Scalable Visual Analytics Solutions and Techniques for Business Applications

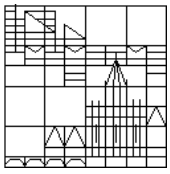
Dissertation zur Erlangung des akademischen Grades des Doktors der Naturwissenschaften an der Universität Konstanz im Fachbereich Informatik und Informationswissenschaft

vorgelegt von

Jörn Schneidewind

Tag der mündlichen Prüfung: 15. Juni 2007

Referent: Prof. Dr. Daniel A. Keim, Universität Konstanz
Referent: Prof. Dr. Robert Spence, Imperial College London



Konstanz, August 2007

II

Parts of this thesis were published in [SMK07, SSK06, SSKS06, KSHD06]

Abstract

The information overload is a well-known phenomenon of the information age, since due to the progress in computer power and storage capacity over the last decades, data is produced at an incredible rate, and our ability to collect and store this data is increasing at a faster rate than our ability to analyze it. This gap leads to new challenges in the analysis process since analysts and decision makers rely on the information hidden within the data. More than ever before, organizations in commercial, government, university and research sectors are tasked with making sense of huge amounts of data.

But because of the complexity and volume of today's data sets, extracting the valuable information hidden within data is a difficult task. New methods are needed to allow the analyst to examine these massive, multi-dimensional information sources to enable them to make effective decisions. To face this challenge, the field of Visual Analytics aims at the integration of data mining technology and information visualization and thus combines two powerful information processing systems the human mind and the modern computer [GEC98].

The basic idea is to aid the capabilities of the human mind by computer systems, which extract and compile information from heterogeneous sources and present them in an appropriate way to the user. The user may then use his flexibility, creativity, and general knowledge to steer the exploration process and to extract relevant patterns. In this context this thesis provides novel scalable analysis techniques that follow the Visual Analytics Mantra in terms of handling massive, heterogeneous volumes of information by integrating human judgment by means of visual representations and interaction techniques in the analysis process.

Novel analysis techniques for a number of analysis tasks are presented, that take the special properties of hierarchical-, time-related and geo-related datasets into account. Application examples from a number of scenarios are presented that show how these techniques are successfully applied in business scenarios, including sla-, business process- and financial analysis. Furthermore the concept of relevance driven Visual Analytics is introduced and based on this concept a visualization process model is provided and evaluated that combines automated analysis and image analysis techniques in order to support the user in creating insightful visualizations.

Experimental results are presented, that show that this concept can improve

the visualization process in terms of scalability and is therefore expected to be useful in many application domains.

Summary (in German)

Das Informationszeitalter ist dadurch gekennzeichnet, dass die Menge verfügbarer Informationen explosionsartig ansteigt. Dies liegt in erster Linie am technologischen Fortschritt, wodurch die Leistungsfähigkeit von Computern stetig wächst und damit die Speicherung und die Verwaltung sehr großer Informationsmengen möglich sind. Ein weiterer Grund für die ständig steigende Informationsflut sind Kommunikationsnetzwerke wie etwa das Internet, die den Zugriff auf riesige Mengen von Informationen ermöglichen.

In Anbetracht dieser Informationsflut wird es zunehmend wichtiger relevante von nicht-relevanten Informationen zu filtern, denn die richtige Information zur richtigen Zeit am richtigen Ort zu bekommen, ist für Analysten, Führungskräfte oder Sicherheitsbehörden essenziell. Das Bereitstellen dieser Informationen aus der Flut von Daten, d.h. das enthaltene Wissen transparent und nutzbar zu machen, stellt jedoch eine große Herausforderung an viele klassische Data Mining Bereiche dar. Wie sollen genau die Informationen herausgefiltert werden, die wirklich wichtig sind, um präzise und schnelle Entscheidungen treffen zu können? Wie können diese Informationen dem Benutzer dann effektiv präsentiert werden?

Neuartige Techniken zur Datenanalyse und zur Datenvisualisierung sind nötig, um auf diese Herausforderungen zu reagieren. Eine Antwort auf dieses Problem bietet *Visual Analytics*. Ihre Methoden unterstützen uns dabei, alle relevanten Daten schnell zu erfassen, indem sie die besondere Fähigkeit des menschlichen Gehirns nutzen, Regelmäßigkeiten zu erkennen. Sie präsentiert uns die vorhandenen Daten in einer Form, in der wir Muster leicht erkennen können. Durch die visuelle Darstellung der Daten macht *Visual Analytics* Zusammenhänge deutlich, die uns sonst verborgen blieben. Aufgrund dieser Darstellung können wir Hypothesen aufstellen, Aussagen treffen und Antworten auf unsere Fragen finden. Die Werkzeuge von *Visual Analytics* können uns dabei unterstützen, sinnvolle und nachvollziehbare Entscheidungen zu treffen.

Die vorliegende Dissertation stellt zahlreiche innovative Methoden und Verfahren vor, die im Kontext von *Visual Analytics* entwickelt wurden, um relevante Informationen aus großen Datenmengen zu extrahieren und dem Benutzer visuell zu präsentieren. Dabei stehen sowohl die Berücksichtigung unterschiedlicher Datentypen, wie etwa Daten mit explizitem Raum- oder Zeitbezug, als auch die Benutzerinteraktion im Vordergrund.

Die einzelnen Verfahren werden anhand von echten Daten, die zumeist aus Geschäftsdaten bestehen, evaluiert um die Vorteile gegenüber existierender Verfahren aufzuzeigen. Als einer der Hauptbeiträge der Arbeit wird in diesem Rahmen ein Verfahren vorgestellt, welches durch eine kombinierte visuelle- und datenbasierte Analyse eine effektivere und damit zielgerichtete Exploration relevanter Informationen ermöglicht. Dieses Verfahren wird detailliert vorgestellt und anhand realer Anwendungen evaluiert.

Acknowledgements

I would like to thank all the people who supported me during the past years while I have been working on my PhD studies.

First of all, my sincere thanks to my supervisor, Prof. Dr. Daniel A. Keim, who introduced me to the exiting world of research in the fields of Visualization, Data Mining and Visual Analytics. His long standing research experiences, his creative ideas and his great support not only made this work possible, but also guided me in acquiring the research skills that are necessary to work successfully on scientific projects, to publish research results and to present them at international conferences. This had given me the opportunity to meet many senior researchers and to discuss interesting research issues with them, which highly influenced this work.

My thanks go also to Prof. Dr. Robert Spence for his interest in my work and his willing to act as the second referee. Discussions with him were always exciting, and really helped me to improve the thesis.

Special thanks to my colleague and valuable friend Dr. Mike Sips, with whom I worked on many successful research projects in the field of Visual Analytics. Our constructive and productive discussions as well as his great support highly contributed to this thesis. I also enjoyed working with Dr. Christian Panse on a number of very interesting research projects.

Thanks to my colleagues from the DBVIS group at the University of Konstanz, who provided me an inspiring and supportive working environment. Their cooperation and constructive comments positively influenced the quality of this thesis. Special thanks to Dr. Tobias Schreck and Florian Mansmann, I had the great pleasure to work together with them in various research topics successfully.

To develop the techniques proposed in this thesis, a large amount of implementation, data preprocessing, and testing were necessary. I thank the students who helped me to manage the various tasks, in particular, Helmut Barro, Jakob Haddick, Henrico Dolfing and Cordula Bauer.

Thanks go also to Ming C. Hao and Umeshwar Dayal from Hewlett Packard Research Labs, Palo Alto, U.S., who gave me the opportunity to work on challenging projects at HP Labs in the fields of Business Analytics and Visualization. I enjoyed working with them, and many ideas from our cooperation are incorporated in this thesis.

Last but not least, I would like to thank my family. My parents who always supported me, my sister for carefully proof-reading this thesis and of course my wife Nicole for her great patience, understanding and her encouragement.

Contents

List of Figures

Part I

Preliminaries

Chapter 1

Introduction

Due to the progress in computer power and storage capacity over the last decade, today's scientific and commercial applications are capable of generating, storing, and processing massive amounts of data. Enterprises and institutions typically spend enormous amounts of money on large scale database management systems and Data Warehouses to store and access their data efficently. Many of these database management systems handle extraordinarily vast data sets of multiple terabytes in size. The Winter Top Ten Program 2005 [Cor05] for example, a worldwide survey that identified the world's largest and most heavily used databases in 2005, established new milestones in database scalability. The survey found out that the validated size of the largest commercial database increased three-fold since the 2003 program, topping the 100 terabyte (TB) mark. The number of database rows/records also rose markedly in the past two years. The 2005 leading result is 2.8 trillion rows (in a Sprint Nextel database), a five-fold increase since 2003. Among the largest commercial databases that Winter found are databases operated by Yahoo! (100 TB), AT&T (93 TB) and Amazon (24 TB). Beyond business, the biggest database that the survey revealed was a 222 TB database operated by the Max Planck Institute for Meteorology. And the information explosion is not limited to a few commercial or scientific applications. Almost all transactions of everyday life such as ATM transactions, credit card purchases or telephone calls are logged by IT infrastructure. And of course the internet, the youngest and fastest growing media in today's world, provides a huge source of web-accessible information, including webpages, web-connected databases or intranets.

Researchers from Berkeley estimate that in 2002 alone about 5 exabytes (5 billion gigabytes) of new information were stored on magnetic storages (hard discs, tapes) [LV06]. This corresponds to almost 800 MB of recorded information per person worldwide each year. Another 18 exabytes of streaming information was produced in 2002 and the study estimates that storage of new information is growing at a rate of 30 % per year. This growth of available information has exploded largely because information can be stored inexpensively since the cost of magnetic storage is dropping rapidly; as of Fall 2005 a gigabyte of storage costs less than $1

and it is predicted that this cost will drop further in the future.

Keeping our networked, digital world running depends on accessing and managing these massive amounts of data that are multiplying and growing dramatically every year. In this reality, the central issue in data analysis has shifted from getting data to making sense of it.

According to a study from M.Lesk [Les97], in the future we will be able to save everything, no information will have to be thrown out, but the typical piece of information will never be looked at by a human being because of the rapidly widening gap between the amount of available data and the amount of attention available to process it. So without effective ways for data analysis, we are drowning in data and dying for information. Thus input will not matter as much as sophisticated analysis and relevant choice since it is critical for analysts and decision makers to have the information they need at their fingertips.

Therefore the natural question then becomes: How can relevant information be extracted from these massive and complex datasets? For most data warehouse environments standard reporting tools are extremely beneficial to individuals who need to easily obtain timely and accurate information from large data sets for decision-making purposes.

Accessing information from these reporting environments often range from ad hoc queries, e.g. by using query languages like SQL, to multidimensional analysis, like provided by OLAP tools. While these forms of data analysis are excellent at answering the questions posed by the information consumer, they do not provide any other insight and are very limited in detecting interesting patterns in the data that are beyond standard queries. Furthermore, users often do not know the data and its distribution, they are often not even exactly sure what information they have. In these situations standard query tools are improper since they provide no information about interesting data relationships and potentially interesting patterns in the data. But it is often exactly this kind of information that analyst try to find since they are then able to identify trends within the data that they did not know existed which may give them a competitive advantage in business life.

Examples are large stores and supermarkets which hold huge databases on customer purchases, initially collected for inventory and financial recording purposes. Analysts may be interested in using information on customer purchasing patterns to increase sales. Insurance companies have huge databases of information on insurance claims, which can be used to adjust estimates of risk, identify fraud, inappropriate treatment or over-treatment, and to detect trends which may lead to an escalation of medical costs. Telephone and banking companies can use their transaction records to analyze customer habits to adapt their marketing strategies accordingly. Companies and organizations have recognized this additional value that lies within the vast amounts of data they are storing.

During the last decades, seeking knowledge from massive data sets has therefore attracted significant commercial and scientific research in several fields, including machine learning, statistics, databases, and data visualization. The latter

provides an interface between two powerful information processing systems, the human mind and the modern computer system, and is in the focus of this thesis. Visualization is the process of transforming data, information, and knowledge into visual form making use of the humans natural visual capabilities [GEC98]. With effective visual interfaces we can interact with large volumes of data rapidly and effectively to discover hidden characteristics, patterns, and trends. Research and development in visualization has fundamentally changed the way we present and understand large complex data sets. The widespread use of visualization has led to new insights and more efficient decision making in many application scenarios. But the tremendous growth of the internet, the overall computerization of the business and defence sectors, and the deployment of data warehouses pose new challenges on Visualization techniques. In the future we will delve deeper into the information age, therefore making sense of even more complex and larger amounts of data becomes critical. In order to respond to this challenge, scalable novel approaches and new visual metaphors are needed, which is the topic of the next chapters.

1.1 Modern Data Analysis

Mining information and interesting knowledge from large databases has been recognized by many researchers as a key research topic and has attracted substantial industry attention as well. The information hidden in large enterprise databases is often of strategic and financial significance for many industrial companies and their extraction is an important area with an opportunity of major revenues: Businesses use data to gain competitive advantage, increase efficiency, and provide more valuable services to customers. Analyzing the huge amount of data obtained from large databases such as credit card payments, telephone calls, environmental records, and census demographics, is however a very difficult task. In fact, as data volumes grow dramatically, manual data analysis and interpretation is becoming completely impractical in most domains. Instead analysts must be supported by sophisticated, scalable, (semi) automated analysis methods which effectively extract and present potentially useful patterns.

Because computers have enabled humans to gather more data than we can digest, it is only natural to turn to computational techniques to help us unearth meaningful patterns and structures from the massive volumes of data [FS97]. One common way to reach this goal is to move more responsibility away from the user towards the computer by employing automated analysis techniques. For that reason the interdisciplinary field of *Knowledge Discovery in Databases (KDD)* which brings together techniques from various areas such as machine learning, pattern recognition, databases, statistics, AI, knowledge acquisition for expert systems, data visualization, or high-performance computing has attracted much research attention to address the issues of analyzing such huge data sets and extracting knowledge from them. According to Fayyad et al.[FPSS96] *KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable*

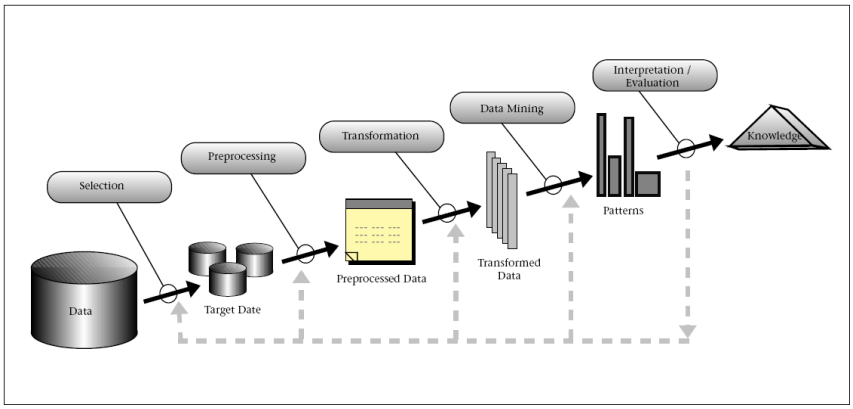


Figure 1.1: Overview of the basic flow of steps that compose the KDD process [FPSS96].

patterns in data.

Figure 1.1 gives an overview of the KDD process which is interactive and iterative, involving numerous steps with many decisions made by the user. The process model is described in detail in [FS97], here we broadly outline its basic steps:

1. **Define tasks/goals**
   First is developing an understanding of the application domain and the relevant prior knowledge and identifying the goal of the KDD process from the analyst's / customer's viewpoint.

2. **Selection**
   Creating a target data set, selecting a data set, or focusing on a subset of variables or data samples, on which discovery is to be performed.

3. **Data Cleaning and Preprocessing**
   Removing noise if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, and accounting for time-sequence information and known changes.

4. **Data Reduction and Projection**
   Finding useful features to represent the data depending on the goal of the task. With dimensionality reduction or transformation methods the effective number of variables under consideration can be reduced or invariant representations for the data can be found.

5. **Data Mining**
   Matching the goals of the KDD process (step 1) to a particular data-mining method, for example summarization, classification, regression, or clustering.

Applying the chosen algorithm to the transformed data set in order to receive a set of patterns extracted from the data.

6. **Evaluation**
   Interpreting mined patterns, possibly returning to any of the previous steps for further iteration. This step can also involve visualization of the extracted patterns and models or visualization of the data given the extracted models. Finally, acting on the discovered knowledge: using the knowledge directly, incorporating the knowledge into another system for further action, or simply documenting it and reporting it to interested parties.

The core step in the KDD process is the application of particular data-mining methods. As mentioned before, Data Mining is an interdisciplinary field, the confluence of a set of disciplines, including databases systems, statistics, machine learning, or visualization. Moreover, depending on the data mining approach used, techniques from other fields like neural networks, fuzzy set theory, or knowledge representation may be applied. Depending on the data to be mined, the data mining system may also integrate techniques from spatial data analysis, image analysis, information retrieval and the like [HK06]. Because of their large variety, it is important to provide a general classification of data mining systems. In [FS97] two goals of data mining are distinguished defined by the intended use of the system: (1) *verification* and (2) *discovery*. With *verification* the system is limited to verifying the users hypothesis. With *discovery* the system autonomously finds new patterns. The discovery goal is further subdivided into *prediction*, where the system finds patterns for predicting the future behavior of some entities, and *description*, where the system finds patterns that can then be presented to the user in an understandable and comprehensible form. In this thesis, we are primarily concerned with discovery-oriented methods. *Data Mining* involves the application of automated methods to fitting models to or determining patterns from observed data. Numerous data mining algorithms have been proposed in the literature, general overviews can be found in [HM01, HK06]. In [HK06] data mining algorithms are classified according to the following primary data mining methods:

- *Characterization and Discrimination*
  Summarization of general characteristics or features of a target class of data and comparision of the general features of that target class data objects with general features of objects from one or a set of contrasting classes.

- *Association Analysis*
  Discovery of association rules showing attribute-value conditions that occur frequently together in a given set of data.

- *Classification and Prediction*
  Process of finding a set of models that describe and distinguish data classes or concepts for the purpose of being able to use the model to predict the class of objects whose class label is unknown.

- *Cluster Analysis*
  Objects are clustered or grouped based on the principle of maximizing the intraclass similarity of objects and minimizing their interclass similarity.

- *Outlier Analysis*
  Finding data objects within a database that do not comply with the general behavior or model of the data.

- *Evolution Analysis*
  Describes and models regularities or trends for objects whose behavior changes over time.

Data mining approaches have been successfully applied in a number of domains, including financial data analysis (loan payment prediction, customer credit policy analysis), telecommunication industry (fraud detection, sequential pattern analysis) or biomedical analysis (association analysis in gene sequences). For a survey of industrial applications of data mining systems see [PSBK+96], for scientific data analysis and research see [FHS96].

1.2 The Need for Visual Data Analysis

For data mining to be effective, it is important to include the human in the data exploration process and combine the flexibility, creativity, and general knowledge of the human mind with the enormous storage capacity and the computational power of today's computers [Kei02]. User knowledge is essential to steer the data mining process or to evaluate and validate extracted patterns / knowledge since only the user can determine whether the resulting knowledge satisfies given requirements.

Therefore, instead of allowing an automated data mining process to iterate in a trial-and-error manner, a natural and effective way to enhance the process is to support human involvement [AEK00]. However, mainstream data mining techniques significantly limit the role of human reasoning and insight [FGW01]. Visual data exploration bridges this gap by integrating the human in the data exploration process, applying its perceptual abilities to the large data sets available in todays computer systems [Kei02].

The basic idea is to present the data in some visual form, allowing the human to gain insight into the data, draw conclusions, and directly interact with the data. The visual data exploration process can be seen as a hypothesis generation process: The visualization of the data allows the user to gain insight into the data and come up with new hypotheses. The verification of the hypotheses can also be done via visual data exploration but it may also be accomplished by automatic techniques. In addition to the direct involvement of the user, the main advantages of visual data exploration over automatic data mining techniques by statistics or machine learning are:

- Visual data exploration can easily deal with highly inhomogeneous and noisy data
- Visual data exploration is intuitive and requires no understanding of complex mathematical or statistical algorithms or parameters.
- Visualization can provide a qualitative overview of the data, allowing data phenomena to be isolated for further analysis

As a result, visual data exploration usually allows a faster data exploration and often provides better results, especially in cases where automatic algorithms fail. In addition, visual data exploration techniques provide a much higher degree of confidence in the findings of the exploration. This fact leads to a high demand for visual exploration techniques and makes them indispensable in conjunction with automatic exploration techniques.

1.3 Outline of the Thesis

This thesis provides novel contributions in the context of Visual Analytics. Since Visual Analytics aims at integrating analytical and Information Visualization techniques, the next chapter gives an overview on Information Visualization techniques and applications and describes how Visual Analytics has evolved from this research field.

Part II of the thesis then gives an introduction to Visual Analytics, explains it's scope and provides a formal definition of the Visual Analytics process. The research challenges in the emerging field of Visual Analytics are introduced, and their relevance to the field of Business Analysis is explained. The motivation for this thesis is based on these challenges. We focus on techniques which combine analytical and visualization techniques in the context of business applications, on one hand in order to increase the scalability of existing approaches and on the other hand to be able to analyze heterogeneous data sources.

Therefore, in Part III we explain the data model for integrating and managing heterogeneous data, and provide novel visual analysis techniques based on this model, which take the special properties of temporal, hierarchical and geo-spatial data into account. When using such techniques to analyze large as well as complex data sets, we identified the challenging task of supporting the user in creating insightful visualizations. In complex and heterogeneous data sets, as we have to deal with in Visual Analytics, it is not clear how to construct visual mappings from the data since the data sets may be too large to find such mappings manually.

In Part IV we therefore present techniques that support the user in constructing insightful visualization based on a combined analytical and visual analysis of the data and visualization space. We provide a formal framework for our approach and present applications that show the potential of the proposed technique.

The thesis closes with a summary of the proposed research contributions and a discussion of possible future research directions in Part V.

Chapter 2

Principles of Visual Data Exploration

Visual Data Exploration aims at the tight coupling of automated data mining techniques and visualization methods and thus combines two powerful information processing systems: the human mind and the modern computer. According to Keim [KAS04, Kei02], Visual Data Exploration usually follows a three step process: *Overview first, Zoom and Filter, and then Details on Demand* , which is known as the Visual Information Seeking Mantra [Shn96].

- *Overview first:* When exploring large data sets, the analyst needs to get an overview of the data first. There he may identify interesting patterns or relevant parts of the data and focus on them.
- *Zoom and Filter:* For investigating the detected patterns the analyst focuses on one or more of them.
- *Details on Demand:* For analyzing the patterns the analyst needs to drill-down and access details of the data.

Effective and expressive visualization techniques play an essential role in this context, because only by employing appropriate visualization the analyst is able to steer the exploration process and to gain insight into the data. Visualization technology may be used for all three steps of the data exploration process: Visualization techniques are useful for giving an overview of the data, allowing the user to identify interesting subsets. In this process, it is important to keep the overview visualization while focusing on the subset using another visualization. As an alternative one can distort the overview visualization in order to focus on the interesting subsets. To further explore the interesting subsets, the user needs a drill-down capability in order to observe the details about the data. Note that visualization technology does not only provide visualization techniques for all three steps but also bridges the gaps between them.

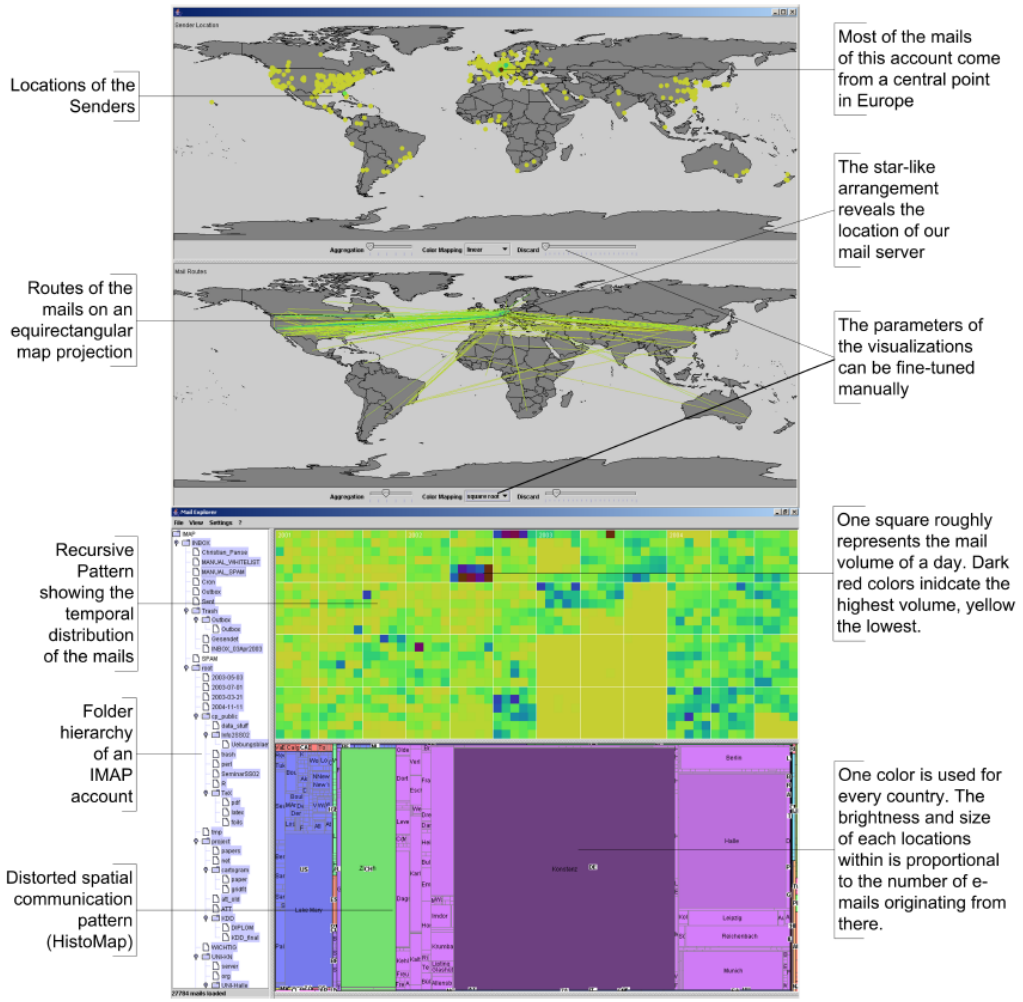


Figure 2.1: *Overview first / Zoom and Filter / Details on Demand*
The Mail Explorer Interface [KMP+05] provides a number of configurable linked views to explore characteristics of received emails. The user has four overview visualizations to analyze temporal and geo-spatial properties of personal mail. He can identify interesting patterns, here for example that the majority of all sender locations are in the US and Europe. This can be explained because of the fact that most of the research partners of the corresponding user are located in the US and Europe. Some e-mails with exotic sender locations (e.g. China) are likely to be spam. Interaction techniques are provided to zoom on relevant patterns and to select details on demand.

When talking about *Visualization* in this thesis we follow the definition given by Card, Mackinlay, and Shneiderman [CMS99]:

- *Visualization:* The use of computer-supported, interactive, visual representations of data to amplify cognition.

Whereas cognition is acquisition or use of knowledge. This definition covers Scientific Visualization as well as Information Visualization. In this thesis we focus on the latter. Scientific Visualization (SciVis) applies visualization to scientific data, typically physical data (the human body, the earth, molecules, or other) to enable scientists to perceive certain phenomena in the data. Information Visualization (InfoVis) on the other hand, focuses on visualizing non-physical, abstract data such as financial data, business information, document collections, and abstract conceptions.

- *Information Visualization:* The use of computer-supported, interactive, visual representations of abstract data to amplify cognition.

More precisely, Information Visualization is a process that transforms data, information, and knowledge into a form that relies on the human visual system to perceive its embedded information. It's goal is to enable the user to observe, to understand, and to make sense of the information [GP01]. This kind of information usually does not have any obvious spatial mapping, which leads to two problems: First of all, it is not clear how to render visible properties of the objects of interest. And secondly, there is the fundamental problem of mapping non-spatial abstractions into effective visual form, which provides the challenge for InfoVis techniques.

2.1 Information Visualization Classics

Although the field of Information Visualization is relatively young, the first IEEE Conference on Visualization took place in 1990, work in data graphics dates from about the time of William Playfair (1759-1823), a Scottish engineer who was one of the first who used abstract visual properties, such as line and area (bar and pie charts), to represent data visually [Pla86, CMS99]. Starting with Playfair, the classical methods of plotting data were developed and thus a number of well known historic application examples can be found in the literature.

In 1983 E. R. Tufte published a theory of data graphics [Tuf83], emphasizing maximizing the density of useful information. Together with Bertin's theory of graphics [Ber67], published in 1967, which identified the basic elements of diagrams and described a framework of their design, these theories became well known and highly influenced the development of Information Visualization as a discipline.

Tufte offers many models of expressive graphics which became *Classics* in InfoVis, like a reproduction of a chart done in 1861 by Charles Joseph Minard,

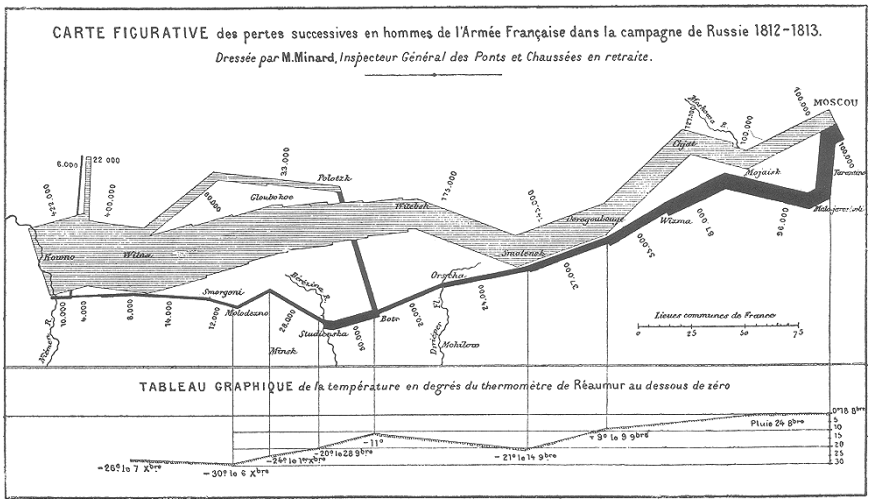


Figure 2.2: Napoleon's march to Moscow: Minard's map portrays the losses suffered by Napoleon's army in the Russian campaign of 1812. Beginning at the Polish-Russian border, the thick band shows the size of the army at each position. The path of Napoleon's retreat from Moscow in the bitterly cold winter is depicted by the dark lower band, which is tied to temperature and time scales [Tuf83]

showing Napoleon's fateful 1812 march to Moscow. Minard displays large amounts of information on one easy-to-read chart: the size of the army from the start of its march through its disastrous retreat, the route of the campaign, the distances traveled, the time frame, and the winter temperatures shown in Figure 2.2.

Tufte shows that, at their best, graphics can be critical to analyzing serious problems and presenting solutions to decision-making bodies. As a positive example, he presents a reprint of the map of London used by Dr. John Snow in 1854 to track down the cause of a cholera epidemic [Sno55, Tuf97]. Dr. Snow started by indicating the number and locations of fatalities on the map, as shown in Figure 2.3. It turned out that there was a high incidence of cases in Londons Broad Street. He investigated the fatalities on Broad Street and the fatalities outside of the Broad Street area, and he found that they both pointed to the Broad Street well. Dr. Snow interviewed Broad Street residents who did not become ill and he found that they did not use the nearby well. Neighborhood brewery workers shunned the water because they received free beer on the job. Inmates at a nearby work house suffered few fatalities because the institution had its own well. Snow presented his readily understandable chart and related information to the city board in charge of the water supply. They took immediate action to remove the source of the epidemic, the pump handle at the well and the epidemic disappeared.

In contrast, Tufte also showed that poor graphical presentation may lead to

Figure 2.3: Dr. Snow's map of the cholera epidemic, for which he is most famous in epidemiology [Sno55]. The map shows that most of the deaths due to cholera clustered around the Broad Street water pump. *From the Department of Epidemiology, UCLA, School of Public Health http://www.ph.ucla.edu/epi/snow.html*, Used by permission of Ralph R. Frerichs

wrong conclusions and false decisions. As an example, he presents the Challenger space shuttle disaster from January 1986. Shortly after the space shuttle lifted off, the Challenger exploded, killing the entire crew. The disaster happened because due to the cold January temperatures the rockets O-rings failed, leading to an explosion of rocket fuel. It turned out that the day before the disaster, engineers at rocket-maker Morton Thiokol were convinced that the space shuttle flight scheduled for the following morning should be delayed, since based on observations from previous flights, they feared that cold temperatures could cause O-ring failures. However, as Tufte demonstrates, they failed to present their data in a meaningful / comprehensible way to the decision makers. They included irrelevant data and used confusing labels, as shown in Figure 2.4. They did not focus on the critical factor of temperature and how it had affected O-rings on previous flights. The en-

gineers' flawed presentation failed to persuade the decision-makers. Tufte instead provided a scatterplot mapping temperature to the x-axis and the O-ring damage index on the y-axis, that clearly shows the risk of O-ring failures in cold weather, which might have prevented the launch of the shuttle. As a consequence he points out, how critical the visual factor is for comprehending data.

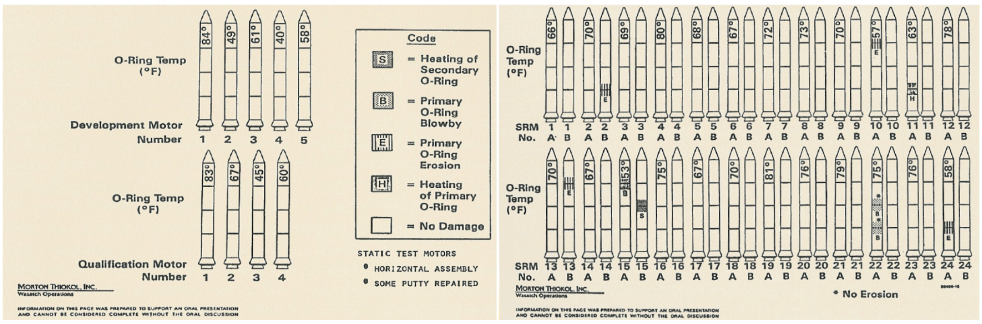


Figure 2.4: Two of the charts used by the engineers before the Challenger disaster. The graphs display tiny pictures of each shuttle booster, lined up in chronological order, showing launch temperatures and any O-ring damage. The most important facts (O-ring damage and temperatures) are buried in a mound of other irrelevant details, thus the chart fails to communicate the link between cool temperature and O-ring damage. [Com86]

More of these classical applications of InfoVis techniques can be found in [Tuf83, Tuf90, Spe01, Spe06]. An overview on origins and milestones of Information Visualization can be found in [CMS99].

2.2 Classification of Information Visualization Techniques

Today there exist a number of well known techniques for visualizing abstract data sets, such as x-y plots, line plots, and histograms. Many of these techniques were developed in a statistical context [Tuk77, Cle93]. These techniques are useful for data exploration, but are limited to relatively small and low dimensional data sets. Caused by the new challenges in analyzing very large and high-dimensional data sets, a large number of novel Information Visualization techniques have been developed over the last years, allowing visualizations of multidimensional data sets without inherent two- or three-dimensional semantics. Good overviews of the approaches can be found in a number of recent books [CMS99] [KK93] [Spe01] [Spe06] [War00].

According to Keim [Kei02, Kei01], these techniques can be classified based on three criteria:

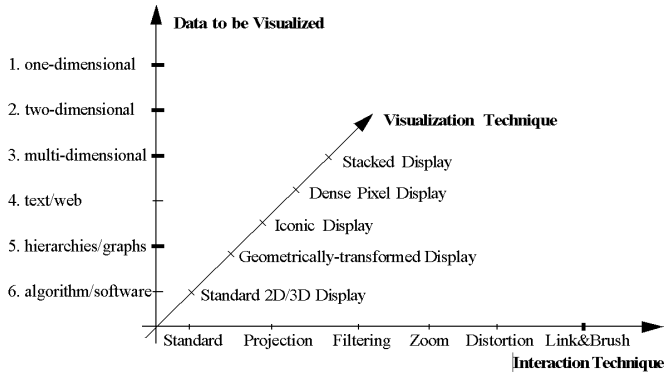


Figure 2.5: Classification of visual data exploration techniques [KW02]

- The data type to be visualized
- The visualization technique used
- The interaction technique used

In the following, we give a brief overview on these criteria; more details can be found in [KW02].

2.2.1 Data Type to be visualized

The data to be visualized usually consists of a large number of records, each consisting of a number of variables or dimensions. Each record corresponds to an observation, measurement, or transaction. The number of attributes is called dimensionality and can differ from one data set to the other. Data sets may be classified as:

- *one-dimensional data*, such as temporal (time-series) data,
- *two-dimensional data*, such as geographical maps,
- *multi-dimensional data*, such as relational tables,
- *text and hypertext*, such as news articles and web documents,
- *hierarchies and graphs*, such as telephone calls,
- *algorithms and software.*

A distinction may also be made between dense dimensions and dimensions that may have arbitrary values. Depending on the number of dimensions with arbitrary values, data are sometimes also called univariate, bivariate or mulivariate [KW02].

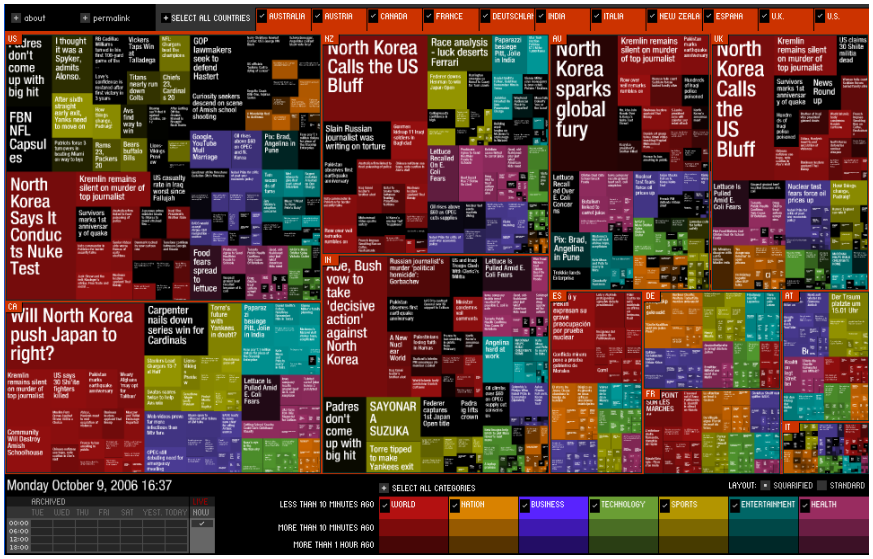


Figure 2.6: Newsmap: Visual reflection of the constantly changing landscape of the Google News using a treemap visualization algorithm. Color is used to indicate news categories. Screenshot was taken from http://www.marumushi.com/apps/newsmap/ on 9th October 06, one day after North Korea's first nuclear test, which is a major topic in the news.

2.2.2 Visualization Techniques

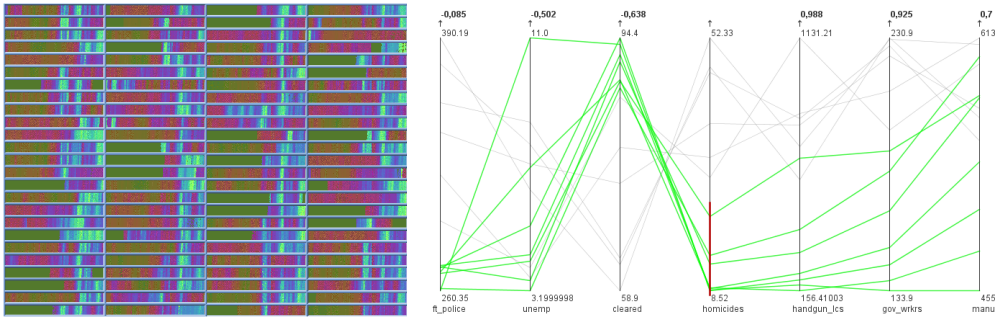
There are a large number of visualization techniques that can be used for visualizing data. In addition to standard plots, there are a number of more sophisticated classes of visualization techniques which can be classified as follows:

- *Standard 2D/3D displays*
  Standard plots such as charts, x-y plots, histograms or maps

- *Geometrically transformed displays*
  Techniques that aim at finding appropriate transformations of multi-dimensional data sets. Examples include scatter plot matrices, Hyperslice [vWvL93] or the well known Parallel Coordinates (Figure 2.7(b)) [ID90]

- *Icon-based displays*
  Techniques that map the attribute values of multi-dimensional data items to features of an icon. Examples are Chernoff faces [Che73] and stick figures [Pic70] [PG88]

- *Dense pixel displays*
  Class of techniques that map each dimension value to a colored pixel and group pixels belonging to each dimension into adjacent areas [Kei00].



(a) Recursive Pattern Technique showing 50 stocks in the Frankfurt Allgemeine Zeitung (Frankfurt Stock Index Jan 1975 - April 1995). This technique maps each stock value to a colored pixel; high values correspond to bright colors. ©IEEE *Used by permission of Daniel Keim*

(b) Parallel Coordinate Plot analyzing the relationship between the annual number of homicides in Detroit (1961-73) and possible impact factors like the number of handgun licenses per 100,000 population (handgun_lcs) or the percentage of homicides cleared by arrests (cleared) [Fis76]. Typical interaction techniques are provided to allow exploratory data analysis, e.g. to select subsets of the data (green lines).

Figure 2.7: Examples for a dense pixel 2.7(a) and geometrical transformed 2.7(b) display

Examples are the Recursive Pattern (Figure 2.7(a)) [AKK95] and Circle Segments [AKK96]

- *Stacked displays*
  Stacked displays are tailored to present data partitioned in a hierarchical fashion. Examples are treemaps (Figure 2.6) [JS91a] [Shn92] and dimensional stacking [War94].

Note that these classes correspond to basic visualization principles that may be combined in order to implement a specific visualization.

2.2.3 Interaction Techniques

For Visualization techniques to be effective, it is important to integrate Interaction techniques. Interaction techniques allow users to directly navigate and modify the visualizations according to the exploration objectives, to look at the same data from different perspectives, as well as to select subsets of the data for further operations. In addition they make it possible to link and combine multiple independent visualizations. Interaction techniques can be categorized based on the effects they have on the display, in particular *Navigation, View enhancement, or Selection*. Examples are:

- *Dynamic Projection*
  Automated navigation operation to dynamically change the projection in

order to explore a multi-dimensional data set. An example for a system that supports dynamic projection is XGobi [SCB98]

- *Interactive Filtering*
  Combination of Selection and View enhancement to interactively partition the data into segments (e.g. via Browsing or Querying) and focus on interesting subsets. Systems that support interactive filtering are Magic Lens or Polaris [STH02].

- *Interactive Zooming*
  View modification technique that allows the display of more details for data subsets of interest while showing irrelevant data at lower resolution. Application examples are the Table Lens [PR96] or Data Space [ADLP95].

- *Interactive Distortion*
  View modification technique that preserves a data overview during data drill down operations. Portions of the data are shown with a high level of detail while others are shown at lower detail level. Examples of distortion techniques are the Bifocal Display [SA81] or Perspective Wall [MRC91]. An overview on distortion techniques can be found in [LA94].

- *Interactive Linking and Brushing*
  Brushing is an interactive selection process that is often combined with linking, a process for communicating the selected (brushed) data to other views. This allows the combination of different visualization techniques to overcome the shortcomings of individual techniques. Tools that support Linking and Brushing are XGobi [SCB98] or the XmdvTool [War94].

Figure 2.5 shows the proposed classification schema. Note that the three dimensions of our classification - data type to be visualized, visualization technique, and interaction technique - can be assumed to be orthogonal. Orthogonality means that any of the visualization techniques may be used in conjunction with any of the interaction techniques for any data type. Note also that a specific system may be designed to support different data types and that it may use a combination of visualization and interaction techniques [KW02].

2.3 Visual Data Exploration Methodology

Since Visual Data Exploration aims at the integration of automated data mining and visualization techniques, interaction is not limited to the visualization techniques, it is important for the whole exploration process. In the literature three common approaches have been proposed on how the human should be integrated in Visual Data Exploration, shown in Figure 2.8. Here we give a brief description, for details we refer to [KAS04, Sip06].

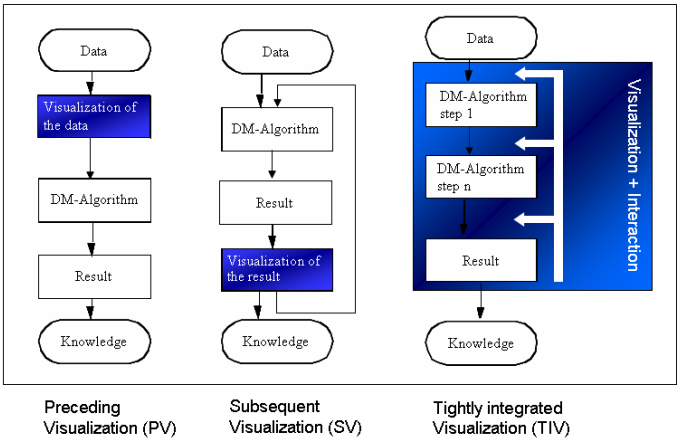


Figure 2.8: Common approaches of human involvement in the Visual Data Exploration process [KAS04]

2.3.1 Preceding Visualization

In this method data is initially visualized and presented to the user. By interaction with the data, e.g. by using interaction methods as provided in the last section, the user may then select subsets of the data for further automated analysis, or define parameters of the automated data mining step based on the observations drawn from the visualization. Then the automated data mining algorithm is started, which finally generates the results of the requested analysis.

2.3.2 Subsequent Visualization

An automated data mining algorithm initially performs the data mining task, the results of this step are presented in visual form. The analyst may then interpret the patterns, adjust the parameters for the data mining algorithm based on the observations from the visualization step, and rerun the automated step.

2.3.3 Tighly Integrated Visualization

Tighly integrated visualization couples automated methods and human interaction as shown in Figure 2.8. An automated data mining algorithm performs the data analysis, but does not produce the final analysis results. Visualization techniques are used to present intermediate results to the user, who is then able to specify user feedback for the next data mining run in form of interaction (parameter justification, selection, filtering,) based on his domain knowledge and his visual capabilities, to steer the automated analysis. Tighly integrated visualization is able to lead to a better understanding of the extracted patterns, since it makes maximum use of automated methods and human problem solving capabilities.

2.4 From Visual Data Exploration to Visual Analytics

Visual Data Exploration techniques have proven to be an important instrument in the exploration of large databases. Today, Visual Data Exploration plays an essential role in many application domains and Information Visualization has evolved into a recognized research field involving user interface and application design. But the information explosion poses a challenge for current techniques in terms of volume and complexity of available data sets. Even visualization techniques which were designed to handle data sets that were considered extraordinarily large some years ago, like treemap approaches as shown in Figure 2.9 or dense pixel display like the *VisDB* system [KK94], which are able to handle hundreds of thousands of items, are limited by the quantity of objects that can be visualized on the available display area. This limit is reached when each single pixel on the display represents an object.

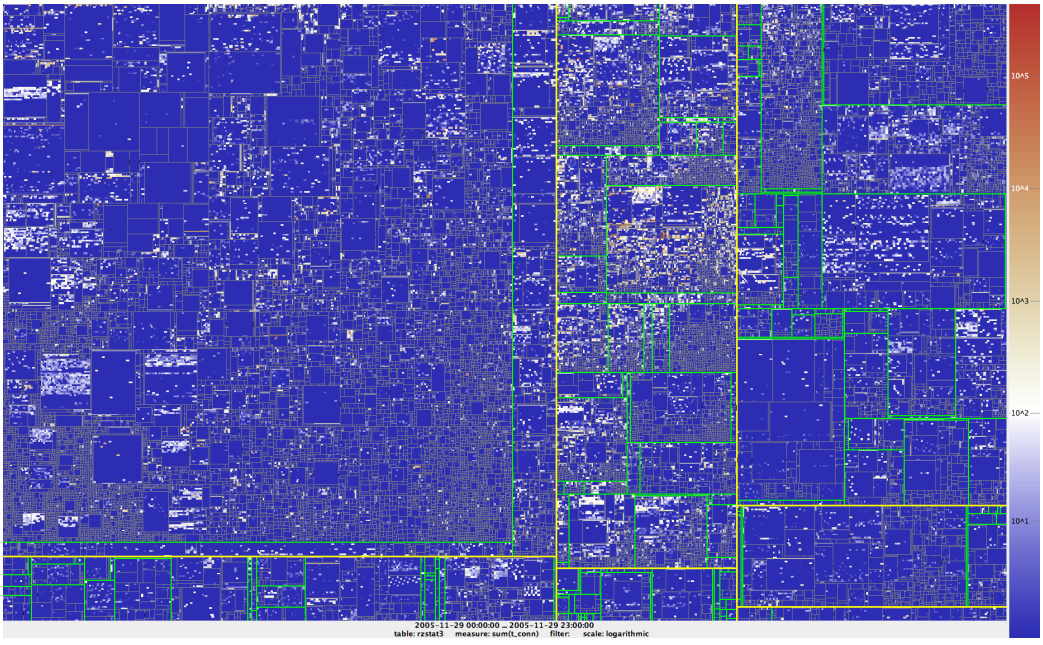


Figure 2.9: Interactive Treemap visualization of network traffic items: Treemap visualization showing (anonymized) outgoing network traffic connections from the gateway computer at the University of Konstanz to all 197427 IP prefixes at a single day (11/29/05). *Used by permission of Florian Mansmann*

Eick and Karr [EK02] proposed a scalability analysis and came to the conclusion that many visualization metaphors do not scale effectively, even for mo-

derately sized data sets. They list factors affecting visual scalability including human perception, monitor resolution, visual metaphors, interactivity, data structures or algorithms, and computational infrastructure. There is a need for novel visual metaphors that take the structure and volume of current data sets in to account.

Furthermore, in the age of massive data sets all three steps of the Information Seeking Mantra: *Overview First, Zoom and Filter, Details on Demand* are difficult to realize. An *Overview* visualization without losing any interesting pattern or obvious subset of the data is difficult to create, since the amount of pixels of modern data display systems to not keep step with the increasing flood of data, and thus aggregation or sampling techniques are necessary to generate visual data overviews. But this always implies the risk of loosing relevant information in the aggregation or sampling step. Additionally, the amount of information hidden in massive data sets and their complexity make it very difficult for the human to understand interesting relationships or detect them interactively via *Zoom and Filter* techniques. Besides that, interactive navigation in large data sets, typically gigabytes, pose a performance challenge on visualization methaphors and computer systems.

