Projections for Visual Analysis of Multivariate Data: Methods for Identification, Interpretation, and Navigation of Patterns

Dissertation zur Erlangung des

akademischen Grades eines Doktors der

Naturwissenschaften

vorgelegt von

Dominik Peter Jäckle

an der

Mathematisch-Naturwissenschaftliche Sektion

Informatik und Informationswissenschaft

Konstanz, 2017

Konstanzer Online-Publikations-System (KOPS)
URL: http://nbn-resolving.de/urn:nbn:de:bsz:352-2-dys499qgqcaj1

Tag der mündlichen Prüfung: 13. Dezember 2017

1. Referent: Prof. Dr. Daniel A. Keim

2. Referent: Prof. Dr. Harald Reiterer

To my parents.

"If you want to go fast, go alone. If you want to go far, go together."

– African Proverb –

Acknowledgments

From my point of view, Visual Analytics is, where creativity and algorithms meet to tell an intriguing story about a possibly unknown, complex problem from a different angle and provide new insight. This is what I have enjoyed during my time as a Ph.D. student – a balancing act between appealing design, theoretical approaches, and applications. I am very thankful for the opportunity I was given by my supervisor Daniel Keim to realize my ambitions and be part of his amazing group. I also like to thank my secondary advisor, Harald Reiterer, who supported my work from an early stage on in various fruitful discussions and collaborations. They not only encouraged me to continue my work after several setbacks successfully but also shared their longstanding experience with me.

After I completed my studies at the University of Stuttgart in 2012, I moved to Konstanz and encountered a great team and great people, who warmly welcomed me. The dedication text *"If you want to go fast, go alone. If you want to go far, go together."* best describes all the fruitful collaborations I had. I want to particularly thank Florian Stoffel, Bum Chul Kwon, Juri Buchmüller, Johannes Fuchs, Fabian Fischer, Dominik Sacha, Michael Hund, Sebastian Mittelstädt, Andreas Stoffel, Michael Behrisch, Hansi Senaratne, and Tobias Schreck, who supported my research agenda. I also like to thank all other colleagues of the DBVis group for all the interesting and sometimes silly discussions in our coffee corner.

Durig two projects, I worked in close collaboration with researchers from Siemens Munich and Hewlett-Packard Laboratories in Palo Alto. I want to thank Daniela Oelke from Siemens, and Wei-Nchih Lee, Ming Hao, Nelson Chang, and Henry Sang from Hewlett-Packard for the very fascinating and productive sessions. I did indeed learn a lot.

Nothing in my life would have ever been possible without the steady support and love of my family and my amazing parents Peter and Liselotte, who always have a place in my heart.

Abstract

Dimensionality Reduction, in particular, projection-based methods transform the data to a lower-dimensional space, yet preserving its main structure. A scatterplot typically depicts the results, presenting a means to make the data space visually accessible to the user. This abstract representation of complex data enables exploration, however, brings in challenges about the analysis and interpretation of patterns because the data is often large-scale, comprises many attributes, or evolves. The present thesis aims to integrate the user into the analysis process using interactive data visualization, and centers around the research question: *How to support people to identify, interpret, and navigate patterns in multivariate projection spaces?* This thesis makes two main computer science contributions to tackle this question based on the assessment of related work concerning the interactive visual analysis of projections of multivariate data.

First, the development and evaluation of interactive visual analysis methods to foster the identification and interpretation of patterns in multivariate data spaces using projections. A user study together with domain experts untrained in advanced statistics shows the effectiveness of projections. The experts mastered the abundance of attribute combinations (subspaces), and thus patterns, by manually deciding on interesting attributes. This behavior motivated the development of novel methods to analyze structural pattern changes among different subspaces visually and to support the interpretation of identified patterns. Patterns can not only change among subspaces but also over time, posing a challenge to identify patterns in general. This thesis proposes sequential one-dimensional projections that make temporal patterns visible, as well as means to interpret identified patterns. Different use cases showcase the usefulness of the methods, including the analysis of survey, crime and computer network data.

Second, the development and investigation of off-screen visualization for context-aware navigation in information spaces spanned by projections. The depiction of a multivariate projection can result in a large information space that is challenging to navigate effectively. Users apply zooming and panning operations to explore the space at a global but also local scale depending on the task at hand. As a result, the users face the inherent trade-off between overview and detail. This work proposes a data-driven overview by surrounding the viewport with a dedicated border region that preserves the relations between off-screen located data objects. Aggregation, thereby, plays a key component to overcome the challenges regarding the visualization of vast amounts of data. Several techniques and use cases are presented in this context. Furthermore, results of a study show that the border can be designed adaptively to improve the awareness of the data space dimensions without negatively influencing the overview perception. The results of the study also suggest projecting off-screen located objects to the border region using the orthographic over the radial strategy.

