N. Rossiter. Understanding visualization: A formal approach using category theory and semiotics. *CoRR*, 2013.

[145] T. Vigen. *Spurious Correlations*, volume ISBN-10: 0316339431. 2015.

[146] R. Walker, A. Slingsby, J. Dykes, K. Xu, J. Wood, P. H. Nguyen, D. Stephens, B. L. W. Wong, and Y. Zheng. An extensible framework for provenance in human terrain visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2139–2148, Dec 2013.

[147] X.-M. Wang, T.-Y. Zhang, Y.-X. Ma, J. Xia, and W. Chen. A survey of visual analytic pipelines. *Journal of Computer Science and Technology*, 31(4):787–804, July 2016.

[148] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[149] C. Ware, N. J. Pioch, and E. K. Jones. Visual thinking algorithms for visualization of social media memes, topics, and communities. 2013.

[150] L. Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag, Berlin, Heidelberg, 2005.

[151] G. Wills and L. Wilkinson. Autovis: Automatic visualization. *Information Visualization*, 9(1):47–69, 2010.

[152] W. Wunderlich. Über ein abwickelbares Möbiusband. *Monatshefte für Mathematik*, 66:276–289, 1962.