In the future, visualization must be tightly coupled with automated methods from the field of KDD, Statistics and Artificial Intelligence to provide effective tools to the analyst for analyzing even large scale data sets. The emerging field of Visual Analytics focuses on facing this challenge of handling these massive, heterogeneous, and dynamic volumes of information by integrating human judgment in the analysis process by means of visual representations and interaction techniques. It is the combination of related research areas including Visualization, Data Mining, and Statistics that turns Visual Analytics into a promising field of research. Thus, Visual Analytics extends the concepts of Visual Data Exploration to face the new challenges arising from the growing flood of information. This thesis focuses on Visual Analytics techniques that take the special structure and complexity of current data sets, especially in the area of business applications, into account. Novel approaches for a number of applications are presented and finally an approach for an automated support of Visual Analytics in large data set is introduced and evaluated.

Part II

Visual Analytics: Scope and Challenges

Chapter 3

Scope of Visual Analytics

3.1 Introduction

In today's applications data is produced at unprecedented rates. While the capacity to collect and store new data rapidly grows, the ability to analyze these data volumes increases at much lower rates. This gap leads to new challenges in the analysis process, since analysts, decision makers, engineers, or emergency response teams depend on the information hidden in the data. The emerging field of Visual Analytics focuses on facing this challenge of handling these massive, heterogeneous, and dynamic volumes of information by integrating human judgment in the analysis process by means of visual representations and interaction techniques. Furthermore, it is the combination of related research areas including Visualization, Data Mining, and Statistics that turns Visual Analytics into a promising field of research.

Today, a selected number of software tools are employed to help analysts to organize their information, generate overviews, and explore the information space in order to extract potentially useful information. Most of these data analysis systems still rely on interaction metaphors developed more than a decade ago and it is questionable whether they are able to meet the demands of the ever-increasing mass of information. In fact, huge investments in time and money are often lost because we still lack the technical feasibility to properly interact with the databases. To scale existing visual representations to meet the escalading data volumens, the state-of-the art in several major areas must be advanced. This includes visual representations of large data collections, support for multi-type information synthesis, or the support for visual exploration of high dimensional spaces.

Visual Analytics aims at bridging this gap by employing more intelligent means in the analysis process. The basic idea of Visual Analytics is to visually represent the information, allowing the human to directly interact with the information, to gain insight, to draw conclusions, and to ultimately make better decisions. The

visual representation of the information reduces complex cognitive work needed to perform certain tasks. People may use Visual Analytics tools and techniques to synthesize information and derive insight from massive, dynamic, and often conflicting data by providing timely, defensible, and understandable assessments.

Visual Analytics focuses on integrating new computational and theory-based tools with innovative interactive techniques and visual representations to enable human information discourse. The design of the tools and techniques is based on cognitive, design, and perceptual principles. After describing the scope of Visual Analytics in the next section, we give a formal description of the Visual Analytics process and introduce in the next chapter the research challenges of Visual Analytics that are in the scope of this thesis.

3.2 Scope of Visual Analytics

The goal of visual analytics research is to turn the information overload into an opportunity. Decision-makers should be enabled to examine this massive, multi-dimensional, multi-source, time-varying information stream to make effective decisions in time-critical situations. For informed decisions, it is indispensable to include humans in the data analysis process to combine their flexibility, creativity, and background knowledge with the enormous storage capacity and the computational power of today's computers. The specific advantage of Visual Analytics is that decision makers may focus their full cognitive and perceptual capabilities on the analytical process, while allowing them to apply advanced computational capabilities to augment the exploration process.

In general, Visual Analytics can be described as "the science of analytical reasoning facilitated by interactive visual interfaces" [TK05]. To be more precise, Visual Analytics is an iterative process that involves information gathering, data preprocessing, knowledge representation, interaction and decision making. The ultimate goal is to gain insight in the problem at hand which is described by vast amounts of scientific, forensic or business data from heterogeneous sources. To reach this goal, Visual Analytics combines the strengths of machines with those of humans. On one hand, methods from knowledge discovery in databases (KDD), statistics, and mathematics are the driving force on the automatic analysis side, while on the other hand human capabilities to perceive, relate, and conclude turn Visual Analytics into a very promising field of research.

Historically, Visual Analytics has evolved out of the fields of information and scientific visualization. According to Colin Ware, the term visualization is meanwhile understood as "a graphical representation of data or concepts" [War00], while the term was formerly applied to the forming of a mental model [Spe01]. Nowadays, fast computers and sophisticated output devices create meaningful visualizations and allow us not only to mentally visualize data and concepts, but also to see and explore a precise representation of the data under consideration on a computer screen.

However, the transformation of data into meaningful visualizations is not a trivial task that will automatically improve through steadily growing computational resources. Very often there are many different ways to represent the data under consideration and it is unclear which representation is the best one. State-of-the-art concepts of representation, perception, interaction and decision making need to be applied and extended to be suitable for visual data analysis.

The fields of information and scientific visualization deal with visual representations of data. The main difference among the two is that scientific visualization examines potentially huge amounts of scientific data obtained from sensors, simulations or laboratory tests. Typical scientific visualization applications are flow visualization, volume rendering, and slicing techniques for medical illustrations. In most cases, some aspects of the data can be directly mapped onto geographic coordinates or into virtual 3D environments.

We define Information Visualization more generally as the communication of abstract data relevant in terms of action through the use of interactive interfaces. There are three major goals of visualization, namely a) presentation, b) confirmatory analysis, and c) exploratory analysis. For presentation purposes, the facts to be presented are fixed a priori and the choice of the appropriate presentation technique depends largely on the user. The aim is to efficiently and effectively communicate the results of an analysis. For *confirmatory analysis*, one or more hypotheses about the data serve as a starting point. The process can be described as a goal-oriented examination of these hypotheses. As a result, visualization either confirms these hypotheses or rejects them. *Exploratory data analysis*, as the process of searching and analyzing databases to find implicit but potentially useful information, is a difficult task. At the beginning, the analyst has no hypothesis about the data. According to John Tuckey, tools as well as understanding are needed for the interactive and usually undirected search for structures and trends [Tuk77].

Visual Analytics is more than mere visualization. It can rather be seen as an integral approach combining visualization, human factors and data analysis. Figure 3.1 illustrates the detailed scope of Visual Analytics [KMSZ06]. Concerning the field of visualization, Visual Analytics integrates methodology from information analytics, geospatial analytics, and scientific analytics.

Especially human factors (e.g., interaction, cognition, perception, collaboration, presentation, and dissemination) play a key role in the communication between human and computer, as well as in the decision making process. In this context, *production* is defined as the creation of materials that summarize the results of an analytical effort, *presentation* as the packaging of those materials in a way that helps the audience understand the analytical results in context using terms that are meaningful to them, and *dissemination* as the process of sharing that information with the intended audience [TK05]. In matters of data analysis, Visual Analytics furthermore profits from methodologies developed in the fields of data management and knowledge representation, knowledge discovery, and statis-

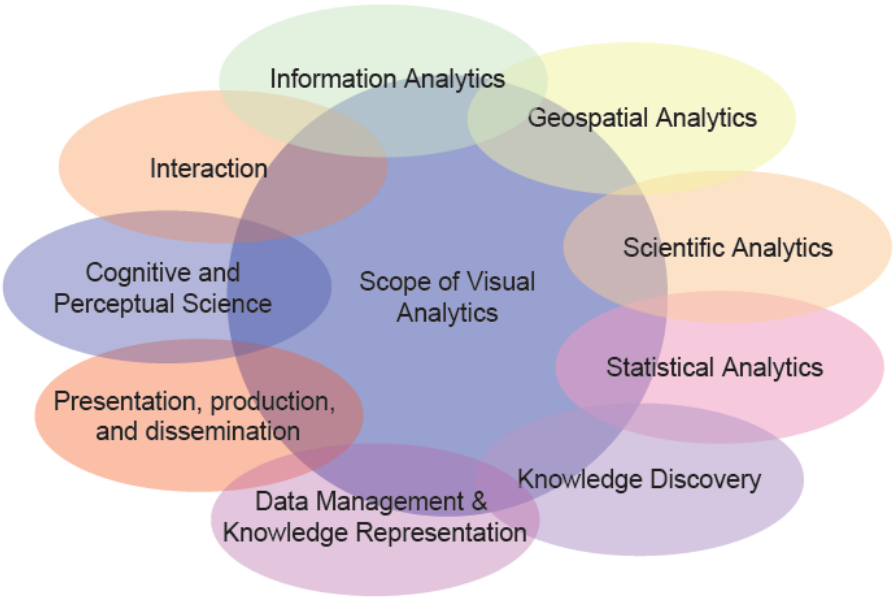


Figure 3.1: Scope of Visual Analytics

tical analytics.

According to Jarke J. van Wijk, "visualization is not 'good' by definition. Developers of new methods have to make clear why the information sought cannot be extracted automatically" [vW05]. From this statement, we immediately see the need for the Visual Analytics approach using automatic methods from statistics, mathematics, and knowledge discovery in databases (KDD) wherever they are applicable. Visualization is used as a mean to efficiently communicate and explore the information space when automatic methods fail. In this context, human background knowledge, intuition, and decision making either cannot be automated or serve as input for the future development of automated processes.

Examining a large information space is a typical Visual Analytics problem. In many cases, the information at hand is conflicting and needs to be integrated from heterogeneous data sources. Moreover, the system lacks knowledge that is still hidden in the expert's mind. By applying analytical reasoning, hypotheses about the data can be either affirmed or discarded and eventually lead to a better understanding of the data, thus supporting the analyst in his task to gain insight. Contrary to this, a well-defined problem where the optimum or a good estimation can be calculated by non-interactive analytical means would generally not be described as a visual analytics problem. In such a scenario, the non-interactive analysis should be clearly preferred due to efficiency reasons. Likewise, visualiza-

tion problems not involving methods for automatic data analysis do not fall into the field of Visual Analytics.

The fields of visualization and Visual Analytics both build upon methods from scientific analytics, geo-spatial analytics and information analytics. They both profit from knowledge out of the field of interaction as well as cognitive and perceptual science. They do differ in so far as Visual Analytics additionally integrates methodology from the fields of statistical analytics, knowledge discovery, data management and knowledge representation as well as presentation, production and dissemination.

3.3 The Visual Analytics Process

In this section we provide a formal description of the Visual Analytics process. As described in the last section the input for the data sets used in the visual analytics process are heterogeneous data sources (i.e., the internet, newspapers, books, scientific experiments, expert systems). From these rich sources, the data sets $S = S_1, \ldots, S_m$ are chosen, whereas each $S_i, i \in (1, .., n)$ consists of attributes $A_{i1}, \ldots, A_{ik}$. The goal or output of the process is insight I. Insight is either directly obtained from the set of created visualizations V or through confirmation of hypotheses H as the results of automated analysis methods. We illustrate this formalization of the visual analytics process in Figure 3.2. Arrows represent the transitions from one set to another.

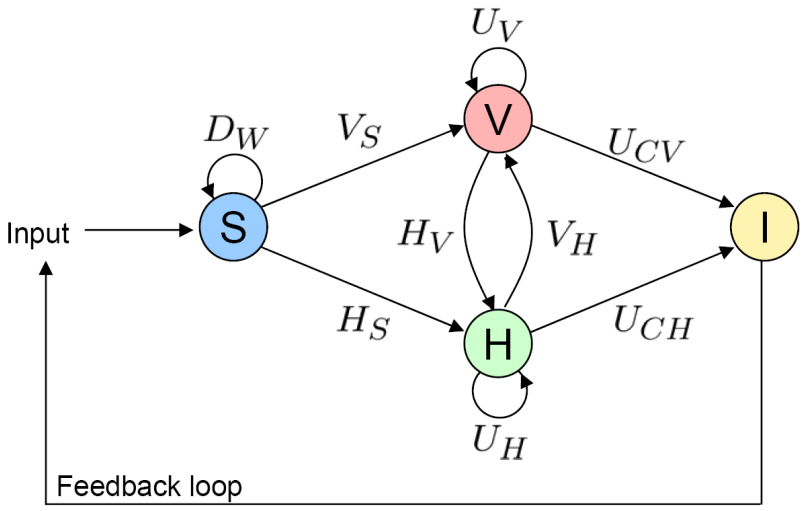


Figure 3.2: The Visual Analytics process: The goal is to get insight I from the input data, either from visualizations V or hypotheses H.

More formal the Visual Analytics process is a transformation $F : S \rightarrow I$, whereas F is a concatenation of functions $f \in \{D_W, V_X, H_Y, U_Z\}$ defined as follows:

D_W describes the basic data pre-processing functionality with $D_W : S \rightarrow S$ and $W \in \{T, C, SL, I\}$ including data transformation functions D_T, data cleaning functions D_C, data selection functions D_{SL} and data integration functions D_I that are needed to make analysis functions applicable to the data set.

$V_W, W \in \{S, H\}$ symbolizes the visualization functions, which are either functions visualizing data $V_S : S \rightarrow V$ or functions visualizing hypotheses $V_H : H \rightarrow V$.

$H_Y, Y \in \{S, V\}$ represents the hypothesis generation process. We distinguish between functions that generate hypotheses from data $H_S : S \rightarrow H$ and functions that generate hypotheses from visualizations $H_V : V \rightarrow H$.

Moreover, user interaction $U_Z, Z \in \{V, H, CV, CH\}$ is an integral part of the visual analytics process. User interaction can either effect only visualizations $U_V : V \rightarrow V$ (i.e., selecting or zooming), or it can effect only hypothesis $U_H : H \rightarrow H$ by generating a new hypothesis from given ones. Furthermore, insight can be concluded from visualizations $U_{CV} : V \rightarrow I$ or from hypotheses $U_{CH} : H \rightarrow I$

The typical data pre-processing applying data cleaning, data integration and data transformation functions is defined as $D_P = D_T(D_I(D_C(S_1, \ldots, S_n)))$. After the pre-processing step either automated analysis methods $H_S = \{f_{s1}, \ldots, f_{sq}\}$ (i.e., statistics, data mining, etc.) or visualization methods $V_S : S \rightarrow V, V_S = \{f_{v1}, \ldots, f_{vs}\}$ are applied to the data in order to reveal patterns as shown in Figure 3.2.

The application of visualization methods can hereby directly provide insight to the user, described by U_{CV}; the same applies to automatic analysis methods U_{CH}. However, most application scenarios may require user interaction to refine parameters in the analysis process and to steer the visualization process. This means that after having obtained initial results from either the automatic analysis step or the visualization step, the user may refine the achieved results by applying another data analysis step, expressed by U_V and U_H.

Moreover, visualization methods can be applied to the results of the automated analysis step to transform a hypothesis into a visual representation V_H or the findings extracted from visualizations may be validated through an data analysis step to generated a hypothesis H_V. $F(S)$ is rather an iterative process than a single application of each provided function, as indicated by the feedback loop in Figure 3.2. The user may refine input parameters or focus on different parts of the data in order to validate generated hypotheses or extracted insight.

We take a Visual Analytics application for monitoring network security as an example. Within the network system, four sensors measure the network traffic resulting in four data sets $S_1, \ldots, S_4$. During preprocessing, the data is cleaned from missing values and unnecessary data using the data cleaning function d_c, integrated using d_i (each measurement system stores data slightly differently), and transformed in a format suitable for our analysis using d_t. We now select

UDP and TCP traffic for our analysis with the function d_s, resulting in $S' = d_s(d_t(d_i(d_c(S_1, \ldots, S_4))))$.

For further analysis, we apply a data mining algorithm h_s to search for security incidents within the traffic generating a hypothesis $h' = h_s(S')$. To better understand this hypothesis, we visualize it using the function v_h: $v' = v_h(h')$. Interactive adjustment of the parameters results in $v'' = u_v(v')$, revealing a correlation of the incidents from two specific source networks. By applying the function h_v, we obtain a distribution of networks where similar incidents took place $h'' = h_v(v'')$. This leads to the insight that a specific network worm tries to communicate with our network from 25 source networks $i' = u_{ch}(h'')$. Repeating the same process at a later date by using the feedback loop reveals a much higher spread of the virus, emphasizing the need to take countermeasures.

The Visual Analytics process aims at tightly coupling automated analysis methods and interactive visual representations, as described before. Without the support of automated methods, the visual analysis of large data sets will become impossible in the future. As a consequence, we extended the classical way of visually exploring data sets as defined by the Information Seeking Mantra ("Overview first, Zoom/ Filter, Details on demand")[Shn96], to the Visual Analytics Mantra [KMSZ06]:

"Analyze First -
Show the important -
Zoom, Filter and Analyse Further -
Details on Demand"

In the age of massive data sets all three steps of the Information Seeking Mantra are difficult to realize. An overview visualization without losing any interesting pattern or subset is difficult to create, since the amount of pixels of modern data display systems do not keep pace with the increasing flood of data. The plenty of information hidden in massive data sets make it very difficult for humans to understand the really interesting or relevant information.

In Visual Analytics it is therefore not sufficient to just retrieve and display the data using a visual metaphor, it is rather necessary to support the analyst by analytically filtering the underlying data by its value of interest, but at the same time providing interaction models which still allow the user to get any detail of the data on demand.

This thesis focuses on providing techniques that follow the Visual Analytics Mantra in terms of combining visualization methods with automated techniques, and provides a number of applications, mainly in the field of business analysis, that show the value of our novel approaches.

Chapter 4

Scope of this Thesis: Research Challenges in Business Applications

The value of each Visual Analytics solution, including Business Intelligence applications, is based on its ability to derive knowledge from data as well as the capability to process large volumes of information and identify patterns, trends, rules, and relationships that are too large to be handled through simple human analysis or standard reporting tools. But the growing complexity and volume of today's data sets pose a challenge for Visual Analytics tools. The most important of them, including application and technical challenges, were defined in the Visual Analytics Agenda [TK05].

In the following we describe the most significant application and technical challenges in business applications which are within the scope of our research. The next chapters then show how this thesis provides research results aiming to face these challenges in a number of application scenarios.

4.1 Visual Scalability

The sheer volume of data generated by e-commerce and the need to incorporate the data from different enterprise systems place high demands on the analysis component of future business analysis solutions. Data Warehouses, which are the typical way to collect non-operational enterprise data for analysis purposes, are optimized for reporting and analysis, e.g. by using Online Analytical Processing (OLAP) tools. Because of technical progress companies today are able to operate Data Warehouses larger than 1TB. According to the Winter Report, multi-terabyte warehouses are more and more the norm in today's enterprises as data, user communities, and workloads rapidly grow. Impressive examples are Yahoo.com or AT&T, which as of 2005 both operate 100TB Data Warehouses [Cor05]. High-

performance analytical tools are needed that can perform calculations and analysis against such huge stores of information, glean trends and insights from this ocean of data, and return results at the speed of thought. Beyond standard report techniques, Visual Analytics tools are needed that allow sophisticated analysis and visual representation of the data, since this data volume is magnitudes larger than volumes that can be handled by most existing techniques.

Eick and Karr [EK02] proposed a scalability analysis and came to the conclusion that many visualization metaphors do not scale effectively, even for moderately sized data sets. Scatterplots for example, one of the most useful graphical techniques for understanding relationships between two variables, can be overtaxed by a few thousand points. Additionally, there are two limiting factors for all visualization techniques: human perception and display area. On one hand, human perception, that means the precision of the eye and the ability of the human mind to process visual patterns, limits the number of perceptible pixels and therefore directly affects visual scalability. One the other hand, monitor resolution affects visual scalability through both physical size of displays and pixel resolution. At a normal monitor viewing distance, calculations in [EK02] suggest that approximately 6.5 million pixels might be perceivable for the human eye, given sufficient monitor resolution. The resolutions for typical PC monitors varies from 800×600 to 1600×1400 pixels resulting in $480,000$ to $1,920,000$ displayed pixels. Although large scale displays, like the Powerwall at the University of Konstanz with a 4000×2000 pixel display, exceed the human perception, in typical application scenarios monitor resolution rather than human vision is the limiting factor.

Based on these facts, the analysis of large data sets reveals two major tasks. The first one is the question, how visualizations for massive heterogenous data sets can be constructed without loosing important information even if the number of data points is too large to visualize each single data point at full detail. The second important task is to find techniques to efficiently navigate and query such massive data sets.

4.2 Analysis of heterogeneous Data Sources

Today's Data Warehouses typically not only have millions of records, but also integrate data sets from heterogeneous data sources. The data is typically represented as a data cube, which is defined by 3 components: place (geo-spatial context in 2-D or 3-D), time (with a continuous direction) and a set of (multivariate) attributes. Analysis techniques have to take the special data characteristics along each dimension (geo-spatial, temporal, multivariate dimensions) into account, and therefore powerful visual metaphors are needed. Furthermore the integrated analysis along all dimensions (geo-spatial, temporal, multivariate dimensions) holds great potential to provide valuable and previously unknown information that can identify complex phenomena, especially multivariate space-time patterns. However, Visual Analytics of geo-temporal data are challenging problems. Dynamic space-time pat-

terns and potentially interesting events in space and time have in practice a much higher complexity than available visual encodings can handle. The data is in general defined over a geo-spatial context with some associated attributes such as numerical statistical parameters, text, images, GPS-data, network logs etc. The analysis involves a wide variety of objects with varying attributes in time; it is often hard to see what is emphasized. Therefore Visual Analytics approaches are needed that are able to explore multivariate spatio-temporal patterns and present them in an intuitive form to support human interpretation and decision making.

4.3 Automated Support for Visual Representations

In business applications, analysts have to deal with large parameter spaces when using visualization techniques to explore large data sets. These parameters control the visual encoding of the data, including the selection of attributes from the input data, the selection of the color scale, algorithm parameters, the selection of visual variables and so on. Finding parameter settings that lead to insightful visualization, is however a challenging task. In Exploratory Data Analysis a good or the optimal parameter setting for a given task is often not clear in advance, which means that the analyst has to try multiple parameter settings in order to generate valuable visualizations. Since such selections can hardly be done manually, the integration of automated methods to support the analyst has been recognized as an important research problem in the context of Visual Analytics in Business Applications.

The problem of automatically supporting the user in constructing insightful visualizations is in practice a two stage problem: 1) Dimension management and 2) Appropriate visual mappings.

4.3.1 Dimension Management

Since high dimensional data sets are commonplace in today's applications such as business analysis, bioinformatics or situation awareness, the user needs tools for effective dimension management when analyzing these data sets. This includes dimension ordering and dimension filtering, since high dimensional visualizations can be cluttered and difficult for users to navigate the data space [YWRH03, YWR03]. A data set with 200 dimension for example, would lead to 40000 plots for Scatterplot Matrices [CM88], 200 axes for Parallel Coordinate plots [ID90] or 200 sub windows for Recursive Pattern [KKA95].

Dimension Ordering

The order of dimensions is crucial for the expressiveness and effectiveness of visualizations [ABK98]. *Expressiveness criteria* identify graphical representations that express the desired information. *Effectiveness criteria* identify which of the graphical representations, in a given situation, is the most effective at exploiting the capabilities of the output medium and the human visual system [Mac99]. Bertin [Ber67] presented some examples illustrating that permutations of dimensions and data items reveal patterns and improve the comprehension of visualizations [YWRH03]. In Parallel Coordinates [ID90], Recursive Pattern [AKK95] or Circle Segments [AKK96], for example, the dimensions have to be arranged in some one- or two-dimensional order on the screen. The selected arrangement of dimensions can have a major impact on the expressiveness of the visualization because relationships among adjacent dimensions are easier to detect than relations among dimensions positioned far from each other [YWRH03].

In [ABK98], the importance of dimension arrangement for order-sensitive multidimensional visualization techniques is pointed out, the concept of similarity of dimensions is defined and several similarity measures are discussed. Basically the problem of determining the similarity of dimensions was characterized as follows: The database D containing N data items with d-dimensions can be described as d arrays $A_i (0 \leq i \leq d)$, each containing N values $a_{i,k}, (a_{i,k} \in R, 0 \leq a_{i,k} \leq N)$. A similarity measure S computes the pair wise similarity between arrays $A_i, A_j, (0 \leq i, j \leq d, i \neq j)$ by mapping the two arrays to a (normalized) real number, whereas zero means identity and lower values mean higher similarity than higher values.

In [ABK98] some specific similarity measures are presented, including partial similarity measures. However, in general computing similarity measures is a non-trivial task, because similarity can be defined in various ways and for specific domains, like similarity measures proposed in the context of time series data [YWY00, FRM94] or similarity measures presented in [HDY99]. In [YWRH03] an approach for hierarchical dimension ordering is presented that allows the generation of default settings for dimension orderings and allows users to interactively control aspects of this dimension management process. Note that dimension ordering is also important for many other fields. The database primitive similarity join for example has been used to speed up applications such as similarity search, data analysis and data mining. Its computational overhead is mostly dedicated to the distance calculation between the feature dimensions [YWRH03]. [BKK02] propose a generic approach to speed up these distance calculations by ordering dimensions according to a probability model.

Of course, many multidimensional visualization systems, like Polaris [STH02] XmdvTool [War94], or CircleView [KSS04a], support manual dimension ordering. Although manual dimension ordering might be sufficient for low dimensional data sets, interactive orderings become tedious or impractical when exploring high dimensional data sets, since the number of possible orderings boost exponentially with increasing number of dimensions. With the exploding volume and dimen-

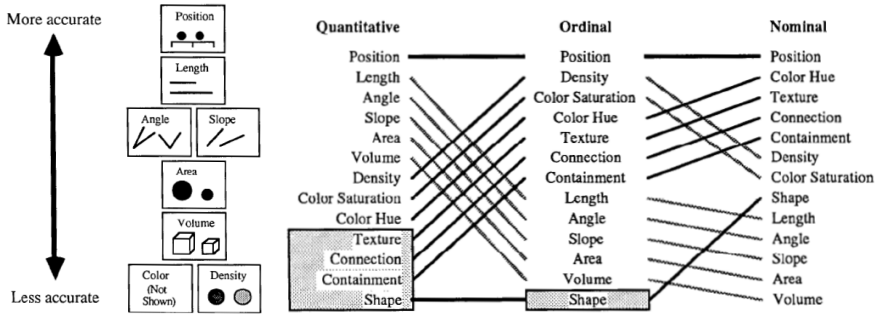
sionality of today's data sets it is therefore more and more important to support the user with automated approaches.

Dimension Filtering

Dimension filtering is an essential task for visualizing high dimensional data sets. Since large numbers of dimensions not only cause clutter in multidimensional visualizations but also make it difficult for users to navigate the data space and are impractical for many common visualization approaches, dimension filtering removes some of the dimensions from the display. Available complex data sets may contain several hundred or more dimensions, which none of the existing visualization techniques can map all at the same time without cluttering the display [YWRH03]. One way to face this problem is to apply dimension reduction approaches like Principal Component Analysis (PCA) [Dun89], Multidimensional Scaling (MDS) [CC01] or Self Organizing Maps (SOM) [Koh97], which are able to condense hundreds of dimensions into a few, typically two or three. As a standard method of visualizing high-dimensional data, its dimensionality is reduced to two or three dimensions, e.g. by using PCA, and then a scatterplot is created with data represented by labeled and / or colored pixels on the screen. However, those resulting dimensions have only little intuitive meaning to the user and allow little user interaction. Moreover, if the data contains explicit space or time attributes, this context is lost if the dimensions are condensed. Therefore tools that employ dimension reduction techniques like the VIS-STAMP system [GCML06] which employs Self Organizing Maps, typically need additional linked views, like Parallel Coordinates, to extract characteristics of the data items in the low dimensional projection.

Dimension filtering in contrast, is more intuitive to users since the remaining dimensions are all original and therefore meaningful dimensions in the data. It is also more flexible to user interaction since it allows selecting or unselecting dimensions to be filtered. The basic idea of dimension filtering techniques is to filter some dimensions to reduce the clutter problem, but at the same time retain most of the information in the dataset. The filtering can be done manually, automatically or semi-automatically [YWRH03]. The manual approach is impractical when the dimensionality is fairly large, therefore automatic and semi-automatic approaches have attracted much research attention. Data Mining methods like Clustering, Classification or Association techniques [HK06] or Correlation / Similarity measures are a common way to automatically filter dimensions to reveal relevant relationships. We integrated these techniques in the *VisImpact* system [KSHD06, KSDH05, KSH+05] to identify dimensions in large multidimensional business data sets that have an impact on certain business metrics.

In [YWRH03] an approach based on dimension hierarchies is proposed that automatically generates a default filtering result based on a combination of dimension similarity and importance. The authors assume that if dimensions are very similar to each other, than only one of them should be mapped to the display, and that



(a) (b)

Figure 4.1: *Ranking of perceptual tasks* - Higher tasks are accomplished more accurately than lower tasks. Cleveland and McGill empirically verified the properties of this ranking for quantitative information (a), Mackinlay extended it to non-quantitative data (b) [Mac99]

dimensions that are fairly unimportant for an users visualization task should not be displayed. Since it is possible that an automated dimension filtering step may filter out some dimensions that a user is concerned about or displays dimensions that are uninteresting for the user, it is important to allow the user to interactively readjust filtering results. This semi-automatic procedure, initial automated filtering first, then manual adjustment, is therefore the most common way for dimension filtering, integrated in tools like Xmdv [War94] or *VisImpact* [KSHD06].

4.3.2 Automated Support for effective Visual Mappings

When using visualization techniques for visual exploration of large multivariate data sets, attribute mapping, i.e. the mapping of dimensions to certain visual properties of the visualization, is a very important issue. Improper mappings might lead to ineffective visualizations and wrong conclusion that might be extracted from the visualization. The difficulty is that the effectiveness not only depends on the properties of the visualization, but also on the capabilities of the perceiver (user) [Mac99]. Since there does not yet exist an empirically verified theory of human perception capabilities that can be used to prove theorems about the effectiveness of visualizations, in [Mac99] a conjectural theory is presented that is both intuitively motivated and consistent with current empirically verified knowledge about human perception capabilities. This theory is based on perception experiments made by Cleveland and McGill [CM84], which revealed that people accomplish perceptual tasks associated with the interpretation of graphical representations of quantitative information with different degrees of accuracy. They

identified and ranked these tasks. Higher tasks are accomplished more accurately than lower tasks. Mackinlay extended the ranking to non-quantitative information and defined the *Principle of Importance Ordering*: Encode more important information more effectively [Mac99]. This means that more important dimensions need to be mapped to more pre-attentive visual attributes [YWRH03], such as more important features of the face in Chernoff Faces [Che73] or outer dimensions in dimensional stacking [LWW90].

Dimension ordering can help to improve the effectiveness of visualizations by giving reasonable orders to the dimensions. However, in Exploratory Data Analysis it is often not clear in advance which dimension are more important than others. Furthermore, there are many other parameters that have an impact on the effectiveness of the resulting visualization, for example the selected normalization to a color scale. Therefore, we propose an approach that on one hand uses analytical techniques for dimension management and takes the state-of-the-art visual mapping heuristics into account, but at the same time analyzes the resulting visualizations with respect to certain user tasks. We present application examples that show how this combination of analysis methods can help to support the user in construction insightful visualizations by automatically extracting potentially useful parameter vectors from the underlying candidate parameter space.

Part III

Visual Business Analytics: Techniques and Applications

Chapter 5

Data Model for Business Data

5.1 Business Data and Data Warehouses

With the need to organize and integrate large data volumes from heterogeneous data sources, Data Warehouses have become common in a variety of scientific and business applications. Corporations usually build large Data Warehouses of historical data on key aspects of their operations. Of course, besides creating standard reports from this data, corporations are interested in extracting deeper knowledge from their collected business data such as interesting structures and patterns or causal relationships.

However, because of the size and complexity of these data sets, this is a challenging task. Interactive calculations or analysis tasks that require visiting each record are not plausible and computationally expensive, nor is it feasible for an analyst to view the entire data set, usually millions of records, at its finest level of detail [STH02]. Therefore Data Warehouses usually employ meaningful hierarchical structure on the data to provide levels of abstraction that can be navigated via roll-up and drill-down functionality.

For this reason Data Warehouses usually form relational databases as n dimensional data cubes. Relational databases organize the data into relations, where each row in such a relation corresponds to a basic entity or fact and each column represents a property of that entity. A relation may, for example, represent credit card transactions, where each row corresponds to a single transaction and each transaction has multiple properties, such as the transaction amount, the transaction time, the location of the transaction (e.g. the shop or the bank that processed the transaction), and the customer.

A row in a relation is referred to as a tuple or record, and a column in the relation as a field. A single relational database will contain many heterogeneous but related relations. The fields within a relation can be partitioned into two types: *dimensions* and *measures*. Dimensions and measures are similar to independent and dependent variables in traditional analysis. For example, in the mentioned

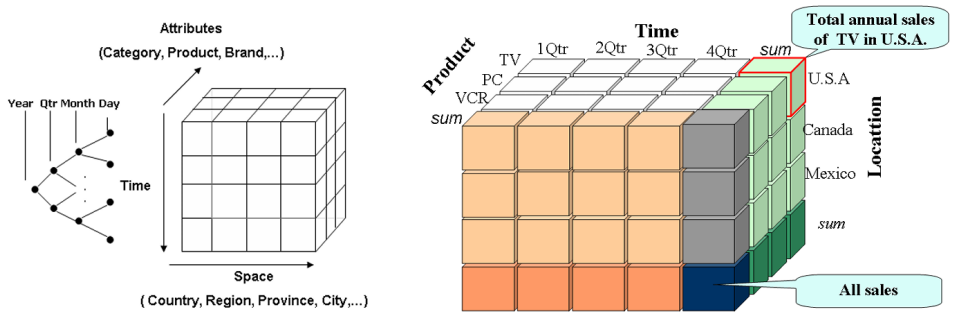


Figure 5.1: Data Cubes are a common way to model business data in Data Warehouse environments. Data Cubes present aggregated data at different levels of detail along a number of dimensions, as shown by the *location, time, product* dimension (right), whereas each cell of the data cube represents a measure (product sales) [HK06].

credit card transaction relation, the customer would be a dimension, while the transaction amount would be a measure. A comprehensive guide to Data Warehouses can be found in [Inm96]. For analysis issues particularly the characteristics of the data cube model have to be taken into account.

5.2 Characteristics of Data Cubes

In Data Warehouses, these multi-dimensional databases are structured as n dimensional data cubes. Each dimension in the data cube corresponds to one dimension in the relational schema. Each cell in the data cube contains all the measures in the relational schema corresponding to a unique combination of values for each dimension.

The dimensions of the data cube usually have a hierarchical structure. This hierarchical structure may be derived from the semantic levels of detail within the dimension or generated from classification algorithms. For example, rather than having a single dimension *location* for locations of credit card transactions, we may have a hierarchical dimension *location* that has levels for country, state, county, and city. If each dimension has a hierarchical structure, then the data must be structured as a lattice of data cubes, where each cube is defined by the combination of a level of detail for each dimension. Using these hierarchies, analysts can explore and analyze the data cube at multiple meaningful levels of aggregation calculated from a base fact table i.e., a relation in the database with the raw data. Each cell in the data cube now corresponds to the measures of the base fact table aggregated to the proper level of detail [STH02].

In Data Warehouses, these hierarchies can be modeled using a star schema. A star schema is characterized by one or more very large fact tables that contain the primary information, and a number of much smaller dimension tables that represent the dimensional hierarchy [HK06]. In this type of hierarchy, there is only one path of aggregation. To model more complex dimension hierarchies where the aggregation path can branch, a snowflake schema is usually employed that uses multiple relations to represent the diverging hierarchies [HK06].

To provide efficient access to the data, Data Warehouses support Online Analytical Processing (OLAP). OLAP allows an effective drill-down / roll-up functionality to the data based on data cubes. In most practical application scenarios these data cubes are typically defined over three components: space (country, states, region,), time (Year, Month, week,) and a set of usually multivariate attributes.

5.3 Requirements for Business Analytics

The given data cube model imposes significant demands on Visual Analysis techniques for the exploration of business data stored in Data Warehouses. Analysis techniques not only have to take the specific data characteristics of temporal and geographic dimensions into account, but also have to support the analysis of hierarchical structures.

To extract a maximum of information from the data, analysis techniques are needed that are beyond the capabilities of standard OLAP and reporting tools. Automated techniques for multivariate analysis along the attribute dimensions need to be tightly integrated with powerful visualization tools for the analysis of temporal and geo-spatial data in order to allow an exploration along all dimensions of a data cube and thus to reveal complex patterns.

Depending on the data type, different demands on the Visual encodings exists in order to provide a comprehensive and effective view to the data. Furthermore, since most visualization techniques can handle much fewer dimensions than the number of dimensions that are modelled in data cubes, the user must be supported by automated methods which identify dimension that have an impact-relationship and which therefore may be subject of detailed visual analysis.

Thus it is important to tightly couple powerful visual encodings with automated data analysis methods and to provide effective dimension management techniques. The user must be able to interactively steer the explorations process to make maximum use of users domain knowledge and perception capabilities.

The next chapters therefore provide novel Visual Analysis techniques for the analysis of temporal, hierarchical, and geo-spatial information that are designed for the analysis of large amounts of business data sets from a number of different application scenarios. Additionally we provide methods that support the user in finding promising attribute settings and thus gaining faster insight on relevant parts of the data.

Chapter 6

Visual Business Analytics of temporal Data

Time related data sets are ubiquitous and appear in many business applications as well as in scientific domains. Examples are finance (stock market data, credit card transactional data), communication (telephone data, signal processing data, network monitoring), or entertainment (music, video).

The analysis of time related data sets is often a key issue in order to gain insight into the data and to identify temporal patterns, trends, or correlations. This knowledge is essential for many data analysis tasks like fraud- and anomaly detection, decision support, prediction, or performance analysis. Visualization techniques have been successfully employed to analyze time related data sets in many application scenarios, and a number of sophisticated visualization methods have been proposed in the past [MS03].

In most analysis tools, however, the most important and most common visualization techniques for time series data still are charts, including line-, bar-, sequence-, point charts and their variations. Financial analysts, for example, use charts almost exclusively to analyze a wide array of assets such as stocks, bonds, futures, commodities, or market indices to forecast future price movements.

However, since in typical business applications there are usually dozens or hundreds of attributes that need to be analyzed over time, simple chart techniques are often inadequate. Impacts and relationships between these time related attributes need to be analyzed, therefore analysis functions must be tightly coupled with visualization tools. Furthermore, time related data sets are often too large in volume to recognize important patterns in the data.

Our research therefore focuses on one hand on techniques that generate compact representations of large time series data that emphazise important data features, and on the other hand on the Visual Analysis of multiple time series over time with intrinsic hierarchical relationships. Additionally we provide an approach for analyzing temporal behavior in large business process workflows.

6.1 Multi-Resolution Visualization

In Chapter 4 the need for scalable analysis methods in order to keep step with the growing flood of information was explained. This issue is especially relevant for time related data, since such data is typically very large in volume, often too large to be visually analyzed at full scale. Most existing visualization approaches do not scale well on such large data sets as visual representation suffers from the high number of relevant data points that might be even higher than the available monitor resolution and does therefore not allow a direct mapping of all data points to pixels on the display. Since in the future Visual Analysis of temporal data will have to deal with even larger data sets, we focus on ways to increase scalability by integrating multi-resolution techniques into the visualization process. Our approach is based on the key concept of compact data representations, showing the data at different object resolutions. In contrast to existing hierarchical aggregation strategies, which show the data at fixed levels of detail, our approach allows a global overview on the data set by providing compact representations of the data and at the same time emphasizes interesting patterns or subsequences by presenting them at higher detail [KS05]. We integrated these multi-resolution techniques into the CircleView System [KSS04a], a recently proposed approach for visually analyzing time related data sets. After introducing the CircleView technique, we describe our multi-resolution approach in detail and show how it can increase visual scalability in terms of data size.

6.1.1 The CircleView Technique

CircleView [KSS04a] is a technique to analyze high-dimensional attributes using an intuitive way of visual representation. The basic idea of *CircleView* displays is to arrange the dimensions of the high dimensional data in a circular layout, by dividing a circle in a number of segments, depending on the number of dimensions of the data set. Each segment is then further divided in subsegments in order to visualize the time related data items of each dimension. Figure 6.1 shows the basic idea of *CircleView* displays, based on an example from stock market analysis. The circle in Figure 6.1 is divided into 30 circle segments according to the number of attributes (30 stocks from the S&P 500 index). Each of the segments is then subdivided into subsegments in order to visualize and to compare the distribution and changes of the attribute values (stock prices) over time. The color of each subarea shows the aggregated value of an attribute at a certain point in time (closing stock price per day). The example shows that a data analyst can easily compare each time slot of a circle segment with the corresponding time slots of the neighboring segments. This makes it very easy to visually analyzing correlations in the data over multiple attributes and time frames.

The illustrating example in Figure 6.2 shows the structure of a single *CircleView* segment. Each segment represents the values / measures of a single attribute over a specific number of time periods $t_0, \ldots, t_n$. *CircleView* supports drill down

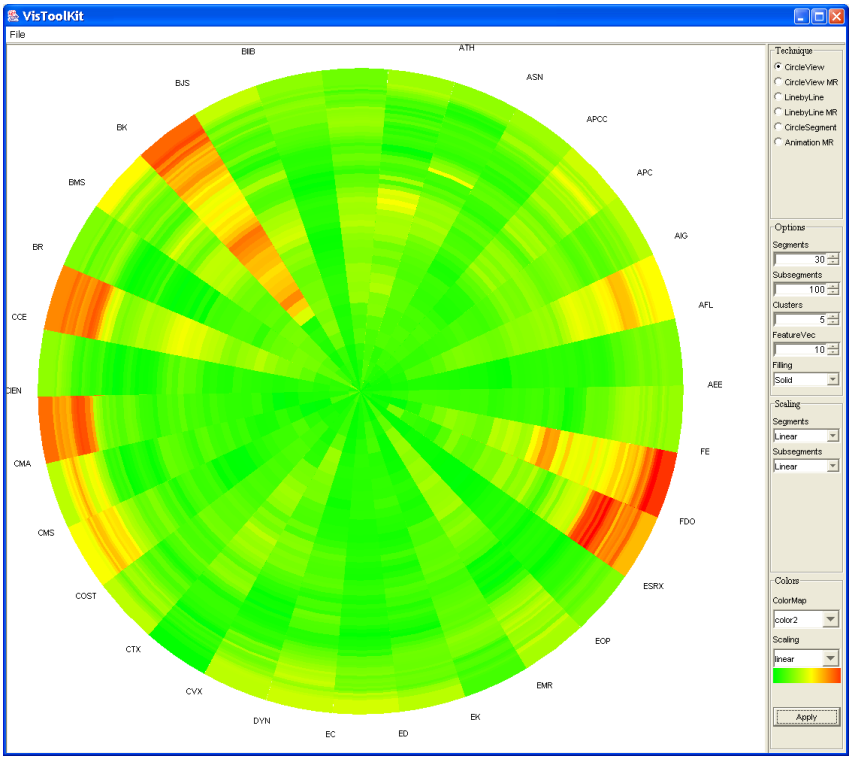
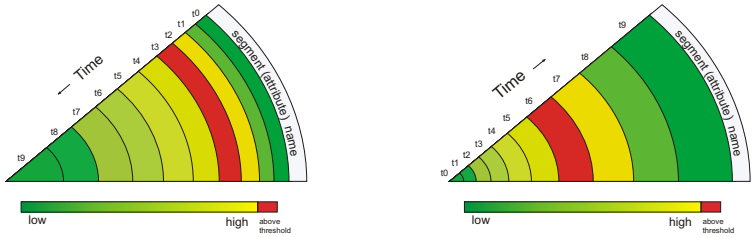


Figure 6.1: Basic Idea: *CircleView* visualizes multi-dimensional time related data by assigning data items to circle segments and subdividing each circle segment in a number of subsegments, one for each time step. The example shows the (min / max normalized) stock prices for 30 stocks from the S&P 500 index, ordered by lexicographic order. Starting in the center of the circle (January 2004), the values for each stock over 100 business days (till June 2004 at the outside of the circle) are shown. Stocks of the FirstEnergy Corp. (FE) for example, had a very good performance.

operations to allow the user to analyze the data at different levels of time related aggregation. For example, if a segment is divided into 24 subsegments, e.g. to visually represent a certain measurement over every hour of the day, the initial *CircleView* layout would map the aggregated values per hour to the corresponding subsegment using a certain aggregation operation like *min*, *max*, or *avg*. The data analyst may now select a certain sub segment, e.g. because he discovers some unusual values, and a new layout would be generated showing the values for each minute of the selected hour.

CircleView supports diverse dimension (segment) ordering and scaling functions in order to adapt the layout to users demands. The goal of *CircleView* displays is on one hand to visualize the high dimensional data in a way that supports



(a) Evolution of the time events from the outside to the center

(b) Evolution of the time events from the center to the outside

Figure 6.2: Illustration of a single CircleView segment: Different ordering / scaling functions are supported in order to adapt the layout to user's demands.

intuitive visual comparison between dimensions, and on the other hand enables the user to easily follow the development of dimension values over time. The first task is supported by providing segment ordering functionality, scaling functions support the second task.

In many scenarios, for example stock market analysis, it is useful to emphasize current time events (current data items), but at the same time show the trend on historic data. Thus *CircleView* provides scaling functions that emphasize current time events by rescaling of the corresponding subsegments. Figure 6.2 shows an illustration of this idea. The current time event is t_9, the oldest time event is t_0. The length of the corresponding subsegment, defined by the start radius and the end radius, depends on the priority of the events depending on their time stamp. It is easy to see that the used scaling function emphasizes actual events. Note that Figure 6.2 shows also different layouts for current and historic values. In the left figure the most current time event is located at the outside of the circle and the oldest data items are located in the middle of the circle, in the right figure it is the other way around.

6.1.2 Interface Functionality

Our *Circle View* Interface supports the following interaction possibilities:

1. Relate and Combine: the analyst can relate and combine *CircleViews* that display data with identical coordinates

2. Navigation: the analyst can modify the visualization of the data on the screen, *CircleView* interface supports manual and automated navigation methods like ordering, scaling, and drill–down

3. Selection: provides analysts with the ability to isolate a subset of the displayed data for operations such as highlighting, filtering, and quantitative analysis

6.1.3 Detection of Correlations and Patterns

The ordering of the segments plays an important role in finding correlations and patterns in the data since it is often crucial for the expressiveness and effectiveness of visualizations [ABK98]. In *CircleView* displays, it is for example easier to compare segments which are located very close to each other. The analyst can directly compare neighboring time slots since all segments contained in the same circle are aligned and have the same number and length of subsegments. Therefore, the goal is to find a circular layout that places similar segments close to each other. To obtain such an ordering, the similarity between pairs of segments must be determined.

However, as explained in [ABK98] computing similarity measures is a non-trivial task, it can be defined in various ways and for specific domains. In *CircleView* we employed ordering functions based on global correlation measures and clustering techniques as heuristics for ordered layout generation. Besides the computation of a segment ordering based on similarity, the user has the option to swap the position of any circle segment interactively in order to compare any segments of interest.

6.1.4 CircleView Application Example

Industrial Data Analysis

In [KSS+05] we applied the *CircleView* technique to analyze industrial data in the context of the InfoVis Contest 2005. The data provided represented technology companies and their characteristics (e.g. location, product sales, employment count) organized by year form 1989 till 2003 and the task was to characterize trends, patterns, or structure in the data that may be of interest. Figure 6.3 shows analysis results using the CircleView approach. The left figure shows the development of the total number of company sales per industry category. The segments at the center of the circle symbolize the sales amount for the year 1989, at the outside of the circle the sales amount for year 2003 is visualized. For intuitive visual comparison, the segments are clustered using the k-means algorithm [HK06], to group similar segments together. Clusters are visualized by small segments at the outside of the Circle. An interesting information the user can extract from the figures is that software (SOF) has an increase in number of sales in 2001 (yellow changed to orange), but a decrease in 2002 (orange changed to yellow). Additionally the clustering reveals that the total number of sales of software (SOF) and automotive industry (AUT) developed in a similar manner. It is also interesting to see that the defence sector (DEF) shows a continuous increase of sales over the

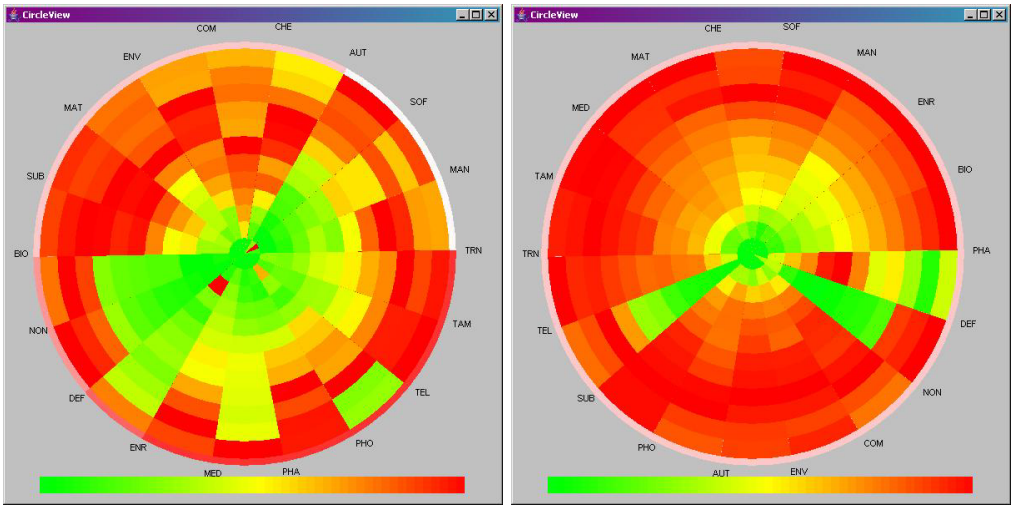


Figure 6.3: *CircleView* applied to the InfoVis 05 Contest data set [GCDT05]. On the left the total number of company sales per industry is shown (clustered), the right figure shows the total number of companies per sector from 1989 till 2003.

years (green changes to read). The figure on the right shows the total number of companies per sector. Here it is for example interesting to see, that almost all sectors had an increase in numbers of companies. Even the "Dot Com" phenomenon had no impact on the increase in the number of software companies (SOF). It is also interesting that the defence sector (DEF) had a high increase in 2000. The only sector where the number of companies decreased after a hype in the mid 90's, is the Pharma sector (PHA).