The present thesis systematically discusses the benefits and challenges of the proposed methods and outlines future directions.

Zusammenfassung

Projektionsbasierte Methoden zur Dimensionsreduktion übertragen die Daten in einen niederdimensionalen Raum, der die Gesamtstruktur erhält. Die Ergebnisse einer Projektion werden häufig in einem Streudiagramm dargestellt, was dem Zweck dient, den Datenraum visuell zugänglich zu machen. Diese abstrakte Darstellung ermöglicht die Exploration der Daten, beinhaltet jedoch Herausforderungen bezüglich der Analyse und Interpretation von Mustern; die Daten sind oft zu viele, umfassen mehrere Attribute oder entwickeln sich über die Zeit. Die vorliegende Arbeit zielt darauf ab, den Benutzer in den Analyseprozess mittels interaktiver Datenvisualisierung zu integrieren und konzentriert sich auf folgende Forschungsfrage: *Wie kann man jemand dabei unterstützen, Muster in multivariaten Projektionsräumen zu identifizieren, zu interpretieren und zu navigieren?* Um diese Forschungsfrage zu beantworten, werden zunächst verwandte Arbeiten hinsichtlich der interaktiven visuellen Analyse von multivariaten Projektionen bewertet. Auf dieser Grundlage legt diese Dissertation zwei wissenschaftliche Beiträge aus dem Gebiet der Informatik dar.

Als Erstes die Entwicklung und Evaluation interaktiver visueller Analysemethoden, die die Identifizierung sowie Interpretation von Mustern in multivariaten Datenräumen mittels Projektionen fördern. Eine Benutzerstudie mit Domänenexperten, die ungeschult in fortgeschrittener Statistik sind, zeigt die Effektivität von Projektionen auf. Die Experten bewältigten die Unmenge an Attributkombinationen (Unterräume) und Mustern, indem sie manuell interessante Attribute auswählten. Dieses Vorgehen motivierte die Entwicklung von neuen Methoden zur visuellen Analyse und Interpretation von Mustern, die sich zwischen verschiedenen Unterräumen entwickeln. Muster können sich jedoch nicht nur zwischen Unterräumen entwickeln, sondern auch mit der Zeit, was das Auffinden von Mustern generell beeinträchtigt. Diese Dissertation empfiehlt die Anwendung von eindimensionalen Projektionen, die sequentiell ausgerichtet werden und somit zeitabhängige Muster sichtbar machen. Darüber hinaus werden visuelle Methoden zur Interpretation angeboten. Mit Hilfe von Umfragedaten, Kriminalstatistiken und Netzwerkdaten wird die Anwendbarkeit der vorgestellten Methoden gezeigt.

Als Zweites die Entwicklung und Untersuchung von sogenannten Off-screen Visualisierungen zur Navigation in Informationsräumen, die von Projektionen aufgespannt werden. Die Navigation kann dabei stark beeinträchtigt werden, falls das Projektionsergebnis in übermäßig großen Informationsräumen resultiert. Um globale sowie lokale Muster in dem aufgespannten Raum zu explorieren, wenden Benutzer sogenannte Zoom- und Pan-Operationen an. Ein Ergebnis ist, dass sich Benutzer ständig zwischen dem großen Ganzen und Detailinformationen bewegen, was einen klaren Nachteil darstellen kann. Diese Arbeit empfielt eine datengetriebene Übersicht, während Detailinformationen exploriert werden. Um diese Übersicht zu erhalten, wird der sichtbare Bereich mit einem dedizierten Bereich umschlossen, der es erlaubt, Relationen zwischen off-screen Objekten zu erhalten. Dabei spielt Aggregation eine Schlüsselrolle, da sie die Herausforderungen, die mit der Darstellung von großen Daten-

mengen verbunden sind, meistert. In diesem Kontext werden verschiedene neue Techniken sowie Anwendungsbeispiele präsentiert. Ergebnisse einer durchgeführten Studie zeigen außerdem auf, dass der dedizierte Bereich adaptiv sein darf, um die Wahrnehmung des zu navigierenden Datenraums zu verbessern. Des Weiteren suggerieren die Ergebnisse, dass off-screen Objekte mit der orthographischen anstatt der radiellen Strategie in den dedizierten Bereich zurückprojiziert werden sollen.

Die vorliegende Dissertation diskutiert systematisch die Vorteile und Herausforderungen der vorgeschlagenen Methoden und umreißt zukünftige Ausrichtungen.