Of course the user can interactively change the attributes for segment partition and color mapping, in order to explore correlations between attributes of interest.

6.1.5 Multi-Resolution Techniques

The basic idea of Multi-resolution visualization is to decompose the data display space into local screen regions with individual object resolutions. These object resolutions control the granularity of the data points within each particular region. To provide and manage the different data granularities, a tree structure is employed. The structure of the tree highly depends on predefined analytical objective functions, which determine the relevance of single- or sets of datapoints. The goal is to provide an initial visual presentation of the whole data set with respect to the available screen space, that gives relevant parts of the data more space on the screen to present them at higher detail. As an illustrating example, we consider a temporal dataset with only ten data points as shown in Figure 6.4. Our approach determines an importance value for each data point. Let the objec-

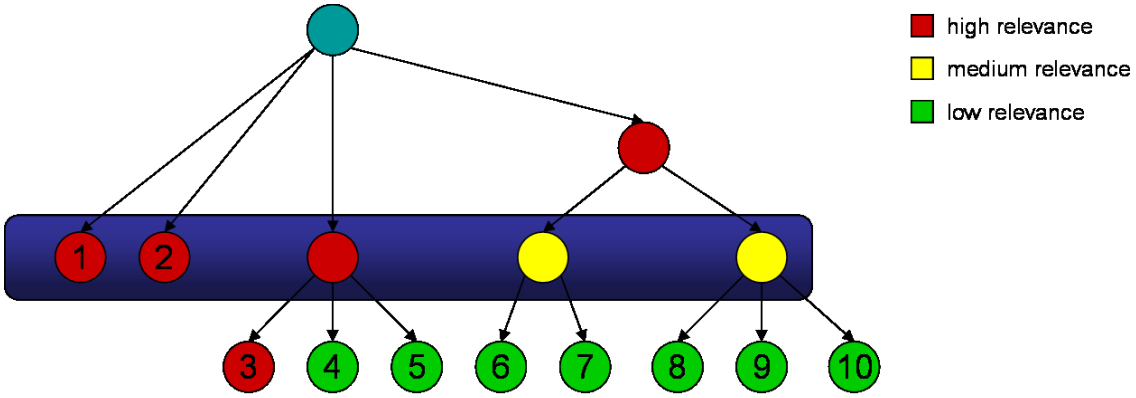


Figure 6.4: Basic idea: A hierarchical data structure is used to generate compact data overviews that take the relevance of the underlying data points into accout. Data points in the blue frame would be visualized.

tive be the detection of data points which have a very high value. Therefore our objective function gives a high importance value to data points which have a high or medium value (red and yellow color in Figure 6.4), while data points with a low value get a low importance value (green color). Based on the importance values of the data points, a hierarchy is created, similar to single linkage clustering. In each step the data subsets with the lowest average importance value are merged, resulting in a tree with different levels of detail like shown in Figure 6.4. To visualize the data set, data objects from the tree structure are selected so that the number and relevance of the selected objects is maximized, depending on the given display space. Each single data point must be included in exactly one selected object. Figure 6.4 indicates a possible selection. The higher the importance value of each single object, the more screen space may be given to visualize the object. Note that the construction of the hierarchy is independent from the underlying visualization, and may therefore be combined with most visualization techniques to increase scalability. The next section describes our approach in detail and gives a more formal definition for the concepts of the scalable visualization paradigm.

Basic Concepts

The relevance function allows us to determine the object resolution of the local screen spaces. The exact definition of relevance functions ψ depends on the application scenario and may be given by the semantic context or may be provided by the user. In general the relevance function can be defined as follows:

Definition 1 *Relevance Function*
Let $D = \{(t_1, x_1), ..., (t_n, x_n)\}$. *The relevance function* $\psi : D \longrightarrow \mathbf{N}$ *assigns every data point* $(t_i, x_i) \in D$ *a relevance value* $\psi(t_i, x_i)$.

With a given relevance function ψ we are able to determine relevant or interesting parts of the data. The goal is then to present less important parts of the data at coarser resolution while presenting relevant parts at higher resolution. Therefore we define Multi-resolution objects:

Definition 2 *Multi-resolution object*
Let $D = \{(t_1, x_1), ..., (t_n, x_n)\}$ *be the input data points and* $\Psi = \{\psi(t_1, x_1), , \psi(t_n, x_n)\}$ *their associated relevance values. A Multi-resolution object MRO is a set of timely close data points with similar relevance values.*

Within every multi-resolution object we define a object resolution level l_i which is application dependent. We suggest to identify the object resolution level l_i as the average of the relevance of all multi-resolution object members. Other application dependent functions (e.g. min or max) may also be used.

Definition 3 *Object Resolution*
Let $MRO_i = \{O_1, ..., O_n\}$ *be a multi-resolution object and* $\Psi = \{\psi(O_1), , \psi(O_n)\}$ *the associated relevance values of the members of the object. The object resolution level* l_i *can be determined as:*

$$l_i = \frac{\sum_{i=1}^{n} \psi_{O_i}}{N}$$

For a given number of maximum allowed multi-resolution objects n_{max} , the Multi-resolution object tree is then constructed bottom up similar to single linkage clustering, by iteratively merging objects with low resolution levels and updating the resolution levels of merged objects until n_{max} is reached.

Wavelet Based Multi-resolution

With the defined concepts, we can now introduce a realization of the proposed ideas. In our experiments we used the Wavelet transformation to construct a Multi-resolution structure. Wavelets deliver a hierarchical structure by iteratively aggregating the data, whereas object resolutions are realized in form of the used compression level. Wavelet Transform (WT), or Discrete Wavelet Transform (DWT) [BGG97] is commonly used in computer graphics and signal processing. In the past it has been successfully applied to temporal data for dimension reduction, data reduction, content-based search and pattern recognition. There are a wide variety of popular wavelet algorithms, including Daubechies wavelets, Marr (or Mexican Hat) wavelets and Morlet wavelets [BGG97]. These algorithms have the advantage of good resolutions for smoothly changing temporal data, but are expensive to calculate. We use Haar wavelets [Haa10] in the context of our Scalable Visualization because:

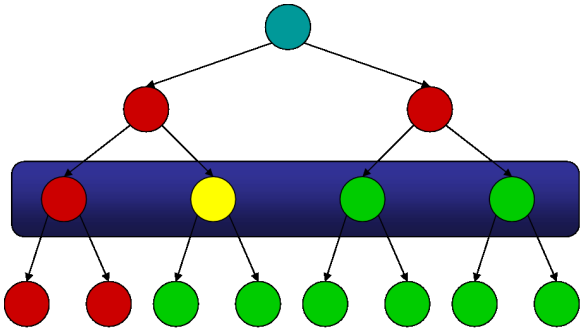


Figure 6.5: Balanced binary tree structure as a result of a Haar wavelet transformation. The blue box indicates the selection of displayed data based on their compression level.

- It is fast. Given a temporal dataset D with length n, with n being an integral power of 2, the complexity of Haar transformation is $O(n)$. Especially when visualizing continuous data streams this is very important.
- It is memory efficient, since it can be calculated in place without a temporary array, which is important considering the huge amounts of data we aim to visualize.
- It is exactly reversible without the edge effects that are a problem with other wavelet transformations. This is especially important for the support of drill down operations

A Haar wavelet transformation is defined as:

Definition 4 *Haar wavelet*

Transformation $\psi_i^j(x) = \psi(2^j x - i), i = 0, ..., 2^j - 1$

$$\textit{where } \psi(t) = \begin{cases} 1 & 0 < t < 0.5 \\ -1 & 0.5 < t < 1 \\ 0 & \textit{otherwise} \end{cases}$$

Together with a scaling function

$$\psi(t) = \begin{cases} 1 & 0 < t < 1 \\ 0 & \textit{otherwise} \end{cases}$$

Figure 6.5 illustrates a simplified balanced binary tree structure as it could be constructed by a Haar wavelet transformation. Analogue to Figure 6.4, red nodes are data objects of high relevance, yellow nodes of medium relevance and green nodes of low relevance. Suppose we have screenspace for four data objects, then we

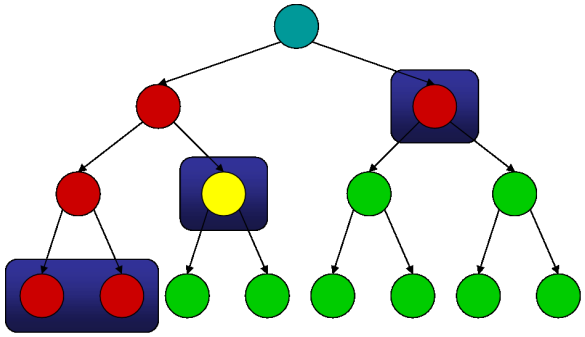


Figure 6.6: Balanced binary tree structure as result of a relevance driven Haar wavelet transformation. Blue boxed nodes indicate the selection of data objects based on their relevance.

choose a compression level in the tree that contains four data objects as illustrated in Figure 6.5.

Although this procedure would reduce the amount of data to be visualized, it would present all data points at the same resolution level, independently of their relevance. This may result in the disappearance of interesting and relevant patterns, since aggregated views do only provide averaged information. To avoid this drawback, we integrated relevance functions and Haar Wavelets, as illustrated in Figure 6.6. There four data objects are selected out of the tree, so that the total relevance is maximized. Thus irrelevant data items are much more compressed than relevant ones. Before we show how we integrated the Multi-resolution approach into the CircleView framework, we present a motivating example of our approach based on the analysis of ECG data.

Electrocardiogram Analysis

Electrocardiograms (ECGs) measure the electric potential between two points on the surface of the body caused by a beating heart over time. The number of data points in an ECG is typically large since data is often collected via long-term ECG's with high sampling rates. The analysis of such data is a very important issue in medicine, e.g. in order to detect heart diseases. Typically the search for discords in such ECG's is done manually by a cardiologist. Time series discords are subsequences of a longer time series that are maximally different to all the rest of the time series subsequences. Keogh et al. proposed the HOT SAX [KLF05] method to obtain those discords from ECG data automatically. This method obtained very useful results, according to a cardiologist. The detected discords correspond very well with anomalies that were found manually by cardiologists in the ECG data. However, by merely highlighting the discords, it may be hard for an analyst to discover the reason for this anomaly because of the high density of data points.

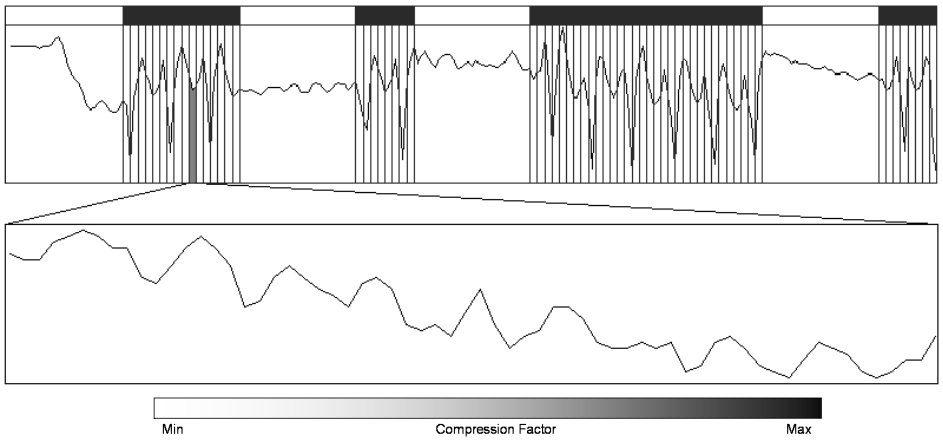


Figure 6.7: Multi-resolution ECG Visual Analytics: A method for finding discords is employed as relevance function in the analysis process. The uncompressed parts are the found discords, corresponding with anomalies in the ECG, the compressed parts offer an overview of the rest of the ECG.

Therefore we integrated automated methods for detecting discords as relevance functions with our Wavelet function, in order to reduce the number of data points and thus point out the discords more clearly. The basic idea is to first find the discords and to show these subsequences at full detail and then to compress irrelevant subsequences with our Wavelet function. By adapting the scaling in the visualization step to the underlying compressing level of the data, discords can easily be analyzed. Discords will be displayed at full scale, while the rest of the data will be compressed on a scale until it fits onto the screen. Since for a cardiologist not only the discords / anomalies are of interest, but also the ECG pattern before or after the discord, we not only display the discord, but also the values directly before and after the discords at full scale. The results of our method are shown in Figure 6.7.

The uncompressed parts are the detected discords; the compressed parts still provide an overview of the whole data set. By visualizing the data this way, the whole ECG can be displayed, while discords can be visually emphasized. Details on demand are supported via drill-down functionality in our visualization, by selecting a compressed part of the data. The underlying original data will be shown if it fits onto the screen, if not, the original data will be compressed with the smallest possible factor so that it fits onto the screen. Note that the stored coefficients of the Haar Wavelet allow a lossless rebuilding of the original uncompressed data.

In many application scenarios, the relevance of single data points is directly related to their time stamp. Usually recent data points are more important for an analyst than historic ones. For example, in real time ECG analysis, a cardiologist is more interested in the actual status, while the trends of the past should still

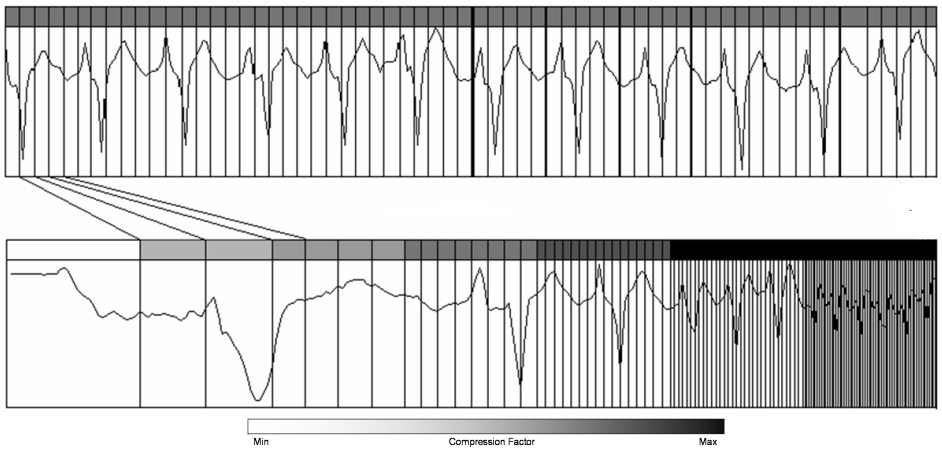


Figure 6.8: Visualization of transformed data T of the ECG dataset, based on a relevance function

be visible. To support this task, we integrated relevance functions that compress the data based on the underlying time stamps. Recent data will be shown at full detail, and historic data will be compressed with higher compression levels. To reach this goal, our method works in two steps. In the first step, the underlying dataset is subdivided into subsequences of length s, with s being an integral power of two. This stepsize s defines how many compression levels should be created and how many data points should be presented at full scale. Based on s the number of different compression levels is determined. The second step then compresses the data. The first subsequence S_1 with length s will be visualized while untransformed. The next subsequence S_2 with length $s \cdot 2^1$ data points will be transformed with a Haar wavelet on level 1 to s data points. The next subsequence S_2 with length $s \cdot 2^2$ will be transformed with a Haar Wavelet on level two to s data points, and so on till $s \cdot 2^{maxLevel}$. The relevance function $\psi : D \longrightarrow \mathbf{N}$ in this context, is defined as:

Definition 5 *Time driven Relevance Function*
Let $D = \{(t_1, x_1), ..., (t_n, x_n)\}$. *The relevance function* ψ *depending on the time stamp* t, *and stepsize* s *is defined as* $\psi((t_i, x_i)) = \frac{1}{((i \mod s)+1}$

Figure 6.8 shows the result after applying this function to the ECG data set. The upper image in Figure 6.8 shows a standard Haar wavelet transform on an ECG dataset. Our example data set contained 4096 data points. With Haar Wavelet transformation we compressed this data set to 256 data points, shown in the upper line chart. Considering the original dataset size and the number of transformed data, each transformed data point presents the average of 16 (4096/256) original data points. The gray boxes above each segment of the line chart indicate the compression factor for the corresponding data points. A cardiologist may use

this visualization to check the ECG over a longer time interval at a very high level of detail, since the important trends in the ECG can still be identified.

The lower image in Figure 6.8 shows the transformed data T of the same dataset, based on the relevance function defined before. The lines between the upper and the lower graph indicate the corresponding time periods. The important trends are still easy to see, but if we look in the left corner of Figure 6.8 we notice a relative long horizontal line, and unfortunately we must conclude that the concerning patient has deceased. In the upper figure that shows the same data set, it is hard to see that the patient deceased. However, both visualizations have their advantages in certain situations, and that is exactly the reason why we are proposing the relevance function based Multi-resolution, since this allows task specific data compressions. The next section shows, how we integrated the proposed time driven relevance function in CircleView for a more effective analysis of financial data.

6.1.6 Application Examples

Stock Market Data Analysis

We used the CircleView technique to analyze a stock market data set containing 2004's S&P 500 stock prices [KS05]. The basic idea of the CircleView visualization technique is to display the distances for the attributes as segments of a circle similar to the Circle Segments technique [AKK96]. If the data consists of k dimensional attributes, the circle is partitioned into k segments, each representing the distances for one attribute. Inside the segments, the distance values belonging to one attribute are arranged from the center of the circle to the outside in a subsegment layout. The size of the segments and subsegments can either be predefined or parameter dependent. Since the size of each subsegment can vary from pixel to segment size, CircleView contains CircleSegments (pixel size) and Pie Charts (segments size) as special cases. In contrast to Circle Segments, CircleView supports ordering and clustering of segments and subsegments, user interaction as well as nearest neighbor searches between segments or within one segment. Therefore it allows intuitive comparison between single segments and subsegments to identify trends and exceptions in the data.

Figure 6.9 presents the basic idea of Multi-resolution CircleView, showing the stock prices of 240 stocks from the S&P 500 over six months, from January 2004 (outside of the circle) to June 2004 (center of the circle). Each segment represents a single stock and each subsegment the stock price of a certain time period depending on the level of detail. The number of data values that can be visualized without aggregation using CircleView is limited by the circle area. In massive datasets this border can easily be reached. To handle this case using multi-resolution, we use the fact that from an analyst's point of view it may be more interesting to analyze recent stock prices than the prices one month ago. Therefore the basic idea is to show only recent stock prices at full detail and to present older values as

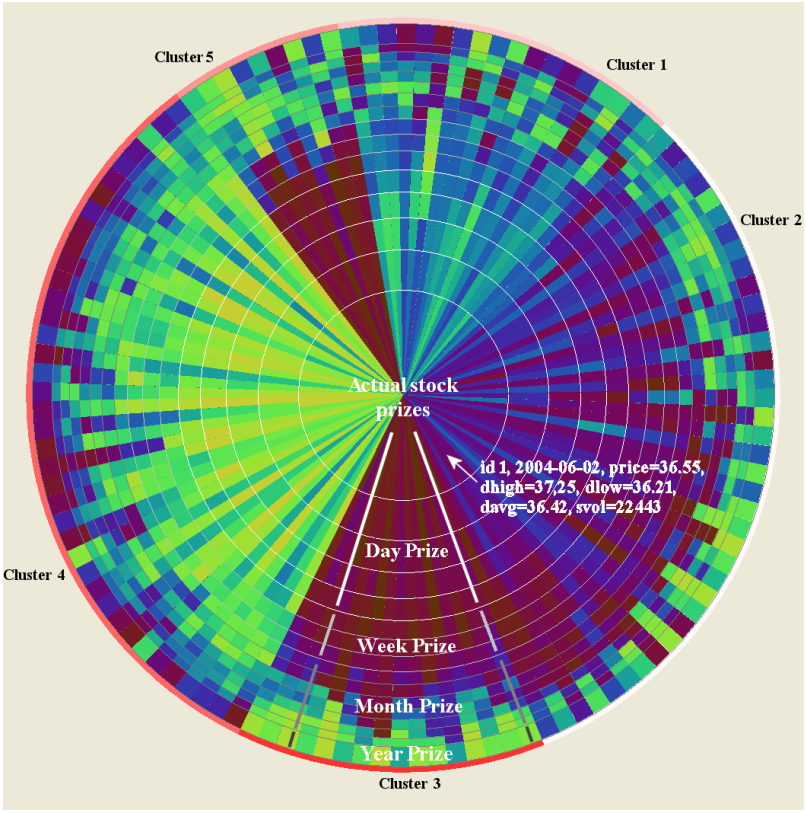


Figure 6.9: CircleView showing stock prices of 240 stocks from the S&P 500 over six months (120 business days). Each Segment represents a single stock and the subsegments represent stock prices. The most recent stock prices are shown in the middle of the circle at full detail. For older stock prices the multiresolution approach presents average prices per week/month/year (outside of the circle).

aggregated high level views. The relevance function is based on this assumption. The level of detail of each data point depends on the available screen space and user parameters.

In the presented example, for the five most recent daily stock prices the closing stock price is shown at full detail. The level of detail as well as the length of the subsegments decreases from the center to the outside of the circle, resulting in higher visual importance of more important values. Since the screen space is limited and it is hard to identify particular stock prices if the subsegments are too small, Multi-resolution is employed to handle this problem. Older stock values are only presented as average values per week, per month or per year depending on the particular data as shown in Figure 6.9. The older the data, the lower the level of detail. Therefore the user is able to access specific information on recent data

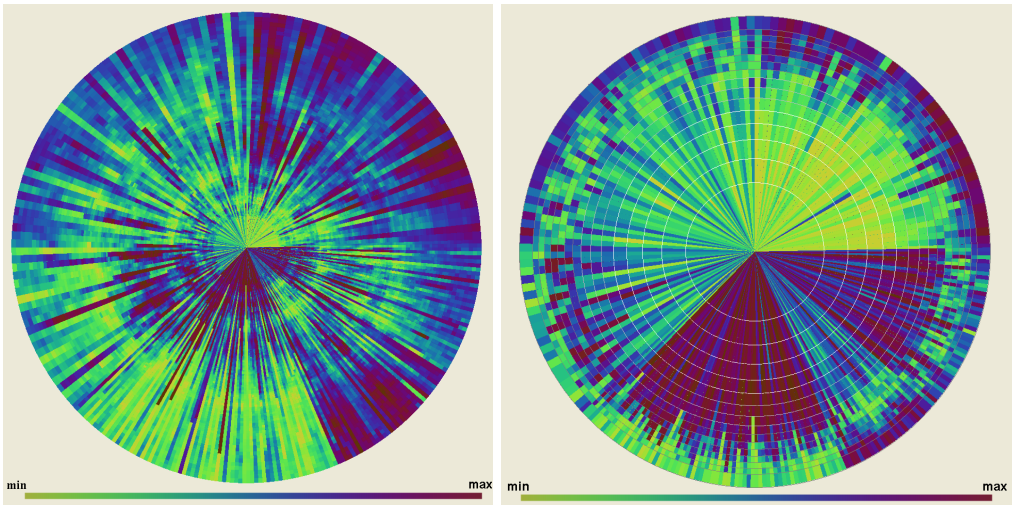


Figure 6.10: Comparison of CircleView (left) vs. Multi-resolution CircleView (right) showing 240 stocks from the S&P 500 over 6 months (120 business days). Although the user gets an overview in both figures, Multi-resolution allows instant access to current stock prices, shown by larger subsegments in the center of the circle, while the global trend is preserved. Clustering helps to identify similar groups of stocks.

instantly, e.g. by mouse interaction and at the same time gets an overview on the whole dataset. Of course there is the possibly to perform drill-down operations on items with lower resolution to get information on historic data on demand.

In Figure 6.10 the described data sample is visualized using k-means clustering with five clusters and compared to standard CircleView. Similar stocks are clearly revealed. Again 240 stocks from the S&P 500 over 120 business days are shown. It is easy to see that Multi-resolution provides better results than standard CircleView in terms of revealing current stock prices but at the same time preserves the global trend. An analyst may easily select and explore the actual stock prices, which is a difficult task in standard CircleView presented on the left side in Figure 6.10. By selecting a week or month value, the user is able to drill-down to the next lower resolution level day and week respectively. The main advantage of this approach in contrast to other techniques is the fact that the level of detail depends on the importance of data values resulting in very flexible visualizations. The user may with little effort change some parameters to get higher level or lower level views or to change segment and subsegment sizes.

6.1.7 Conclusion

We focused on scalability issues of existing visualization techniques in the context of visual exploration of large temporal data sets. Since many of today's visual exploration techniques do not scale well on large data sets, we provided a hierarchical technique for reducing data size by relevance driven data accumulation. We integrated these techniques into an ECG data analysis framework and the CircleView approach and applied them to ECG and financial data. Our Multi-resolution technique can in general be combined with most visualization techniques, which may be a topic for future research.

6.2 VisImpact: Business Process Analysis

Business operations involve many factors and relationships and are modelled as complex business process workflows. Figure 6.11 shows such a workflow for a simplified product order scenario. The execution of these business processes generates vast volumes of complex time related data. The operational data are instances of the process flow, taking different paths through the process. The goal is to use the complex operational information to analyze and improve operations and to optimize the process flow.

We introduce a new visualization technique, called *VisImpact* that turns raw operational business data into valuable information. *VisImpact* reduces data complexity by analyzing operational data and abstracting the most critical factors, called impact factors, which influence business operations. The analysis may identify single nodes of the business flow graph as important factors but it may also determine aggregations of nodes to be important. Moreover, the analysis may find that single nodes have certain data values associated with them which have an influence on some business metrics or resource usage parameters. The impact factors are presented as nodes in a symmetric circular graph, providing insight into core business operations and relationships. A cause-effect mechanism is built in to determine "good" and "bad" operational behavior and to take action accordingly. We have applied *VisImpact* to real-world applications, fraud analysis and service contract analysis, to show the power of *VisImpact* for finding relationships among the most important impact factors and for immediate identification of anomalies. The *VisImpact* system provides a highly interactive interface including drill-down capabilities down to transaction levels to allow multilevel views of business dynamics.

6.2.1 Introduction

What is the cause of an unfulfilled contract? If the problem continues for a day, how much will it cost? Which customer orders will be impacted? To answer such questions, many research efforts have focused on how to transform the business process data, as logged by the IT services, into valuable information. However, due to multiple factors and the complexity of business operations, analysts face the challenge of understanding the underlying data and finding the important relationships from which to draw conclusions.

Charts, scatter plots, and spreadsheets are commonly used, but are inappropriate to visualize relations between multiple business parameters in a single view. Thus, analysts usually need to view pages of charts and reports to obtain the information they need to make informed decisions. However, the degree of difficulty increases as business operation complexity grows. Therefore analysts need tools that provide them insight into important business factors in the context of their overall business operations, e.g. business operations these business factors depend

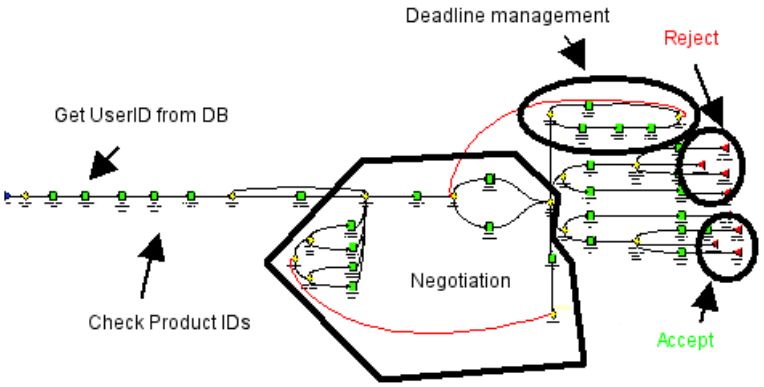


Figure 6.11: A Product Order Activity Workflow

on or the impact of the changes of business factors.

In this context, visualization technology plays an important role and a number of advanced visualization techniques for business impact analysis have been proposed in the past. Parallel coordinates [ID90], for example, is a geometric projection technique used for multidimensional visualization and automatic classification. SeeSoft's line representation technique [ESS92] is used to visualize Year 2000 program changes. ILOG JViews is a tool for analyzing workflow processes. E_Bizinsights is used to provide dimension-based structures for web path analysis. All these example techniques can effectively display data for business decision making. The techniques start with the presentation of whole business operation and provide drill-down capabilities to the analyst to obtain detailed information.

We introduce a new visualization technique called *VisImpact*. This technique extends existing approaches by providing an integrated analysis of business process schema and business process instances in order to improve business operations. First, *VisImpact* analyzes the business process schema and data to identify the important factors which influence the business metrics and resource parameters. These important impact factors and the corresponding business process instances are then represented as nodes and lines on a symmetric circular graph to display the important relationships and details of the process flows.

VisImpact is different from existing workflow visualizations in that existing techniques display the process workflow schema as a standard graph and use standard charts to plot highly aggregated summaries of the workflow execution data. Our approach in contrast uses advanced analysis techniques to determine the important impact factors which are then visualized as circular graphs.

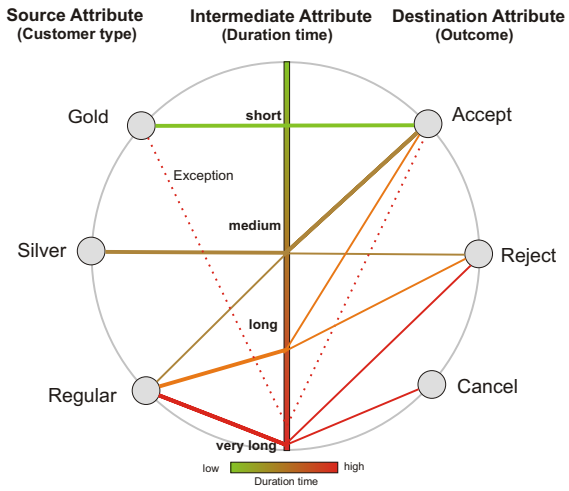
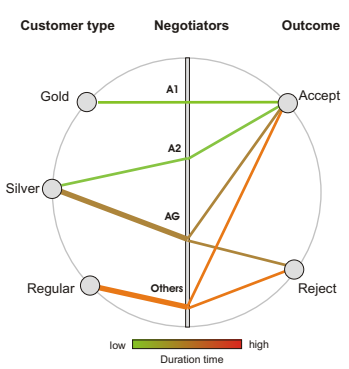


Figure 6.12: Product Order Activity by Process Duration Times (color represents process duration time)

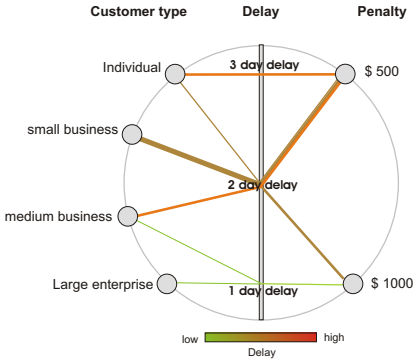
6.2.2 Basic Idea of *VisImpact*

Business operation data is inherently complex, most often too complex to be directly visualized. Usually the business operations consist of many steps and alternatives and every data instance may take a different path through the process. To understand the business process and the impact of certain parameters, in the first step *VisImpact* analyzes the process data. As an introductory example, Figure 1 shows a simplified version of a product order activity process schema. Note that this business process is a very simple one; realistic business processes are at least 10 times larger. To simplify the complexity of the visualization, *VisImpact* automatically abstracts important attributes, called impact factors, using data mining techniques ranging from statistical correlation analysis and partial matching to techniques used in clustering and classification analysis. Additionally, the analyst may steer the analysis process by interactively selecting business attributes and metrics of interest manually, and the system detects impact factors for the corresponding selection. The analysis results are then stored in an impact factor matrix.

Finally, *VisImpact* transforms these impact factors to nodes, with lines between nodes on a symmetric circular graph, representing a portion of the business operation. The goal was to provide an appropriate visual layout, that is able to represent an abstraction of the underlying complex workflow, but at the same time shows the relationships between discovered relevant business impact factors for further visual analysis. In general, each business workflow can be modelled as directed graph containing three different categories of nodes: start nodes, end nodes, and inner nodes including different node types like decision or action nodes. The edges



(a) Order Activity by Negotiators



(b) Order Activity by Delay

between the nodes define the process flow. Typically the process flow starts at a start node, passes certain inner nodes and ends at an end node. We adapted this principle by generating circular graph layouts consisting of three types of nodes, representing three types of nodes of the underlying business process schema. The edges between the nodes represent the process instances and their color represents an additional attribute, e.g. a business metric. Thus, each circular graph visualizes four different attributes:

- *Source Attribute* for partitioning the left side of the circle
- *Intermediate Attribute* for partitioning the center axis
- *Destination Attribute* for partitioning the right side
- *Color Attribute* (colored lines) for visualizing business metrics, such as response time, dollar amount, and contract fulfillment degree.

Each process instance is shown as a line connecting source, intermediate and destination nodes. *VisImpact* records the process instances in the process flow map , so that analysts are able to track the cause-effect paths in real time. Each circular graph represents a special impact relationship, and multiple circular graphs are linked together to show whole business process operations.

In the product order activity example, *VisImpact* uses a clustering algorithm to identify types of customers (Gold, Silver, Regular) and order processing duration times (very long, long, medium, short) and present it in relationship to the outcome of the decision (Accept, Reject, Cancel) as shown in Figure 6.12.

The lines are colored according to the duration time. In the example, it is obvious that the higher the rank of the customer, the faster the processing of the order and the higher the likelihood of an acceptance decision. Since the data is shown on an instance level (and not in an aggregated form) it becomes clear that for a regular customer the duration time varies largely and that the likelihood of

a reject decision increases with higher processing durations. Note that there are also some exceptions to this general tendency which can be seen by the red line that ends in the accept node. The second *VisImpact* graph (see Figure 6.13(a)) shows a second impact relationship, namely between specific instances of the negotiation nodes, duration time, and outcome of the decision. Note that in this case the negotiators are partitioned into two specific negotiators *A1* and *A2*, then a group of negotiators AG, and all other negotiators *Others*. The four groups have been identified by the automatic algorithm based on the similarity of the business process data instances.

In the third graph (see Figure 6.13(b)), we show how *VisImpact* can be used to analyze product order contract fulfillment, i.e. the ability to deliver certain products within a specified time period. There is a penalty cost if a contract is delayed beyond a certain number of days. The *VisImpact* system determines that penalty cost is closely related to specific delay times. Note that for correlating penalty cost and delay time to customers, the system proposes a different classification of the customer nodes, namely *large enterprise, medium business, small business,* and *individual*.

To make large volumes of data easy to explore and interpret, *VisImpact* links multiple circular graphs and records the path of process instances in the process flow map. Therefore, analysts can quickly click on a node or a line to observe all linked operation flows across different graphs in real-time.

6.2.3 Formal Definition of *VisImpact*

In the following, we formally introduce the techniques used to generate *VisImpact* visualizations. The first step is the identification of all relevant impact relationship. For this step we perform a global correlation analysis and use partial matching, cluster and classification analysis techniques. The result of this step are triples of related attributes which are then visualized as nodes and the instances are represented as edges of a graph. The problem is to find a good graph layout that supports human problem solving and decision-making processes. There are some general requirements that graph layouts for human consumption should fulfil, known as aesthetics criteria [KD01]. Some important criteria for *VisImpact* are display symmetry, edge crossing reduction, uniform vertex distribution, and uniform edge lengths. Additionally, the layout should present an ordering of the nodes corresponding to the business parameters and present an abstraction of the data. In addition, the layout should allow a visualization of large data volumes. A circular layout is chosen, because it provides a good compromise between all requirements [Eic99].

Determining the Impact Relationships

The first step of *VisImpact* is to determine the most important impact relationships. For this step we use (semi-) automatic data mining techniques, namely

statistical correlation analysis, partial matching techniques, as well as cluster and classification analysis.

Statistical Correlation and Similarity Analysis

First, we determine the pair-wise global correlations among all measurements as given by Pearson's correlation matrix [Pea96]. Pearson's correlation coefficient r between bivariate data, A_{1i} and A_{2i} values $(i = 1, \ldots, n)$ is defined as

$$r = \frac{\sum_{i=1}^{n}(A_{1i} - \bar{A_1})(A_{2i} - \bar{A_2})}{\sqrt{\sum_{i=1}^{n}(A_{1i} - \bar{A_1})^2 \sum_{i=1}^{n}(A_{2i} - \bar{A_2})^2}}$$

where $\bar{A_1}$ and $\bar{A_2}$ are the means of the A_{1i} and A_{2i} values, respectively. If two dimensions are perfectly correlated, the correlation coefficient is 1, in case of an inverse correlation -1. In case of a perfect correlation, we can omit one of the attributes since it contains redundant information. In most cases, however the correlations are not perfect and we are interested in high correlation coefficients and select sets of three highly correlated attributes to be visualized in *VisImpact*. Other statistical correlation coefficients such as the Spearman correlation [KG90] are provided in *VisImpact* as well.

An available alternative for adjacently depicting similar dimensions is to use the normalized Euclidean distance as a measure for global similarity Sim_{global} defined as

$$Sim_{Global}(A_i, A_j) = \sqrt{\sum_{i=0}^{N-1}(b_i^1 - b_i^2)^2} \tag{6.1}$$

where $b_i^j = \frac{a_i^j - MIN(A_j)}{MAX(A_j) - MIN(A_j)}$

In order to become more robust against outliers, instead of using MAX (the 100%-quantile) and MIN (the 0%-quantile), we use the 98% and 2% quantile of the attribute. The global similarity measure compares two whole dimension such that any change in one of the dimensions has an influence on the resulting similarity. The defined similarity measure allows it to determine triples of similar attributes for the successional visualization. Since in general, computing similarity measures is a non-trivial task, because similarity can be defined in various ways and for specific domains, the modular design of the *VisImpact* system allows the integration of specific similarity measures with little effort, like similarity measures proposed in the context of time series data [AFS93, ALSS95] or similarity measures presented in [BKK97].

Partial Similarity In real business process applications global similarities are rare, since in most cases correlations only occur for certain subsets of the data.

Imagine for example two business measures over time, like the duration time for Gold and Silver customers in Figure 6.12. There may be short periods where the two measures show a similar behavior e.g. because of some global development. However, it is unlikely that they behave similar over days or weeks. In the impact relationship analysis we therefore have to analyze the data for partial similarities. In our application scenario, we are especially interested in periods where two attributes behaved similar. Thus, given the two variables A_k and A_l, the synchronized partial similarity[ABK98] measure is employed , to detect pairwise attributes with periods of similarity in the data:

$$Sim_{Syn}(A_k, A_l) = \underset{i,j}{max}\left\{ (j-i) | (0 \leq i < j < N) \wedge \sqrt{\sum_{z=i}^{j} c_z} < \epsilon \right\} \tag{6.2}$$

where $c_z = (b_z^k - b_z^l)^2$

with b_i^j defined as above and ϵ is some maximum allowed dissimilarity. This partial similarity measure uses the length of the longest sequence which is at least ϵ-similar (under scaling and translation invariance). Triples of attributes with pairwise maximum Sim_{Sync} values are then selected for *VisImpact* analysis. Depending on the application, the partial similarity may also be an Unsynchronized Partial Similarity[ABK98]. In this case, two dimensions do not have to be similar at the same "time but in an arbitrary time frame of the same length. Since computing partial matchings is a time-consuming process, most approaches like [YWY00, FRM94] also use some heuristics and index structures to speed up the computation [HDY99], that will be considered in future extensions of *VisImpact*.

Cluster analysis For some attributes, the parameter values are continuous (such as dollar amount), for others, there are large numbers of categorical values (such as expense requestors). In order to perform a useful impact analysis, it is important to partition the value ranges appropriately. Cluster analysis can help to do this based on the characteristics of the data instances. The cluster analysis may, for example, find out that - based on the characteristics of their product order flows - the companies may be partitioned into three groups (gold, silver, regular) and the negotiators into two single ones (A1, A2) and two groups (AG, Others).

There are a large number of clustering methods which have been proposed in the literature. One of the most general techniques is kernel density estimation [HK99]. In kernel density estimation, the influence of each data point is modelled using a kernel function, and the overall density of the data is calculated as the sum of the kernel functions of all data points. Clusters can be derived from a density function by density based single linkage or hierarchical clustering. Due to the large number of analyses which need to be performed in the *VisImpact* framework, we have to use an efficient implementation of kernel density estimation, and therefore the DENCLUE algorithm [HK98] is employed.

Classification Analysis In some applications, the goal of the data exploration is to understand the relationship between the business process data and some specific business metrics such as response time, dollar amount, or degree of contract fulfillment. If the analyst is interested in a specific business metric, we can perform the automatic analysis with the business metric as target attribute. The task is to find the business process parameters which are best predicting the outcome of the target attribute. A well-known heuristic for this task is the GINI index, which is also used in decision tree construction. Given a business metric B which is partitioned into a disjoint set of k classes (e.g. accept, reject) or value ranges (e.g. large, medium, small) denoted by $C_1, \ldots, C_k$, $(B = \bigcup_{i=1}^{k} C_i)$, then the GINI index of an attribute A which induces a partitioning of A into $A_1, \ldots, A_m$ is defined as

$$InfoGain_{GINI}(B, A) = \sum_{i=1}^{m} \frac{|A_i|}{|B|} GINI(A_i) \tag{6.3}$$

where

$$GINI(A_i) = 1 - \sum_{j=1}^{k} \left[\frac{|C_j|}{|A_i|} \right]^2 \tag{6.4}$$

The *InfoGain* is determined for all attributes and attribute combinations and the two attributes with the highest *InfoGain* with respect to the target attribute B are chosen for visualization. Alternatively, we use the attribute A_x with the highest *InfoGain* and then repeat the calculation with A_x as target attribute to find the second attribute to be displayed.

The Circular *VisImpact* Graph

The business impact visualization is defined as a graph $G = (V, E)$, where V is a set of nodes connected by edges E. The node set V is partitioned in k subsets $V_1, \ldots, V_k$ depending on k partitioning attributes. Each edge $(u, v) \in E$ implies either $u \in V_i$ and $v \in V_{i+1}$ or $u \in V_{i+1}$and $v \in V_i$, $i \in 1, \ldots, k-1$. The nodes V represent the set of data items for the corresponding k classes of V and the edges represent the relationships and interactions between them. An edge can have at least two attributes, showing characteristics of the relationship, represented by width and color of the edge.

In the *VisImpact* System, a special case of circular graph is used, where the node set V of the graph consists of three subsets V_1, V_2, V_3, $V = V_1 \cup V_2 \cup V_3$, $(V_i \cap V_j = \emptyset \Rightarrow i \neq j)$. The set of source nodes V_1, is determined by the first attribute (source attribute). The second attribute (intermediate attribute) determines the subset V_2 of intermediate nodes, and the third attribute (destination attribute) determines V_3, the set of destination nodes. Corresponding to the definition of the general circular graph, there exist only edges $e = (u, v) \in E$ between V_1 and V_2 or V_2 and V_3. In order to present the given nodes and edges in a circular layout, let $C = (x, y, r)$ be a circle with center (x, y) and radius r in the $2D$-plane. We

introduce a screen positioning function $f : V \rightarrow R^2$, which determines for each node $v \in V$ the x/y-position $(v.x, v.y)$ on the circle.

Since we want to visualize the relations and interactions between three sets of nodes, we divide the circle C in three regions to place the nodes from the three sets V_1, V_2, V_3. The nodes of V_1 are placed on the left side and the nodes of V_3 are placed on the right side of the circle, which means for all nodes $v \in V1 \cup V3$ holds:

$$C.r^2 = (v_x - C_x)^2 + (v_y - C_y)^2$$

For all nodes $v_i \in V_1$ is $v_i.x - C.x < C.x$ and for all nodes $v_j \in V_3$ is $v_j.x - C.x > C.x$. The nodes of V_2 are placed on the center axis of the circle, which means on a line from Point $P1(C.x, C.y - C.r)$ to Point $P2(C.x, C.y + C.r)$, so that for all $v_j \in V_2$

$$v_j.x = C.x \text{ and } C.y - C.r < v_j.y < C.y + C.r$$

Computing the Node Positions The placement of nodes on the circle axis is straight forward and depends only on the selected mapping. To place nodes on the left or right half of the circle, the positioning function f employs the radian ϕ to compute the position for each node depending on the selected mapping, as shown in Figure 6.13. For quantitative data, linear mapping is used to map the data points to the left side, the right side or the center axis of the circle, and the radian is determined accordingly. Optional, the data points can be placed in an ordered equidistant manner. This is especially useful for categorical data or in cases where the analyst is more interested in the process flow than in exact node values. The radian ϕ for a node $v_i \in V_1$ is then defined as follows:

$$\phi = \pi - \alpha \left[-\frac{1}{2} + \frac{i}{n} \right] , i = 0, \ldots, n$$

The angle α, $0 < \alpha < \pi$, describes the positioning area of the nodes, shown in Figure 6.13. In order to position the nodes of V_1 on the left side of the circle, (i.e. $0.5\pi < \phi < 1.5\pi$), we set $\alpha = c\pi, 0 < c < 1$. For placements on the right side of the circle, i.e. for positioning of all nodes $v \in V_3$, $\pi - \alpha$ has to be replaced by α in the equation above. The parameter c separates the nodes on the right and left half of the circle and the nodes in the middle. The term $\frac{i}{n}$ divides the drawing area, given by α, in n equidistant locations in order to place the n nodes from V_1. The radian ϕ is used to compute a position for each node $v_i \in V_1$:

$$v_i.x = C.x + \cos(\phi) \cdot C.r$$

$$v_i.y = C.y + \sin(\phi) \cdot C.r$$

The node positions for the nodes $v \in V_3$ can be computed analogical. Color coding and tool tip techniques are used to represent relevant node attributes.

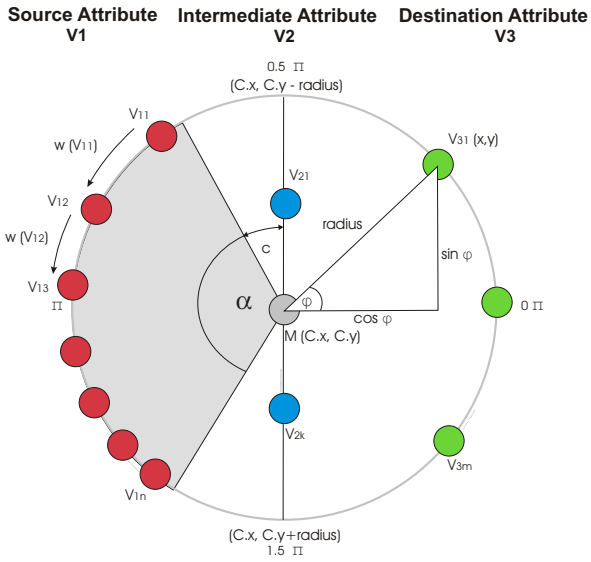


Figure 6.13: Computation of node positions on the circular *VisImpact* layout

Weighted Node Positions

In order to give important information in our visualization more attention, an optional weight function can be used. Instead of just ordering the nodes according to their values and then place the nodes on the circle in an equidistant manner, this weight function gives important nodes more space on the screen while less important nodes get less space, realized by a weighted computation of the radian ϕ, as shown in Figure 6.13. The weight $weight_i$ of a node $v_i \in V_l, i \in (1, \ldots, N)$, depends on a forth attribute A. We define the weight by the ratio of v_i's attribute $a_i \in A$ and the sum of all attributes $a_j \in A$, $|A| = N$, were $i \in (1, \ldots, N)$:

$$weight_i = \frac{a_i}{\sum_{j=1}^{N} a_j}$$

After computing a weight for each node, *VisImpact* orders the nodes by their weights and places them by starting at the top of the circle. The weighted positioning function w distributes the available space on the circle to the nodes by calculating a weighted radian ϕ_{weight} for each node $v_i \in V_1, i \in (1, \ldots, N)$:

$$\phi_{weight} = \pi - \alpha \left[-\frac{1}{2} + w(i) \right] \; , \; w(i) = \sum_{j=0}^{j<i} weight_j$$

In order to place the nodes in V_3 on the right side of the circle, $\pi - \alpha$ has to be replaced by α in the formula above.

Placement of Categorical Attributes In cases where the ordering of nodes in the *VisImpact* visualization is not implicitly given by the node values, e.g. for categorical attributes like *customer name* or *customer type*, the analyst is typically only interested in the process flow between certain attributes. The goal then is to find a circular node layout that reduces edge crossings, since they may reduce the readability of the resulting graph. Therefore a placing method that reduces edge crossings by rearranging single nodes is integrated into the *VisImpact* system to place nodes with no implicit ordering. Since in general, the problem of finding vertex orderings that minimize edge crossings in a layered graph is NP-hard, even for 3-layered graphs as used by *VisImpact* [EW94], heuristics are needed to solve even moderately sized problems.

Let $G = (V, E), V = V_1 \cup \ldots \cup V_k, V_i \cap V_j = \emptyset \Leftrightarrow i \neq j$, be a general circular graph as described above. An ordering layer $V_i, i \in (1, \ldots, k-1)$ is specified by a permutation π_i of V_i. We express the ordering of V_i by the permutation π_i. Let $cross(G, \pi_1, \ldots, \pi_k)$ be the number of edge crossings in a straight line drawing of G given by $\pi_1, \ldots, \pi_k$. The minimum number of edge crossings that can be achieved by reordering the vertices in $V_1, \ldots, V_k$ is denoted by $opt(G)$:

$$Opt(G) = \min_{\pi_1,\ldots,\pi_k} cross(G, \pi_1, \ldots, \pi_k)$$

Having three sets of nodes V_1, V_2, V_3, *VisImpact* computes a minimal edge crossing by dividing this 3-layered crossing minimization problem in two 2-layered One Sided Crossing Minimization Problem:

$$Opt'(G) = \min_{\pi_i,\pi_{i+1}} cross(G, \pi_i, \pi_{i+1}) \ , \ i = 1, 2$$

$Opt'(G, \pi_i)$ denotes the minimal attainable number of edge crossings by fixing the permutation of V_i and reordering the nodes of V_{i+1}. The Barycenter heuristic [Sug81] is used to compute such a node ordering. The basic idea of this heuristic is to simply compute the average position, i.e., the Barycenter, for each node and then sort the nodes according to these numbers. In typical application scenarios not all 3 attributes will be nominal or categorical without given orders, which restricts the crossing minimization process.

6.2.4 The *VisImpact* System

System Architecture and Components

To analyze large volumes of transaction data with many impact factors, *VisImpact* has been integrated into the visual data mining system *VisMine*[HDH99]. The system uses a web browser with a Java activator to allow real-time interactive visual data mining over the web. The *VisImpact* architecture contains four basic components:

1. Abstract Component

   The abstract component of *VisImpact* derives an *impact factor matrix* from the input data. The system can then automatically show a number of VisImpact visualizations based attributes with high impact relationships. Alternatively, the analysts can select a pair of impact factors from the matrix based on the knowledge of the user and the application requirements. Then, from the pair of impact factors, *VisImpact* automatically abstracts the third impact factor that has the highest impact value with the selected pair of impact factors. In addition, *VisImpact* provides a *control window* to allow analysts to real-time select impact factors for further analysis.

2. Layout Component

   This component orders, groups, maps, and weights the abstracted impact factors and transforms their relationships to lines between two nodes according to the *process flow map*. Nodes and lines are laid out on a symmetric circular graph. The width of the line represents the number of lines with the same process flow. The color of a line is the average value of a data item. Nodes and lines on the circular graph are weighted by the average value of a data item. Nodes with higher weights are given more space on the graph. The label of the nodes with the highest weights is colored red. Lines with the highest weights are drawn last to avoid overlapping.

3. Interaction Component

   To make the circular graph easy to explore and interpret, *VisImpact* provides fade-in, fade-out, clicking, and drill down capabilities.

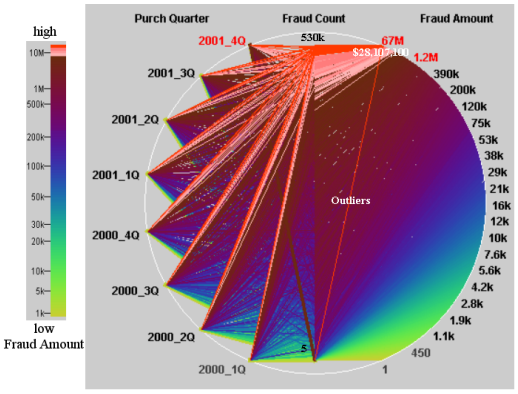
4. Extension Component

   Additional circular graphs are generated as the number of selected attributes grows.

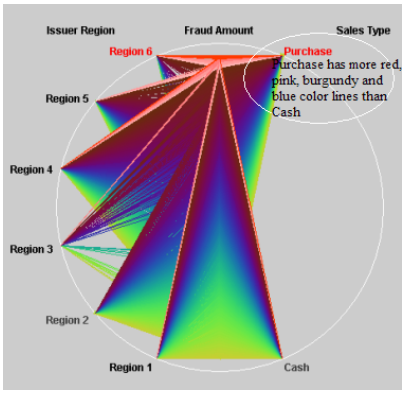
Anomaly Detection

VisImpact employs a *process flow map* to link related process flows and relationships across different circular graphs. *VisImpact* instantly detects exceptions by finding red lines (exceed threshold) or crossed lines (anomaly) in a graph. For example, in fraud analysis (Section 5.1), a high fraud amount usually is associated with a high fraud count. There could be a potential problem if a high fraud amount occurs with a low fraud count as shown later in Figure 5A. *VisImpact* provides the following visual capabilities:

- Fade-In and Fade-out
  VisImpact allows analysts to focus on an outlier and fade in related process paths. The unrelated nodes and flows are faded out. The analyst can easily



(a) *Fraud Distribution by Purchase Quarter* Impact Factors are Quarter, Fraud Amount and Fraud Count. Each line is a transaction, color represents the value of the Fraud Amount: Fraud increases with time (e.g. more red lines in each quarter). 2001_4Q has the highest fraud amount(red label). An outlier: a red line crosses from low Fraud Count to high Fraud Amount (other lines are nearly parallel - high correlation)



(b) *Fraud Distribution by Region* Using Fraud Amount from Figure 6.14(a) and choose two other impact factors: Region and Sales Type. Region 6 (red) has the highest fraud amount (on the top of the circular graph, more red, pink, and burgundy; less green lines). Most fraud comes from Purchase vs. Cash. Purchase has more red, burgundy, and blue lines than Cash.

Figure 6.14: Fraud Analysis based on credit card data

discover the source of the problem by tracing the lines starting from the anomaly.

- Drill Down
  The analysts select a single node or a single line to drill down to the transaction level to display multilevel views of business dynamics.

6.2.5 VisImpact Applications

We have experimented with *VisImpact* for fraud analysis, a case study is presented in [KSDH05], and service contract analysis using real-word business data.

Fraud analysis

Fraud is one of the major problems faced by many companies in the banking, insurance, and telephony industries. Over $ 2 billion in fraudulent transactions are processed yearly on electronic payments. Transforming raw transaction data into valuable business operation information to enable fraud analysis will save companies millions of dollars. Fraud analysis specialists require tools that help

them to better understand fraud behavior and impact factors as well as to identify unusual exceptions. Typical questions in fraud analysis are:

1. What is the fraud growth rate in recent years and what are the impact factors?

2. Which sales region and sales type has the most fraud?

3. Are there any outliers and what is their cause-effect?

To address these three questions, *VisImpact* first selects three highly correlated attributes from the impact factor matrix such as Purchase Quarter, Fraud Amount (aggregated), and Fraud Count. Using these three impact factors, *VisImpact* lays out the nodes and flows in a circular graph as shown in Figure 6.14(a). Figure 6.14(a) shows that there are high correlations (more parallel lines) between Fraud Amount and Fraud Count. Colors represent the value of the Fraud Amount; red represents a fraud amount that is in the top 10%. Most important, there is an outlier (a red line) crossing from low Fraud Count (5 counts) to a very high Cash advance with a fraud amount of $ 28,107,100. This exceptional transaction might be a potential problem or error.

To understand which sales regions or sales have the most fraud, the analyst selects the Region as the source node and Sales Type as the destination node from the user domain knowledge and draws a second circular graph as shown in Figure 6.14(b). From Figure 6.14(b), it can be learned that Region 6 has the highest Fraud Amount with more red, pink, burgundy and blue lines than other regions. Purchase has a higher fraud amount than Cash advance. This is because Purchase has many more red and burgundy lines than Cash advance.

To find outliers and their root-cause, *VisImpact* uses the process flow map to identify related operation paths of a transaction record and to discover exceptions. Most interestingly, an outlier is seen as a red line crossing from Cash to the high Fraud Amount. The analyst can easily move the pointer to find detailed information about this outlier, such as the amount and purchase quarter. Investigating further, the analyst can select the Region 6 node to focus only on Region 6 fraud. The fraud from other regions is faded out as shown in Figure 6.15. The analyst can quickly see that the outlier comes from Cash advance. This capability to trace the process flow of a transaction is crucial for finding the cause-effect relationship of outliers. Using the above information, the company is able to place strict control on certain regions (countries) and credit card usages. After better understanding the source of the fraud, the company will be able to take preventive action.

Service Contract Analysis

All businesses have relationships with customers and suppliers; they execute business processes to obtain services from suppliers and add value to deliver services

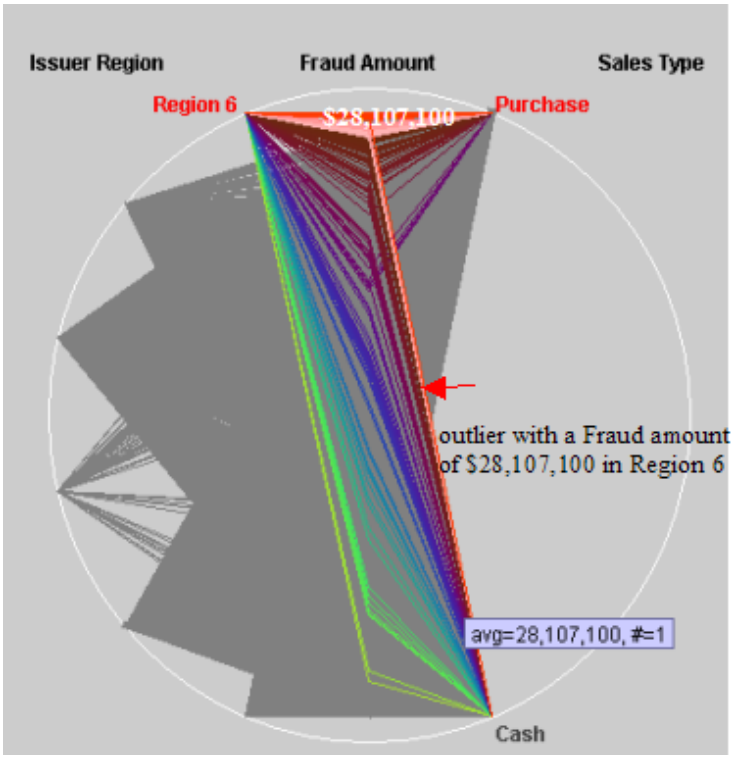
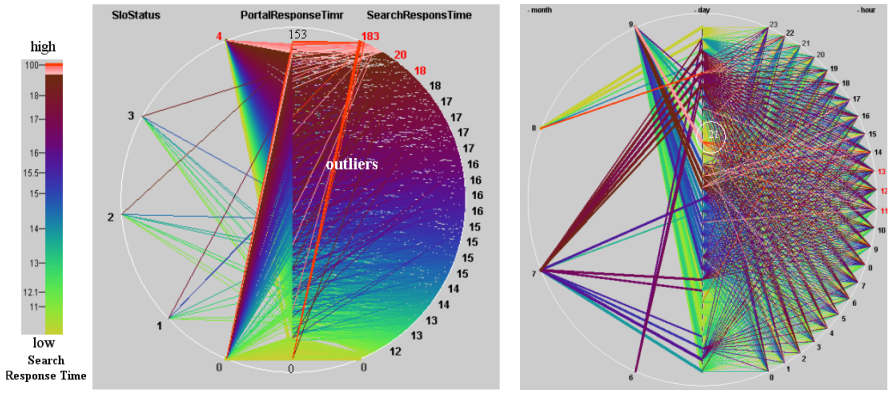


Figure 6.15: *Finding the Cause of the Outlier*
Region 6 from Figure 6.14(b) is selected. The outlier is seen as a red line crossing from Cash to the amount of $ 28,107,100. The outlier is linked to Region 6 from the left half of the graph. The outlier is also linked to Fraud Count 5 and 2000_4Q in Figure 6.14(a). The cause of the outlier is a cash advance which happened in 2000_4Q, Region 6 with a Fraud Amount of $ 28,107,100 and Fraud Count of 5.

to customers. Such service processes are usually modelled by Service Level Objectives (SLOs) and contracts [Sah02], stipulated between customers and suppliers. A contract typically contains SLOs defining what service should be delivered with what level of quality and within what time period. An important question business managers need to pursue is whether their business operations are fulfilling the SLOs. This is a difficult problem, often complicated by service performance (e.g. response time, server availability) involved in the execution of business operations.

We have applied *VisImpact* to a real-world, large-scale data set of service contract analysis in an effort to better understand SLO operational flows, distributions, and anomalies. In this application, the SLO status indicates the probability of a contract becoming unfulfilled (violated), with 0 being the most probable and 4 being the least probable. The data set contains 10,061 service transactions with over 50 SLO impact factors such as SLO status, portal response time, search re-

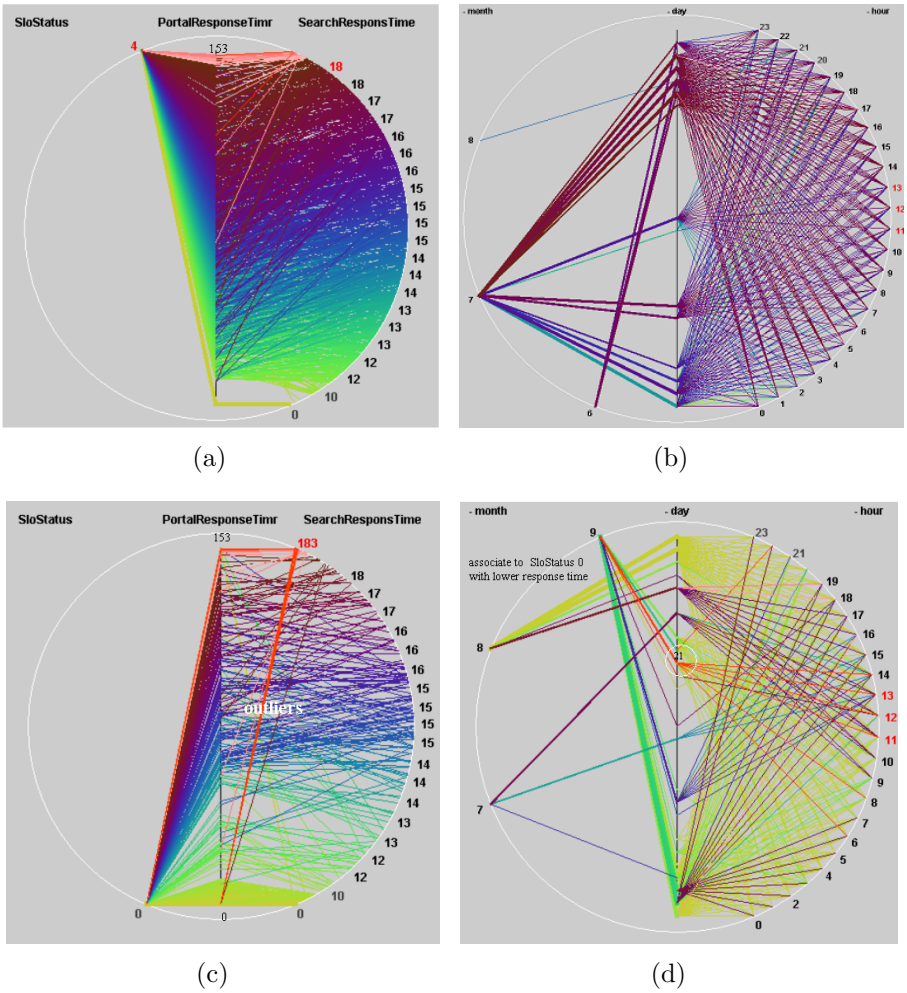


(a) (b)

Figure 6.16: *Service Contract Process Flows and Distribution Over Time* Portal Response Time and Search Response Time are highly correlated as seen by nearly parallel lines in 6.16(a). Lines with the highest Search/Portal Response Times (top 10%) are colored red. Outliers are shown by lines crossing from low Portal Response Time to High Search Response Time). SLO Status 4 has the highest Search Response time in 6.16(a) (more red, pink, burgundy). Month 7 and day 21 have the highest Search Response Time - more blue and burgundy in month 7, most red lines in day 21. High Search Response Times occurred after the 10th day of a month (more red, pink, burgundy, and blue).

sponse, month, day, and hour. *VisImpact* maps nodes to SLO impact factors, lines to service transactions, line widths to the number of service transactions, and colors to the values of selected impact factors (i.e., search response time). Nodes are placed in order according to the selected impact factors.

Operational Flows and their Distribution *VisImpact* first abstracts the three most highly correlated factors from the impact factor matrix and constructs a circular graph as shown in Figure 6.16(a). The source nodes show *SLO Status*, the intermediate nodes *Portal Response Time*, and the destination nodes *Search Response Time*. Nodes are connected with lines from the process flow map. The color of a line is the value of the *Search Response Time* (ms). Figure 6.16(a) shows that the SLO Status is highly impacted by both Portal and Search Response Times. A transaction with high values for Search Response Time often has high values for Portal Response Time as shown by the numerous nearly parallel lines. Note that there are major outliers, shown by the red lines crossing from a low Portal Response Time to a high Search Response Time.

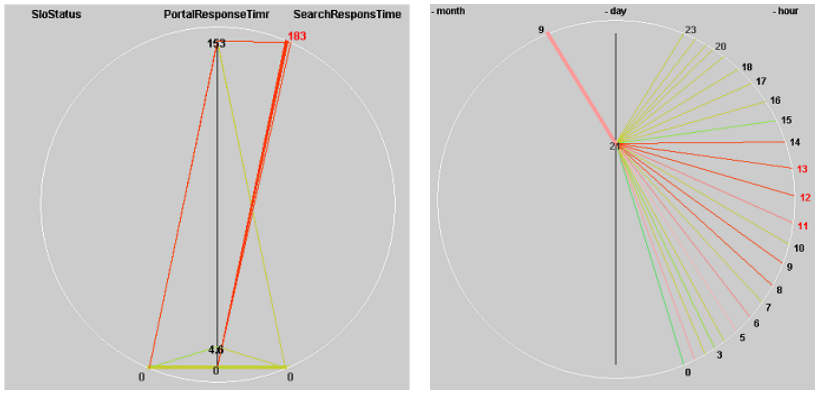


(a) (b)

(c) (d)

Figure 6.17: *Process Flows and Relationships between Multiple Impact Factors*

Graphs are generated when the analyst selects SLO Status 4/Status 0 in 6.16(a). 6.17(a) and 6.17(b) show that SLO status 4 is associated with higher response times, as seen by blue and burgundy. 6.17(c) and 6.17(d) show that SLO status 0 is associated with lower response times as seen by the yellow and green colors. An outlier is detected in Figure 6.17(c) and a highest Search Response Time node day 21 in 6.17(d).



(a) (b)

Figure 6.18: *Discover the Cause of Anomalies (outliers)*
6.18(a) and 6.18(b) are generated when the analyst selects the node on month 9 day 21 (linked with most red lines in 6.18(b)). The lines from the anomalies in 6.18(a) are linked to the month 9, day 21, and hours (8-14) node in Figure 6.18(b). All unrelated lines are faded out for easy identification. The analyst is allowed to move the pointer on the red lines and nodes to display transaction record level information, such as finding server availability in this case.

Time Dependency *VisImpact* generates a second circular graph to show the time dependency as presented in Figure 6.16(b) from user domain knowledge. The transaction process flows in Figure 6.16(b) are tightly linked to SLO Status in Figure 6.16(a). In Figure 6.16(b), the source nodes are month (6,7,8,9), the intermediate nodes are days (1-31), and the destination nodes are hours (0-23). Figure 6.16(b) shows search response time distribution over time. The color denotes the Search Response Time.

Process Flow Relationships between multiple Circular Graphs *VisImpact* helps to discover that the Search Response Time is the potential root-cause of the unfulfilled SLOs. This is seen through the linking of process flows across two circular graphs. As shown in Figures 6.17(a)/6.17(b) and 6.17(c)/6.17(d), we are able to verify this relationship because (a) in Figure 6.17(a)/6.17(b) the higher probability SLO status (e.g., SLO Status 4) is associated with slower response times, as seen in the blue and burgundy colors in Figure 6.17(b) that show the correlation and (b) in Figures 6.17(c)/6.17(d) the lowest probability SLO status (e.g. SLO Status 0) is associated with faster response times, as seen in the yellow and green colors in Figure 6.17(d).

Detection of anomalies among impact factors One of the key functions of *VisImpact* is to detect process flow anomalies including outliers. In Figure 6.16(a), *VisImpact* helps to detect outliers which are shown as thick red lines drawn from high Search Response Times to low Portal Response Times. After the analyst selects the red line and fades out all unrelated connections, a serious search response time problem (occurring in month 9, day 21) is clearly shown in Figures 6.18(a) and 6.18(b). The analyst can move the pointer to drill down to the detail transaction level to find out that the problem occurred at a time when a search engine was unavailable, which caused a long search time for all previously entered transactions. Using *VisImapct* as a real-time monitoring system, these anomalies can be addressed immediately before the SLO violation probability becomes worse.

6.2.6 Evaluation and Comparison

In this section we evaluate the results achieved by *VisImpact* and compare our technique to existing approaches. Of course it is a difficult task to measure the quality of visualization results, since a definitive and strong set of methodologies for measuring the "goodness" or the value of a given visualization is still lacking [MHNW97]. First attempts for providing quality measurements for scatterplots are provided in [WAG05], but there is still a lot of research in this field necessary. Therefore our evaluation focuses on the comparison of *VisImpact* with other techniques for multivariate data analysis, in particular Scatterplots and Parallel Coordinates [ID90].

To investigate the usability of the *VisImpact* approach, future work will include an user study, as first user feedback was very promising. The main advantage of the proposed *VisImpact* approach is the integration of automated analysis techniques into the visualization process to focus on relevant attributes or groups of attributes in the underlying multivariate datasets and thus provide abstract presentations of relevant parts of the data. This allows it to break complex Business operations down to groups of relevant Business attributes which allows a better Business analysis. The analysis results are provided using an intuitive radial visualization layout. The analyst may interact with the system for further data analysis, e.g. by using drill down and roll up capabilities. To show the benefit of the new technique we used the dataset introduced in Section 5.2 and produced Scatterplot matrices and Parallel Coordinates to analyse the data. Both techniques are commonly used for analysing multivariate data.

Figure 6.19 shows a scatterplot matrix of the data set introduced in Section 5.2. The diagonal contains the attribute names and histograms showing the data distribution. Above the diagonal the scaled absolute correlation values of the corresponding attributes are presented, below the diagonal the datapoints are shown. Since the scatterplot matrix shows as many scatterplots as there are pairs of parameters, it may be hard for the user to detect relevant patterns if the number of scatterplots is to high. The figure shows that there is a high correlation between

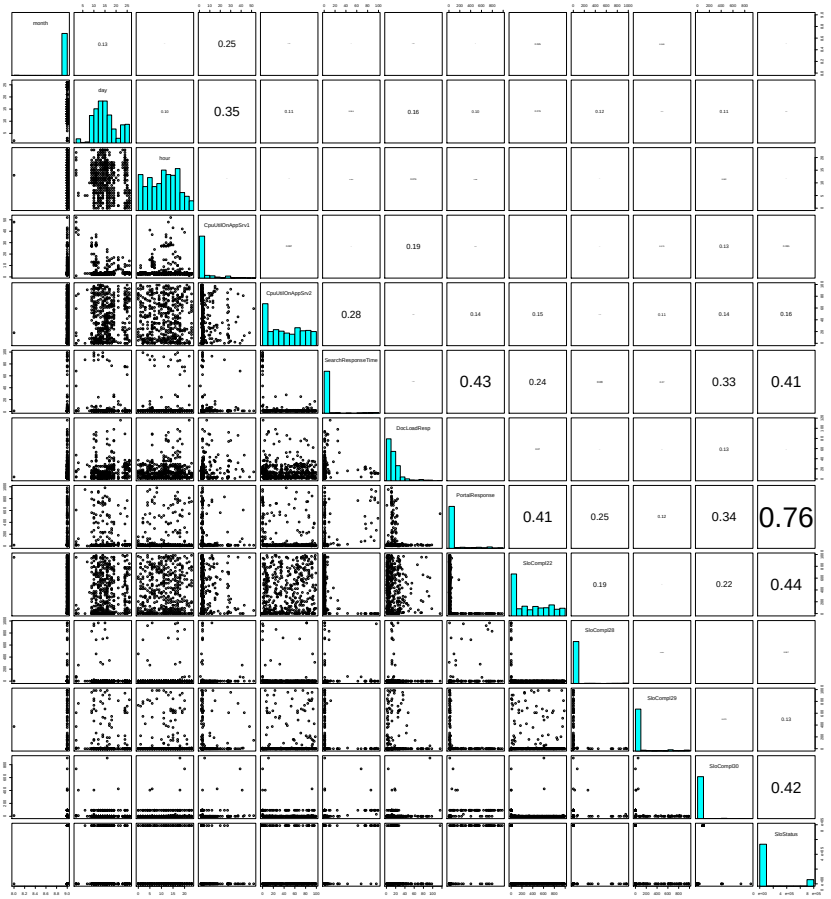
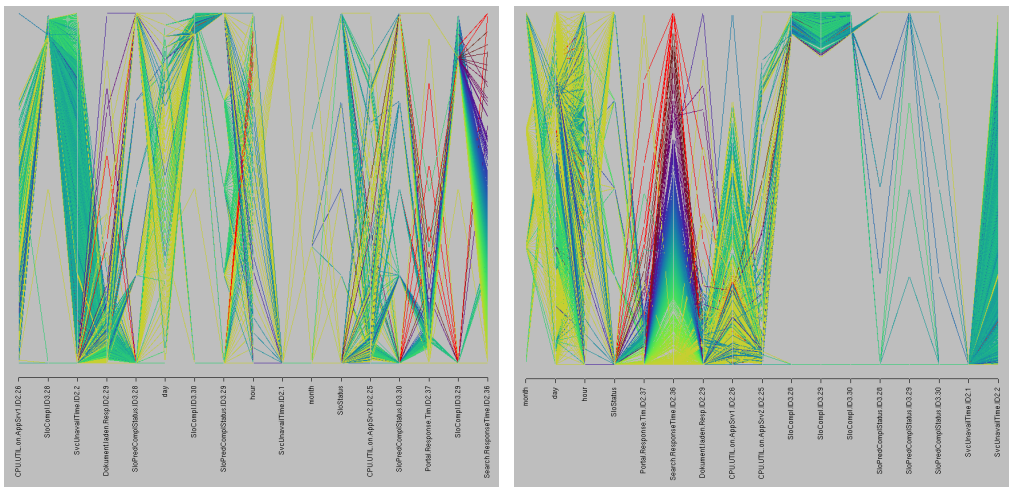


Figure 6.19: Scatterplot matrix showing a process flow data set and the correlation of it's attributes. The diagonal histograms show the data distribution

the attributes PortalResponseTime and SloStatus (0.76) and between PortalResponseTime and SearchResponseTime (0.43). Assuming that PortalResponseTime is the measurement of interest and the analyst is interested in attributes which are highly correlated to this measure, he has to construct a new scatterplot matrix that only shows highly correlated attributes, i.e. PortalResponseTime, SearchResponseTime and SloStatus. In *VisImpact* this step is done automatically as shown in Figure 6.16. Additionally, user feedback indicates that the circular layout in combination with the node ordering options in the *VisImpact* layout provides a much easier way for the analyst to get insight into the relations between the attributes, since due overplotting effects scatterplots are often hard to read without additional interaction techniques like brushing.

Another common method for analyzing and visualizing multidimensional data sets are Parallel Coordinates [ID90] and their various extensions. Basic idea of this

technique is to take all the axis of the multidimensional space and to arrange them in order but parallel to each other. Figure 6.20 shows the Parallel Coordinate plot of the Business data set introduced in Section 5.2. Experts in the analysis of such Parallel coordinate plots may derive a great deal of data understanding from these plots, but the interpretation can be strongly influenced by the order of the axes [Spe01]. There exist some tools which improve the usability of Parallel Coordinates plots by providing interaction and analysis techniques like the Xmdv tool [War94], but they do not take the special needs of business analysis processes, described in Section 6.2.2, into account and do therefore not provide the same functionality as the *VisImpact* system. And there is still the drawback that it is hard to identify correlations or clusters between attribute axes which are not located next to each other. Therefore the visflow graph visualization focuses on 3 single or aggregated attributes in each single view and provides automated techniques and interaction capabilities for selecting them from the available parameter set.



(a) (b)

Figure 6.20: Parallel Coordinate Plot for the Business data example introduced in Section 6.2.5, Color represents Search Response Time (same colormap as in Figure 6.16) . For non-analysis experts it may be difficult to extract similar information as shown in Figures 6.16, 6.17 and 6.18 from this Parallel coordinate plot due to the missing visualization capabilites provided by *VisImpact*.

For example it is difficult to detect relationships between the three attributes SloStatus, PortalResponse time and SearchResponse time in Figure 6.20(a), therefore the user has to rearrange the axis either automatically based on a correlation measure as shown in Figure 6.20(b) or manually. *VisImpact* automates this process, and presents for a given task in each single view only the 3 most rele-

vant attributes or groups of them and fades out unrelated attributes. Additionally the circular flow map layout provides more space for placing the data points e.g. according to the weight function, assuming that the diameter of the circle corresponds to the length of a single Parallel Coordinate axis. Ordering techniques integrated in *VisImpact* additionally help to improve the readability of the resulting visualization for a better visual information representation.

6.2.7 Conclusion

VisImpact is one of the first approaches to analyze complex time related workflow process data, by combining data mining techniques, which identify anomalies,relationships and relevant impact factors, and a visualization approach which provides meaningful data abstraction levels. The basic idea is to include the underlying workflow model as well as the workflow instance data, which correspond to certain paths through the workflow at runtime, in the analysis step. We are sure that this provides a more accurate way for analysing and optimizing complex business processes.

The *VisImpact* approach uses a new symmetric circular layout to generate graphs. To simplify the complexity of business operations, *VisImpact* uses correlation analysis, partial matching, cluster, and classifcation analysis techniques to abstract important business impact factors. *VisImpact* presents different impact factors in multiple graphs to view the data from different perspectives. We have addressed the special node placement and coloring problems to make the visualizations useful for discovering patterns and exceptions, and find the cause of business problems by tracing process paths of a transaction record. We applied the *VisImpact* technique to real data sets from a wide variety of applications, including fraud analysis and service contract analysis. The current experimental studies show significant advantages of the *VisImpact* technique in comparision to existing techniques in performing root-cause analysis. The *VisImpact* technique may be also very useful in other areas such as capacity planning and business financial activities.

Chapter 7

Visual Business Analytics of hierarchical Data

Many business data applications involve several hierarchies reflecting the inherent structure of the underlying domains. Typically, these hierarchies are presented through the data cube model, representing hierarchies as a lattice of cubes, with drill-down and roll-up capabilities. The ubiquitousness of hierarchical data structures reveals the importance of hierarchy visualization. Such visualization should convey the hierarchical structure and provide support for a wide variety of navigation operations, such as zooming / panning, and drilling-down / rolling-up. Most of recent hierarchy visualization research has concentrated on the challenge of displaying large hierarchies in a comprehensible form [YWR02]. In this context a number of techniques have been proposed, like node link diagrams [CAH87], cone trees [CRM91], radial layouts [SZ00], or hyperbolic layouts [Mun97]. Significant research attention was paid to the space-filling hierarchy visualization. These techniques aim to use display space very efficiently. Among the space-filling approaches, there are rectangular space-filling techniques, such as treemaps and its variations [JS91b, BSW02, BHvW00] and radial space-filling techniques [YWR02]. Compared to the rectangular space-filling techniques, the radial methods have been shown to work better in revealing hierarchical structures [BN01]. In our research we focused on the task of adapting hierarchical visualization techniques to Visual Analytics in terms of integrating automated methods that emphasize hierarchical features and then present them in visual form. As an application example we present a tool for interactive frequent pattern mining. Furthermore, we propose an approach to adapt hierarchical space filling layouts to temporal data, where it is not only important to convey the hierarchical structure, but it is also necessary to find appropriate layouts and alignments for effective analytical reasoning.

7.1 Visual Analytics of Frequent Patterns

Frequent pattern mining plays an essential role in many business analysis tasks including association-, correlation-, and causality analysis. In association analysis for example, the goal is to find interesting patterns and trends in a data base. Association rules are statistical relations between two ore more items in a data set which tell the analyst that the presence of some items in the database imply the presence of some other items in the same transaction with a certain probability called confidence c.

One of the most common application examples is market basket analysis, where analysts are interested in products which customers purchase together which in turn might be used to optimize the shelf layout of a supermarket or to adapt business marketing strategies accordingly. A popular analysis result was found in data from a US grocery chain. Using Data Mining software to study the behavior of their customers, interesting relationships were found between diapers, beer, gender of the buyer, and day of the week. Analysts found that on Thursdays and Saturdays males who buy diapers also buy beer [FHS96]. Such information that is not evident at first look may be used to relocate the merchandises to more strategic places, in this example, keeping diapers and beer close to each other. Another application example is web click stream analysis, where a web administrator might be interested in detecting killer pages.

Mining association rules from large databases is usually a two-step process:

1. Find all frequent itemsets in the database
2. Generate association rules from the itemsets

In the first step, all itemsets (e.g. groups of products in a supermarket) in the database need to be found that occur at least as frequently together in an transaction as a pre-defined minimum support count *minsup*. From these so-called frequent itemsets, the association rules with minimum confidence c can be easily computed. Thus the performance of association rule mining is determined by the first step. Numerous techniques for mining frequent patterns have been proposed in the past, like the Apriori algorithm or the FP-growth [HPY00] approach. These approaches take a given support and confidence level as input parameters and return the corresponding association rules.

The problem, however, is that it is difficult to determine the appropriate support and confidence levels a-priori. For low support and confidence levels the number of resulting association rules is usually very high, so that it is difficult to identify really interesting ones. On the other hand, if support and confidence levels are too high, important rules may be overlooked.

Therefore it is very important to allow the user to get a general idea of frequent patterns contained in the data, and to interactively explore these patterns by changing important parameters, like the minimum support *minsup* threshold

in order to discover the effects immediately or to select only special items and thus discover only frequent itemsets which contain these items. To provide such functionality it is important to integrate Frequent Pattern Mining algorithms and interactive Visualization techniques. Therefore we developed the FP-Miner tool. The basic idea of this approach is to provide a radial hierarchical layout, similar to the Sunburst [SZ00] and Interring technique [YWR02], to visually represent the computed frequent patterns, and to allow the user to get details on demand by providing drill-down and selection capabilities. The visual interface is tightly coupled with the *FP-growth* frequent pattern mining technique. We applied the technique to visually analyze sales patterns, co-authorship in Digital Libraries, and network patterns.

7.1.1 Basic Concepts

The main concept of our radial hierarchical visualization technique is the visualization of frequent pattern hierarchies based on the support of objects.

Therefore we first give some basic definitions:
Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. Let D be a set of database transactions where each transaction T is a set of items $T \subseteq I$. Each transaction is associated with an identifier TID. Let X be a set of items. A transaction T is said to contain X if and only if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ holds in the transaction set D with support *sup*, where *sup* is the percentage of transactions D that contain $X \cup Y$. The rule $X \Rightarrow Y$ has confidence *conf* in the transaction set D, if *conf* is the percentage of transactions in D containing X that also contain Y.

Usually the analyst is only interested in rules that satisfy both a minimum support threshold *minsup* and a minimum confidence threshold *minconf*, so-called *strong* rules. A set of items is referred to as itemset, an itemset that contains k items is an k-itemset. The frequency of an itemset is the number of transactions that contain the itemset. An itemset is called *frequent*, if the frequency of an itemset is greater or equal to the product of *minsup* and the total number of transactions in D. As mentioned in the last section, the main step in association rule mining is the finding of these frequent itemsets. Once these frequent itemsets are found, it is straightforward to generate strong association rules using the following equation for confidence:

$$confidence(X \Rightarrow Y) = \frac{support(X \cup Y)}{support(X)}$$

Based on this equation, association rules can be found by generating all nonempty subsets s for each frequent itemset l, and then check if $s \Rightarrow (l - s)$ satisfies the minimum confidence threshold. Finding the frequent itemsets, and thus selecting appropriate *minsup* thresholds, is therefore a critical step in the process.

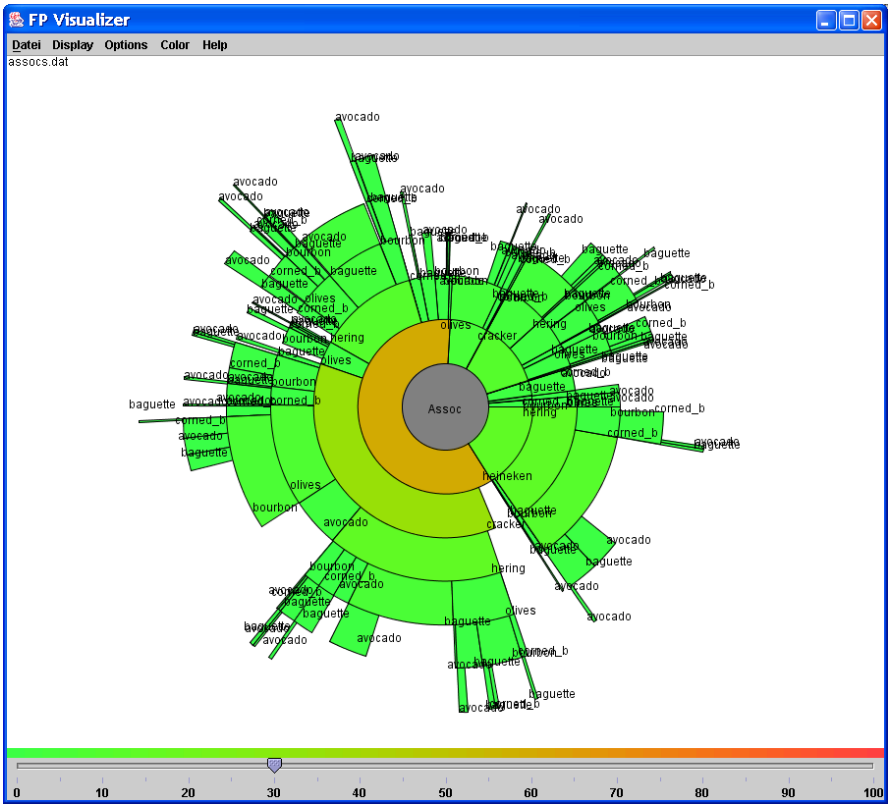


Figure 7.1: FP-Miner Framework: Radial Hierarchical Frequent Pattern Visualization. Shown is the FP tree visualization for the SAS Assocs example. Color shows item support. The slider allows the adjustment of the *minsup* threshold.

The FP-Miner framework supports the user in finding relevant frequent patterns by providing a visual interface that gives visual feedback on computed frequent patterns, allows an interactive adoption of relevant parameters like the *minsup* value, and discover the resulting changes immediately. The FP-Miner framework, shown in Figure 7.1, is based on the idea to share methods from applied data mining and visualization.

The basic idea is to apply data mining techniques to mine frequent patterns in transactional databases. In particular, the widely used FP-growth algorithm is used to compute so-called frequent pattern trees from which frequent patterns can than be extracted. After such a hierarchy is computed, a hierarchical visualization technique is used to visualize the structure content. The provided interactive system supports user navigation and interaction and an effective online exploration of large databases.

7.1.2 Mining Frequent Patterns

The mining component of our FP-Miner Framework is based on a modified version of the FP-growth algorithm [HPY00]. This approach uses a so-called Frequent Pattern Tree (FP-Tree), a prefix tree structure, for efficient frequent pattern mining by pattern fragment growth. The input for the algorithm are a transactional database D and a minimum support threshold *minsup* and the output is the complete set of frequent patterns. Basically the algorithm works in two steps:

1. Construction of the FP-tree
2. Mining of the FP-tree using the FP-growth procedure

The second step is performed as follows: Starting from each frequent length-1 pattern, construct its conditional pattern base which consists of the set of prefix paths in the FP-tree co-occurring with the suffix pattern. Then construct a (conditional) FP-tree and perform mining recursively on such a tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree. For the detailed algorithm we refer to [HPY00].
An important observation is that the appropriate selection of the *minsup* parameter is critical, since it directly influences the size and complexity of the FP-tree and thus the number of frequent patterns. But since the algorithm works automatically the user has to find an appropriate parameter in an trial and error manner, since there is no possibility to get insight into the data or the resulting FP-tree. To bridge this gap, the FP-Miner allows an interactive analysis of the resulting FP-trees. The analyst can interactively change the *minsup* threshold, and analyse the resulting FP-tree immediately. Before we describe our technique in detail, we give a short intro to the FP-Tree.

Definition 6 (FP Tree) *A frequent-pattern tree (FP-Tree) is a tree structure with the following properties:*

1. *It consists of one root labelled as "null", a set of item-prefix subtrees as the children of the root, and a frequent-item-header table*
2. *Each node in the item-prefix subtree consists of three fields: item-name, count and node-link.item-name registers which item this node represents, count registers the number of transactions represented by the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name, or null if there is none.*
3. *Each entry in the frequent-item header table consists of two fields, item-name and head of node-link.*

Given a transactional database D and a minimum threshold *minsup*, the FP-tree is constructed in two steps. First D is scanned once to collect the set of

frequent items and their support. These items are sorted in support descending order in a list L. Second the root of the tree is created and labelled with null. In the second scan, for each transaction in D the frequent items are then selected and sorted according to their support. This path is then inserted into the tree, either by creating a new path, if it does not exits yet, or by increasing the counter of each node for existing paths and updating the link table. For a detailed description of this step see [HPY00]. The basic idea of the FP-Miner is to visualize these FP-tree structures directly using radial hierarchical visualizations. In this way, the user is able to get insight into the data and interactively steer the pattern mining process.

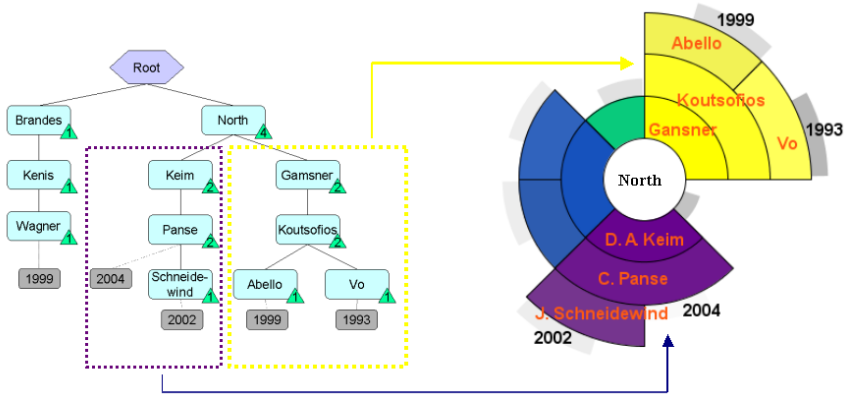


Figure 7.2: From Frequent patterns to Visualization, shown on the example of co-authorship

7.1.3 The Visual Interface

In practice, FP-trees resulting from large data sets are complex and are difficult to understand. Since FP-trees are classical tree structures, this tree structure can be used for effective visualizations. The basic idea is to visualize the tree directly using a radial hierarchical visualization as shown in 7.1. To provide a visual representation of the FP-tree, the nodes of the tree are mapped to circle segments as shown in Figure 7.2.

The root of the tree is visualized by a circle in the middle of the visualization. Starting from the root, each subtree is visualized by a circular ordering of circle segments, where each circle segment represents a single node in the tree. Segments which have the same distance from the center of the visualization have the same tree level in the underlying hierarchical structure. The order in which circle segments of same level are placed, depends on an additional attribute (e.g. support, lexicographic order,). Another attribute can be mapped to the color of

the segments, e.g. to indicate the support of an item, specific item properties like item names, or to indicate items that belong to the same subtree.

This concept as shown in Figure 7.3, can be applied to any tree structure. Since the implementation of the system is object oriented, the visualization is independent from the underlying database format, so that the system can be customized for any type of input data and many application scenarios.

The FP-Miner system allows an efficient and effective exploration and manipulation of hierarchical structures. The main goal of the visualization is on one hand to avoid confusing the data analyst by keeping the overview visualization intuitive and simple, but on the other hand to show effectively as many patterns as possible.

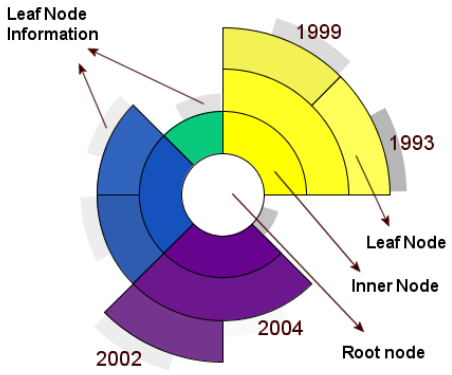


Figure 7.3: Mapping hierarchies to radial hierarchical layouts

The exploration process starts by defining an initial *minsup* threshold via interactive slider. The corresponding FP-Tree is then constructed and the analyst can get an initial overview on the data. The analyst may then interactively select single items of interest to construct conditional FP-Trees or readjust the *minsup* parameter through interactive query slider to see the changes in the extracted patterns immediately. The sliders are directly linked and brushed with the visualization, that means, the results of all selections, manipulations and refinements trigger the automated mining components and the results can be directly observed. This provides an significant improvement in comparison to the fully automated approach, since the analyst can interactively select promising *minsup* thresholds or visually explore patterns of interest. The next sections show how we applied the FP-Miner technique successfully in different application domains.

7.1.4 Applications in Market Basket Analysis

The classical application example of frequent pattern mining is market basket analysis, where analysts are interested in determining what products customers

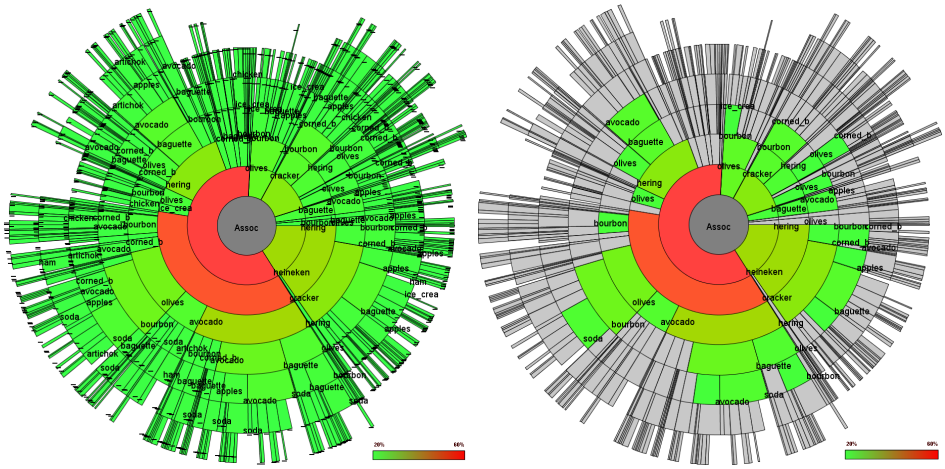


Figure 7.4: The left figure shows the FP-Viz technique applied to the SAS Assocs data with $minsup = 30$. In the right figure the $minsup$ is then set to 40, infrequent items are faded out. Color shows support "heineken"(sup=60) and "cracker"(sup=49) are the most frequent items.

purchase together. Therefore we applied the FP-Miner technique to the SAS Assocs sales data example. This dataset contains 1000 itemsets containing seven items (products) each and we are interested which items typically occur together. Here we can directly apply the technique proposed in the last section, color is used to indicate the support of items.

After the user has defined the $minsup$ threshold, e.g. by moving an interactive slider, FP-Miner computes a FP-tree from the data set that contains only items whose support is $\geq minsup$, as shown in Figure 7.4 (left). Here the $minsup$ was set to 30. The figure shows that the first level of the FP tree has seven nodes, shown by seven arcs in the innermost circle segment. Item "heineken"has the highest support, it is contained in approximately 60 percent of all transaction. It turns out that "heineken"usually occurs together with "cracker". So the user gets a first impression about interesting itemsets contained in the data and may now adjust the $minsup$ e.g. to 40 as shown in figure 7.4 (right) and items below the threshold are instantly faded out. The user may also select special items for detailed analysis.

Suppose the user moves the mouse on the arc labelled "heineken", since it may be the most interesting item, then the support for "heineken"is shown. If the user then clicks the mouse, a new (conditional) FP-tree for item "heineken"will be constructed and shown immediately, as shown in Figure 7.5. This new tree contains only items which occur together with "heineken". It turns out that "heineken"usually occurs together with "cracker", "hering"and "olives". Color can either be used to distinguish the different items, by determining a unique

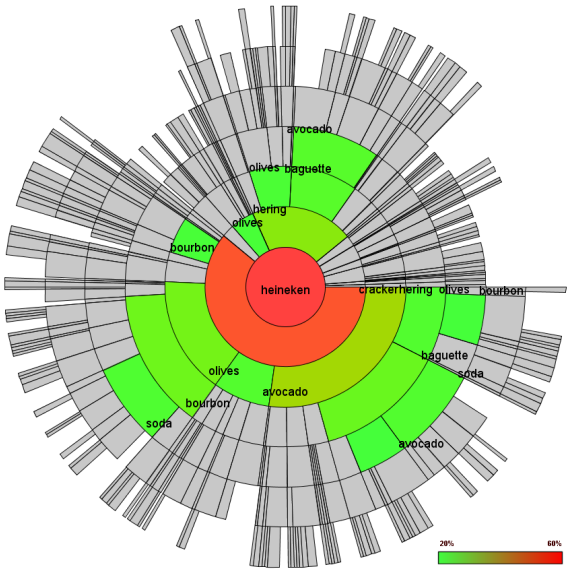


Figure 7.5: Visualizing the FP-tree for item "heineken"from SAS Assocs data set with $minsup = 40$. Color shows the support of each FP-tree node

color for each item, or to visualize the support of itemsets as shown in Figure 7.4.

7.1.5 Applications in Co-Authorship Analysis

In the second example we applied the FP-Miner for analyzing and exploring co-authorship in large digital document libraries, specifically the DBLP library. DBLP is the Digital Bibliography and Library Project [Ley05] and provides bibliographic information on major computer science journals and proceedings. The server indexes more than 520.000 articles with about 250.000 authors and contains several thousand links to homepages of computer scientists. The DBLP is available in XML format with a volume of about 57 Megabyte, which we used as input data for our framework. Using the FP-Miner framework shown in Figure 7.6, the user is able to detect co-authorships and their development over time in a single view. This allows us to answer questions like *Are there groups of authors which usually publish together?* or *How does the co-authorship of a single author change over time?*. To use the FP-Miner to answer these questions, we can transform the co-authorship analysis task to a frequent pattern mining task by considering an author's publications and its co-authors as a transaction database D, whereas each publication is a transaction T, and the support of each item (co-author) as the number of collaborations with a certain author. Thus we use the FP-Tree to detect authors which have a high number of collaborations, and based on this information we determine the structure and layout of our visualization. Therefore we have to use a modified version of the FP-tree, described in the next section.