Contents

1 Introduction

1.1 Motivation

THE notion of *Multivariate Data* characterizes information that comprises various observations, each described by multiple attributes [39]. Data, or information, of this type, are collected at large-scale in all areas of our day-to-day life: computer network logs, surveys of natural disasters or inhabitants, crime reports, financial statistics, or any tabular data that consists of multiple observations and attributes. The main tasks in understanding such complex data are to identify, interpret, and navigate interesting, discernible areas (patterns). Examples include dense groups, outliers, correlations, or any anomaly among attributes or observations that provide insight into the data structure. Real-world data, however, is often large-scale, comprises many attributes, or evolves over time, posing a challenge to provide appropriate methods to get insight and generate knowledge about the data.

Visual Analytics suggests involving the user in the analysis process using interactive data visualization [177]. By leveraging the human capabilities to explore the data, visual analytics facilitates finding relevant information and fosters sensemaking. This thesis, furthermore, follows the idea of *Explorative Data Analysis*, which was coined by John Tukey [204]. The main idea is to explore the data and form new hypotheses without having specific knowledge about the data. The exploration of multivariate data, in particular, poses a challenging task. Firstly, because one needs to find appropriate methods to identify and interpret patterns in multivariate data to provide insight. Secondly, because the visual representation of the data possibly spans a large physical space. There is a need for preserving the main data characteristics, as well as the data topology to enable the user's awareness of spatial relations. This thesis focuses on Dimensionality Reduction as means to visually explore the data. I investigate the interpretability of *multivariate data projections*, as well as novel methods for identifying patterns and for navigating the resulting information space.

1.1.1 Multivariate Data

There exist different terms that describe tabular data. Thereby, one has to distinguish between the description of the data as a whole and the columns. Commonly used terms to label the data are *multivariate*, *multi-dimensional*, or *high-dimensional*. According to Chan [37], multi-dimensional data comprises independent dimensions, whereas multivariate data consists of dependent variables. "Nevertheless, a set of multivariate data is in high dimensionality and can possibly be regarded as multi-dimensional because the key relationships between the

attributes are generally unknown in advance. The multi-dimensional property is therefore implied in common usage" [37, p. 8]. Furthermore, the terms *attribute* and *feature* can be used to denote either *dimensions* (independent) or *variables* (dependent). In this work, I stick with the concept of multivariate data that comprises multiple attributes.

1.1.2 Multivariate Data Projections

Dimensionality Reduction (DR), in particular, projection-based methods are a means to make multivariate data visually accessible to the user. The general idea of DR is to transform the data to a lower-dimensional space, preserving its main structure. Results are typically depicted in a two-dimensional scatterplot, in which proximity between points indicates similarity. The DR pipeline is illustrated in Figure 1.1 [178]. The pipeline consists of four consecutive steps enabling the user to interact with each of them. Especially the latter three steps represent ongoing research, which involves several challenges.

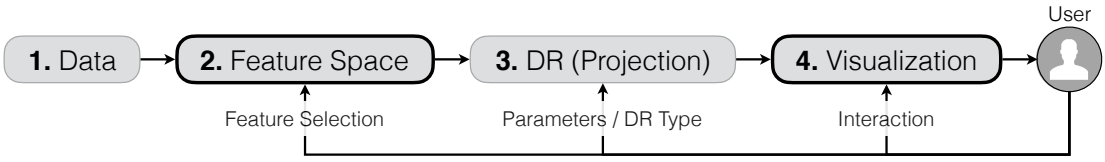


Figure 1.1: *The DR pipeline according to Sacha et al. [178] describes the classical way to transfer multivariate data into a visualization. This thesis contributes methods to tackle the challenges introduced in the steps (2)* feature space *and (4)* visualization*. There is a clear need to support users with appropriate interpretation and navigation methods.*

The actual DR, the projection, is performed in the third step. A well-known and researched projection method is the *distance-preserving projection* Multidimensional Scaling (MDS) [47]. MDS maps the data to a lower-dimensional representation and enables discovering structures while the pairwise distances between observations are preserved. In contrast to other methods, MDS can also take distances for nominal (values are $=$ or $\neq$), ordinal (values follow a natural ordering), or any quantitative (can do arithmetic on values) attributes [33] into account. Because real-world data often comprises different data types, this thesis showcases methods and results using MDS. All presented analysis and visualization methods are also applicable using other DR methods that operate on purely numerical data.