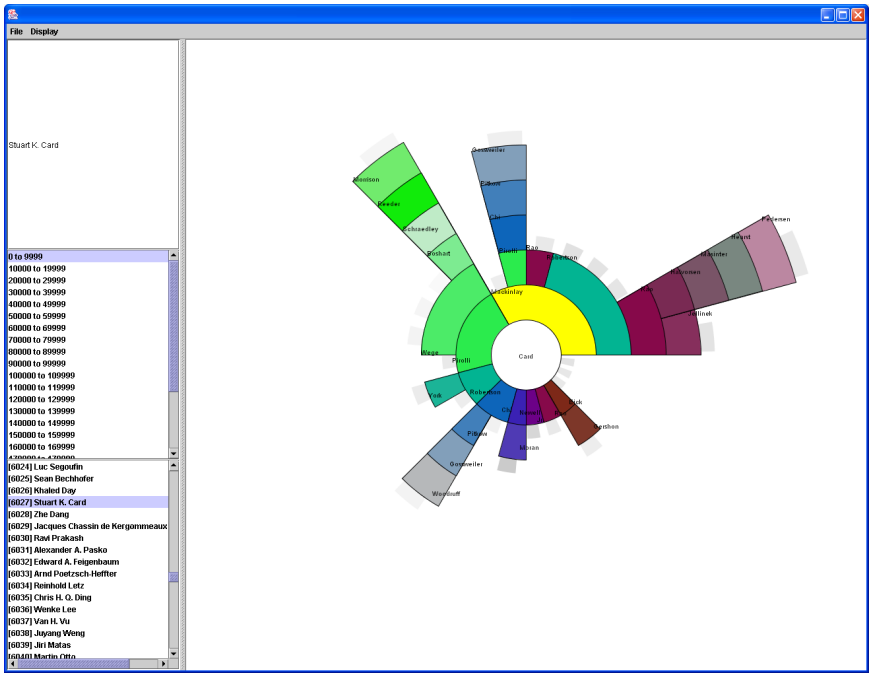


Figure 7.6: FP-Miner: Efficient browsing of Digital Libraries. The figure shows the co-authors of Stuart K. Card based on DBLP publications. The framework supports interactive navigation as well as the keyword search for certain authors.

Mining Co-Author Structures using an extended FP-Tree structure

To fit the special needs of co-authorship visualization and their development over time we modified the FP-tree. In particular, the major differences are as follows:

Definition 7 (DigLib FP-tree) *The DigLib FP-tree is an FP-tree with additional properties:*

1. *There are no node-link links between nodes in the FP-tree carrying the same item-name*
2. *There is no frequent-item header table necessary*
3. *The minimum support threshold* $minsup = 0$

Property 1 guaranties a hierarchical ordering of authors or papers in our visualizations. If one author, paper or keyword occurs multiple times, there is no need to follow each link to another subtree, since each subtree is visualized independently. Since the visualization operates directly on the FP-tree, the frequent item header table becomes redundant. Finally the initial minimum support threshold $minsup = 0$, since it is important to include all objects (authors, papers,...) in

the visualization and not only frequent ones. In general, our visualization employs the FP-tree for detection of collaborative authors or papers and of the ordering of them, depending on the frequency of their occurrence.

Using the Co-Authorship FP-tree, we have the following tree construction algorithm:

Algorithm 1 (DigLib FP-tree Construction)
Input: *A transactional database*
Output: *DigLib FP-tree*
The DigLib FP-tree is constructed as follows:

1. *Scan the database DB once. Collect the set of frequent items F, and the support of each frequent item. Sort F in support-descending order as a frequent item list FList.*

2. *Create the root of an FP-tree R and label it as null. For each transaction T in DB do:*
   Select the frequent items in T and sort them according to the order of FList.
   Insert the sorted frequent itemset in the FP-tree.

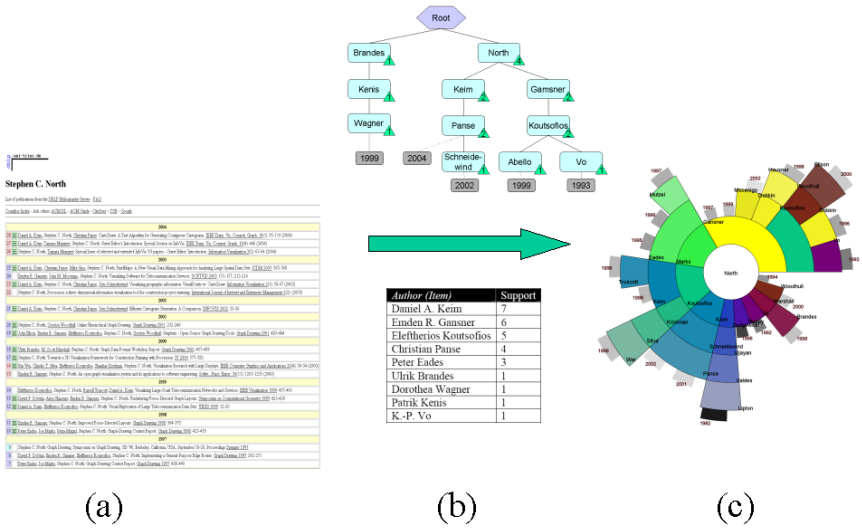


Figure 7.7: Visualizing author collaboration: (a) Extract all publications of single author from *DB* (b) Extract frequent item sets from query result, compute support for each item, create DigLib FP-tree (3) Visualize DigLib FP-tree using radial technique

The insertion into the FP-tree is performed as follows. Let the sorted frequent-item list in T be $[p \mid P]$, where p is the first element and P is the remaining list. If

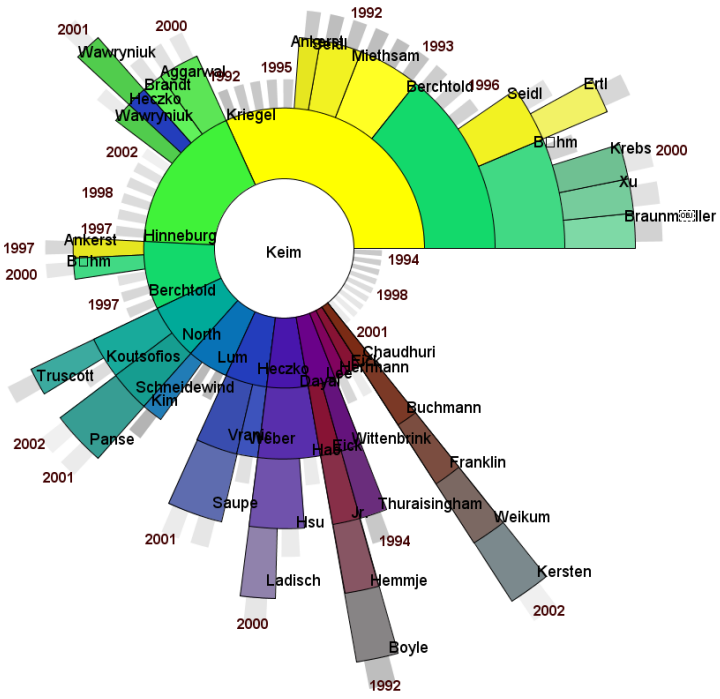


Figure 7.8: Visualization showing co-authors of Daniel A. Keim ranked by their frequeny and year.

R has a child N such that N.item-name=p.item-name, then increment the counter of N by 1, else create a new node N, with a initial count of 1 and a parent link to R. If P is not empty, then repeat the insertion recursively. Note that the construction of the tree structure takes exactly two scans of the database. For bibliographic libraries each transaction represents a publication with authors as items.

Co-Authorship Examples

Figure 7.8 shows the Authorring constructed from the DBLP database for Prof. Daniel A. Keim. The radial layout contains 63 publications, starting from 1992 till today. This may lead to the conclusion that his research career started in 1992. D. A. Keim wrote most publications together with H. P. Kriegel (20 co-publications), S. Berchthold (14 co-publications) and A. Hinneburg (11 collaborations). There are a lot of publications between 1994 and 2002 were D. A. Keim was single author.

His first publications were written together with H. P. Kriegel at the University of Munich, but there is no co-authorship after 2000. The reason is that D. A. Keim moved to the University of Halle (Germany). There he wrote most publications

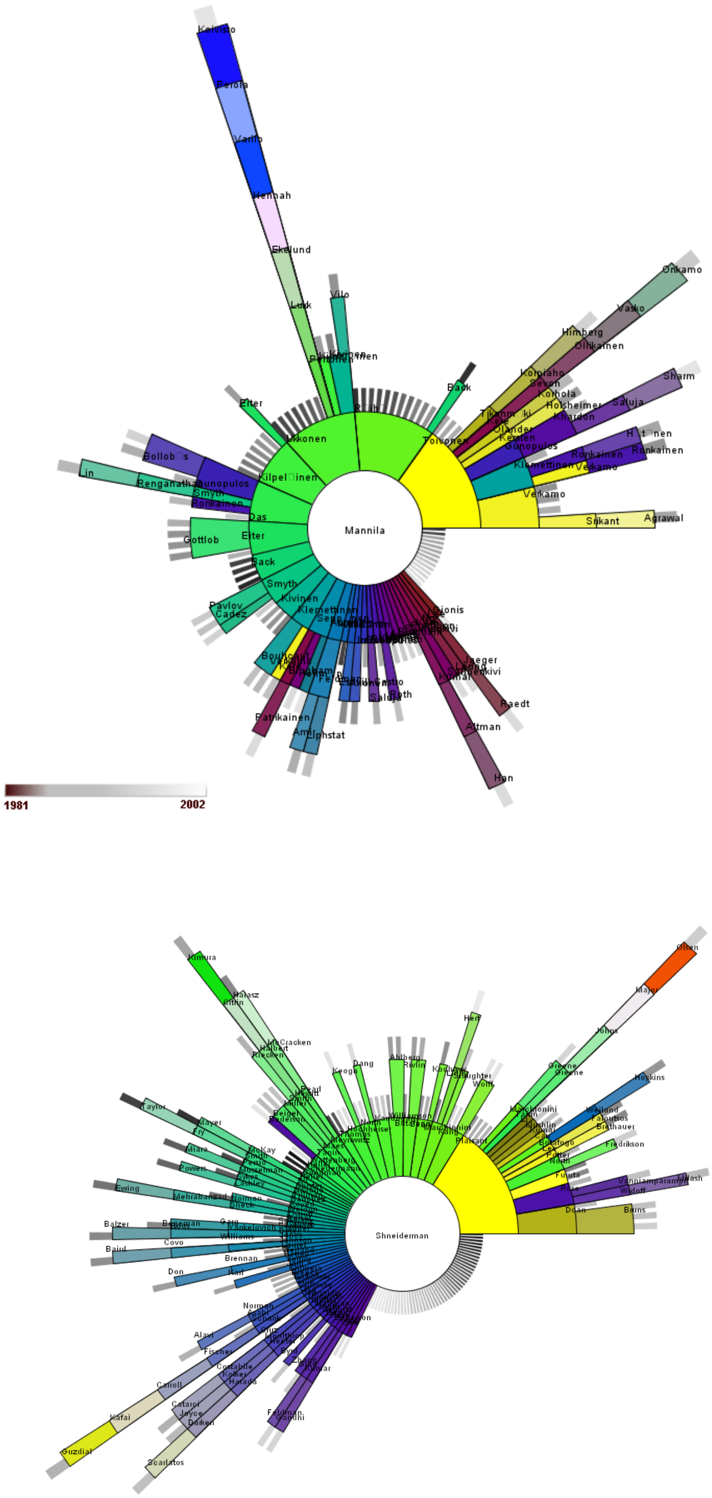


Figure 7.9: Co-Authorship for H.Mannila (top) and B. Shneiderman (bottom) based on DBLP publications

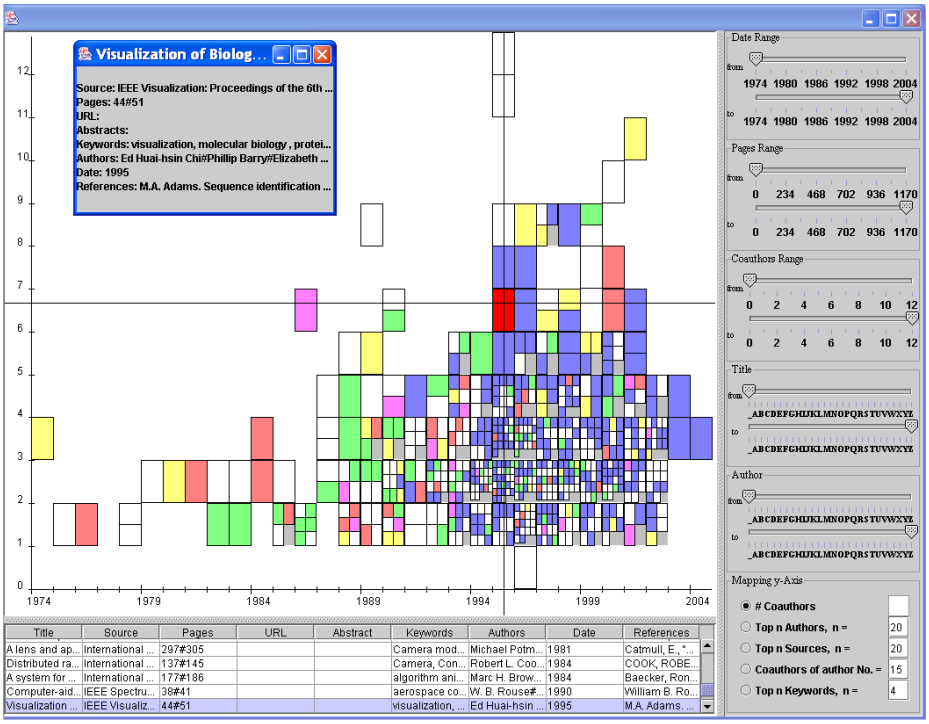


Figure 7.10: Paperfinder analyzing the InfoVis 2004 contest data set [KSS+04b]. Via interactive query sliders the user may search / filter documents based on certian criteria (publication year, title, topic, number of co-authors). Once a paper has been selected, an authorring can be constructed to analyze the relations between the authors.

together with A. Hinneburg. Publications together with members from AT&T Labs, USA (Stephen C. North) and from HP Labs, USA (U. Dayal, M. Hao) indicate that he worked close together with this researchers. In fact D. A. Keim had a research year at AT&T Labs in 2001. In 2002 D.A. Keim started to work as Professor at the University of Konstanz, Germany. Therefore the number of collaborations with A. Hinneburg decreased. Since the system is interactive, the user can select any co-author of Daniel A. Keim, by choosing the corresponding circle segment in the underlying visualization (i.e. via mouse click), and instantly get the radial layout for the selected co-author. Two more examples, the co-authorship for H.Mannila and B. Shneiderman, are shown in figure 7.9

Since the Authorring technique is a very useful tool in exploring co-authorship, we integrated it in our Paperfinder Framework [KSS+04b]. The Paperfinder shown in Figure 7.10 is a tool for exploring digital document libraries which we proposed in the context of the 2004 InfoVis. It provides interaction functionality through dynamic queries. This allows an interactive search for digital documents and their detailed analysis. Via interactive query sliders the user may search for documents

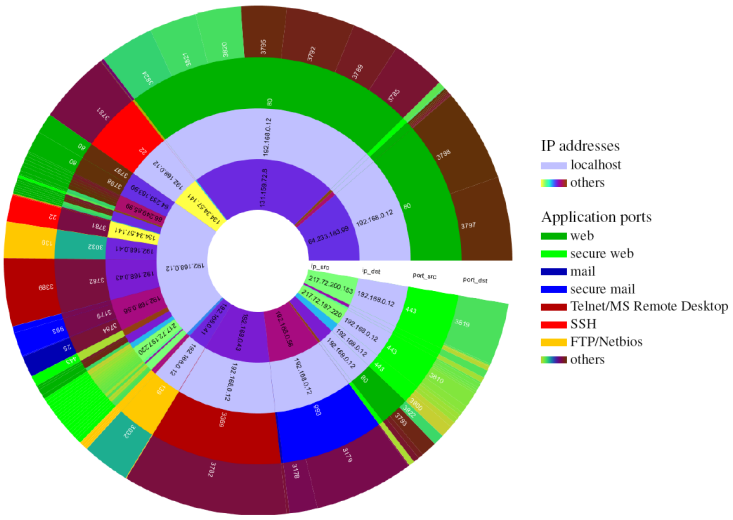


Figure 7.11: RTA display showing the distribution of network traffic of a local computer. We maintain an overview by grouping the packets from inside to outside. The inner two circles represent the source and destination IP addresses, the outer two circles represent the source and destination ports. Traffic originating from the local computer can be recognized by the lavender colored circle segment in the inner ring. Traffic to this host can be recognized by the lavender colored segments on the second ring. Normally, ports reveal the application type of the respective traffic. This display is dominated by web traffic (port 80 - colored green), remote desktop and login applications (port 3389 - red, port 22 - bright red) and e-mail traffic (blue).

from certain publication years or belonging to certain topics. The Authorring provides a very helpful extension in this context, since it allows a effective exploring of author relations in such digital libraries.

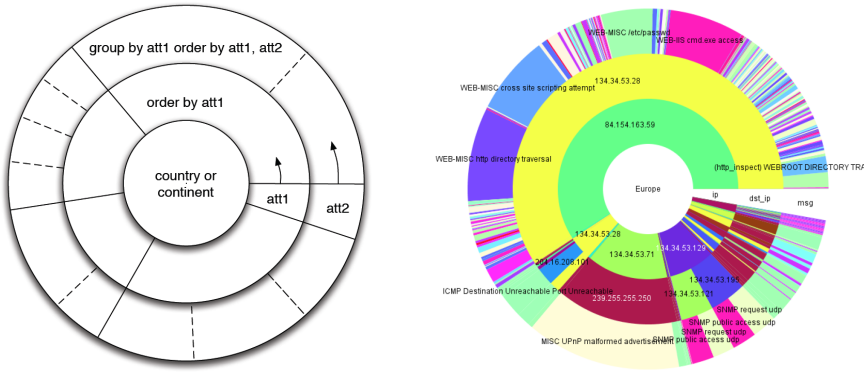
7.1.6 Applications in Network Analysis

Extensive spread of malicious code on the internet and also within intranets has risen the users' concern about what kind of data is transferred between one's personal computer and other hosts on the network. Visual Analytics of this kind of information is a challenging task due to the complexity and volume of the data type considered and requires special design of appropriate visualization techniques. In [MSSK06] we therefore considered the problem of visually analyzing important characteristics among the communication flows between hosts on the internet. This requires the analysis of large data volumes, that may contain complex interrelationships and that vary over time.

We tackle the problem by abstracting the internet communication flow to the

network (packet) level as defined by the TCP / IP Reference Model. This model considers information flows on a network by means of packets (atomic information units) which are moved though the network from a given source host using a source port to (usually) a destination host using a destination port.

We focus on visualizing packet level communication properties, as the packet level defines a simple data structure in terms of source and targets of hosts and ports. From its port information, we can usually conclude the type of service addressed by the packet, e.g., port 80 usually indicates WWW traffic, port 22 indicates Secure Shell (SSH) Traffic, and so on. Thus this level is a viable option to consider for visual network communication monitoring.



(a) (b)

Figure 7.12: (a) Visual Layout of the radial traffic analyzer (b) Displaying security alerts from the intrusion detection system Snort. After discarding ICMP Router Advertisements, ping and echo alerts, we can clearly see that host 134.34.53.28 (yellow) was attacked by 84.154.163.59 using various methods (outer ring).

We used our radial layouts to visualize the distribution of given communication volumes along these main four packet-based attributes. The basic idea is to provide the radial layout, to visually represent the frequent patterns in a high level view, and to allow the user to get details on demand by providing drill down and selection capabilities. Combining the radial layouts with an appropriate colormap, the user gets a compact informative summary of inbound and outbound packets with respect to a given host on a network.

In [MSSK06] we also proposed an extension of this approach where we combined the radial network packet layouts by a second Treemap like rectangular layout technique, to relate the traffic patterns to geo-locations as derived from IP-packets respective IP-addresses.

To analyze network packets using the radial layout, we place the most important attribute, as chosen by the user, in the inner circle, and arrange the values

in ascending order, to allow better comparisons of close and distant items. The subdivision of this ring is conducted according to the proportions of the measurement (i.e., number of packets or connections) using an aggregation function over all tuples with identical values for this attribute. Each further ring displays another attribute and uses the attributes of the rings further inside for grouping and sorting, prioritized by the order of the rings from inside to outside as illustrated in Figure 7.12(a).

In the default configuration, we used four of these rings. The visualization is to be read from inside to outside, starting from the innermost ring for the source IP-addresses, the second ring for the destination IP-addresses, and the remaining two rings for the source and the destination ports, respectively. In Figure 7.12(a) beginning on the right, we map the fractions of the payloads for each group of network traffic counter-clockwise on the rings while sorting the groups according to ip_{src}, ip_{dst} , $port_{src}$, and $port_{dst}$. Beginning with grouping the traffic according to ip_{src}, we add another grouping criteria for each ring further outside. This results in a finer subdivision of each sector on the next ring.

To facilitate a better understanding of the rings, sectors representing identical IP addresses (inner two rings) are drawn in the same color, ports (outer two rings) are colored respectively. To further enhance the coloring concept, we created a mapping function for ordinal attributes that maps a number x (i.e., the port number, or IP address number) to the indices of an appropriate colormap: $c(x) = x mod n$ (n: number of distinct colors used). Prominent ports (e.g., HTTP=80, SMTP=25, etc.) are mapped to colors that do not show up in our colormap for easier identification. This mapping function facilitates to correlate close IP addresses or ports. To differentiate between traffic that is transferred over an unsecured and a secured channel, we modify the brightness of the color (i.e., HTTP/80 = green, HTTPS/443 = light green, etc.). To map numeric attributes (e.g., number of connections, time, etc.) to color, it makes more sense to normalize the data values and then map them to a colormap with light to dark colors or vice versa. Different colormaps were used for the attributes, and should clarify the comparability of rings. An IP address appearing as a sending host in the innermost circle and reappearing as a receiving host in the second circle should be colored identically, whereas this color should then not be used for a port.

To make maximum use of the available display space, different grouping operations are useful and are realized by assigning the chosen dimension (i.e., source IP, destination IP, source port, destination port) to the inner rings. On one hand a grouping according to the hosts might be useful when determining high-load hosts communicating on different ports, while on the other hand a grouping according to the target ports clearly reveals the load of each type of traffic. To compensate for the strict importance rating according to the inner circles, the positioning and thus importance within the sorting order can be interactively changed using drag and drop mouse interaction.

As soon as many different circle segments are drawn, some segments become

too small to plot labels into. Therefore, we cut long labels and employ Java tooltip popups showing the complete label and additional information like the host name for a given IP-address, and the possible application programs corresponding to the respective port . As filtering is a common task, a simple mouse click triggers a filter that discards all traffic with the chosen attribute values. Detailed information about the data tuples represented through a circle segment is accessible using a popup menu. Transferred bytes is not the only available measure when analyzing network traffic. When investigating failed connections, for example, the measure transferred bytes would not show the data tuples of interest on the ring, as they all have 0 bytes for the attribute. In this situation, the measure "number of connections" would be useful to correctly size the circle segments.

Experts often compare transferred bytes to the count of sessions on a set of active hosts. High traffic with only few sessions is considered to be a download resource, whereas medium traffic on many sessions it typical for more medium-bandwidth applications like WWW. The RTA display is flexible to display many different data sets and can be adjusted to the data at hand on the fly. An example is to configure the inner two rings with the source and target IP-addresses and the outer ring with security alerts generated by an intrusion detection (IDS). Alternatively, one can extend the IP address dimension through the use of associated higher-level network attributes (e.g., IP network block, autonomous system, etc.) to investigate whether e.g., a denial of service (DOS) attack originates from a certain network block, or to assess the danger of a virus spread from neighboring autonomous systems.

7.1.7 Conclusion

The analysis of hierarchical information is an important topic in many application scenarios. We presented approaches that combined analytical and visual methods to analyze and represent such hierarchical data sets. Our visualization was based on a radial layout that was applied to frequent pattern analysis, co-authorship analysis and network analysis. The present application examples showed the usefulness and functionality of our techniques. In the future our Circular Layouts could be integrated in frontends for certain online libraries like the DBLP to visually explore co-authorship online in such libraries.

7.2 VisMap

7.2.1 Introduction

Business Data Warehouses usually represent the data at different aggregation levels using the data cube model, as described in Chapter 5. These data cubes typically contain, among others, a time dimensions, leading to a hierarchy of time related information. Network analysis, business process analysis or financial analysis for example, require the analysis of a large number of time series in parallel, which show intrinsic hierarchical relationships and varying degrees of relevance with respect to the analysis task.

The *VisMap* technique focuses on exploring such hierarchies of time related data, by providing appropriate aligned layouts and meaningful levels of abstraction and by supporting effective user interaction.

Classical techniques for visualization of hierarchical data like the well-known treemap approaches [Shn92, BSW02, BHvW00], focus on visualizing the hierarchical structure and the proportion of single items in such data sets, but typically do not take the special needs of time series analysis, like aligned layouts, compact representations at different resolutions, highlighting or ordering of time series according to their relevance, into account. Since treemap algorithms use a fixed recursive layout algorithm (e.g. slice and dice [Shn92]), it is hard to consider the special properties of time series data adequately.

Therefore we proposed the *VisMap* approach that automatically generates aligned visual map layouts for large hierarchical time related data sets and at the same time emphasizes the hierarchical structures of the data. The basic idea is to employ a generic tree structure to represent the hierarchical relationships among the data items and then to automatically generate a layout that allows a visual comparison of multiple time series data with respect to their underlying hierarchical structure. Our application of these techniques to real world business applications, including sales analysis and stock market analysis, shows the wide applicability and usefulness of our visualization approach. We have experimented with *VisMap* for service contract analysis and financial analysis using real business data.

7.2.2 Analysis of hierarchical time related Data

To date, the visual analysis of time-related data has received significant research attention in Information Visualization and a number of advanced visualization techniques for various analysis tasks have been proposed [MS03]. Examples are the Spiral technique [WAM01] to identify periodic patterns in time series data, the Cluster and Calendar based Visualization technique [vWS99] to identify patterns and trends on multiple time scales simultaneously, or the TimeSearcher [HS04] to visually explore time related data sets via interactive queries.

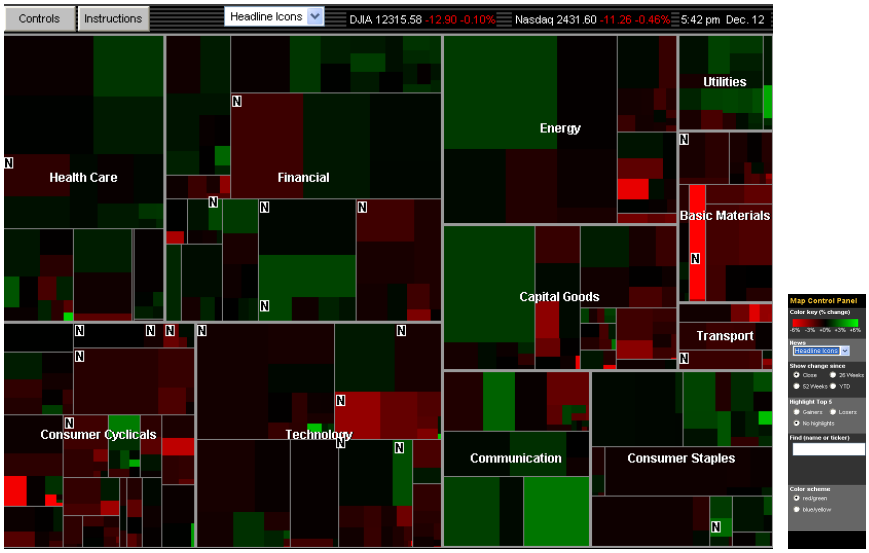


Figure 7.13: Smartmoney's Map of the market allows to watch more than 500 stocks at once: Each colored rectangle in the map represents an individual company. The rectangle's size reflects the company's market cap and the color shows price performance [Sma06].

Due to the increasing volume and complexity of time related data sets, many techniques consider the problem of simultaneously visualizing large as well as long time series data, while maintaining perceptibility of the time- and the value dimension. The Recursive Pattern technique [AKK95], for example, visualizes very large time related data sets by employing a pixel based rendering paradigm. Other techniques, like proposed in [BM04, Chu98], use numeric aggregation to scale with the size of time-related data sets.

In the context of financial data analysis, we proposed the *Growth Matrix* visualization technique [KNS+06], to analyze and visually compare the performance of groups of assets over multiple time intervals. An example is given in Figure 7.14, representing three technology funds composed of technology stocks. Each pixel in the matrix represents a certain time interval and the color represents the performance (growth rate) of the corresponding fund compared to the overall performance of this sector. Green colors indicate that the fund had a better performance than its sector, red color indicates a worse performance. Thus, the growth matrix allows an intuitive analysis of good and bad overall performance of single stocks / funds compared to the market. In the presented example, the overall *Growth Matrix* patterns are similar and reflect the "dot-com" phenomenon, however the leftmost fund asset generally exhibits better maximal performance. It has the largest growth rates prior to the technology crisis and recovers more quickly from it. Thus, this particular fund has been best managed to control losses. Conversely,

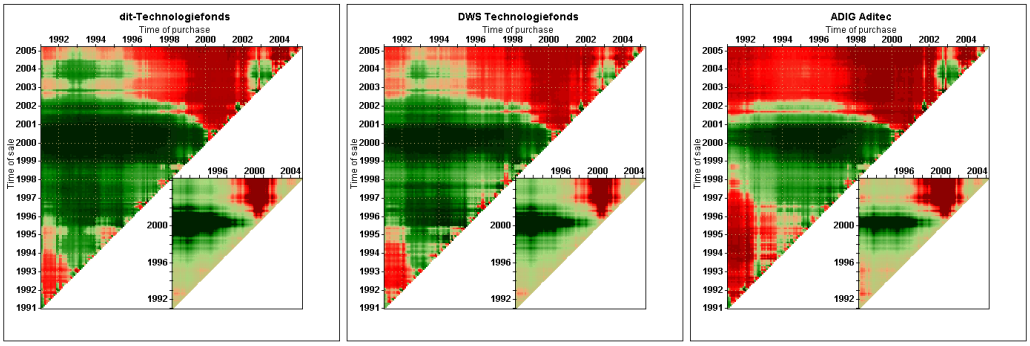


Figure 7.14: *Growth Matrix* employed to analyze three funds composed of technology stocks. By simultaneous visualization of a set of similar assets strong and weak candidates can be identified: While the overall *Growth Matrix* patterns are similar and reflect the"dot-com" phenomenon (red areas), the leftmost fund asset generally exhibits better maximal returns. It has the largest growth rates prior to the technology crisis, and recovers more quickly from it [KNS+06].

the rightmost fund performed worst in managing the technology crisis.

Although most of the proposed techniques are powerful tools for analyzing multiple time series simultaneously, they have a limited ability to take the intrinsic hierarchical relationships into account, e.g. the stock market classification of assets into sectors or the segmentation of sales data into certain product categories.

To visualize such hierarchical information spaces, stacked displays, in particular treemap approaches [Shn92, BSW02, BHvW00] have become very popular. An example is Smartmoney's Map of the market [Sma06] shown in Figure 7.13, a technique to visually analyze stock market data. It is very effective in exploring the performance of single stocks or whole finance sectors at a certain point in time. But since it is hard to discover changes or correlations of their performance over time using treemap approaches, the tool provides linked views that provide charts for selected stocks. However, it would be very useful to visualize time series data and their hierarchical relations in a single view for intuitive visual comparison. In [ESS92] a space filling approach for visual analysis of software systems is proposed, that uses a hierarchical visualization techniques to visualize software changes. This was one of the first attempts to combine hierarchical and time related visualization techniques.

When analyzing complex time series data, it would be also useful to have multiple views on the data at different levels of details. Therefore we proposed an approach that semi-automatically generates aligned visual map layouts for large hierarchical time related data sets, with respect to the underlying data properties and different detail levels.

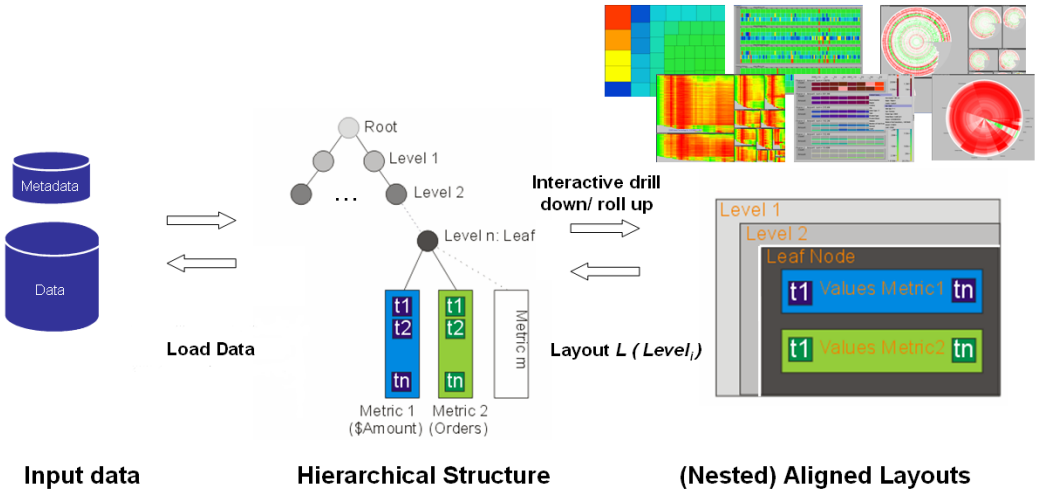


Figure 7.15: VisMap basic idea – From hierarchical input data to automated interactive Visual Map Layouts

7.2.3 The VisMap System

Basic Concepts

The goal of *VisMap* is to extend existing hierarchical space filling approaches for an intuitive analysis of hierarchical time series data. Therefore our idea is to combine hierarchical visualization techniques, in particular space-filling approaches, to decompose the screen space into individual components given by the hierarchical structure, and to visually align the given time related information of each individual component using intuitive chart techniques for straight forward visual comparison.

In general the generated layout should fulfil certain constraints which are fundamental for an effective visual exploration of the data. In particular, visual components, i.e. subregions of the display, must be always proportional to a statistical parameter. This is the fundamental property of treemap approaches, which allows an easy comparison of the relevance of certain display subregions. In Figure 7.13 for example, the rectangle's size reflects companies market cap, which makes it very easy to visually identify large companies.

In *VisMap* this constraint is fulfilled by using treemap algorithms to partition the data. The second important constraint is given by the analysis task of visual comparison of time series data belonging to certain components of the hierarchy.

In Figure 7.13 for example, the analyst would be interested to compare the performance of all stocks belonging to the energy sector over time, to identify stocks that have a good overall performance. (Note that in Figure 7.13 it is only possible to analyze the performance rates at a single point in time). Therefore the time

series data of each sector must be aligned and intuitive visualization techniques must be used. In *VisMap* we use intuitive techniques including line- and bar charts as well as the *CircleView* approach to visualize the data. Furthermore we align the data of each subregion for visual comparison. Since we usually deal with very large and complex amounts of data, it must also be possible to present the data at different levels of details and to focus only on parts of the data, which is our third constraint. By providing interactive drill down and roll up functionality the user may then navigate through different aggregation levels to explore detailed patterns in the data. Figure 7.15 shows the basic idea of *VisMap*.

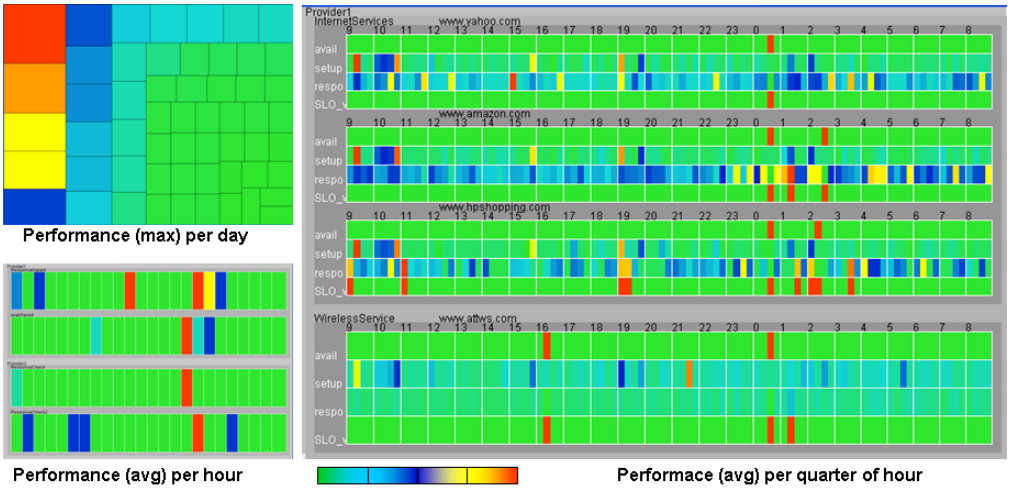


Figure 7.16: Hierarchical Aligned Layouts: Using *VisMap*, the analyst is able to interactively explore server performance at different levels of detail.

An example scenario is shown in Figure 7.16, where we employed the *VisMap* approach to control the performance of certain department servers. In the initial high level view shown in the upper left figure, the maximum service violation (i.e. a violation is caused if some server metrics are outside of predefined ranges) of each server per day is visualized, using a treemap layout. Red color indicates that there was a server with very high service violation. The analyst may now drill down to that particular server to get more details, shown in the lower left figure. The figure now shows the four services that run on that server, aligned by hour of the day. From the image it turns out that there were certain hours of the day with very high service violations. The analyst may now drill down again to analyze the reason for these server problems by looking at the metrics for certain services. He can now see that there were certain server problems shortly after midnight caused by server unavailability and high response times. Since we used real world data, we know that these problems where caused by server maintenance which was performed after midnight. This example shows how *VisMap* supports analytical reasoning by combining hierarchical aligned layouts with drill-down /

roll-up functionality.

VisMap Functionality

In this section we describe the constraints given before as well as the *VisMap* functionality in more detail. For an effective analysis of hierarchical time related data, *VisMap* supports a number of features that are novel in and highly contribute to the field time related data analysis. In particular *VisMap* supports:

- Hierarchical Aligned Time Series Layouts
- Adaptable Visualization Metaphors
- Time Series Ordering and Analysis Functionality
- Appropriate Levels of Abstraction
- Interactive Drill-down/Roll-up and Content Query Functionality

The combination of these basic functionalities makes *VisMap* a powerful exploration tool. In the following we describe these points in more detail.

Hierarchical Aligned Time Series Layouts

To analyze correlations and patterns across multiple time series with hierarchical relationships, it is important to (a) use a visual layout that emphasizes the hierarchical relations and (b) aligns the data appropriate.

To take these issues into account, *VisMap* combines two different layout techniques. To visualize the hierarchy we use rectangular space-filling techniques, in particular adapted treemap approaches [JS91b, BSW02, BHvW00]. These techniques are not only able to convey the hierarchical structure of the data, they are also able to reflect certain statistical properties of the data, e.g. the importance or relevance of a data item. To compare multiple data sets that belong to a certain hierarchical entity, it is important to align the data appropriately. *VisMap* aligns the data based on the time stamp of each data item. This allows to make visual comparisons at a glance. To visualize the data items, a number of different visualization metaphors are integrated into the system.

Adaptable Visualization metaphors

Many visualization metaphors can be used to present aligned time series layouts. We used techniques like line charts, bar charts and our *CircleView* technique to produce aligned visual layouts. The user can interactively switch between the different layouts, to adapt the visualization according to his demands.

Time Series Ordering and Analysis Functionality

VisMap uses a number of analytical functions to order, arrange, and analyze the underlying data. Ordering functions are employed to arrange the time series data according to user demands, a number of normalization functions highlight certain patterns in the data and clustering techniques group selected data items according to their properties.

Interactive Drill-down/Roll-up and Content Query Functionality

Interactivity is an important aspect of analyzing time series data in business Warehouse environments. To make large volumes of multi-attribute time series easy to compare and interpret, *VisMap* provides several interaction capabilities. The most important feature is the layered drill-down. The user is able to drill-down from the current structure level to the next hierarchical level. Based on the number of underlying data items, either a new aggregated view is created (e.g. using a treemap layout), or a aligned layout is produced (e.g. using line charts in a line by line layout). The user can select different aggregation functions (min,max,avg), that the visual layout of aggregated layouts should reveal.

Technical Details

To take the proposed constraints into account, *VisMap* is based on a generic tree structure, that is adaptable to certain application scenarios. The basic idea is to create a tree from the given input data, whereas the inner nodes of the tree represent the hierarchical information of the data and the leaf nodes contain the time related information. Note that the hierarchy can be extracted from the Data Cube model, if *VisMap* is used to explore Data Warehouse data.

This structure is than automatically mapped to a space filling 2D layout, whereas treemap like algorithms are used for inner nodes and techniques for the visualization of time series are used for leaf node. Furthermore the user can interactively adapt the visual layout, e.g. by changing the layout of certain sub regions to visually compare the time related data of certain subregions, or by drill-down an roll-up functionality to analyze aggregated views of the data or analyzing only parts of the data.

VisMap Algorithm

Based on the technical information given before, we can now give a detailed description of the *VisMap* algorithm. The algorithm is similar to treemap algorithms, but the important difference compared to treemaps is the ability to allow a flexible use of layout metaphors, whereas treemaps use a fixed layout algorithm like *Slice and Dice*. *VisMap* in contrast can react on the data properties of each hierarchy level, it can especially switch from hierarchical partition to aligned layout techniques. Moreover our algorithm is able to change the layout at a certain level of

the hierarchy and then present the underlying data at certain levels of aggregation, which is an important feature to adapt the Visualization component to the underlying Data Cube model. Thus *VisMap* is more flexible than existing treemap approaches.

Algorithm 2 VisMap Layout algorithm

Input: Rooted Tree T where each leaf node L contains a time series ts_i. Inner nodes V decribe the hierarchy levels; the path from *root* to ts_i describes the hierarchical information for ts_i. Each Node V contains a label and an aggregated value computed from its child nodes.
LayoutFunction for InnerNodes $Layout_{inner}$
LayoutFunction for LeafNodes $AlignLayout_{Leafs}$
Output: Space filling Layout of T where all $ts_i \in L$ with same parent V are aligned
Procedure *Compute Initial VisMap Layout*
TreeNode root=getRoot(createTree(data D, metadata H, aggregate f_a))
root.area = setStartRectangle(Display.width,Display.height)
currentNode = root
if currentNode.childs $\neq$ LeafNodes **then**
  Layout=$Layout_{inner}$
else
  Layout=$AlignLayout_{Leafs}$
VisMap(currentNode,currentNode.area, Layout)
{$drawArea$ (currentNode.area, Layout)
ComputeChildAreas(currentNode.area, currentNode.childs, Layout)
for each node $\in$ currentNode.childs **do**
  if node.childs $\neq$ LeafNodes **then**
    Layout=$Layout_{inner}$
  else
    Layout=$AlignLayout_{Leafs}$
  VisMap(node,node.area, Layout)
}

The next section now shows, how we successfully employed *VisMap* in financial and service level analyis.

7.2.4 VisMap Application Examples

SLA Analysis

Service Level Agreements (SLAs) are very common in business scenarios. SLAs are service contracts which are signed between customers and suppliers to guarantee

that suppliers deliver certain goods and services to customers. Such a contract typically contains Service Level Objectives (SLOs) defining what service should be delivered with what level of quality and within what specified time period. An important question business managers need to understand is whether their business operations are fulfilling SLOs and if not, which SLO is violated and what is the cause of the violation. Since the violation of SLO's may result in enormous costs for service providers, it is of course an important issue to be able to visually analyze and monitor SLA's. At HP Research Labs we successfully applied *VisMap* to real world SLA data, for visual root cause analysis.

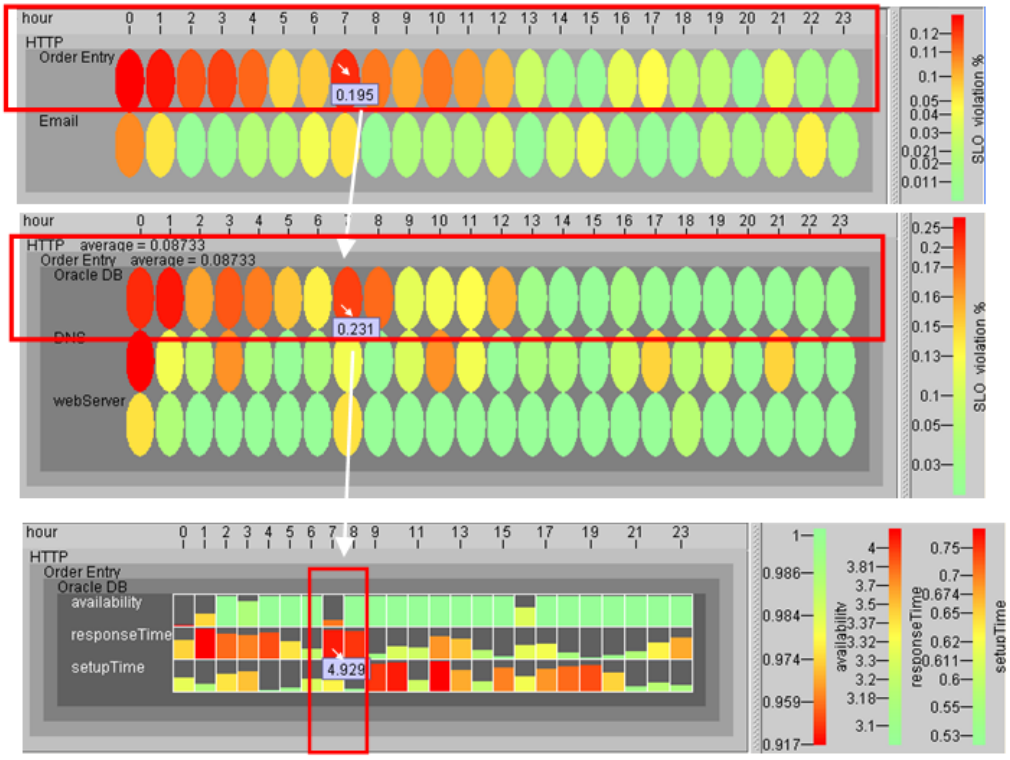


Figure 7.17: Visual Service Contract Analysis of an HTTP Order Entry SLO: VisMap allows different views on the data via Multi-Attributes Layered Drilldown and Correlation Analysis

Figures 7.17 illustrates a series of drilldowns to find the cause of an HTTP Order Entry SLO violation. The initial *VisMap* Layout aligns therefore all SLO's (OrderEntry, Email) based on their violation levels per hour. To identify SLO violations, the maximum violation level per hour is mapped to color. For example, in the upper image in Figure 7.17, the analyst discovers an SLO violation (OrderEntry) at the 7th hour. Next, the analyst drills down from the Order Entry, discovers that the cause of the violation is an Oracle database, shown in the mid-

dle image in Figures 7.17. It's SLO violation level is 0.231 percent. To find the cause for this violation the analyst drills down from Oracle DB, to the DB's logged system performance metrics (availability, response time, and setup time) and finds that the cause of the violation is that system availability is low, which makes the response time long, i.e. above the allowed threshold, and thus the Order Entry contract is violated. After the system support increases the system availability, the response time is reduced and the SLO violation measurement drops. Analysts can observe those changes along the time line.

Business people want to have a set of reports to compare their yearly sales. They want to answer questions like which region, countries, and product has the most sales and which have the least sales. *VisMap* generates a sequence of maps to answer their questions as illustrated in Figure 7.17

Stock Market Analysis

The financial sector is an important domain dealing with complex time dependent data sets. The visual analysis of these kinds of data is an essential issue in technical finance market analysis to support asset performance analysis and decision making processes. In financial analysis, however, the most important and most common visualization techniques for time series data are chart diagram, typically line- and bar charts, since they provide an intuitive way to get insight into price fluctuations of securities and assets. These charts may be enriched by overlaying aggregate plots, e.g. moving averages. One of the most important asset price series characteristic from an analyst's point of view is *Return*. Regarding *Return*, analysts and investors are interested in growth rates of an asset price series within certain, often multiple, different time frames. Briefly, growth rate is defined as the ratio between the asset price at the end and the starting point of a time frame interval.

Since there are typically thousands of assets in the market separated in different categories, a performance analysis of each one over time using charts is a difficult task, since numerous charts have to be constructed and compared. Therefore we used *VisMap* to analyze such large numbers of assets over time. We used the Squarified Treemap approach [BHvW00] to visualize the different finance sectors, similar to Figure 7.13. However, in contrast to the technique in Figure 7.13, we now use visual aligned layouts to show the asset performance over time, and not only at a certain point in time. This makes it easy to identify correlations of asset performance over time and to identify good and bad assets. We analyzed the performance of 920 funds contained in the Lipper Bond Index. In this database, all prices were sampled on a monthly basis during March 2002 and March 2005, whereas not each asset covered the whole time frame. The database represents European and international funds composed of stock assets partitioned into several sectors. The initial layout is shown in Figure 7.18. The sectors of the layout represent the different financial sectors whereas the size of each box reflects the number of funds belonging to that sector. As shown in the Figure, most funds

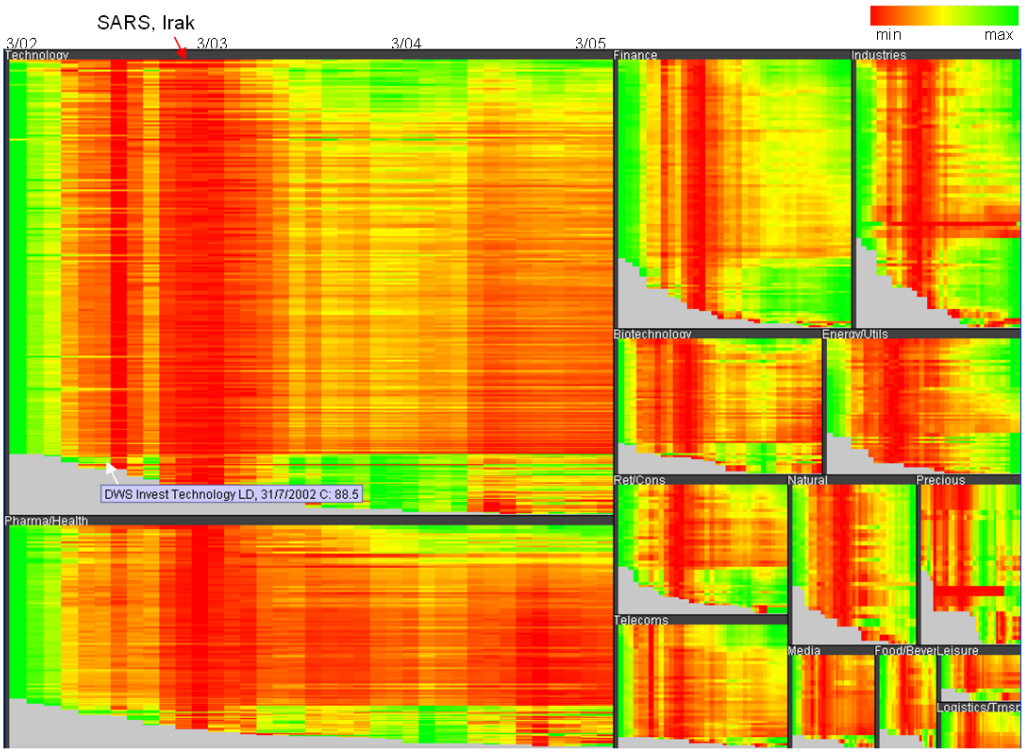


Figure 7.18: *VisMap* using Rectangular Layouts to analyze 920 Lipper Bond Stocks per sector via aligned hierarchical layouts. It is easy to identify strong (green) and weak (red) periods for each sector / asset. The funds per sector are aligned per day, and sorted by overall performance. It is easy to see the negative impact of the SARS epidemic and the Iraq war in early 2003 to the stock market.

belong to the technology sectors. Color shows the normalized stock price per asset over the whole time span.

Different ordering functions are applied to improve the visual representation. At first the funds are aligned according to their time stamps. It is easy to identify funds that were floated after March 2002. It is also easy to see that the SARS epidemic and the begin of the war in Iraq had a major impact on the stock market, leading to falling stock prices across all sectors. However, it is easy to see that fund of the finance and industry sector recovered much faster from these negative effects, than the technology or pharma sector. As shown in Figure 7.18 interactive tools like tool tip functionality can be used to get detailed information for certain funds. *VisMap* supports different visual methaphors. Suppose that the analyst is now interested to compare the average performances of all sectors compared to a certain query point (e.g. point of purchase). Therefore he selects the *CircleView* metaphor [KSS04a] and performs a roll-up operation, as shown in Figure 7.19. The

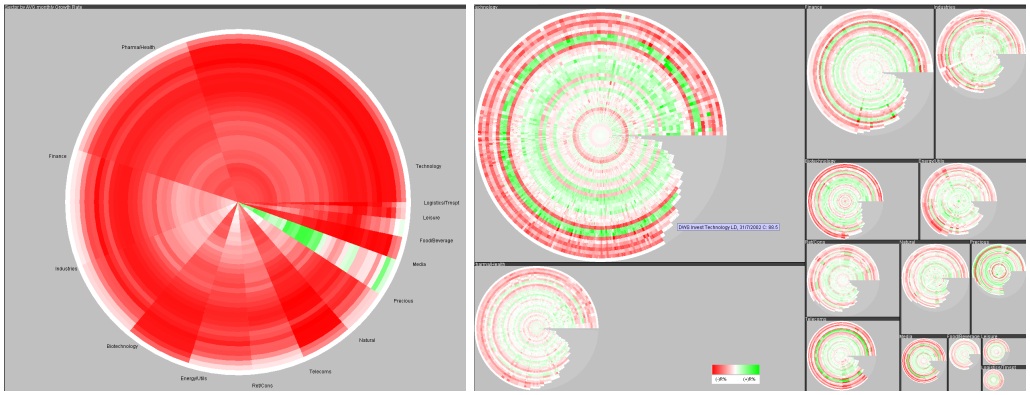


Figure 7.19: VisMap using Circular Layouts to analyze Lipper Bond stocks per sector. The left figure shows the average performance of each sector. The right figure shows the avg asset price differences between each month (e.g. point of sell) and the first month of the fund (March 2002, e.g. point of purchase) after the drill down on asset level.

left figure shows the average growth rate between the query point (March 2002, shown at the outside of the circle as white Circle Segment) and each month till March 2005. Green color indicates positive growth rates, red color negative ones. The figure shows that, if one would had invested in the technology sector in March 2002, there would have been a good chance they might have lost money till 2005. The precious sector however would have been a good investment. If the analyst now wants more details, he may drill down to the next deeper level, and analyze particular stocks. Suppose the analyst would be interested in short term trades, than he may change the colormap to reflect monthly growth rates, shown in the right figure. The figure shows, that even in this bad overall performance period 2002 - 2005, there were possibilities to make money with short term (monthly) trades, indicated by green areas. In this manner the analyst can adapt the *VisMap* layout to his demands, and even may generate line charts for single funds using linked views to get details on demand.

7.2.5 Conclusion

The *VisMap* technique is a novel approach to visualize multi-attribute time series, by transform hierarchical time related data to a hierarchical structured visual map. To speed up visual comparisons, time series are ordered and aligned according to certain criteria, such as total, average, maximum, or correlation. The experimental studies show significant advantages of the *VisMap* technique at discovering patterns, trends, and historical problems in comparison to existing space filling layout approaches.

Chapter 8

Visual Business Analytics of spatio-temporal Data

In many application scenarios, data is collected and referenced by it's geo-spatial location. The analysis of geo-spatial patterns in such data sets is an important task in many business analysis scenarios. Today even common activities of everyday life such as telephone calls or credit card payments are logged by enterprise IT-infrastructure and most of these data sets contain geo-referenced data, like addresses / cell-phone zones of caller / callee for telephone calls or the place of purchase for credit card transactions. This results in large volumes of geo-related data, stored in Data Warehouse environments. Every thorough business analysis must take these geographical information into account when looking for patterns within the data. For decision makers and analysts it is essential to rapidly extract relevant information from this flood of data in order to turn raw data into valuable knowledge. However, due to the data's complexity and volume, they are confronted with an urgent need for new methods and tools that can intelligently and automatically transform geographic data into information and additionally synthesize geographic knowledge. An example is credit card fraud protection where the geographic information of credit card transactions at certain points in time can help to prevent fraud. Credit card companies may verify customer authorizations for those transactions which show a great difference in the distance of transactions in a very short time or transactions that have been processed in high risk countries within a short time period (countries that are well known sources for credit card fraud). Therefore, effective methods for the analysis and visual presentation of geo-spatial information are needed. Moreover, it is often not sufficient to identify geo-patterns in the data, but the analyst is also interested in changes of these patterns over time. Taking all these dimensions into account results in the challenging task of analyzing multivariate space-time patterns. This makes it necessary to develop integrated analysis methods which take the attribute-, geo- and time dimensions into account, which leads to new challenges in Visual Analytics.

8.1 Introduction

Business Analysis of geo-spatial data create special challenges for the development of powerful analysis methods and for representing discovered geographic knowledge. Geo-spatial data is special since it describes objects or phenomena that are related to a specific location in the real world. Geo-spatial data consist of two essential parts, locations and attributes. The attributes describe the characteristics or properties (median household income, number of sold products,...) for a certain location. Locations are typically referenced to longitude and latitude [Sip06]. In many practical scenarios locations also refer to geographical regions (country, county, city,...) which are then mapped to x / y locations based on certain semantics. While classical KDD or Visualizaiton techniques involve highly dimensional information spaces, geographic data is unique since the location dimensions of the information space are interrelated and provide the measurement framework for the remaining attribute dimensions. Thus when analyzing geo-related data, the goal of the exploration process is the identification of geographic phenomena, and their effective visualization, using a 2-D projection. According to [Sip06], these phenomena can be classified into point phenomena, line phenomena and area phenomena.

- Point Phenomena: This category has no spatial extend and can be specified by a pair of coordinates (longitude, latitude) with one or more statistical values. Examples are census demographics with statistical values for certain households or telephone call data with x / y location of callee/caller

- Line Phenomena: These phenomena have a length but essentially no width and can be specified by a series of coordinate pairs (longitude, latitude), equally to paths of a graph. Examples are large telecommunication networks or the internet infrastructure

- Area Phenomena: Area phenomena have both length and width, and can be specified by a series of coordinate pairs, which describe the bounding polygon of the phenomena, and corresponding statistical values. Examples are the analysis of election results (votes per state, region) or sales analysis per county or state.

Up to day, the analysis of these geo-related phenomena has received significant research attention. A number of novel and interesting methods and techniques have been proposed, in the context of geographical information systems (GIS), cartography and geo-spatial data mining. The next section gives a brief outline of the most important of these techniques, a detailed overview can be found in [AA05, MDK05, Sip06]. In the context of geo-spatial data mining we proposed several novel approaches for the various phenomena [KSSP04b], which we briefly introduce in the next section too.

8.2 Geo-spatial Analysis Techniques

Visualization of Point Phenomena

The general idea when visually analyzing geo-spatial point phenomena is to represent relevant information through pixels at the corresponding geo-spatial position and to use color to encode the statistical information. These simple visualization are known as Dot Maps and are a very useful and familiar way for visualizing the spatial distribution of statistical parameters. The analysis may also involve the analysis of multiple statistical parameters, resulting in multiple Dot Maps- These techniques have been successfully used in health statistics, crime analysis, telecommunication, and census demographics [Sip06].

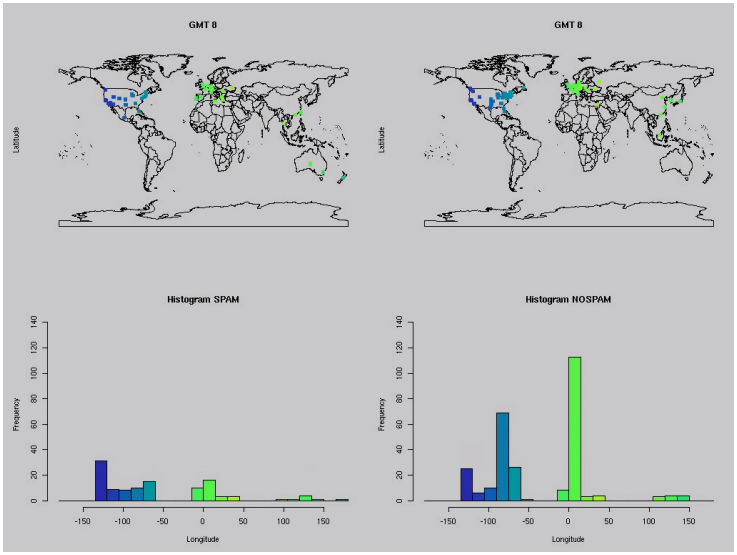


Figure 8.1: Analysis of spatial email distribution: We visualized geo-locations of email-senders, by mapping email IP's to x / y locations on a map using a geo-locator database. We used the SPAM filter classification result, to distinguish between SPAM / NO-SPAM mails. The Figure shows the emails that the author received during one week between 8.00 and 8.59 GMT. Most SPAM mails arrived from the US-westcoast, most regular mails arrived from US-eastcoast and Europe.

In our research we proposed such techniques in the context of network analysis. We developed a method for email analysis in order to identify email sender locations, which has been shown to be a very useful way to identify locations of SPAM mailers [KSSP04a, KSS+04c]. Additionally we combined the Pixel Bar Chart techniques [KHDH02] with Dot Maps in order to link sales patterns to certain spatial locations [KSDH03].

A problem with Dot Maps are occlusion effects, since usually the data is non-uniformly distributed. This may lead to the effect, that over plotting of pixels occurs in some areas of the map, while other parts of the map stay empty. Several techniques have been proposed that take these issues into account, like the Visual Points systems [KH98a], which avoids over plotting by local repositioning of pixels or the PixelMap approach [Sip06], which is based on map distortions.

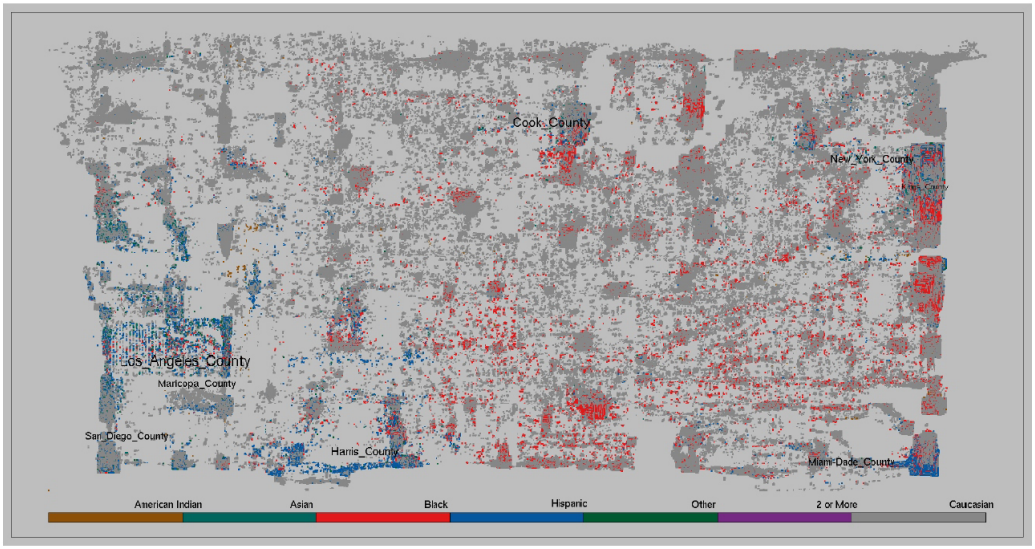


Figure 8.2: PixelMap applied to the InfoVis Contest 06 dataset (US Census Data) to explore the Global and Local Distribution Pattern of the Major Ethnic Groups, which was on of the Contest tasks: **Caucasian**: live in almost every neighborhood in the USA, **American Indian**: major pattern can be observed in Oregon, **Black:** major pattern at the West Coast and the South, **Hispanic:** Miami (Exile Cubans), West Coast, California and Texas (dynamic map labeling shows the eight most populated areas in the USA) [SSKN06]

Visualization of Line Phenomena

The basic idea to visualize line phenomena is to represent the coordinate pairs (longitude, latitude) as nodes that are mapped to the corresponding x / y location on a 2-D map and to represent the edges between nodes as lines between the x / y locations. This kind of techniques is commonly used in network analysis, where the nodes represent servers and the lines represent the server connections. Color and shape may be used to represent statistical network information. Based on such visualization it is than possible to analyze the structure of networks or to identify network traffic patterns. A tool that employs network maps is for example

AT&T's SWIFT 3D system [KKN99].

In [KSSP04a] we proposed an approach for visual analysis of SPAM mail routes that employs network maps. There we focus on the origin of SPAM email senders and traced there way from the origin to our email server. Figure 8.3 shows the regular and SPAM email path of the author. The email paths displayed in the plot have been stored since 2000. Each spatial location corresponds to a computer system from which the emails were sent. Each line segment represents the path of an email message between two computer systems. The figure on the right displays only Spam emails.

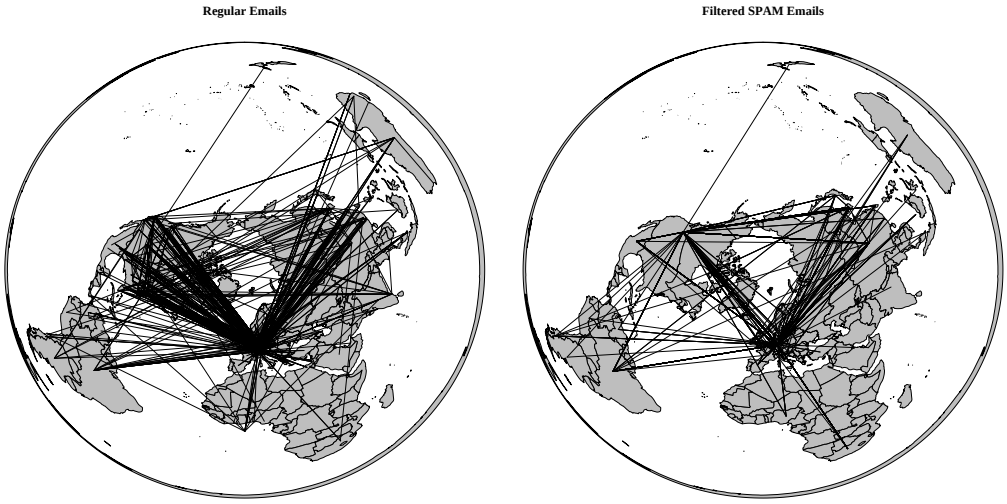
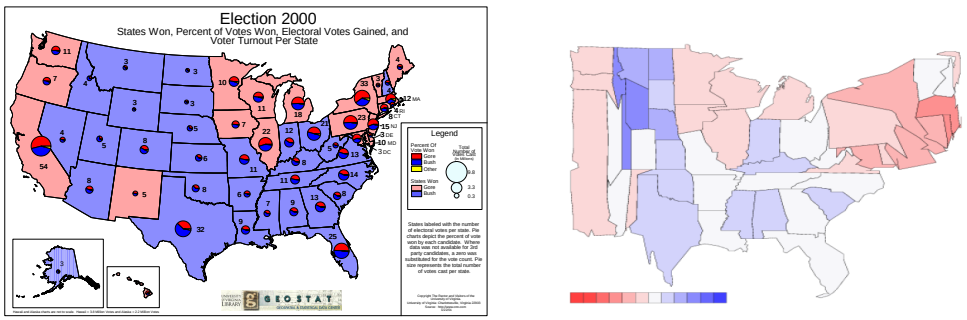


Figure 8.3: The figures display the worldwide NO-SPAM / SPAM email routes of one of our department IMAP users. The IMAP server is located in Konstanz, Germany (37 41.0N / 09 08.3E). In our department, SPAM hits one fourth of our email traffic.

It is easy to see, that most of the emails received by our department located in Konstanz (Germany) come from Europe and North America, while almost all of the emails received from South America, Africa or Asia are SPAM. This information could be used to adapt the SPAM filter. It is interesting to see that a large part (25 percent) of the emails are SPAM.
If the networks to be visualized become large and complex, network maps may again suffer from occlusion problems. An example is the internet graph [CB], which makes it hard to identify single networks. In such cases, appropriate solutions, like appropriate aggregation levels, must be found.

Visualization of Area Phenomena

Visualization of Area Phenomena is based in the presentation of closed contours. Typically a set of coordinates defines the points of a bounding polygon, which



(a) Traditional Map Design [Cen02] (b) Population Cartogram *(VP-Carto)*

Figure 8.4: Results of the 2000 US Presidential Elections (Bush (blue) vs. Gore (red)) visualized with a traditional Map and VP-Carto. The traditional map gives the impression that the blue candidate (Bush) clearly won the election, but the cartogram shows that the election was in fact really close.

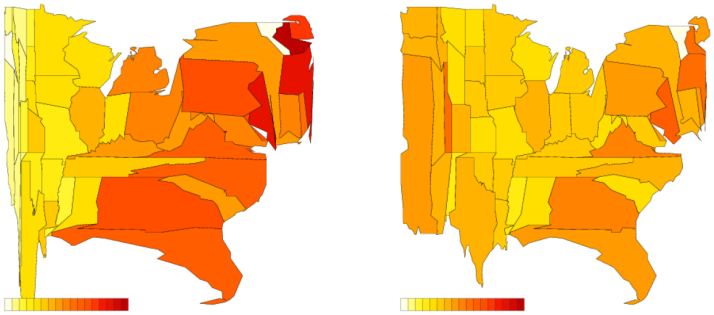
determines the area of the phenomena. Contours may be countries, states, or counties. Color and shape may than be used to represent statistical values of each contour. A typical way in GIS systems to visualize area phenomena are thematic and choropleth maps, where regular maps are used to represent the spatial distribution of statistical parameters by color coding certain sub- areas of the map. A problem with such maps is that they emphasize regions with larger areas, rather than regions with higher statistical weights [KSPN02].

To bridge this gap, a number of distortion techniques have been proposed in the literature that aim at distorting regular maps according to the statistical weights of sub regions rather than on there area proportion [Tob76, GZT95, Den96].

Cartograms for example rescale polygonal elements of a map (like countries, states,...) according to a statistical parameter, but at the same time try to preserve the shape of the global map and local regions as well as the topology of the original map. So in cartograms, the area of regions correspond to statistical parameters instead of their geographic area in the original map. In demographical analysis for example this is an important feature since it is important to visually analyze the distribution of statistical parameters rather than the geographic area of regions.

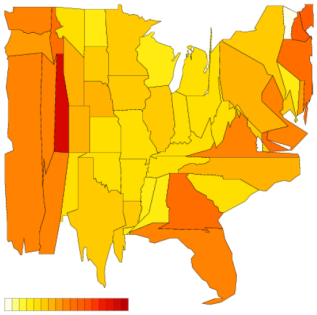
The construction of cartograms, however, is difficult to achieve in the general case because it is impossible even just to retain the original map's topology [KNP02]. Therefore a number of heuristics have been proposed [GZT93, KH98b, KNP02].

In [KSPN03] we proposed the *VP-Carto* approach for the construction of cartograms. This approach uses a quadree structure to manage and rescale the subregions of the underlying map, is very efficient and produces good results with respect to topology and shape preservation of the original map. Figure 8.4 shows an example of election data visualized with VP-Carto. In Figure 8.5 we visual-

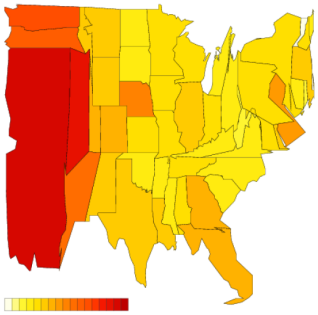


(a) 6:00am

(b) 12:00am



(c) 6:00pm



(d) 12:00pm

Figure 8.5: Long Distance Call Volume Data computed with *VP–Carto*.

ized call volume data using *VP-Carto*. The telephone call volume (normalized by population) is visualized at four different times during one day. Color is redundantly mapped to the normalized call volume with brighter colors corresponding to smaller call volumes. The resulting visualizations clearly reflect the different time zones of the US, and show interesting patterns of phone usage as it proceeds during the day. For example, we see the western part of the country shrink in size in the early part of the day (6 am EST) and slowly increase in size as the day goes on, reflecting increasing traffic originating in that part of the country. It is interesting that the call volume is especially high in the morning and in the evening (6 am on the east coast and 0 am on the west coast), while it is slightly lower during the day.

We can conclude that a number of sophisticated geo-spatial visualization techniques have been proposed. However, in the context of Visual Analytics, it is getting more and more important to not only visualize single statistical parameters over a geographical context, is is rather important to show the dynamics of multivariate attributes over space and time.

8.3 Visual Analytics of Space-Time Patterns

Although a number of effective visualization techniques for geo-spatial have been proposed, as indicated in the last section, most of these visualization techniques have limited capabilities to analyze data across geo-spatial, temporal, and multivariate dimensions. They rather analyze single attributes along the space dimension. Since in Data Warehouse scenarios the data is typically modelled along all three dimensions (space, time, multivariate attributes), an effective combination of data mining and interactive visual encodings is needed to support decision making. Therefore, it is not sufficient to show the geo-spatial distribution of single attributes at a certain point in time. It is rather important to show the dynamics of multivariate attributes over space and time.

The integrated analysis along all dimensions (geo-spatial, temporal, multivariate spaces) holds great potential to provide valuable and previously unknown information that can identify complex phenomena, especially multivariate space-time patterns. However, Visual Analytics of geo-temporal data are challenging problems, since dynamic space-time patterns and potential interesting events in space and time have in practice a much higher complexity than available visual encodings can handle.

Space-time-patterns can be seen as a series of multivariate profiles. The research challenge is to provide effective visual encodings in multi-dimensional data spaces that allow to identify multivariate geo-patterns, identify their relationship, follow their changes over time, and understand why patterns are changing. Effective visual reasoning is based on the visual understanding of patterns in an environment with multiple dimensions and the projection of their future status. To support interactive decision making, effective tools should therefore support the following tasks:

- Presentation of multivariate patterns to the data analyst using data mining and abstraction techniques. Support of interaction techniques to adjust the result by selecting central themes and dimensions
- Visualizing of uncertainty and stability of the patterns and their temporal behavior
- Highlighting temporal behavior in different perspectives and levels using coordinated views
- The projection of their status in the near future

We focus on the combination of automated data analysis methods and smart visual encodings to face this problem. The aim of our approach is to analyze real-world Data Warehouses to support the analyst in the process of decision making. We support the data analyst in analyzing multivariate patterns by providing interactive exploration of spatio-temporal properties.

8.3.1 Background

Exploring and analyzing large spatio-temporal data sets is an challenging task because of data complexity and the challenge of providing appropriate visual mappings. First approaches have been proposed in [SS06], where the authors proposed a framework for interactive mining for multi-variate patterns. Some efforts have been made in visually mining spatio-temporal patterns with focus on spatial distribution of temporal behavior [AA05]. Recently an interesting approach was proposed by MacEachren et al. [GCML06]. The authors propose a novel inquiry system for exploring space-time pattern. The system is based on a number of different views on the data, to take the different characteristics of temporal- and geo-spatial data into account. The tools combines computational methods, in particular self-organizing maps to analyze the multivariate data, and visual methods, in particular a reorderable matrix and a map matrix, to visualize temporal and geo-spatial patterns. The authors used the tool to analyze the InfoVis 2005 Contest data set.

Although these proposed methods may hold great potential to increase the value of existing analysis tools, little research has been done so far to integrate these techniques into Data Warehouse Analysis tools. The Polaris system [STH02] was designed to explore data cubes at multiple meaningful levels of aggregation, but the system focuses on analyzing multivariate patterns in the data rather than take the time and space dimensions into account.

Our aim was to support such an exploration of space-time patterns in Data Warehouse environments. Such an analysis of space-time-attributes data requires the tight integration of automated methods and interactive methods into the exploration process. Our approach provides a suite of easy-to-understand visual encodings that are able to highlight geo-spatial patterns and their interconnectedness over time stored in a Data Warehouse environment. It allows the interactive exploration of the data by providing drill-down and roll-up functionality.

8.3.2 The Visual Interface

The goal of our research is to support Visual Analytics of space-time patterns in Data Warehouse environments. Therefore we provide a visual front-end, *DWVis* shown in Figure 8.6, that allows an interactive navigation based on the underlying Data Cube structure. The interface is able to create standard reports using common chart techniques (e.g. bar charts as shown in the upper right window in Figure 8.6). However, we focused on extending theses standard functionalities of classical Data Warehouse report tools, in order to reveal more complex information from the underlying data. Therefore our exploration process follows the Visual Analytics Mantra, that means we incorporate automated methods to extract and analyze multivariate patterns and the user is than able to refine the results or to get details on demand.

The interface allows the data analyst to select a certain level of detail in the

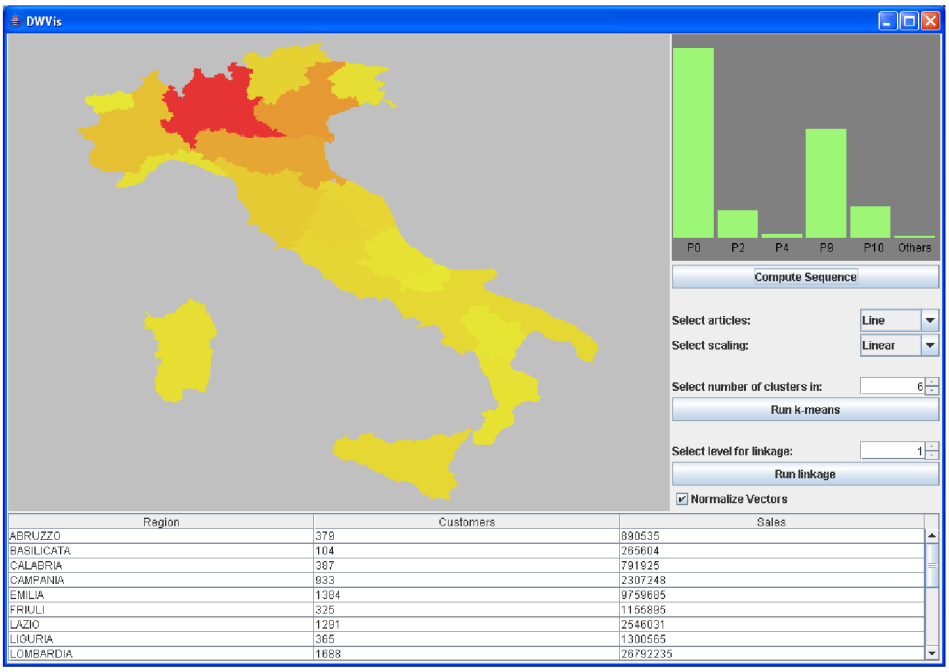


Figure 8.6: *Warehouse Interface* – Selection of space-time-attribute slices from the Data Warehouse based on Data Cube queries. The data is aggregated by grouping it into different geo-spatial entities at different geographic scales.

underlying data cube structure via drill-down / roll-up functionality. For example, when analyzing sales data over certain years, the selection (country, year, sales) would analyze the sales of a product per country over the year. The user could then drill down to (county, year, sales) to analyze the sales for all counties of a selected country over the year. Our goal is of course to visualize this geographical distribution of attributes using geo-spatial techniques. As described in the last section, most geo-spatial analysis tools are only able to visualize the geo-spatial distribution of single attributes at a certain point in time. This corresponds to the analysis of a single slice in the data cube model, e.g. the visualization of sales per country of a single product at a specific year. An example data set is shown in Figure 8.6, where the total yearly sales (2001) of products for Italian regions are visualized. The visualization is based on real-world sales data from an Italian company. We integrated geo-spatial techniques, in particular Thematic Maps, to visualize the geo-spatial distribution of attributes at a certain time step. It is easy to see that the Italian province *Lombardia* in the north of Italy, has most sales (26 million). The upper right sub windows show some additional measures, like the histogram of products that contribute to the total sales (Product 0 was sold most, note that product names are anonymized). In addition to the core

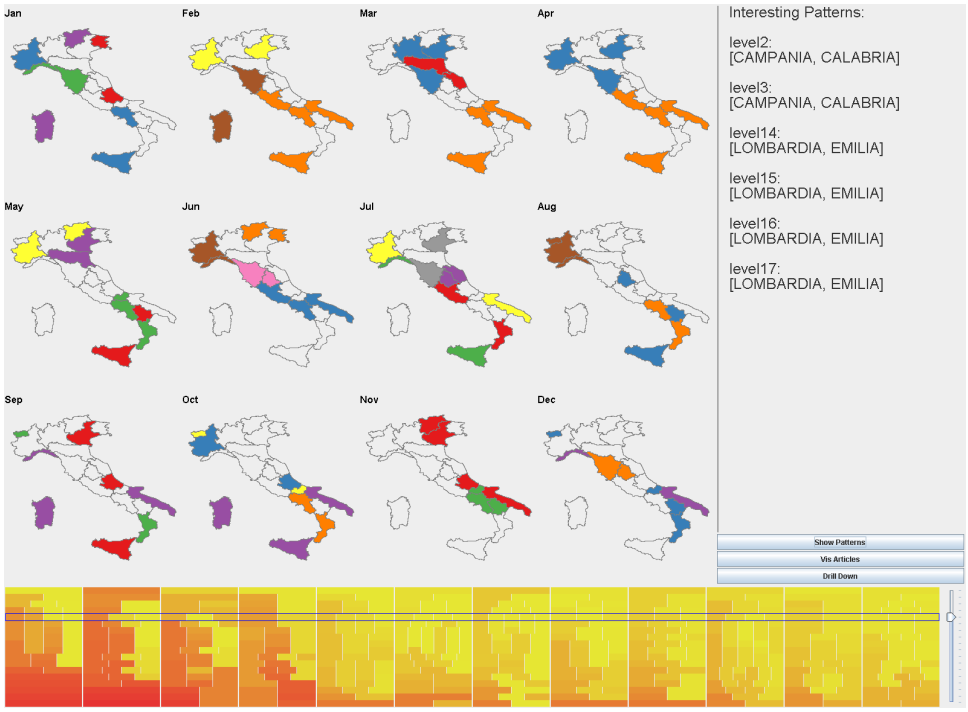


Figure 8.7: *Analysis Interface* – Highlighting the evolution of geo-spatial pattern in time. The regions Sicilia, Puglia, Campania and Lazio are grouped together in a common cluster *(orange cluster)* that is defined from February until April (some little changes in March).

visualization technique, we provide interaction techniques to enable effective data exploration. Our interface allows the data analyst to directly create and select different views to the data, e.g. by interactive drill-down operations along all data cube dimensions. However, the main benefit of our tool is its ability to analyze and visualize complex space time patterns. The basic idea is to combine thematic maps with techniques for multivariate analysis, in particular clustering techniques. The idea is to compute clusters at each time step and then to compare the patterns of neighboring time steps with respect to their pattern stability and changes in the patterns. The next sections describe our approach in detail.

8.3.3 Highlighting Space-Time Patterns

The objective of this research is to develop new methods and techniques to discover the spatial inter-connectedness of information in time over a geo-spatial context.

The complexity of the visual analysis boosts with the time dimension, and the idea is to appropriately combine smart visualizations with automatic analysis methods.

Exploration of Patterns in the Attribute Space

The basic idea of our approach is to analyze the n-dimensional geo-spatial observations at every time step, and then to analyze and visualize the changes between these single time steps.

More precisely, we identify each geographic entity as a point in the n-dimensional attribute space A_t at a particular time step t. Thus, for each geo-graphical entity and each time step we can determine a multivariate profile $(geo_{id}, t_i, a_0, \ldots, a_n)$, whereas geo_{id} defines the id of a geo-entity, t_i defines the particular time step and $a_0, \ldots, a_n$ defines the attribute properties of the geo-entity at that particular time step.

We then group similar multivariate profiles together based on their attribute properties. Since the multivariate profiles typically contain more than one parameter with more then two dependent variables, we use multivariate analysis methods to extract relevant patterns from the data. In this context, we define a geo-pattern as a group of geographical entities that have similar multivariate profiles.

In general, every multivariate analysis technique can be integrated in our tool like well known PCA, MDA or LDA techniques [Dav73]. We used clustering methods to detect multivariate patterns, which basically group the multivariate profiles into clusters with similar properties. A number of clustering techniques have been proposed in the literature [HK06, HK99], recent approaches for analyzing multivariate geo-patterns employed also Self Organizing maps (SOM) [GCML06]. In our approach we used agglomerative clustering, in particular an adapted version of Single Linkage Clustering [HK06], to detect geo-patterns. This class of clustering techniques allows us, to interactively refine the clustering by readjusting the number of clusters, and thus explore patterns at different data granularities.

Multivariate Attribute Analysis

Now we take a deeper look on the analysis algorithm. After extracting the multivariate profiles (e.g. product sales) for a certain geo-spatial level of detail (country, state, county,...) from the Data Cube via interactive query, we compute a initial proximity matrix M based on all profiles.
The $N \times N$ proximity matrix M contains the distances between clusters. Since we use agglomerative clustering, we start by considering each geo-profile as a single cluster containing only one item. M contains initially the distances between the generated geo-profiles, with $M(i,j) = dist(profile_i, profile_j)$, and $i, j \in N_0$ are assigned sequence numbers. Note that the distance between geo-related clusters can be determined in many different ways. In general two groups can be distinguished: Distance measures that are only defined over the attribute dimensions (like Euclidean or Manhattan metric) and distance measures that consider additionally the geo-spatial dimensions (like Geographical neighborhoods). In our application scenarios we focus on distances defined over attribute dimensions, in particular Euclidean distance, and emphasize the geo-related distances in our vi-

sualization step. The algorithm for identifying multivariate geo-patterns is then defined as follows:

Algorithm 3 (Clustering of multivariate profiles)
Input: *n geo-spatial profiles with* $profile_{id} = (id, t_i, a_0, , a_n)$*, proximity matrix M*
Output: *Hierarchical Clustering C*
Initialize n Clusters with $C_i = profile_i$
While: *Not all Objects are in one Cluster* **Do**

1. *Find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to* $dist[(r),(s)] = min(dist[(i),(j)])$ *where the minimum is computed over all pairs of clusters* $(i),(j)$ *in the current clustering.*

2. *Increment the sequence number :* $m = m+1$*. Merge clusters* (r) *and* (s) *into a single cluster to form the next clustering m. Set the level of this clustering to* $L(m) = dist[(r),(s)]$

3. *Update the proximity matrix M, by deleting the rows and columns corresponding to clusters* (r) *and* (s) *and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted* (r,s) *and old cluster* (k) *is defined in this way:* $d[(k),(r,s)] = mind[(k),(r)],d[(k),(s)]$

End While

The result of the algorithm is a hierarchy of clusterings (dendogram). The user can now interactively navigate through this hierarchy, to analyze different levels of details. Note that we can prune the highest (all objects belong to one clusters) and lowest clustering levels (each object represents a cluster) from consideration, since it will give the user no information on relevant pattern.

An example is show in Figure 8.8. The Figure shows the analysis of sales pattern. It is easy to see, that there are regions that have similar patterns.

This computation of clusters for a certain time step i corresponds to the analysis of a slice in the underlying data cube model along the geographical and attribute dimension and a single time step. To analyze the patterns over time, we therefore analyze the patterns for each time step, i.e. all data cube slices along the time dimension. We can now analyze the changes of patterns over time.

Inter-connectedness in Time

We are now interested in changes of patterns over time. Therefore we explore at each time step the geo-patterns over the attribute spaces A_t, and we are interested in patterns that are defined over a time period $t_i, \ldots, t_j$ with some minor changes in their cluster members. We can describe this task more formal:
Let C_{t_i} a cluster defined in attribute spaces A_{t_i}. C is inter-connected in the time period $t_i, \ldots, t_j$ if the $C_{t_i} \cap \ldots \cap C_{t_j} \neq \emptyset$. Please note, the inter-connectedness

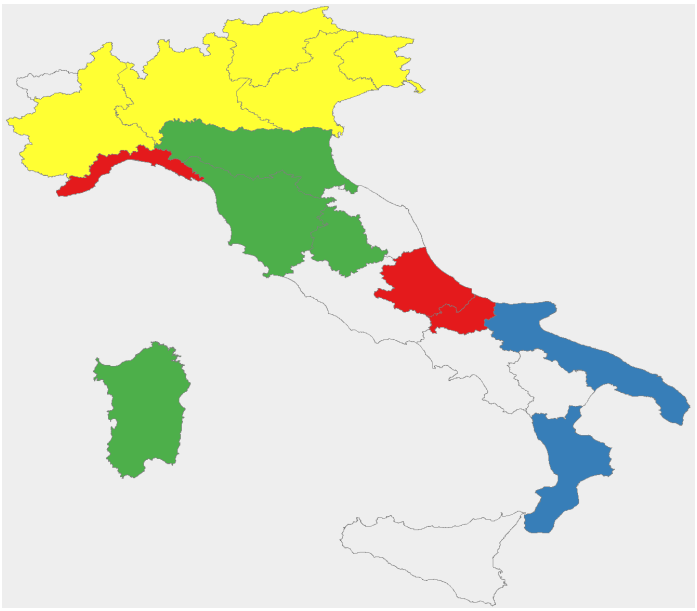


Figure 8.8: Mutlivariate analysis of sales data: Clusters are defined by color. The figure shows that there are Italian regions that have similar sales patterns.

allows to analyze the growth of patterns in both the geographic space and attribute spaces A_{t_i} because $C_{t_i} \cap \ldots \cap C_{t_j} \neq \emptyset$ if $C_{t_i} \in C_{t_j} \forall t_j$. Another interesting problem is to visualize changes in the attribute values directly. This can also be done because clusters which are not interconnected in time, change their attribute values significantly. More precisely, every geographic entity that is contained in such clusters, may significantly change its properties and may be subject of further analysis.

Visualization of the Patterns

We visualize the inter-connectedness of information in time over a geo-spatial context by highlighting and tracking events and patterns at every single time step. The basic idea is to use color to indicate the stability of patterns over time. At time step t_0 for a selected number of clusters, a clustering is computed and visualized using a certain solid color for each cluster. The clustering in time step t_1 computes the same number of clusters, and the intersection with clusters of the previous time step is computed. Clusters that are similar are visualized using the same color as in the previous clustering, if clusters are below a similarity threshold a new solid color is assigned. By showing a Thematic Map of cluster for each time step, the stability of clusters is emphasised an the user can easily recognize stable patterns. The visualization result is shown in Figure 8.7. The ordering

according to the time steps goes from left to right with line breaks at the edges of the available screen. The individual patterns are uniquely encoded by color. If an event, alert or pattern occurs over a certain time frame (event, alert, or pattern is interconnected in time) the color remains constant (to enable an efficient visual awareness). Our approach performs a comparision of trees (dendograms) resulting from the clustering for each time step, in order to idendify *stable* clusters.

Let $H(t)$ and $H(t+1)$ be two trees computed at the time steps t and $t+1$. Let l the current level in the exploration process and $\{H(t)[l]\}$, $\{H(t+1)[l]\}$ the set of all nodes defined on level l. The pair wise comparison follows the two basic steps: Starting in the top parent level we compute the pair wise set intersection for all nodes for each level l. If the intersection set of two nodes contains more than ψ percent common elements than we highlight this cluster in both time steps. The threshold ψ is given by the user; the default value is 70%.

Multi-Scale Visualization

Since Data Cubes represent hierarchical aggregations of the underlying data, it is essential to take these hierarchies into account when analyzing Data Warehouse data. Our approach allows an automated as well as an interactive drill down to analyze details. When analyzing space-time patterns on country level for example, our algorithm automatically computes space time patterns at the next lower level, for countries belonging to stable clusters. Alternatively the user may select a country and the algorithm computes space-time patterns on all states that belong to this country. This approach is a synthesis of the degree of interest proposed by [Fur88] and multi-resolution approaches by [KS05, KSS06], with extension of some ideas from scalability techniques proposed by [SSKS06].

An example is shown in Figure 8.9. Italian sales data is clustered according to product sales patterns at a certain time step t. In the north of Italy there is the largest cluster (yellow color) containing 5 states. Our algorithm now automatically computes the clustering for this cluster on the next lower level, shown in the right figure. There a number of subregions with similar characteristics that can be identified.

8.3.4 Application Examples

Interactive Sales Analysis

Based on the technical details described before, we now present an application of our approach to real world sales data. The data set contains Italian product sales. The basic functionality is shown in Figure 8.7. The main window shows the space-time patterns based on the selected hierarchy level and the selected clustering level. The lower window shows the dendogram, i.e. the cluster hierarchy. The color indicates the average distances between objects in on cluster. Shades of red indicate clusters that contain very dissimilar objects, yellow indicates similar

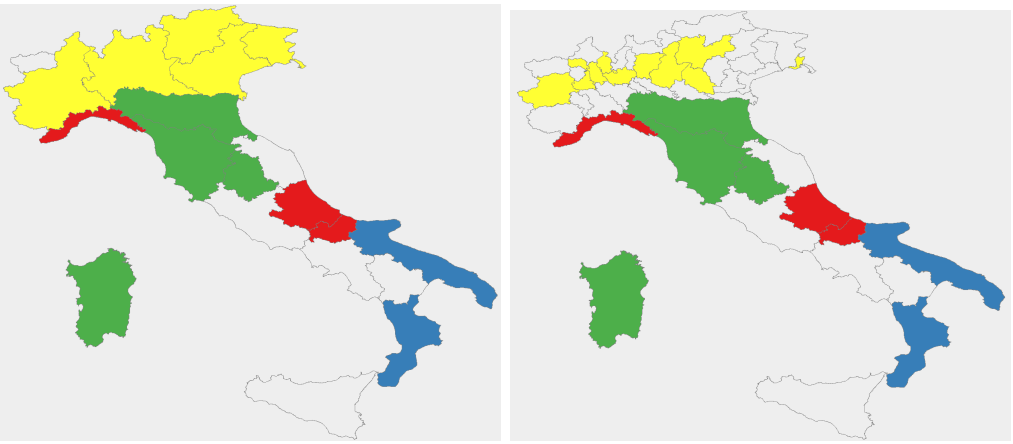


Figure 8.9: *Enhance interesting clusters automatically* – if events, alerts or patterns occur in neighboring spatial entities than the highlighting algorithm analyses the next lower level (from state into county level) to visualize more details to enable visual awareness

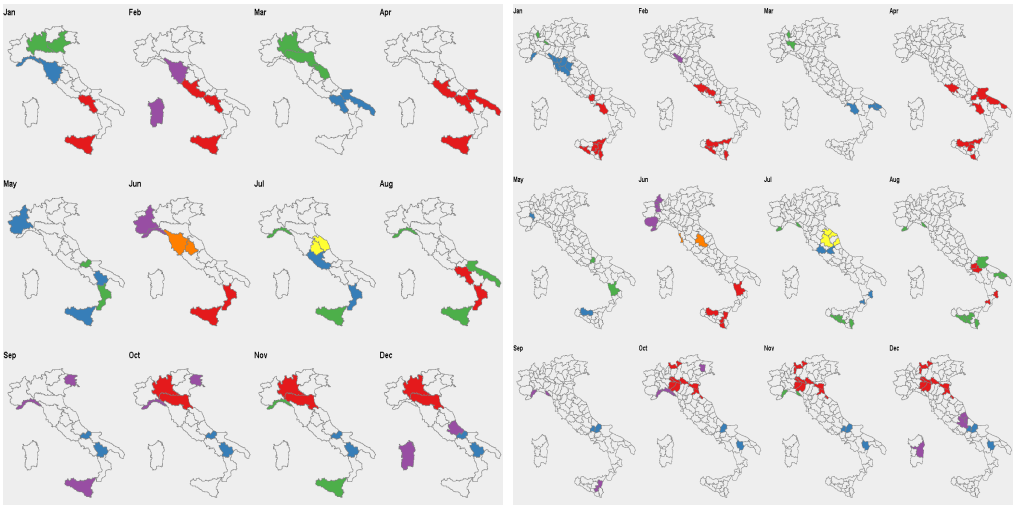
objects. This representation may help the user to find appropriate cluster levels, to find relevant patterns.

Tracking and Comprehension of Space-Time Patterns

The goal is to visualize objects, resources and their activities within a combined temporal and geo-spatial display. Figure 8.10(a) and 8.10(b) show the interconnectedness of the sales data example. In practice, at first the overview highlighting is presented to the analyst and he may then identify some potential interesting pattern for further detailed investigation. Our interactive approach allows the data analyst to explore the interestingness of patterns by navigating through the levels of the hierarchical clustering.

Customer Sales Analysis

Figure 8.10(a) shows the global spread of two major customer sales patterns. The sales pattern of the first product type (*red cluster*) starts in January and ends in April and is located in the south of Italy (Sicilia, Campania, Lazio). It shifts to the north regions Lombardi and Emilia-Romagna (see figure 8.10(b)) in the last three months. We can identify two important sales periods of this product type and it may allow a data analyst to adjust sales policies. We can see that the *green cluster* sales pattern (second product type) has a vice versa behaviour.



(a) Highlighting based on regions (Overview) (b) Top-Down Highlighting into provinces (Detail)

Figure 8.10: *Tracking Space-Time Pattern* – Highlighting Space-Time Pattern at different geographic scales. The sales pattern of the first product type (*red cluster*) starts in January and ends in April and is located in the south of Italy (Sicilia, Campania, Lazio). It shifts to the northern regions Lombardi and Emilia-Romagna in the last three months.

8.3.5 Conclusion

Geo-related data sets are ubiquitous and appear in many application scenarios. A number of sophisticated techniques have been proposed that aim at visualizing geo-related data in an appropriate way. Most of these techniques do, however, focus on the visualization of single statistical parameters or single attributes over a geo-spatial context. A typical example is the visualization of sales data, where for every country or state the number of sold items is visualized, e.g. by using color coding in Thematic Maps. In todays applications it is more and more important to not only show the geo-spatial distribution of single variables at a certain point in time, rather we have to deal with much more complex data sets which may contain multivariate patterns. Furthermore these patterns may change over time, which makes it necessary to provide powerful methods that are able to detect multivariate patterns and at the same time analyze their geo-spatial relation as well as their behaviour over time. The newly available complex, high-dimensional data sets pose a demand on Visual Analytics techniques that integrate automated and interactive methods, in order to face the challenge of analysing these space-time patterns in today's applications.

We proposed a method that is able to detect patterns in multidimensional data sets, in particular clusters in the data, and which presents these patterns over a

geo-spatial context. Furthermore our method is able to analyze the changes of these patterns over time. Our techniques follow the Visual Analytics Mantra in terms of combining automated and interactive exploration methods. We showed that our approach worked on real world business data. An important issue in our future work is to determine the probability of events and their projection to their near future, for example in case of emergency management scenarios.

Part IV

Relevance Driven Visual Analytics

Chapter 9

Introduction

9.1 Basic Concepts

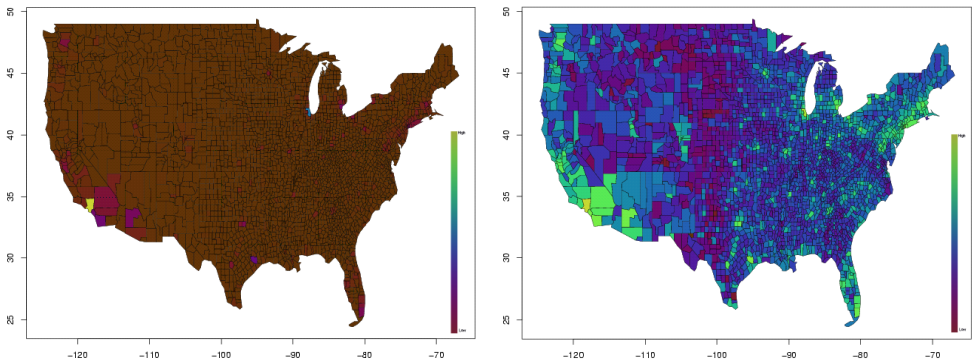
A wide variety of advanced visual exploration and visualization methods have been proposed in the past. These techniques have proven to be of high value in supporting researchers and analysts to obtain insight into large data sets and to turn raw data into useful and valuable knowledge by integrating the human in the exploration process. However, with the increasing volume and complexity of today's data sets, new challenges for visualization techniques arise. To keep step with the growing flood of information, visualization techniques are getting more sophisticated, e.g. by integrating automated analysis methods or providing new visualization metaphors as proposed in the context of Visual Analytics [TK05].

But this also means, that visualization techniques are getting more complex, forcing the user to set up many different parameters to adjust the mapping of attributes to visual variables on the display space. In classical data exploration, playing with parameters to find a promising parameter setting is an important part of the exploration process, but with the increasing number and diversity of the parameters it becomes more and more difficult to determine a good parameter setup, which is vital for insightful visualizations.

For example, if we have 50 attributes (or attribute dimensions) and 4 parameters for the visual mapping, as e.g. in Pixel Bar Charts [KHDH02] employed in Section 11.1.2, then we have over 5 million possible mappings and it is very unlikely to find useful ones interactively.

Suboptimal parameter settings or the investigation of irrelevant data dimensions make the exploration process tedious and an interactive search impossible. In general, finding a good parameter setup is a challenging task for the analyst, since it is often not clear what is the best parameter setting for a given task, due to the huge parameter- and attribute space [Spe01].

Dimension management techniques can help the user by giving reasonable selections or orders of the dimensions that might be relevant for visual analysis.



(a) Linear Colormap (b) Logarithmic Colormap

Figure 9.1: *A typical application scenario* – The visual analysis of a census data set involves different normalizations to a color scale. Although both visualizations are based on exactly the same input data, the right figure provides more insight since a logarithmic color scale is more suitable for the underlying data distribution

However, in Exploratory Data Analysis it is often not clear in advance which dimensions are more important than others. Furthermore, there are many other parameters that have an impact on the effectiveness of the resulting visualization, for example the selected normalization to a color scale.

A simple application scenario is shown in Figure 9.1. The figure shows two choropleth maps visualizing USA population density data at county level. The two maps are based on the same input data, but created with two different parameter settings. More precisely, in the left figure a linear color mapping was chosen, in the right figure a logarithmic color mapping was chosen. It is easy to see that the linear mapping provides much less insight to the data than the logarithmic mapping, because the data is highly non-uniformly distributed.

For example, very high populated areas around Los Angeles, Chicago or Manhattan cause uniform dark colors for the remaining USA and it is almost impossible to see fine structures or differences in population density among them. In practice, the analyst does not know a priori which normalization function is best suited for a given dataset and he may test some preferred ones. Of course, there are typically much more parameters that have to be selected.

But the growing data complexity and data volumes do not allow such playing with data by hand anymore. Therefore we aim at supporting the user in finding promising parameter setups from the available parameter space to speed up the exploration process. We present a framework that employs automated analysis methods to detect potentially useful parameter settings for a given pixel-based visualization technique and an associated input data set, with respect to a given user task like clustering or outlier detection.

9.2 Related Work

In many application scenarios, analysts have to deal with large parameter spaces when using visualization techniques to explore large data sets. These parameters control the visual encoding of the data, including the selection of attributes from the input data, the selection of the color scale, algorithm parameters, the selection of visual variables and so on. The problem is that the optimal parameter setting for a given task is often not clear in advance, which means that the analyst has to try multiple parameter settings in order to generate valuable visualizations. Since such selections can hardly be done manually, the integration of automated methods to support the analyst has been recognized as an important research problem in the context of Visual Analytics [TK05].

As explained in Chapter 4, the problem of automatically supporting the user in constructing insightful visualizations is in practice a two stage problem: Dimension management and appropriate visual mappings. A number of approaches for dimension management, including dimension ordering and filtering techniques, have been proposed as shown in Chapter 4.

However, since these techniques focus exclusively on the analysis of the data and not on the visualization, they do not guarantee that the resulting visualization provides much insight. Often these techniques do not consider local phenomena in the data or additional visual parameters like the color coding or color scaling. Since there does not yet exist an empirically verified theory of human perception capabilities that can be used to prove theorems about the effectiveness of visualizations [Mac99], it is usually not even clear how to map the selected data to visual parameters. Therefore, we propose an approach that on one hand uses analytical techniques for dimension management and takes the state-of-the-art visual mapping heuristics into account, but at the same time analyzes the resulting visualizations with respect to certain user task.

First approaches in this direction have been proposed in the context of visualization. In [HBW05] a semi-automated technique to search the visualization parameter space with applications in surface texturing is presented that focuses especially on perceptual and aesthetic concerns. The basic idea is to employ a genetic algorithm to guide a human-in-the-loop search through the parameter space. The approach produces some initial visualizations constructed from different parameter settings (parameter vectors). The resulting visualizations are rated by the user and this rating is then used to guide the progress of the genetic algorithm. This technique follows approaches proposed in [Sim91, Gre05] which coupled image generations with user feedback in the context of genetic algorithms. As a result the approach builds a database of rated visualization solutions. Data mining techniques may then be used to extract information from the database. The drawback of these approaches is that the user is still involved in the evaluation stage, that means that the number of visualizations that can be evaluated is rather limited.

In the field of InfoVis some techniques were proposed which avoid this problem

by exclusively applying automated methods. In [WAG05, Tuk77, TT85] graph theoretic approaches to analyze scatterplots were proposed. This work called Scagnostics highly influenced our work. Since scatterplot matrices contain as many scatterplots as there are pairs of parameters (attributes), they do not scale well to high numbers of dimensions. Therefore it would be useful to reduce the number of scatterplots by pruning irrelevant ones with respect to a given task.

The goal of the mentioned approaches was to find interesting attribute relationships by creating scatterplot matrices from the data and then analyze each scatterplot that reveals a relationship between two attributes for certain properties using graph theoretic methods. The basic idea is to construct geometric graphs based on the data points of each scatterplot and then to compute relevance measurements from these graphs. For example, properties of the convex hull and the minimal spanning trees of the scattered points are used for outlier or cluster analysis. These techniques have shown that automated analysis works well to filter relevant from irrelevant scatterplots.

With our approach we extend this idea to a broader set of visualization techniques. We provide analysis functions to analyze both patterns in the data using data analysis techniques as well as the patterns contained in the images by using image analysis techniques. Hence we suggest a general process model for automated parameter space analysis and show how we applied this model to pixel based visualization techniques namely Pixel Bar Charts and Space Filling curves. Although data mining methods are commonly used for data analysis, our approach is novel since only little research has been done on analyzing visualizations with respect to their information content using image analysis methods.

The next chapters focus on the problem of automatically searching through visualization parameter space to support the user in finding promising parameter settings to speed up the exploration process [SSK06]. Since we deal with pixel images resulting from visualizations instead of scatterplots, we call our approach Pixnostics instead of Scagnostics.

Our approach on one hand uses analytical techniques for dimension management and takes the state-of-the-art visual mapping heuristics into account, but at the same time analyzes the resulting visualizations with respect to their potential relevance for the user. We present application examples that show how this combination of analysis methods can help to support the user in construction insightful visualizations by automatically extracting potentially useful parameter vectors from the underlying candidate parameter space.

Chapter 10

Automated Parameter Space Analysis

10.1 Problem Definition

10.1.1 Visualization Parameter Space

The challenging task in generating expressive visualizations is to find an adequate visual encoding of the input data set in the data display space. The visual encoding depends on the input data, the employed visualization technique and the visual variables given by a particular parameter setting. In classical data exploration, the mapping to visual variables described by Bertin [Ber67], such as position (x,y), size, or color is manually controlled by the user.

More precisely, the central process of visualization V can be described as

$$I(t) = V(D, S(P), t)$$

where data D is transformed using a specification S into a time varying image $I(t)$, according to the work proposed by Wijk [vW05]. To simplify matters, we aim at determining initial parameter settings for non-animated visualizations, therefore the time t can be excluded from our considerations. (Note that the complexity of the problem would be boosted exponentially, if we take time into consideration). Figure 10.1 illustrates this classical visualization pipeline.

Based on the formula above, the data set D is given as input data within a database environment with $D = (d_1, \ldots, d_n)$. In general, the employed visualization technique V is defined by the application scenario and is given by the user. In [Mac99] an approach is proposed where the visualization technique is determined automatically, but this approach is limited to relational data.

In the following we use novel pixel-based techniques namely Pixel Bar Charts [KHDH02] and Jigsaw maps [Wat05] as examples to explain our new idea, and we focus mainly on demographic data provided by the US Census Bureau to evaluate

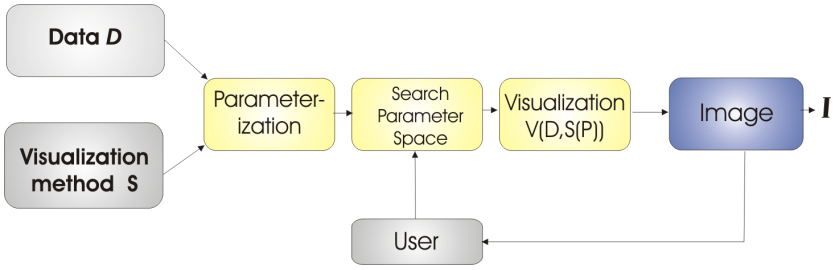


Figure 10.1: Classical Visualization process: The user has to find an optimal parameter setting manually. Ideally such a setting should produce an insightful visualization I

our approach.
Pixel Bar Charts (described in Section 11.1.2) have four parameters and more to adjust the visual encoding. It uses one parameter to separate the data into bars, two parameters for ordering pixels in x and y direction and one parameter for color coding. Obviously, the manual selection of useful parameter settings does not scale with practical data sets, since the number of parameter settings which the user may try is limited.
In our setting the input data D and the visualization technique V is given by the user and $P = \{P_i = (p_i^1, \ldots, p_i^m)\}$ as instance of a parameter setting generating image $I(S(P_i))$ is determined by the system.

10.1.2 Limits and Problem Complexity

Most visualization techniques can handle less attributes than provided by the dimensionality of the input data set, so in the visualization step potentially useful attribute combinations must be selected from the data. As an example, we consider a data analyst who wants to use the Pixel Bar Chart technique to analyze real world customer purchase data. Such data sets typically contain at least 20 dimensions (attributes), including name of item, price of item, name / id of the customer, status and so on. The number of possible parameter settings $P_i = (p_i^1, \ldots, p_i^m)$ that control the visual encoding and therefore the number of possible images I_{P_i} is defined by the number of different attribute combinations of size m from the available number of dimensions d, given as:

$$\left|\{I(S(P_i)) = V(D, S(P_i), t)\}\right| = \frac{d!}{(d-m)!}$$

For the Pixel Bar Chart example, we may specify $m = 4$ attributes at once from the input data with 20 dimensions, which would result in 116,280 mappings. This number may be increased by additional parameters like different colormaps or different scalings. The equation above also shows that increasing dimensionality boosts the parameter space exponentially. If we have 50 attributes the number of

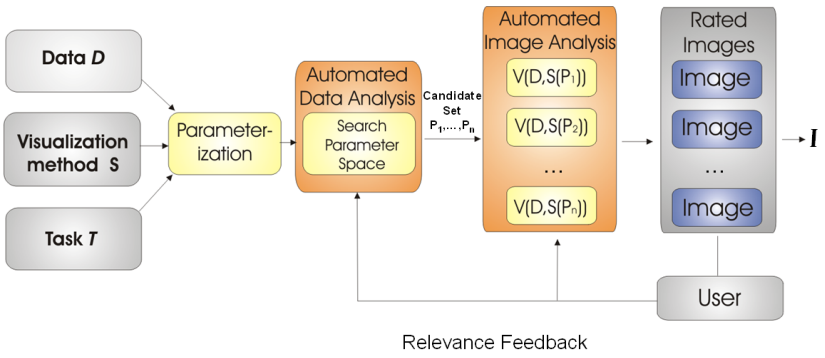


Figure 10.2: Pixnostics process model: Combining task-dependent automated data- and image analysis techniques.

possible mappings is 5.5 million. Then the analyst faces the problem of how to determine interesting subsets from the available data dimensions for visual analysis that could reveal interesting relationships.

10.2 The Process Model

Our Pixnostics approach follows a three step process based on the current task-at-hand:

- **Analytical Filtering and Pruning** of the set of possible images $\{\{I(S(P_i))\}$ by analyzing the parameter space P_i. The aim is to extract useful attribute selections and useful parameter settings automatically (candidate set CS),
- **Image Analysis** of the remaining candidate set CS – generating visualizations using the determined candidate attributes
- **Ranking and Output** of the candidate set CS – providing a ranking of candidate images I_{CS}

Figure 10.2 illustrates the Pixnostics process model. In classical visual exploration, the user visually analyzes a collection of data items to find answers to various questions (analysis tasks). In our framework, an analysis task T describes conditions which the data items needs to fulfil in the resulting visualization, which is the input of the Pixnostics pipeline together with the data D and the specification of the visualization method S (left side in Figure 10.2). Then data analysis and image analysis methods are applied to select potentially interesting visualizations and present them to the user (center / right side in Figure 10.2). An issue for future work is to adapt these functions by integrating the user feedback in form of a relevance feedback loop.

Since the applied methods highly depend on the selected task, it is one of the major challenges to identify the most common tasks, identify their impact on unique visual properties in the resulting image I, and finally to find adequate analysis functions for each of the tasks i.e. to find good predictors for these properties in images. Unique properties in images are homogenous areas, color outliers, edges and segments etc. Previous studies on visualization design proposed a range of different analysis goals and tasks [Shn96] [KK93] [Cas91] [Chi00] [Kei00]. They propose individual taxonomies of information visualizations using different backgrounds and models, so that users and analysts can quickly identify various techniques that can be applied to their domain of interest. Based on the proposed approaches we identified the following generic tasks: identify, locate, cluster, associate, compare, correlate, match and sort whereas we focus in our experiments mainly on cluster- and outlier analysis.

In the following we describe the individual steps in more detail.

10.2.1 Step 1: Analytical Filtering and Ordering

In practice, the number of attributes is greater than the capabilities of most visualization techniques. The first step of our Pixnostics approach is therefore to determine relevant relationships among the different attributes analytically using dimension management techniques. In our experiments we use classical data mining techniques and statistic measures, more precisely, correlation analysis, partial matching techniques, classification techniques, and cluster analysis to accomplish this step, but of course other task specific analysis techniques may be used as well.

Correlation Analysis

Attributes that are correlated may be interesting for detailed analysis, because they may reveal relevant impact relationships. Therefore we employ correlation analysis to find groups of correlated attributes. We determine the pair-wise global correlations among all measurements as given by Pearson's correlation matrix. Pearson's correlation coefficient r between bivariate data A_{1i} and A_{2i} with $(i = 1, \ldots, n)$ is defined as

$$r = \frac{\sum_{i=1}^{n}(A_{1i} - \bar{A_1})(A_{2i} - \bar{A_2})}{\sqrt{\sum_{i=1}^{n}(A_{1i} - \bar{A_1})^2 \sum_{i=1}^{n}(A_{2i} - \bar{A_2})^2}} \tag{10.1}$$

where $\bar{A_1}$ and $\bar{A_2}$ are the means of the A_{1i} and A_{2i} values, respectively.

If two dimensions are perfectly correlated, the correlation coefficient is 1, in case of an inverse correlation it is -1. In case of a perfect correlation, we can omit one of the attributes since it contains redundant information. In most cases, however the correlations are not perfect and we are interested in high correlation coefficients and select sets of highly correlated attributes to be visualized.

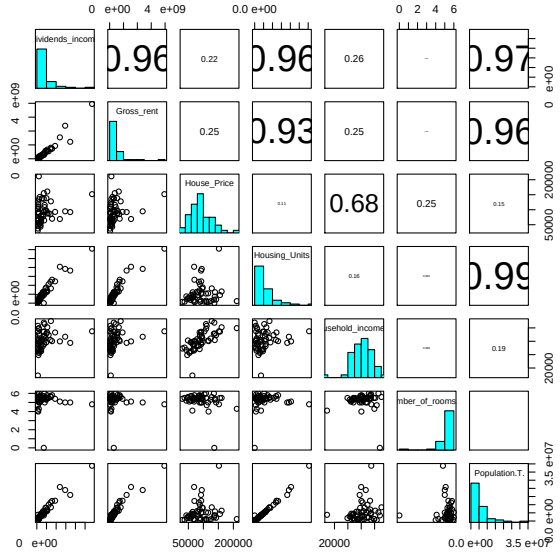


Figure 10.3: Identifying correlations in census housing data on US state level: Besides trivial correlations (e.g. Population and Number of House Units), some interesting correlations are revealed, e.g. between population and gross rent (because of demand and supply effects)

In Figure 10.3 an example from the census housing data [oC] on U.S. state level is shown, correlation coefficients for pairs of attributes are shown in the upper right half of the matrix, histograms in the diagonal show the data distribution. The data set contains, for example, information about US education levels, crime rates, housing or household incomes on different levels of detail (country, state, county, block level). Typical exploration tasks focus on the extraction of information about housing neighbourhoods for particular areas within the US including the identification of correlations between statistical parameters like household income, house prices, education levels and crime rates. The figure clearly shows that states with high total population have high gross rents (0.96) or that Median Household incomes are correlated with Median House prices per state (0.68). The analyst may now investigate such relations in more detail.

An available alternative for adjacently depicting similar dimensions is to use the normalized Euclidean distance as a measure for global similarity Sim_{Global} defined as:

$$Sim_{Global}(A_i, A_j) = \sqrt{\sum_{i=0}^{N-1} (b_i^1 - b_i^2)^2} \tag{10.2}$$

where $b_i^j = (a_i^j - min(A_j))/(max(A_j) - min(A_j))$

The global similarity measure compares two whole dimension such that any change in one of the dimensions has an influence on the resulting similarity. The defined similarity measure allows it to determine groups of similar attributes for the following visualization. Since in general, computing similarity measures is a non-trivial task, because similarity can be defined in various ways and for specific domains, special measures may be included for specific tasks.

Cluster Analysis

In order to perform a visual analysis, it is important to have the option to partition the data appropriately and then to focus on certain parts of the data. Cluster analysis can help to do this based on the characteristics of the data instances. The cluster analysis may, for example, find out that the data instance of a data set may be partitioned into different groups, which may then be independently analyzed using visualization techniques. Since attribute parameter values may be continuous (sales amount) or categorical values (item name) the clustering approach has to take these properties into account. There are a large number of clustering methods which have been proposed in the literature ([HK99] presents a nice overview). In the Pixnostics prototype, we employed *k-means* clustering [KMN+02], one of the most popular approaches.

Classification Analysis

In some applications, for example in visual root-cause analysis, the goal of the data exploration is to understand the relationship between data attributes and some specific target attribute, e.g. which attributes have an influence on the target attribute. The task is to find the attributes which are best predicting the outcome of the target attribute. A well-known heuristic for this task is the GINI index [Zyt02], which is commonly used in decision tree construction.

Given a target attribute (e.g. a business metric) A_T which is partitioned into a disjoint set of k classes (e.g. accept, reject) or value ranges (e.g. large, medium, small) denoted by $C_1, \ldots, C_k$, $(B = \bigcup_{i=1}^{k} C_i)$, then the GINI index of an attribute A which induces a partitioning of A into $A_1, \ldots, A_m$ is defined as

$$InfoGain_{GINI}(A_T, A) = \sum_{i=1}^{m} \frac{|A_i|}{|A_T|} GINI(A_i) \tag{10.3}$$

where

$$GINI(A_i) = 1 - \sum_{j=1}^{k} \left[\frac{|C_j|}{|A_i|} \right]^2$$

The *InfoGain* is determined for all attributes and attribute combinations and the attributes with the highest *InfoGain* with respect to the target attribute A_T are chosen for visualization. These attributes are best predicting the outcome of the target attribute and therefore they may be relevant for detailed analysis.

10.2.2 Step 2: Image Analysis

Once we have selected candidate parameter settings $P_i, i = 1, ..., max$ based on promising attribute selections where max is the number of parameter settings, we generate visualizations by computing all possible mappings of the candidate parameter set to visual variables.

We then apply image analysis methods to determine the relevance, i.e. the potential value of each image (visualization), with respect to the given task. In comparison to existing semi-automated methods where the user is forced to rate the generated visualizations, the image analysis is completely automated. The goal is to process the images in order to generate some measurements of its relevance with respect to a given task T. The challenge is to find adequate image analysis functions to reach this goal. Numerous image processing operations for many different tasks exist in the literature, e.g. for Image Segmentation, Image Retrieval, Edge Detection, Image Denoising or Image Inpainting [Man01]. Many of these techniques may be very useful in visualization analysis.

In our initial experiments we focused mainly on the Information content of each resulting visualization $V(D, S(P_i))$ and employed this measure to compute the relevance of each visualization for certain tasks. A very promising way to extract such information from an image, besides well known color histograms, is Shannon's entropy measure [EM95]. It is frequently used in image processing and analysis. In [KLO04] an entropy based approach for image cluster analysis is proposed; a more general approach for image retrieval using entropy is introduced in [ZIB01].

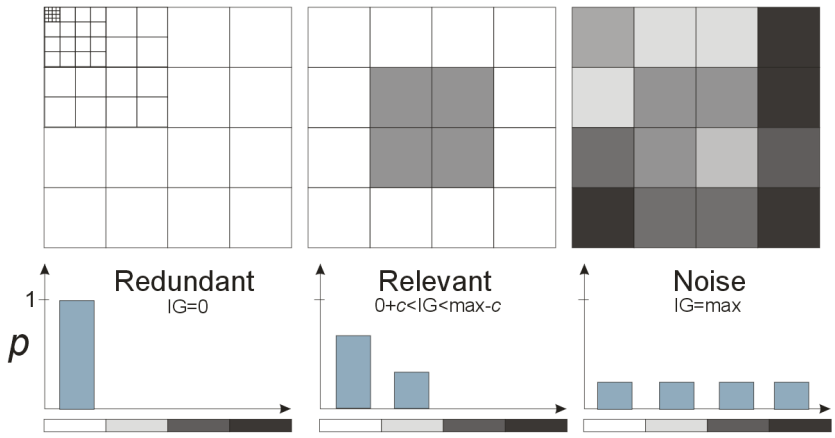
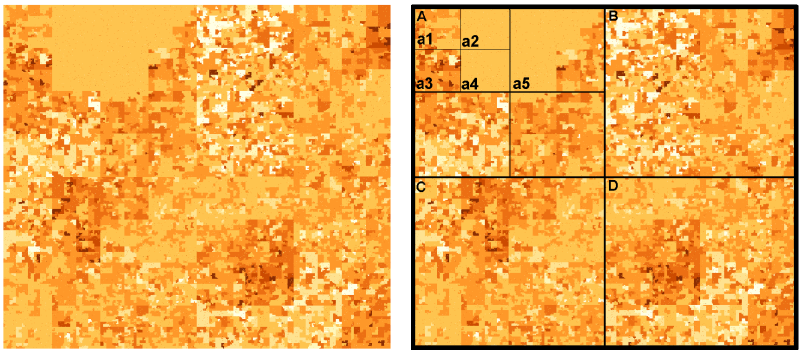


Figure 10.4: Information content (IG) of different gray level images. From an analyst's point of view, interesting images should have an Information content in a certain range c between 0 and IG_{max}

In the first step, we generate and store $I_{Pi} = I(P_i) = V(D, S(P_i))$ as a matrix U of scalars representing gray scale values, the pixel-matrix representation with

$U = (u_{i,j}), i \in [0, ..., I_{width}], j \in [0, ..., I_{height}]$. (color images are converted to gray values). The base for our analysis is the distribution of gray values within the image. Thus we are interested to know the pixel distribution H in certain areas



(a) Original NY state Jigsaw (b) Generated Image Gridcells

(c) Coarse Grid Resolution (d) Fine Grid Resolution

Figure 10.5: Basic idea of grid based information content: Based on the entropy values for certain grid resolutions, measurements for the relevance of the image are generated. Darker gray levels correspond to higher entropy values.

of the image as a function of gray levels g. Image histograms are an efficient way to reach this goal. The histogram of the $2D$ image $g(U)$ can be seen as a $1D$ function $H[g]$ where the independent variable is the gray value g and the dependent variable is the number of pixels H with that level. We can then use the histogram properties to make assumptions about the information contained in the image. For example, if most pixels in an image are contained in a small range of gray levels, the image can be seen as redundant since it provides little new information and thus the underlying parameter setting would not lead to insightful visualizations. If there are too many different gray levels, the image represents noise and it is not likely that it contains relevant information. An image with a bimodal histogram (i.e. a histogram with two peaks) may contain clusters and may be relevant for

visual exploration. Since all pixels in the image must have some gray value in the allowed range, the sum of populations of the histogram bins must be equal to the total number of image pixels N:

$$N = \sum_{g=0}^{g_{max}} H(g)$$

where g_{max} is the maximum gray value ($g_{max} = 255$ for an 8-bit quantizer). The histogram function is equal to the scaled probability distribution function $p(g)$ of gray levels in that image:

$$p(g) = \frac{1}{N} H(g) \text{ with } \sum_{g=0}^{g=max} p(g) = 1$$

Based on the probability distribution, we can now compute Shannons Entropy, which is equal to the minimum number of bits which are required to store the image. If the probability of gray level g in the image $u(i, j)$ is represented as $p(g)$, the definition of the quantity of information in the image is:

$$E(g) = -\sum_{g=0}^{g_{max}} p(g) \log_2(p(g)) \tag{10.4}$$

From this definition, it is easy to show that the maximum information content E is obtained if each gray level has the same probability; in other words, a flat histogram corresponds to maximum information content. The minimum information content $E = 0$ is obtained if the image contains only one gray level. Since minimal information content means redundancy and maximum information content means information overload or noise, the interesting images should have an information content in between, e.g. in a task-dependent range c shown in Figure 10.4

Alternatively, we use the standard deviation *stdev* as a measure of spread of gray levels g in a given image I with N as the number of different gray levels in the image:

$$stdev(I, g) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (g_i - \overline{g})^2} \tag{10.5}$$

Since we want to analyze the images not only in whole, but also find interesting local patterns in the image, we use a regular grid to separate the image in regular grid cells and then apply the methods mentioned above to compute values for each grid cell. The computation of the information content of each cell is identical to the methods described above. The only difference is that from the individual grid values we then compute a single relevance value for each image, described in the next section. To adapt our method to given application scenarios, we do not use a fixed grid-resolution, instead the grid resolution can be adapted by a parameter as shown in Figure 10.5 (b).

Algorithm 4 Entropy-based Image Analysis

Input: Candidate Set $C = \big\{\{A_D^1, \dots A_D^h\}, \{P_1, \dots, P_l\}\big\}$
with Candidate Data Attributes $A_D^1, \dots A_D^h$ $(h < n)$,
Candidate Parameter Settings $P_1, \dots, P_l$: $(l < k)$,
Visualization V with Specification S
Performed Task T
Output: Ranking Scores $R(\big\{I(S(P_i)) = V(\{A_D^1, \dots A_D^h\}, S(P_1, \dots, P_l)\big\})$

Procedure Visualization Analysis
Generate $\big\{I_i = V(\{A_D^1, \dots A_D^h\}, S(P_1, \dots, P_l)\big\}$
for $i \leftarrow 1$ to $|C|$ **do**
  ComputeRegularGridonImage(I_i)
  for each GridCell $GC(I_i)$ **do**
    ComputeEntropyValues $f(GC(I))$
  for all Entropy values $f(GC(I_i))$ **do**
    R =ComputeRankingScore($f(GC(I_i)), T$)
Return Ranking Scores $R(I_i)$

10.2.3 Step 3: Ranking and Output to the User

Having a function f, such as $E, stdev$, which measures the information content of an image, we can now compute rankings from the candidate parameter sets with respect to a given user task T. To show the basic concepts of our approach, we focused two major tasks that are common in most analyses processes, namely outlier analysis, including the search for local outliers or values of interest (e.g. find all counties or cities that have similar household income or unexpected household income), and cluster analysis (e.g. find areas with similar statistical parameters). In our prototype framework, we provide ranking functions for both tasks and show how we applied it to real world data sets. Of course, the user may also use other ranking functions for specific tasks, which can be easily integrated into the Pixnostics framework.

Computing the Global Ranking Score

To determine the relevance of each visualization, we have to compute a global ranking score based on the grid cell values of each image. There can be done in many ways, depending on the user task. Besides simple aggregation functions like means or stdev, which have shown to be very useful to prune a large number of irrelevant visualizations in a pre-processing step, we employed several more sophisticated functions. For cluster analysis for example, we initialize each regular grid cell $GC(I_i)$ with its information content score $f(GC(I_i))$.

Since we are interested in images that provide good cluster properties, a high

ranking score should be given to images where locally close grid cells have similar low grid values. Therefore we assign value 1 to grid cells where the content score $f(GC(I_i))$ is below a threshold e and 0 otherwise. The threshold e defines whether a grid cell is considered to have good or bad cluster properties. This value can be defined by the user or determined automatically by a noise level factor [HK98]. There exist also a treshold for the lower bound, which can be set, but in the clustering example this value is set to zero. After values 1 (good cluster properties) or 0 (bad cluster properties) is assigned to each cell, we can now employ an autocorrelation measure, to analyze if there exist large connected areas in the image with good clustering properties. Again many methods can by employed in that step. The BB Score (or BlackBlack count) [Dav73] which is commonly used in geospatial analysis, is a common way for that task. It analyzes the neighbouring cells of each grid cell, and compares their values. Formally, the BlackBlack count is defined as:

$$BB = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}u_{ij}$$

where $u_{ij} = Z(c_i) * Z(c_j)$

$$w_{ij} = \begin{cases} 1 & \text{if cells i and j are connected} \\ 0 & \text{else} \end{cases}$$

$$Z(c_i) = \begin{cases} 1 & \text{if } e_1 < W_i < e_2 \\ 0 & \text{else} \end{cases}$$

where e_1, e_2 thresholds and $W_i =$ value of cell i

This approach takes the (Entropy) values of each grid cell $c(i, j)$ as input and returns a measure that indicates if adjacent cells have similar values. Images where adjacent cell have similar values have a higher BB score than cells where they have different values. Therefore we rank images with high scores higher, since they may provide better cluster properties.

Depending on the task at hand, other relevance functions may be employed. Besides standard aggregate operations like average, stdev, min and max values, more complex functions can be used to compute a relevance score for each image. The inverse normalized term frequency, for example, is a common choice for outlier analysis. Image processing techniques, like edge detection or object recognition algorithms may be used to rank images according to the number of contained objects. We defined and evaluated a number of relevance functions for a number of visualization techniques and user tasks. The next chapter will decribe these functions and experiments in detail.

Chapter 11

Evaluation and Application

11.1 Application Examples

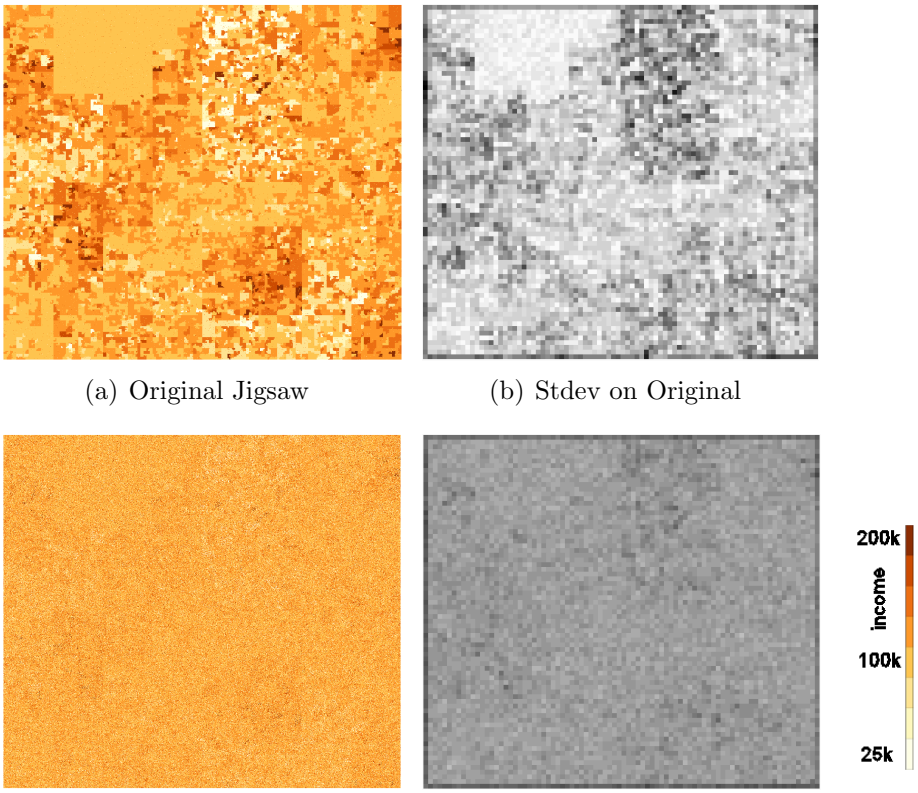
To show the usefulness of our approach we applied the Pixnostics technique to Jigsaw maps Pixel Bar Charts and Parallel Coordinates. The proposed experiments show how Pixnostics can steer the visual exploration process in an unsupervised manner in order to increase the efficiency of the exploration process and to actively support the analyst to reduce the effort of getting insight in the data.

11.1.1 Jigsaw Maps

Our first application example analyzes U.S. census data [oC], in particular median household income for the state of New York on block level. We generated visualizations using Jigsaw maps [Wat05], a pixel based technique based on space filling curves. The basic idea is to map the census data into the 2D plane in such a way that properties like locality and clusters in the data are preserved by using a space filling curve. To verify our proposed techniques, we generated a Jigsaw map from the New York state census median household income data on block level which should preserve the clusters in the data (clusters of areas with high/low income) and their spatial location, as shown in Figure 11.1(a). Figure 11.1(a) reveals a large cluster (Queens) with households that have similar high income (100k). Figure 11.1 (c) shows how this cluster disappears if we insert noise into the data.

We permutated the data points at different permutation rates. This should of course destroy or at least reduce the clustering / locality properties. Based on the permutation rate from 0 to 100 percent, we can now generate an expert ranking of the Jigsaw maps. The original map is the plot with the best clustering properties, if we permutate all data points (100 percent) we get the worst plot, since all clustering properties are destroyed.

Now we applied our automated analysis function based on the clustering task



(a) Original Jigsaw (b) Stdev on Original

(c) 90% Permutation (d) Stdev Permutation

Figure 11.1: Visualization of Information content: Jigsaw maps generated from NY median household income data, darker colors correspond to higher income. Gray levels show the information content of image sections, darker gray levels correspond to higher information content. The permutated image has significant higher information content, which indicates bad clustering properties

that ranks the underlying figures according to their clustering properties. The original Jigsaw should of course be the image with the highest ranking since it provides the best clustering properties. The more permutation there are in the image, the lower the relevance of the image, i.e. its rank, should be.

Thus, we consider the permutation rate as our input parameter and want to find input parameters which produce visualizations with good clustering properties. Figure 11.2 shows the experimental results. The upper figure shows the unordered input data set, a set of Jigsaw images. It is easy to visually identify images with good clustering properties, i.e. images having a cluster with low income in the upper left corner surrounded by high income areas.

In the lower figure, the result after the analysis step is shown. It is easy to see that figures with good clustering properties are ranked first, while Images containing more noise have lower relevance. To determine the ranking either the Entropy or the standard deviation of the pixel gray levels in combination with the regular grid cell hierarchies is employed. Figure 11.1 shows the rationale for the ranking. An image that provides a good clustering has areas with very low Entropy or low Stdev of grey levels while the complete figure does not necessarily have low Entropy. Therefore we start with a fine grid and determine the information content of each cell like shown in Figure 11.1(b). Then we hierarchically compare neighboring grid cells similar to Single Linkage Clustering and try to extend clusters. Finally we aggregate the information content of the clusters using our BB-score and order the images according to their information content values.

In Figure 11.3 the result of the evaluation is shown. We compared the optimal ranking with our computed rankings for a different number of images and depending on the size of our grid cells. As a measure for the quality of our computed rankings, we measured the normalized total mismatches compared to the optimal ranking:

$$RankingError = \frac{\sum_{i=1}^{n} mismatch(R_{1i}, R_{2i})}{n}$$

The function $mismatch(R_{1i}, R_{2i})$ returns 1 if images of rankings R_1 and R_2 at position i are different, 0 otherwise. n is the number of images. Note that in our experiments we produced equidistant permutated images, i.e. if we used six images for comparison we constructed Jigsaw images with $0, 20, 40, 60, 80$ percent permutation. The images should of course be ranked in that order, therefore we count the mismatches in comparison to our computed ranking. As shown in Figure 11.3, we achieved very low ranking errors given sufficient grid resolution (given in percent of the available image length). With increasing grid size, it is harder to distinguish local structures, therefore the ranking error increases. The same observation can be made for increasing number of images, since then the distances between images decrease. For 50 images for example, the difference in permutation rates between adjacent images is only 2 percent, which is of course hard to detect automatically. However, given sufficient grid resolution, we achieved

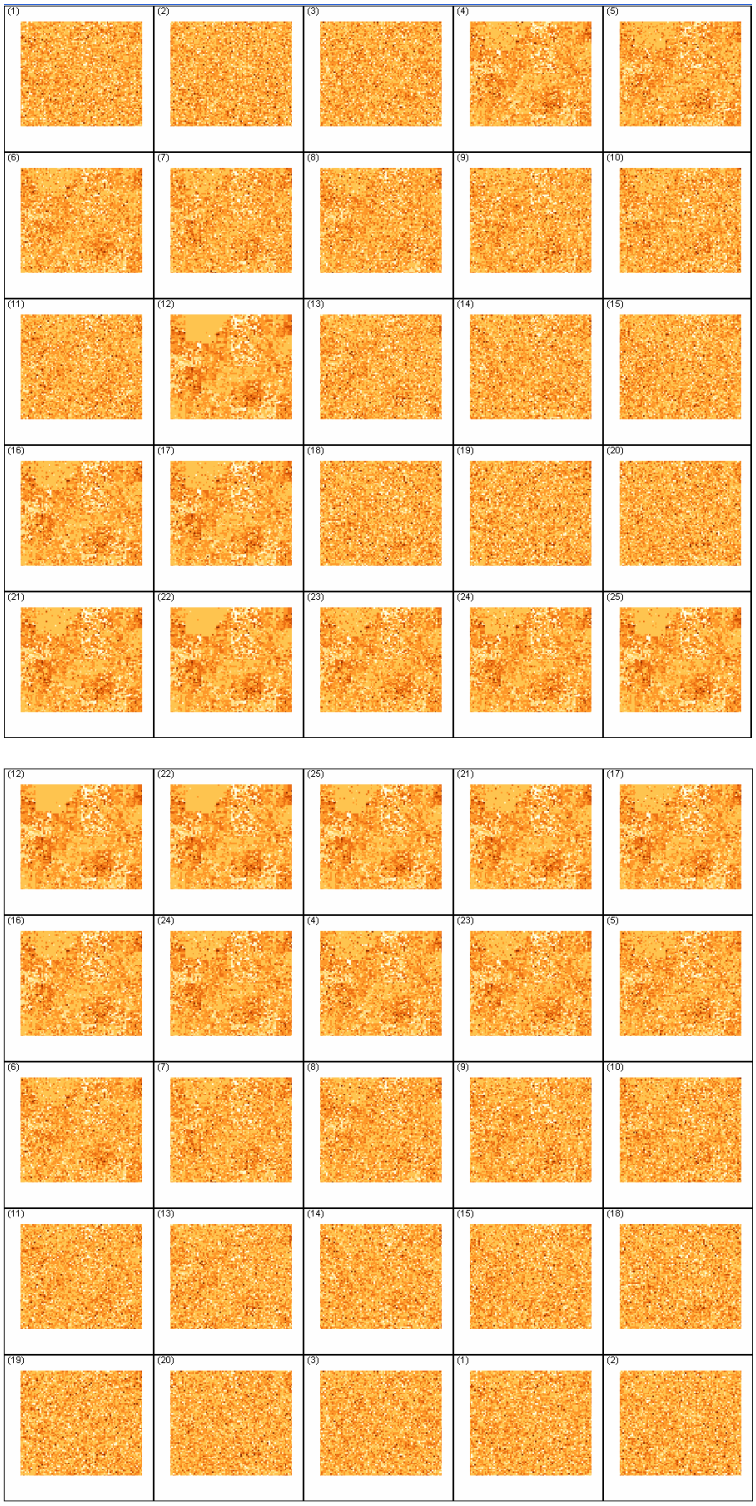


Figure 11.2: Census data Jigsaw unsorted (top) und sorted (line by line starting at top left corner with most relevant, id of each image reflects the position) by ranking function based on Entropy and clustering task (bottom, id of each image reflects position before reordering)

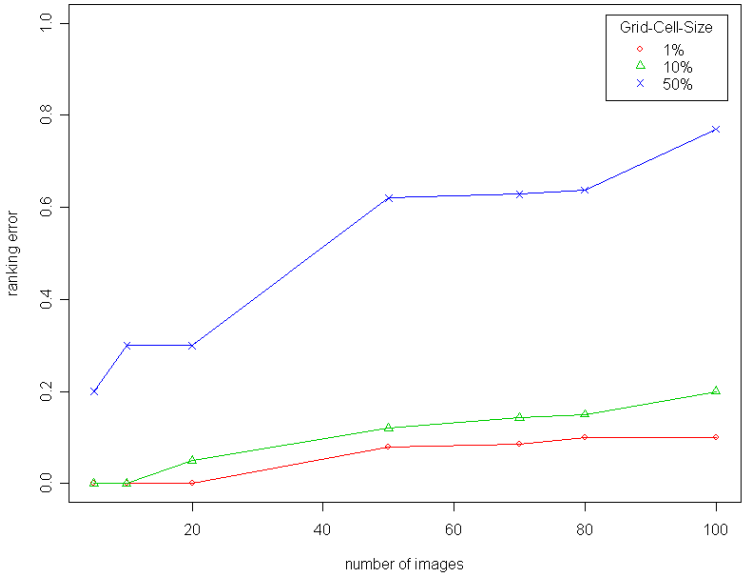


Figure 11.3: Effectiveness depends on grid resolution: Finer grid resolutions lead to more accurate rankings. The chart shows the normalized ranking error based on the matches between the computed and the optimal ranking. The error grows with increasing size of grid cells and increasing number of images. If the number of images increases, the differences between them decrease, which makes automated ordering harder.

very low ranking errors even for 50 or 100 images.

11.1.2 Pixel Bar Charts

Pixel Bar Charts [KHDH02] are derived from regular bar charts. The basic idea of a pixel bar chart is to present the data values directly instead of aggregating them into a few data values by representing each data item by a single pixel in the bar chart.

The detailed information of one attribute of each data item is encoded into the pixel color and can be accessed and displayed as needed. To arrange the pixels within the bars, one attribute is used to separate the data into bars and then two additional attributes are used to impose an ordering within the bars along the x and y axes. The pixel bar chart can be seen as a combination of traditional bar charts and x-y diagrams.

Although Pixel Bar Charts have been successfully applied to explore large data sets (see [KSS+03]), the analyst has to choose selections of attributes for separation, ordering, and color coding of data points from the underlying data manually, according to his analysis tasks.

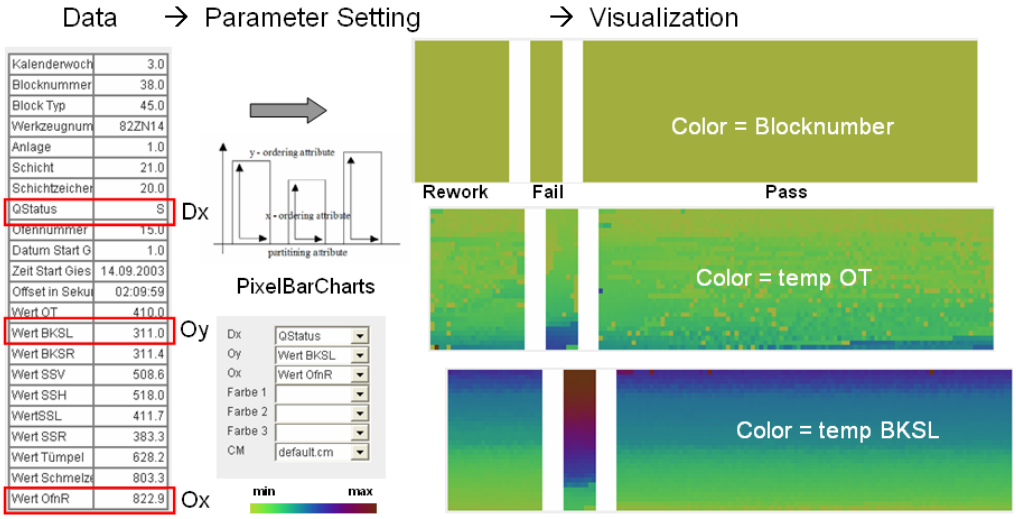
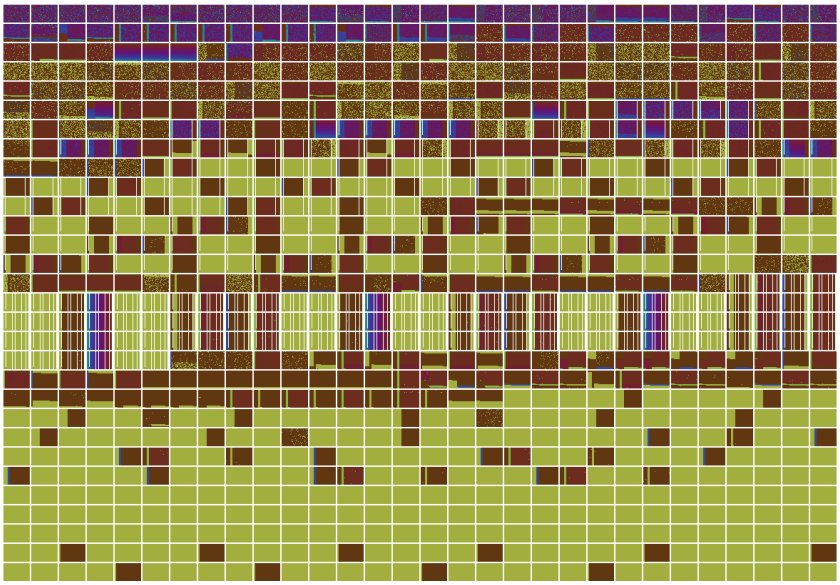


Figure 11.4: Possible mappings of attributes to Pixel Bar Chart parameters: Depending on the selected mapping, the visualization provides more or less insight.

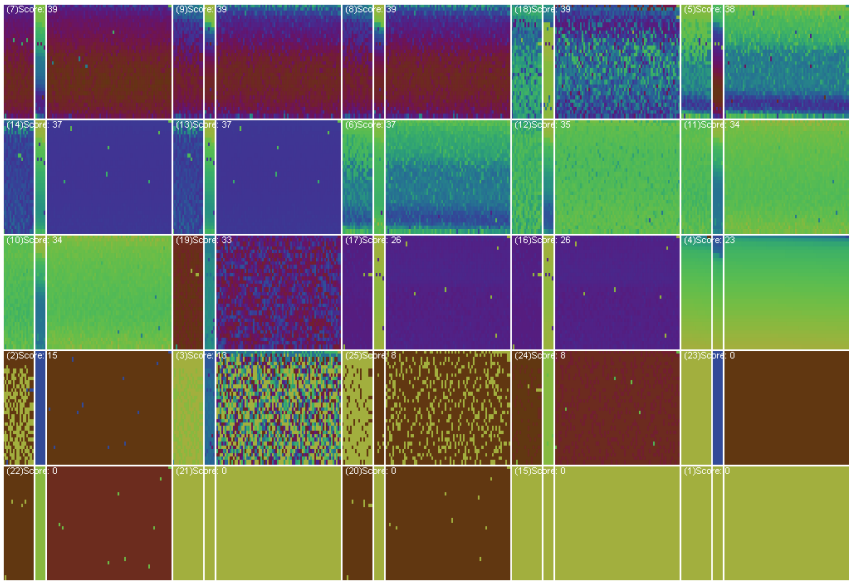
On one hand, this is time consuming since he has to try multiple parameter settings even those that do not reveal interesting patterns, on the other hand he may overlook interesting patterns since only a few attribute combinations can be analyzed manually. To face this problem we applied Pixnostics to Pixel Bar Charts, to guide the analyst through the exploration process and indicate potentially interesting parameter settings.

We applied our approach to a production data example. The data set contains data from an assembly line, in particular measurements from different stages of the assembly line like cast temperatures, part measurements and the quality of the output. All in all the data set contains 22 attributes. The output parts are classified into three groups: accept, reject, rework. Parts that are grouped "accept" pass the quality check, "rework" parts need to be reworked to pass the quality check and "reject" parts must be rejected because of defects. The analysis of such data is an important task in order to reduce rejected parts and thus to reduce production cost. Using Pixel Bar Charts, the analyst faces the problem of how to find groups of attributes that may influence the quality of the output. There are 175, 560 possible combinations to choose four attributes as visual variable from 22, even if the target attribute "Quality" is fixed for separation of the bars, there are still over 9000 combinations for selection of three attributes out of 22, which cannot by checked manually.

Therefore we first apply our automated analysis tools to determine attributes that most influence the "Quality" variable, using correlation and classification



(a) Ranking of images generated from promising attribute selections. Each box shows a thumbnail of a pixel bar chart based on a certain attribute selection, ordered by information content (desc).



(b) Ranking after image analysis: The top 25 images are shown with a fixed target attribute as splitting attribute for the bars.

Figure 11.5: Pixel Bar Chart showing top 25 results after image analysis using entropy measure. It is easy to see that the bar in the middle ("reject" parts) show significant differences in comparison to the 2 other bars.

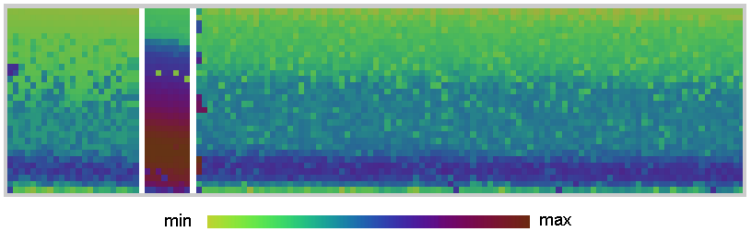


Figure 11.6: Pixel Bar Chart constructed from the output result of Pixnostics (from the chart in the upper right corner in Figure11.5 (b)).

analysis. Of course we can additionally prune all parameter settings where "Quality" is not involved, since these will not bring us any new insight.

From the remaining combinations, we either generate images and order them by information content directly, like shown in Figure 11.5 (a) or we can filter Pixel Bar Charts where the target attribute is fixed as the splitting attribute and select the most valuable ones from them, as shown in Figure 11.5 (b).

The figure shows the 25 most relevant Pixel Bar Charts having "Quality" as the splitting attribute. Note that the left bar shows parts that are "rework", the middle bar shows "reject" parts, and the right bar shows "accept" parts. It is easy to see that the "reject" bars look significantly different than the rest. The analyst may now select a single image from the provided images, and a Pixel Bar Charts is created from this selection as shown in Figure 11.6. The analyst can now easily discover relevant patterns by visual based root cause analysis. In the image the color shows the temperature of a particular casting mold and the ordering in y direction shows the duration of the part at this stage. It is easy to see that the casting mold had a significantly higher temperature for "reject"parts, which is a potentially reason for a damaged part.

In this manner the analyst may investigate further high ranked images, which provides a more efficient way of visual analysis than manual feature selection.

11.1.3 Parallel Coordinates

Parallel Coordinates [ID90] are commonly used for the analysis of high dimensional data sets. The basic idea is to represent the dimensions of the data as parallel axis in the parallel coordinate plot, and to connect the (parallel) coordinates of each data point by lines. Using this technique, certain relationships between variables (dimensions) can be deduced from the appearance of the plot. However, since in Parallel Coordinates the dimensions have to be arranged in some one dimensional order on the screen, the selected arrangement of dimensions can have a major impact on the expressiveness of the visualization because relationships among adjacent dimensions are easier to detect than relations among dimensions positioned far from each other [YWRH03].

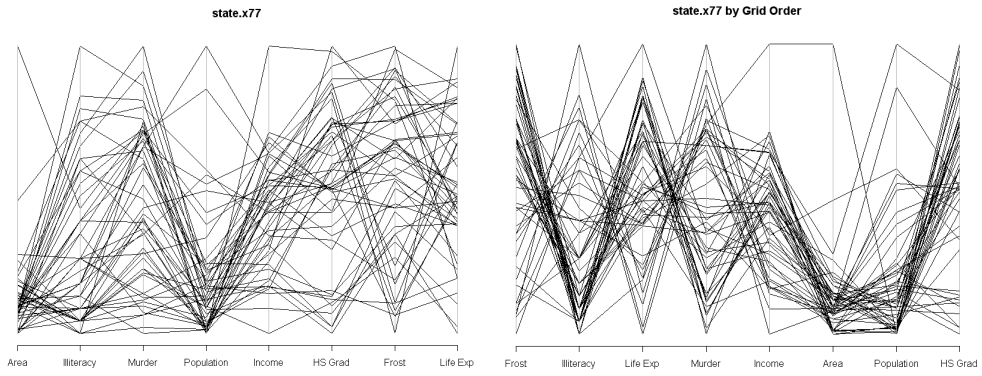
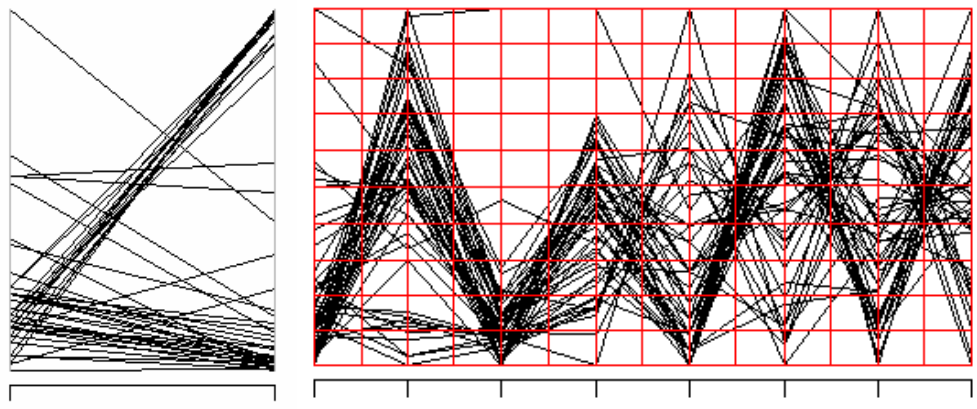


Figure 11.7: *Effect of axis ordering:* The figure shows Parallel Coordinate Plots for the state.x77 dataset. This dataset has eight dimensions $d_1, ..., d_8$. Without effective ordering functionality it may be hard to detect relevant patterns like shown in the left figure. In the right figure we applied our approach for dimension ordering $(d_7, d_3, d_4, d_5, d_2, d_8, d_1, d_6)$, which reveals significant correlations and clusters.

As mentioned in Section 4.3 a number of dimension management techniques have been proposed to arrange the dimension axis in Parallel Coordinates appropriately [War94, YWRH03]. A common way to produce an arrangement of axis is to compute the correlation matrix for the dimensions of the data, and then to map the dimensions to parallel axis so that the correlation between adjacent axis is maximized [ABK98].

Although this approach can help to increase the expressiveness of Parallel Coordinate plots, it is often not sufficient, since global correlation measures do not take special local properties of the data into account. An example is shown in Figure 11.8 (a). These two axis have a very low correlation (Pearson Correlation $= 0.15$), and it would not be likely that these axis would be arranged next to each other by a dimension ordering approach based on correlation measures. However, they reveal interesting local correlations and clustering properties of their data values, which would of course be a reason to locate them next to each other. Our grid based approach shown on the right in Figure 11.8 (b) in contrast, allows to take even local phenomena into account. Note that in order to analyze Parallel Coordinate Plots, we aligned the grid according to the axis, that means we investigated the properties of the lines between the axis, not the axis itself. In many application scenarios other parameters may be involved, like color as an additional coding technique, which are difficult to consider in analytical dimension management. This is another reason why our approach can produce better orderings than using global correlation measures, since depending on our employed relevance function we can take such special parameters into account.



(a) (b)

Figure 11.8: *PC plot analysis* – Since Global Ordering Measures are limited in finding local phenomena, we used our grid based approach to identify good axis arrangements. In the left image the axis reveal a very low correlation (Person Correlation of 0.15), although the analyst can observe some interesting local patterns. The grid based analysis (right) allows the identification of such local patterns.

To take the special properties of Parallel Coordinate Plots into account, we adapted the approach introduced in Section 10.2.2 to this technique. First of all, the employed grid was adapted to the distances of the parallel axis, since in our analysis we focused on the properties of the lines between the axis, and not on the axis itself, shown in Figure 11.8 (b). Besides our Entropy measure, we employed a number of other relevance functions to generate rankings for Parallel Coordinate Plots. Since Parallel Coordinates are not space-filling, in the sense that usually not all pixels of the plot are covered with lines, these ranking functions are based on the ratio of background and foreground pixels (line pixels) of each grid cell. This gives us an idea how many lines cross each grid cell. Based on these values we can then get a measure for the distribution of the lines by computing the standard deviation of all grid values. Higher standard deviation would indicate more non-uniformly distributed data, which corresponds to more structure. An example showing the analysis of the state.x77 data set is shown in Figure 11.9.

To take also color distributions into account, we proposed a second function that computes the global ranking score based on the standard deviation of the color distribution as shown in 11.10. Based on the user task, these functions can be weighted to emphasize structure in the line layout, or structure in the color layout or both.

The evaluation of our ranking results is of course difficult since there does not yet exist an empirically verified theory of human perception capabilities that

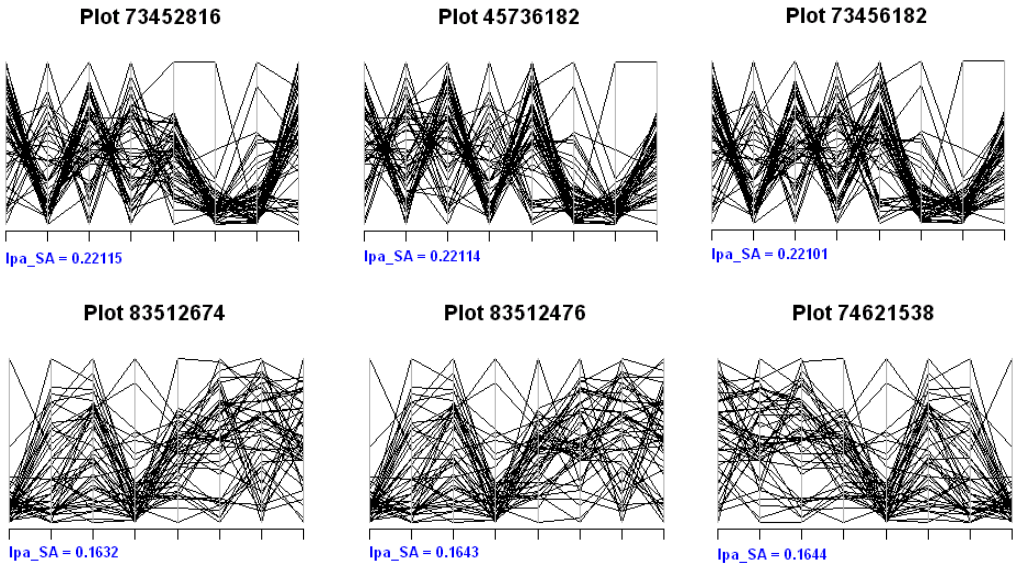


Figure 11.9: The three best (top) and worst (bottom) ranked plots are shown after application of our approach to the state.x77 data set.

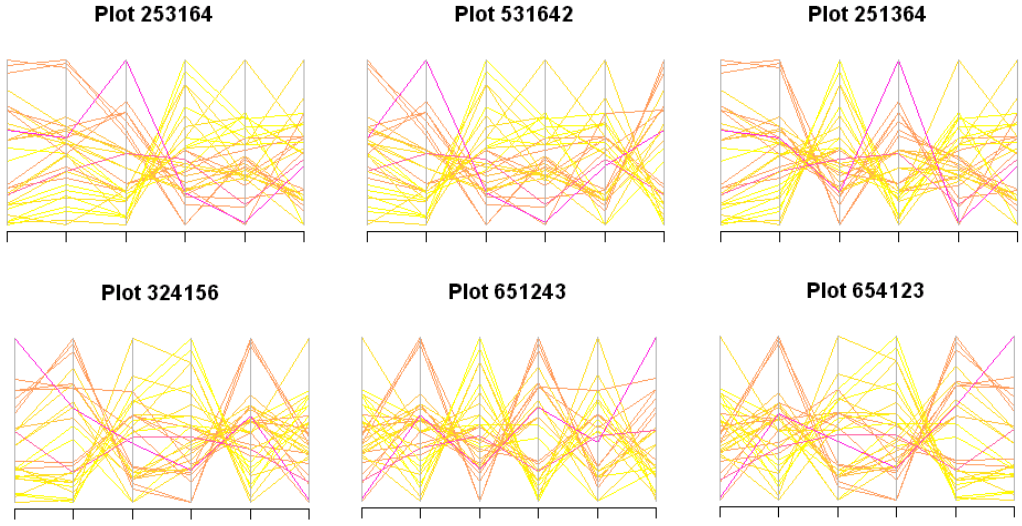
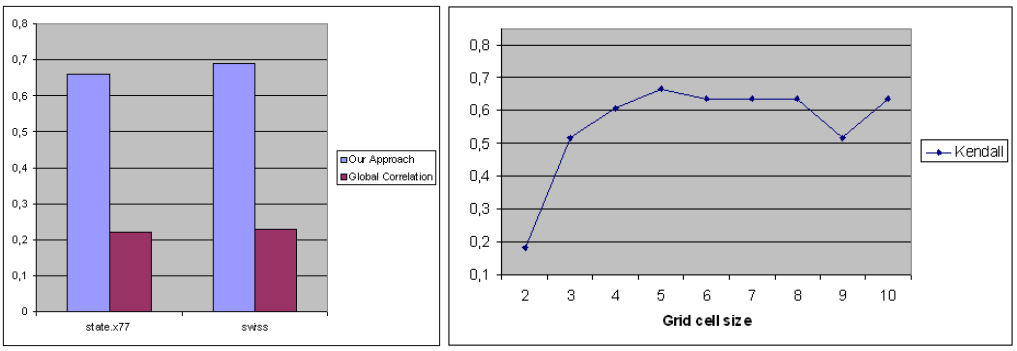


Figure 11.10: The three best (top) and worst (bottom) ranked plots are shown after application of our approach to the mtcars data set. The relevance function now also takes the deviation of color values into account.



(a) (b)

Figure 11.11: Correlation (Kendall) between expert ranking and our approach (blue) and Global Correlation ordering (red) show that our approach produces significant better results for 2 sample data sets. The right image shows the impact of the grid cell size (in percent of the total space between two axis) in our approach on the correlation result.

can be used to prove theorems about the effectiveness of visualizations [Mac99] as explained in Chapter 4.

Therefore, we generated a set of Parallel Coordinate plots by permutations of the dimension axis for the state.x77 and swiss data set. Both data sets are contained in the *R* statistical package. We classified these plots manually according to their structural properties. Plots that contained local patterns like negative/positive correlations, clusters, or outliers got a higher ranking, than plots that contain less obvious structures. Finally we compared the expert ranking with our computed ranking and the ranking produced by ordering the axis according to their pairwise global Pearson correlation using Kendalls Tau measure [Ken48].

Figure 11.11 shows the result of this evaluation. The left figure shows that the ranking produced by our approach has a significant higher correlation to the expert rating than the ranking produced by correlation ordering, for the state.x77 as well as for the swiss data set. The reason is that our grid approach is able to take local phenomenon into account and rank such plot higher, these plots are of course also interesting for analysts and domain experts. The right figure shows again, that the ranking might be influenced by the size of the grid cells.

All in all, we have shown that our technique can be combined with Parallel Coordinates and was able to identify relevant plots by analysis of a set of candidate plots. We have shown that our approach produced better results than the commonly used correlation dimension ordering for two example data sets. This leads us to the conclusion that a combination of analytical and image based approaches has the potential to improve existing dimension management techniques, in order to speed up the exploration of interesting patterns using Parallel Coordinates.

11.2 Conclusion

Integrating automated analysis methods into the visual exploration process is an important challenge in the age of massive data sets and has been recognised as a major research area in the context of Visual Analytics. Therefore the aim of the research proposed is to show how unsupervised analysis functions can help to speed up the visual exploration process by supporting the user with task driven relevance functions for a more effective data analysis. The basic idea of the proposed method is to measure the relevance of the resulting visualization with respect to input parameters and user tasks and to provide a ranking of potentially useful initial visualizations and initial parameter settings. This helps the analyst to focus on relevant parts of the data and relevant parameter settings and leads to an improved exploration process. We provided a formal definition of our work and showed how the technique can be used with Jigsaw maps, Pixel Bar Charts and Parallel Coordinates.

Future work will focus the improvement of the proposed technique and its application to a variety of visualization techniques, not only pixel based but also geometric and iconic techniques. Furthermore we will include the user in the analysis process in form of relevance feedback to dynamically adapt our relevance functions. One big issue for future work is also the evaluation of the proposed methods using real user studies that evaluate the quality of our relevance functions.

Part V

Conclusions

Chapter 12

Summary and Future Directions

12.1 Summary of Contributions

The rapidly increasing amount of data stored in todays database environments requires efficient and effective methods to make full use of the collected data, i.e. to extract interesting and potentially useful patterns from the flood of data. More than ever before, organizations in commercial, government, university and research sectors are tasked with making sense of huge amounts of data. But due to the complexity and volume of todays data sets, extracting the valuable information hidden within data is a difficult task. New methods are needed to allow the analyst to examine these massive, multi-dimensional information sources and make effective decisions. To face this challenge, the field of Visual Analytics aims at the integration of data mining technology and information visualization and thus combines two powerful information processing systems the human mind and the modern computer. This thesis contributes to the field of Visual Analytics by providing novel methods that take the special properties of hierarchical-, time-related, and geo-related datasets into account. Application examples from a number of scenarios are presented that show how these techniques are successfully applied in real world business scenarios. Furthermore, a method that combines automated analysis and image analysis techniques in order to support the user in creating insightful visualizations is presented and evaluated. This approach makes maximum use of automated techniques and human visual abilities, which is a key issue in exploring large data sets.

In the following, we provide a detailed summary of these contributions.

12.1.1 Introduction

The preliminaries in Part I illustrate the topic and the background of this work. After a very general introduction to modern data analysis, we give an overview on Visual Data Exploration techniques. A classification of visualization techniques is presented and some classical visualization examples are provided. Furthermore a Visual Data Exploration methodology is introduced. The end of this chapter motivates the need for Visual Analytics and shows how this research field has evolved from classical research fields like KDD and Visual Data Exploration.

12.1.2 Visual Analytics

Part II introduces the research field of Visual Analytics in detail. The main concepts are introduced and the Scope of Visual Analytics is described. We provide a formal description of the Visual Analytics process as well as a process model. We also identify major research challenges for Visual Analytics, which are in the focus of this thesis.

- **Visual Scalability**
  The analysis of large data sets reveals two major tasks. The first one is the question, how visualizations for massive data sets can be constructed without loosing important information even if the number of data points is to large to visualize each single data point at full detail. The second important task is to find techniques to efficiently navigate and query such massive data sets.

- **Analyzing heterogeneous data sources**
  Analysis techniques have to take the special data characteristics along each dimension (geographic, temporal, multivariate spaces) into account, and therefore powerful visual metaphors are needed. Furthermore the integrated analysis along all dimensions holds great potential to provide valuable and previously unknown information that can identify complex phenomena, especially multivariate space-time patterns. Therefore Visual Analytics approaches are needed that are able to explore multivariate spatio-temporal patterns, and present them in an intuitive form to support human interpretation and decision making.

- **Automated support for Visual Representations**
  Typically we have to deal with large parameter spaces when using visualization techniques to explore large data sets. These parameters control the visual encoding of the data, including the selection of attributes from the input data, the selection of the color scale, algorithm parameters, the selection of visual variables and so on. Finding parameter settings that lead to insightful visualizations, is however a challenging task. In Exploratory Data Analysis a good or the optimal parameter setting for a given task is

often not clear in advance, which means that the analyst has to try multiple parameter settings in order to generate valuable visualizations. Since such selections can hardly be done manually, the integration of automated methods to support the analyst has been recognized as an important research problem in the context of Visual Analytics in business applications.

12.1.3 Visual Business Analytics: Techniques & Applications

Part III presents novel Visual Analytics methods that focus on the identified research challenges. Techniques for the analysis of heterogeneous data spaces are introduced that take scalability issues into account. The introduced techniques are evaluated based on business applications from many different applications scenarios. In particular, we present original and new techniques for the analysis of temporal-, spatial-, and hierarchical data sets and show how an integrated analysis of space-time-patterns can be performed. Our application examples include the analysis of financial data, business process data, sales data and quality management data.

- **Visual Business Analytics of temporal data**
  Time related data sets are ubiquitous and appear in many application domains in business and science, including finance (stock market data, credit card transactional data), communication (telephone data, signal processing data, network monitoring), or entertainment (music, video). The analysis of time related data sets is a key issue to get insight into the data, to identify patterns, trends or correlations. Since challenges arise from the volume as well as from the complexity of these data sets, novel approaches for Visual analytics in time related data sets are needed. We provide the *Circle View* and the *VisImpact* technique. *CirlceView* allows an easy visual comparison of multidimensional time related data sets, supports user interaction and integrated data mining techniques to increase scalability. The technique is evaluated based on financial- and industrial data. The *VisImpact* technique was developed to analyze large business process flows, by integrating data mining techniques to identify important impact factors in large business process flows and combine them with powerful visualization techniques to visually analyze their relations and their development over time.

- **Visual Business Analytics of hierarchical data**
  In real-world business applications, data sets have, besides a time or space dimension, often a hierarchical context. Network analysis, business process analysis, or financial analysis for example require the parallel analysis of a large number of time series, which show intrinsic hierarchical relationships. To visually analyze such data sets, novel Visual Analytics techniques are needed. In this thesis, we therefore provide techniques that allow such an

analysis. In particular, we provide techniques for visually exploring frequent patterns and large citation networks, and provide the *VisMap* Framework that allows the analysis of hierarchical time related data sets by combining stacket displays and alignment techniques.

- **Visual Business Analytics of geo-spatial data**
  Many existing and emergent business applications collect and reference data automatically by their geo-spatial location. Even simple transactions of every day life such as paying by credit card or using the telephone are typically recorded by company computers. For each credit card purchase transaction many parameters are usually recorded such as name and price of the bought product items, and both the place of purchase and the purchaser. Telephone call records include the addresses and sometimes cell phone zones as well as geo-coordinates. Census tables are another well-known example that besides different data values also include addresses and other geo-related information. So when analyzing business data, the geographical dimension has to be taken into account. Therefore, we provided a number of techniques for visually analyzing geo-related data and to reveal the distribution of geo-related phenomena. Furthermore we presented an approach that is able to detect multivariate space-time patterns by combining multivariate analysis and geo-spatial visualization techniques.

In summary, Part III introduces many novel techniques that contribute to the field of Visual Analytics. These techniques take the temporal-, hierarchical as well as geo-spatial data aspects into account. Application examples from many different domains show the benefit of the proposed techniques.

12.1.4 Relevance Driven Visual Analytics

Based on our observations and techniques proposed in Part III, we focus in Part IV on a research challenge that inevitably appears when using visualization techniques for analyzing large and complex data sets. Since Visual Analytics integrates automated methods and Visualization techniques, techniques are getting more complex, forcing the user to set up many different parameters to adjust the mapping of attributes to visual variables on the display space. In classical data exploration, playing with parameters to find a promising parameter setting is an important part of the exploration process, but with the increasing number and diversity of the parameters it becomes more and more difficult to determine a good parameter setup, which is vital for insightful visualizations. Therefore we introduce an approach for automated user support in constructing insightful visualization based on a combined analytical and visual analysis of the data- and visualization space. Our technique supports the user in finding meaningful parameter settings based on the current task at hand. We provide a formal definition of our approach and show application examples that indicate the value of the proposed technique.

12.2 Future Work

The end of this thesis discusses possible future research directions of the proposed approaches. Since in the future we will have to deal with even larger data sets, there will be of course new challenges for Visual Analytics. For the Visual Analytics techniques introduced in this thesis we see the following opportunities for future research:

Analytics of temporal data

Since temporal data sets are typically large in nature, it will be very important to filter important information form these large data streams. Therefore it will be very important to generate initial compact representations of such data sets that emphasize relevant information. We integrated such relevance functions for financial data in the CircleView framework, but in the future it would be useful to provide a set of relevance functions that compute, depending on the current task at hand, compact multi-resolution representations for more application domains. For *VisImpact* which is especially designed to analyze workflow process data, it will in the future be very important to tightly integrate the underlying workflow model and the workflow instance data, which can be seen as particular paths through the workflow, in order to automatically optimize the process flows.

Visual Analytics of hierarchical data

Many data sets contain intrinsic hierarchical relationships, so the development of Visual Analytics techniques for hierarchical data will stay an important issue for future research. In the future, our circular layouts could be integrated in front-ends for certain online libraries like the DBLP to visually explore co-authorship in such libraries. Future work for *VisMap* could include its generalization from analyzing time series data to any categorical data. Multi-resolution and similarity layout algorithms may be integrated to place even long time series in one single display. *VisMap* may also be applied to other applications such as supply chain and capacity planning. Ongoing research also focuses on Intelligent queries in *VisMap*, i.e. after computing an initial aligned hierarchical layout the user may select interesting regions and automated data mining algorithms are then used to extract interesting patterns or to run similarity queries based on these subsets.

Visual Analytics of geo-spatial data

A great challenge in geo-spatial visualization is to provide methods that significantly improve the perception of activities , events, and links as they change in time over a geo-spatial context and that at the same time clearly show the broad

picture. Our provided approaches contributed in this direction by combining automated data analysis techniques with visualization techniques to track geo-spatial patterns over time and to project their future status. We mainly focused on cluster analysis to identify geo patterns. In the future other data mining techniques for multivariate analysis may be integrated. Furthermore, the visual representation may be improved by novel visual metaphors.

Automated support for construction of visualization

Integrating automated analysis methods into the Visual Exploration process is an important challenge in the age of massive data sets and has been recognized as a major research area in the context of Visual Analytics. When using sophisticated Visualization techniques to analyze these complex data sets, there is of course a need for supporting the user in constructing effective and expressive visualizations. Our proposed methods measure the relevance of the potential visualizations with respect to input parameters and user tasks and provide a ranking of potentially useful initial visualizations and initial parameter settings. This helps the analyst to focus on relevant parts of the data and relevant parameter settings and leads to an improved exploration process. Our experiments showed the potential of these methods on real-world examples. Future work will focus on the improvement of the proposed technique and its application to a variety of Visualization techniques, not only pixel based but also geometric and iconic techniques. Furthermore the user may be involved in the analysis process in form of relevance feedback to dynamically adapt our relevance functions. A big issue for future work is also the application of a broader field of analysis functions and methods to extract relevant features from the data and the corresponding data space in order to automatically construct visualizations from potentially interesting patterns in the data.

Bibliography

[AA05] Gennady L. Andrienko and Natalia V. Andrienko. Visual exploration of the spatial distribution of temporal behaviors. In *9th International Conference on Information Visualisation, IV 2005, 6-8 July 2005, London, UK*, pages 799–806, 2005.

[ABK98] Mihael Ankerst, Stefan Berchtold, and Daniel A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *INFOVIS '98: Proceedings of the 1998 IEEE Symposium on Information Visualization*, page 52. IEEE, 1998.

[ADLP95] V. Anupam, S. Dar, T. Leibfried, and E. Petajan. Dataspace: 3-d visualizations of large databases. In *INFOVIS '95: Proceedings of the 1995 IEEE Symposium on Information Visualization*, page 82, Washington, DC, USA, 1995. IEEE Computer Society.

[AEK00] Mihael Ankerst, Martin Ester, and Hans-Peter Kriegel. Towards an effective cooperation of the user and the computer for classification. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 179–188, New York, NY, USA, 2000. ACM Press.

[AFS93] R. Agrawal, C. Faloutsos, and A. N. Swami. Efficient similarity search in sequence databases. In *FODO 1993*, pages 69–84, 1993.

[AKK95] M. Ankerst, D. A. Keim, and H.-P. Kriegel. Recursive pattern: A technique for visualizing very large amounts of data. In *Proc. Visualization '95, Atlanta, GA*, pages 279–286, 1995.

[AKK96] M. Ankerst, D. A. Keim, and H.-P. Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *Visualization '96, Hot Topic Session, San Francisco, CA*, 1996.

[ALSS95] R. Agrawal, K. I. Ling, H. S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *VLDB 1995*, pages 490–501, 1995.

[Ber67] J. Bertin. *Semiology of graphics*. University of Wisconsin Press, Madison, Wisconsin, 1967.

[BGG97] C. Sidney Burrus, R. A. Gopinath, and H. Guo. *Introduction to Wavelets and Wavelet Transforms, A Primer*. Prentice Hall, 1997.

[BHvW00] M. Bruls, K. Huizing, and J.J. van Wijk. Squarified treemaps. In *In Proc. of Joint Eurographics and IEEE TCVG Symp. on Visualization (TCVG 2000)*, pages 33–42, 2000.

[BKK97] S. Berchtold, D. A. Keim, and H.-P. Kriegel. Using extended feature objects for partial similarity retrieval. *VLDB Journal*, 6(4):333–348, November 1997.

[BKK02] Christian Boehm, Florian Krebs, and Hans-Peter Kriegel. Optimal dimension order: A generic technique for the similarity join. In *DaWaK 2000: Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, pages 135–149, London, UK, 2002. Springer-Verlag.

[BM04] L. Berry and T. Munzner. Binx: dynamic exploration of time series datasets across aggregation levels. In *Poster Compendium to Proc. of the IEEE InfoVis Symposium '04*, pages 215–217. IEEE Computer Society, 2004.

[BN01] Todd Barlow and Padraic Neville. A comparison of 2-d visualizations of hierarchies. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 131, Washington, DC, USA, 2001. IEEE Computer Society.

[BSW02] Benjamin B. Bederson, Ben Shneiderman, and Martin Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Trans. Graph.*, 21(4):833–854, 2002.

[CAH87] Stuart K. Card and Jr. Austin Henderson. A multiple, virtual-workspace interface to support user task switching. In *CHI '87: Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*, pages 53–59, New York, NY, USA, 1987. ACM Press.

[Cas91] Stephen M. Casner. Task-analytic approach to the automated design of graphic presentations. *ACM Trans. Graph.*, 10(2):111–151, 1991.

[CB] Bill Cheswick and Hal Burch. Internet mapping project.

[CC01] M.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman and Hall, 2001.

[Cen02] History Central. www.multied.com/elections, Mar. 2002.

[Che73] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal Amer. Statistical Association*, 68:361–368, 1973.

[Chi00] E. H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the 2000 IEEE Symposium on Information Visualization*, pages 69–76. IEEE Press, 2000.

[Chu98] M.C. Chuah. Dynamic aggregation with circular visual designs. In *Proc. of the IEEE InfoVis Symposium '98*, pages 35–43. IEEE Computer Society, 1998.

[Cle93] William S. Cleveland. *Visualizing Data*. Hobart Press, Summit, New Jersey, U.S.A, 1st edition, 1993.

[CM84] W.C. Cleveland and M.E. McGill. Graphical perception: Theory, experimentation and application to the development of graphical methods. *J. Am. Stat. Assoc.*, 79(387):531–554, 1984.

[CM88] William C. Cleveland and Marylyn E. McGill. *Dynamic Graphics for Statistics*. CRC Press, Inc., Boca Raton, FL, USA, 1988.

[CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in Information Visualization Using Vision to Think*. Morgan Kaufmann, 1st edition, 1999.

[Com86] Rogers Commission. Report of the presidential commission on the space shuttle challenger accident, 1986.

[Cor05] Winter Corporation. 2005 topten program summary: Select findings from the topten programs, September 2005. http://www.wintercorp.com.

[CRM91] Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. The information visualizer, an information workspace. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 181–186, New York, NY, USA, 1991. ACM Press.

[Dav73] John C. Davis. *Statistics and Data Analysis in Geology*. John Wiley & Sons, Inc., New York, NY, USA, 1973.

[Den96] Borden D. Dent. *Cartography: Thematic Map Design, 4th Ed., Chapter 10*. William C. Brown, Dubuque, IA, 1996.

[Dun89] G.H Dunteman. *Principal Component Analysis*. Sage Publications, 1989.

[Eic99] S. G. Eick. Visualizing multi-dimensional data with advisor/2000. *Visualinsights*, 1999.

[EK02] S. Eick and A. Karr. Visual scalability. *In J. of Computational and Graphical Statistics*, 1(11):22–43, 2002.

[EM95] M.D. Esteban and D. Morales. A summary of entropy statistics. *Kybernetika*, 31(4):337–346, 1995.

[ESS92] Stephen G. Eick, Joseph L. Steffen, and Eric E. Sumner. Seesoft-a tool for visualizing line oriented software statistics. *IEEE Trans. Softw. Eng.*, 18(11):957–968, 1992.

[EW94] P. Eades and N. C. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4):379–403, 1994.

[FGW01] Usama Fayyad, Georges G. Grinstein, and Andreas Wierse. *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[FHS96] Usama M. Fayyad, David Haussler, and Paul E. Stolorz. KDD for science data analysis: Issues and examples. In *Knowledge Discovery and Data Mining*, pages 50–56, 1996.

[Fis76] J.C. Fischer. Homicide in Detroit: The role of firearms. *Criminology*, 14(1):387–400, 1976.

[FPSS96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. Knowledge discovery and data mining: Towards a unifying framework. In *Knowledge Discovery and Data Mining*, pages 82–88, 1996.

[FRM94] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proc. SIGMOD Conference 1994*, pages 419–429, 1994.

[FS97] Usama Fayyad and Paul Stolorz. Data mining and kdd: promise and challenges. *Future Gener. Comput. Syst.*, 13(2-3):99–115, 1997.

[Fur88] George Furnas. Generalised fisheye views. In *ACM SIGCHI '86 Conference on Human Factors in Computing Systems*. ACM, 1988.

[GCDT05] G. Grinstein, U. Cvek, M. Derthick, and M. Trutschl. IEEE InfoVis 2005 Contest –technology data in the us, 2005.

[GCML06] D. Guo, J. Chen, A. M. MacEachren, and K Liao. A visuaization system for space-time and multivariate patterns (vis-stamp). *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1461–1475, November–December 2006.

[GEC98] N. Gershon, S.G. Eick, and S. Card. Information Visualization. *ACM interactions*, 5(2):9–15, 1998.

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

[Gre05] Gary R. Greenfield. Computational aesthetics as a tool for creativity. In *C&C '05: Proceedings of the 5th conference on Creativity & cognition*, pages 232–235, New York, NY, USA, 2005. ACM Press.

[GZT93] Sabir Gusein-Zade and Vladimir Tikunov. A new technique for constructing continuous cartograms. *Cartography and Geographic Information Systems*, 20(3):66–85, 1993.

[GZT95] Sabir Gusein-Zade and Vladimir Tikunov. Map transformations. *Geography Review*, 9(1):19–23, 1995.

[Haa10] A. Haar. Zur theorie der orthogonalen funktionen-systeme. *Mathematische Annalen*, 69:331–371, 1910.

[HBW05] Donald H. House, Alethea Bair, and Colin Ware. On the optimization of visualizations of complex phenomena. In *IEEE Visualization*, page 12, Minneapolis, MN, 2005.

[HDH99] M. C. Hao, U. Dayal, and M. Hsu. A java-based visual mining infrastructure and application. In *Proc. Information Visualization, 1999, San Francisco, CA*, 1999.

[HDY99] J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. In *Proc.Int. Conf. Data Engineering (ICDE'99)*, 1999.

[HK98] A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. *KDD 1998*, pages 58–65, 1998.

[HK99] A. Hinneburg and D. A. Keim. Clustering techniques for large data sets From the past to the future. In *KDD '99: Tutorial notes of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 141–181, New York, NY, USA, 1999. ACM Press.

[HK06] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[HM01] David J. Hand and Heikki Mannila. *Principles of Data Mining.* Bradford Book, 2001.

[HPY00] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD international conference*, pages 1 – 12, 2000.

[HS04] Harry Hochheiser and Ben Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.

[ID90] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proc. Visualization 90, San Francisco, CA*, pages 361–370, 1990.

[Inm96] W.H. Inmon, editor. *Building the Data Warehouse. 2nd edition.* John Wiley & Sons, New York, 1996.

[JS91a] B. Johnson and B. Shneiderman. Treemaps: A space-filling approach to the visualization of hierarchical information. In *Proc. Visualization '91 Conf*, pages 284–291, 1991.

[JS91b] Brian Johnson and Ben Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *VIS '91: Proceedings of the 2nd conference on Visualization '91*, pages 284–291, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[KAS04] Daniel A. Keim, Mihael Ankerst, and Mike Sips. *Visualizaiton Handbook by Johnson and Hanson*, chapter Visual Data Mining Techniques, pages 813–825. Elsevier Science Publishing, 2004.

[KD01] M. Kaufmann and D.Wagner. *Drawing Graphs: Methods and Models.* Springer Verlag Berlin, 2001.

[Kei00] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 6(1):59–78, January–March 2000.

[Kei01] D. A. Keim. Visual exploration of large data sets. *Communications of the ACM (CACM)*, 44(8):38–44, 2001.

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

[Ken48] M. Kendall, editor. *Rank Correlation Methods.* Charles Griffin Company Lmd, 1948.

[KG90] M. Kendall and J.D. Gibbons. *Rank correlation methods.* Oxford University Press, 1990. 5th edition.

[KH98a] Daniel A. Keim and A. Herrmann. The gridfit algorithm: An efficient and effective approach to visualizing large amounts of spatial data. *IEEE Visualization, Research Triangle Park, NC*, pages 181–188, 1998.

[KH98b] Christopher J. Kocmoud and Donald H. House. Continuous cartogram construction. *Proceedings IEEE Visualization*, pages 197–204, 1998.

[KHDH02] Daniel A. Keim, Ming C Hao, Umesh Dayal, and Meichun Hsu. Pixel bar charts: a visualization technique for very large multi-attribute data sets. *Information Visualization*, 1(1):20–34, 2002.

[KK93] Peter R. Keller and Mary M. Keller. *Visual Cues - Practical Data Visualization.* IEEE Press, 1st edition, 1993.

[KK94] D. A. Keim and H.-P. Kriegel. Visdb: Database exploration using multidimensional visualization. *Computer Graphics & Applications*, 6:40–49, Sept. 1994.

[KKA95] D. A. Keim, H.-P. Kriegel, and M. Ankerst. Recursive pattern: A technique for visualizing very large amounts of data. In *Proc. Visualization 95, Atlanta, GA*, pages 279–286, 1995.

[KKN99] D. Keim, E. Koutsofios, and S. C. North. Visual exploration of large telecommunication data sets. In *Proc. Workshop on User Interfaces In Data Intensive Systems (Invited Talk), Edinburgh, UK*, pages 12–20, 1999.

[KLF05] E. Keogh, J. Lin, and A. Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *The Fifth IEEE International Conference on Data Mining.*, 2005.

[KLO04] Markus Koskela, Jorma Laaksonen, and Erkki Oja. Entropy-based measures for clustering and som topology preservation applied to content-based image indexing and retrieval. *17th International Conference on Pattern Recognition (ICPR'04)*, 2:1005–1009, 2004.

[KMN+02] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu. An efficient k-means clustering algorithm: analysis and implementation, 2002.

[KMP+05] D. A. Keim, F. Mansmann, C. Panse, J. Schneidewind, and M. Sips. Mail explorer - spatial and temporal exploration of electronic mail. In *EuroVis 2005: Eurographics/IEEE-VGTC Symposium on Visualization, Leeds, UK, June 1st-3rd*, 2005.

[KMSZ06] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler:. Challenges in visual data analysis. In *IEEE Information Visualization, London, UK*, 2006.

[KNP02] Daniel A. Keim, Stephen C. North, and Christian Panse. Cartodraw: A fast algorithm for generating contiguous cartograms. Information Visualization Reseach Group, AT&T Laboratories, Florham Park, 2002.

[KNS+06] D. A. Keim, T. Nietzschmann, N. Schelwies, J. Schneidewind, T. Schreck, and H. Ziegler. A spectral visualization system for analyzing financial time series data. In *EuroVis 2006: Eurographics/IEEE-VGTC Symposium on Visualization, Lisbon, Portugal, 8-10 May*, 2006.

[Koh97] T. Kohonen. *Self Organizating Maps*. Springer Verlag, New York, 1997.

[KS05] D. A. Keim and J. Schneidewind. Scalable visual data exploration of large data sets via multiresolution. *JUCS Special Issue on Visual Data Mining*, 11(11):1766–1779, 2005.

[KSDH03] D. A. Keim, J. Schneidewind, U. Dayal, and M. C. Hao. Geo pixel bar charts, poster-paper. In *VIS 2003, IEEE Visualization, Seattle, Washington*, 2003.

[KSDH05] D. A. Keim, J. Schneidewind, U. Dayal, and M. C. Hao. Visbiz: A business process visualization case study. In *EuroVis 2005: Eurographics/IEEE-VGTC Symposium on Visualization, Leeds, UK, June 1st-3rd*, 2005.

[KSH+05] D. A. Keim, J. Schneidewind, M. C. Hao, U. Dayal, and P. Wright. VisImpact: business impact visualization. In *IST SPIE Visualization and Data Analysis (VDA) Conference 2005, San Jose, California, USA, 16-20 January*, 2005.

[KSHD06] D. A. Keim, J. Schneidewind, M. C. Hao, and U. Dayal. Business process impact visualization and anomaly detection. *Palgrave Macmillan Information Visualization*, 5(1):15–27, 2006.

[KSPN02] D. A. Keim, J. Schneidewind, C. Panse, and S. C. North. Efficient cartogram generation: A comparison. In *InfoVis 2002, IEEE Symposium on Information Visualization, Boston, Massachusetts*, 2002.

[KSPN03] D. A. Keim, J. Schneidewind, C. Panse, and S. C. North. Visualizing geographic information: Visualpoints vs cartodraw. *Palgrave Macmillan Information Visualization*, 2(1):58–67, 2003.

[KSS+03] D. A. Keim, J. Schneidewind, M. Sips, C. Panse, U. Dayal, and M. C. Hao. Pushing the limit in visual data exploration: Techniques and applications. In *Advances in Artificial Intelligence, 26th Annual German Conference on AI, KI 2003, Hamburg, Germany, September 15-18, Lecture Notes in Artificial Intelligence, Vol. 2821*, 2003.

[KSS04a] D. A. Keim, J. Schneidewind, and M. Sips. Circleview - a new approach for visualizing time related multidimensional data sets. In *ACM Advanced Visual Interfaces, International Working Conference, AVI 2004, May 25-28, Gallipoli (Lecce), Italy*, 2004.

[KSS+04b] D. A. Keim, J. Schneidewind, M. Sips, C. Panse, and H. Barro. Exploring and visualizing the history of infovis, contest-poster: 2nd place award. In *IEEE Symposium on Information Visualization 2004 (InfoVis04), Austin, Texas, USA*, 2004.

[KSS+04c] D. A. Keim, J. Schneidewind, M. Sips, C. Panse, and R. Heilmann. Finding spatial patterns in network data. In *From Data to Patterns: Int. Workshop on Pattern Representation and Management, PaRMa'04, Heraklion - Crete, Greece, March 18*, 2004.

[KSS+05] D. A. Keim, J. Schneidewind, M. Sips, C. Panse, H. Dolfing, J. Haddick, and F. Dill. Ivc05 exploration toolkit, contest-poster: Honorable mention. In *IEEE Symposium on Information Visualization (InfoVis 2005), Minneapolis, MN, USA, October 23-25*, 2005.

[KSS06] D. A. Keim, J. Schneidewind, and M. Sips. Scalable pixel-based visual data exploration. In *IEEE EMBS Visual Information Expert Workshop (VIEW 06), Paris, France*, 2006.

[KSSP04a] D. A. Keim, J. Schneidewind, M. Sips, and C. Panse. Analyzing large collections of e-mail. In *IKE'04 - The 2004 International Conference on Information and Knowledge Engineering, June 21 - 24, Las Vegas, Nevada, USA*, 2004.

[KSSP04b] D. A. Keim, J. Schneidewind, M. Sips, and C. Panse. Geo-spatial data viewer: From familiar land-covering to arbitrary distorted geo-spatial quadtree maps. In *WSCG 2004, The 12-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, February 2 - 6, Plzen, Czech Republic*, 2004.

[KW02] Daniel A. Keim and Matthew Ward. *Intelligent Data Analysis, an Introduction by D. Hand and M. Berthold*, chapter Visual Data Mining Techniques. Springer Verlag, 2 edition, 2002.

[LA94] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comput.-Hum. Interact.*, 1(2):126–160, 1994.

[Les97] Michael Lesk. How much information is there in the world?, 1997. Technical report, www.lesk.com/mlesk/ksg97/ksg.html.

[Ley05] M. Ley. The dblp bibliographic digital library, September 2005. http://www.dblp.de.

[LV06] Peter Lyman and Hal R. Varian. How much information, September 2006. Retrieved from http://www.sims.berkeley.edu/how-much-info-2003.

[LWW90] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring n-dimensional databases. In *Proc. Visualization '90, San Francisco, CA*, pages 230–239, 1990.

[Mac99] J.D. Mackinlay. Automating the design of graphical presentaions of relational information. In *Readings in Information Visualization*, pages 66–81, San Francisco, CA, USA, 1999. Morgan Kaufmann.

[Man01] Steve Mann. *Intelligent Image Processing*. John Wiley and Sons, November 2 2001.

[MDK05] A. MacEachren, J. Dykes, and M.-J. Kraak, editors. *Exploring Geovisualization*. Academic Press/Elsevier, 2005.

[MHNW97] Nancy Miller, Beth Hetzler, Grant Nakamura, and Paul Whitney. The need for metrics in visual information analysis. In *NPIV '97: Proceedings of the 1997 workshop on New paradigms in information visualization and manipulation*, pages 24–28, 1997.

[MRC91] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: detail and context smoothly integrated. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 173–176, New York, NY, USA, 1991. ACM Press.

[MS03] W. Müller and H. Schumann. Visualization methods for time-dependent data - an overview. In *Proceedings of Winter Simulation (WSC03)*, 2003.

[MSSK06] F. Mansmann, J. Schneidewind, T. Schreck, and D. A. Keim. Monitoring network traffic with radial traffic analayzer. In *IEEE Symposium on Visual Analytics and Technology (VAST 2006), Baltimore, Maryland, USA,, October 29 - November 3*, 2006.

[Mun97] T. Munzner. H3: laying out large directed graphs in 3d hyperbolic space. In *INFOVIS '97: Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)*, page 2, Washington, DC, USA, 1997. IEEE Computer Society.

[oC] United States Department of Commerce. US Census Bureau website. http://www.census.gov, Sep. 2003.

[Pea96] K. Pearson. *Mathematical contributions to the theory of evolutions III.Regression, heredity, and panmixia.* Phil Trans R Soc Lond Series, 1896.

[PG88] R. M. Pickett and G. G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Proc. IEEE Conf. on Systems, Man and Cybernetics, IEEE Press, Piscataway, NJ*, pages 514–519, 1988.

[Pic70] R. M. Pickett. *Visual Analyses of Texture in the Detection and Recognition of Objects.* Academic Press, New York, 1970.

[Pla86] William Playfair. *The Commercial and Political Atlas.* London, 1st edition, 1786.

[PR96] P. Pirolli and R. Rao. Table lens as a tool for making sense of data. In *AVI '96: Proceedings of the workshop on Advanced visual interfaces*, pages 67–80, New York, NY, USA, 1996. ACM Press.

[PSBK+96] G. Piatetsky-Shapiro, R. Brachman, T. Khabaza, W. Kloesgen, and E. Simoudis. An overview of issues in developing industrial data mining and knowledge discovery application. In *In Proceedings of the Second Internation Conference on Knowledge Discovery and Data Mining.* AAAI Press, 1996.

[SA81] R. Spence and M. Apperley. Data base navigation an office environment for the professional. *Behavior and Information Technology*, pages 6–1 to 6–9, 1981.

[Sah02] A. Sahai. Automated sla monitoring for web services. *IEEE/IFIP DSOM 2002, Montreal, Canada,*, 2002.

[SCB98] D. F. Swayne, D. Cook, and A. Buja. XGobi: Interactive dynamic data visualization in the X Window System. *Journal of Computational and Graphical Statistics*, 7(1):113–130, 1998.

[Shn92] Ben Shneiderman. Tree visualization with tree-maps: A 2-D space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[Shn96] B. Shneiderman. The eye have it: A task by data type taxonomy for information visualizations. In *Visual Languages*, 1996.

[Sim91] Karl Sims. Artificial evolution for computer graphics. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 319–328, New York, NY, USA, 1991. ACM Press.

[Sip06] Mike Sips. *Pixel-based Visual Data Mining in Large Geo-Spatial Point Sets*. Series in Computer Science Vol. 8, Hartung-Gorre Verlag, 2006.

[Sma06] Smartmoney. Map of the market, 2006. http://www.smartmoney.com/marketmap/.

[SMK07] J. Schneidewind, M.Sips, and D. A. Keim. An automated approach for the optimization of pixel based visualizations. *Palgrave Macmillan Information Visualization*, 6(1):X, 2007.

[Sno55] John Snow. *On the Mode of Communication of Cholera*. John Churchill, New Burlington Street, London, England, 1855.

[Spe01] Robert Spence. *Information Visualization*. ACM Press Books, Pearson Education ltd.,UK, 2001.

[Spe06] Robert Spence. *Information Visualization: Design for Interaction*. Prentice Hall, 2006.

[SS06] Jinwook Seo and Ben Shneiderman. Knowledge discovery in high-dimensional data: Case studies and a user survey for the rank-by-feature framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(3):311–322, 2006.

[SSK06] J. Schneidewind, M. Sips, and D. A. Keim. Pixnostics: Towards measuring the value of visualization. In *IEEE Symposium on Visual Analytics and Technology (VAST 2006), Baltimore, Maryland, USA,, October 29 - November 3*, 2006.

[SSKN06] M. Sips, J. Schneidewind, D. A. Keim, and Stephen C. North. Information at your finger tips: Exploring the us census data. In *IEEE Symposium on Information Visualization, Contest Poster, Baltimore, Maryland, USA,, October 29 - November 3*, 2006.

[SSKS06] M. Sips, J. Schneidewind, D. A. Keim, and H. Schumann. Scalable pixel-based visual interfaces: Challenges and solutions. In *IEEE Information Visualization, London, UK*, 2006.

[STH02] Chris Stolte, Diane Tang, and Pat Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002.

[Sug81] K. Sugiyama. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 1(2):109–125, 1981.

[SZ00] J. T. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the IEEE Symposium on Information Visualization*, page 57, 2000.

[TK05] J.J. Thomas and K.A.Cook. Illuminating the path:Research and Development agenda for Visual Analyics. In *IEEE*, pages 79–86, October 2005.

[Tob76] W.R. Tobler. Cartograms and cartosplines. *Proceedings of the 1976 Workshop on Automated Cartography and Epidemiology*, pages 53–58, 1976.

[TT85] J.W. Tukey and P.A. Tukey. Computing graphics and exploratory data analysis: An introduction. In *Proceedings of the Sixth Annual Conference and Exposition: Computer Graphics85*. Nat. Computer Graphics Assoc., 1985.

[Tuf83] Edward R. Tufte. *The visual display of quantitative information*. Graphics Press, 1st edition, 1983.

[Tuf90] Edward R. Tufte. *Envisioning Information*. Graphics Press, 1990.

[Tuf97] Edward R. Tufte. *Visual Explanations*. Graphics Press, 1997.

[Tuk77] J.W. Tukey. *Exploratory Data Analysis*. Addison Wesley Publishing, Reading, MA, 1977.

[vW05] J.J. van Wijk. The value of visualization. In *Proceedings of IEEE Visualization*, pages 79–86, Minneapolis, MN, October 2005.

[vWS99] J.J. van Wijk and E.R.V. Selow. Cluster and calender based visualization of time series data. In *Proc. of the IEEE InfoVis Symposium '99*, pages 4–9. IEEE Computer Society, 1999.

[vWvL93] J. J. van Wijk and R.. D. van Liere. Hyperslice. In *Proc. Visualization '93, San Jose, CA*, pages 119–125, 1993.

[WAG05] Leland Wilkinson, Anushka Anand, and Robert Grossman. Graph-theoretic scagnostics. In *INFOVIS '05: Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 21, Washington, DC, USA, 2005. IEEE Computer Society.

[WAM01] Marc Weber, Marc Alexa, and Wolfgang Mueller. Visualizing time-series on spirals. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, pages 7–14, Washington, DC, USA, 2001. IEEE Computer Society.

[War94] M. O. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proc. Visualization 94, Washington, DC*, pages 326–336, 1994.

[War00] Colin Ware. *Information Visualization - Perception for Design.* Morgan Kaufmann, 1st edition, 2000.

[Wat05] Martin Wattenberg. A note on space-filling visualizations and space-filling curves. In *Proceedings of the 2005 IEEE Symposium on Information Visualization.* IEEE Computer Society, 2005.

[YWR02] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proceedings of the IEEE Symposium on Information Visualization*, page 77, 2002.

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

[YWRH03] J. Yang, M. O. Ward, E. A. Rundensteiner, and S. Huang. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003*, pages 19–28, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[YWY00] J. Yang, W. Wang, and P. Yu. Mining asynchronous periodic patterns in time series data. In *Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, pages 275–279, 2000.

[ZIB01] John Zachary, S. S. Iyengar, and Jacob Barhen. Content based image retrieval and information theory: a general approach. *J. Am. Soc. Inf. Sci. Technol.*, 52(10):840–852, 2001.

[Zyt02] J. M. Zytkow. Types and forms of knowledge (patterns): decision trees,. In *Handbook of data mining and knowledge discovery, Oxford University Press, Inc.*, 2002.