This thesis particularly addresses challenges related to the steps (2) *feature space* and (4) *visualization*. The feature space and the visualization are typically considered together because the configuration of the feature space directly impacts the visual layout after projecting the data. The inherent problem is that attributes have a different impact on the projected data points' positions and thus the perceived patterns. One does initially not know which attributes make up a pattern. In particular, for users not trained in advanced statistics, this poses a challenge. Current methods provide intriguing statistic-driven solutions but fall short showing their applicability.

On distance-preserving projections, the more attributes are considered, the less discriminative the distance between projected data points is; significant differences become blurred. Finding the most expressive attributes, or the attributes building an interesting pattern, still

poses an ongoing key challenge. Furthermore, multivariate data can be dynamic. For example, threats in computer networks are multivariate and evolve over time, which poses a challenge to detect patterns visually. We need to adapt projection methods to be able to make patterns over time visible.

There is a clear need to support users with appropriate interpretation methods and to give evidence whether they understand the multivariate projection.

1.1.3 Overview-Preservation

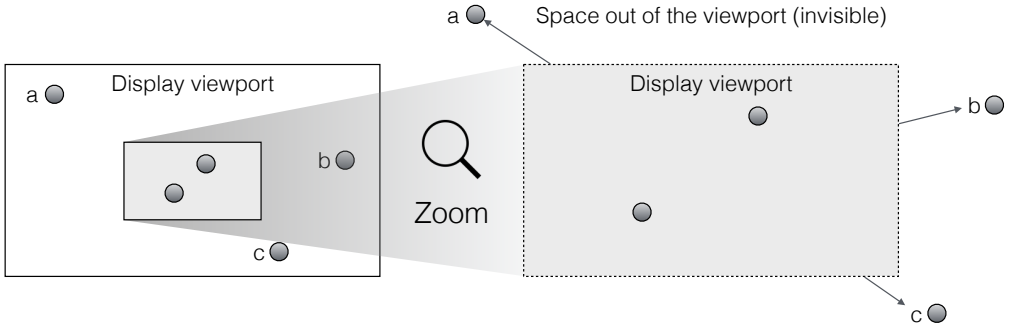


Figure 1.2: *Illustration of the overview-preservation problem. Objects move out of sight through zooming and/or panning interactions. In this example, a zooming interaction causes the objects* a*,* b*, and* c *to move away from the visible area (the display viewport). Users forfeit their overview awareness caused by currently invisible objects.*

In this work, the (4) visualization step of the DR pipeline, depicted in Figure 1.1, shows the projection results as a scatterplot, and thus the structure of the underlying data and the respective feature space configuration. The continuous collection of data poses high demands to the visualization of multivariate data, as well as to the user, due to the limited screen real estate. The problem is depicted in Figure 1.2. Within the limited space, one performs effective interaction techniques to aggregate information for an overview and to focus on areas of interest back and forth. In the event users apply zooming or panning operations to explore large data spaces, the operations have one important commonality: both zooming and panning imply that the user is only analyzing and/or looking at one specific area in detail and other possibly relevant information moves out of the display viewport. In such situations, users face the inherent trade-off between overview and detail as Jerding and Stasko defined in the following way [112, p. 43]:

> *"Visualizations which depict entire information spaces provide context for navigation and browsing tasks; however, the limited size of the display screen makes creating effective global views difficult."*

It is still ongoing, unsolved research how to providing overview and context while showing an area in detail. Multivariate data comprises possibly multiple attributes and data types, posing a challenge to provide a data-driven context effectively. Despite the advancement made in image-based approaches, I argue that data-driven, context-preserving visualizations have not been sufficiently considered for multivariate data, yet.

1.2 Research Trajectory

The Information Mural by Jerding and Stasko [112] was one of the first research papers I read as a postgraduate. Although the paper was published in 1998, the problem of having too few pixels to display large information spaces entirely still appears relevant. The resolution of displays has increased, but so has the amount of data. In the era of Big Data, we process and present vast amounts of data in constant pursuit of insight. This applies in particular to DR techniques such as planar projections of data with many attributes. Projections span a huge space and can hide patterns due to sequential and/or attribute-wise dependencies. Interactive visual analysis helps to tackle this problem and proposes to let the human steer the projection parameters and visual representation beyond automatic capabilities, leading to the driving **research question** of this thesis:

> ***"How to support people to identify, interpret, and navigate patterns in multivariate data spaces using interactive visual analysis?"***

The analysis of *patterns in multivariate data space* – to which I count the identification, interpretation, and navigation – poses the key challenge of this thesis. Projections of multivariate data represent a means to make the data space accessible to the user. However, a projection expresses similarities between objects through aggregation of attributes entailing a loss of information. To enable the identification and interpretation of patterns in the projected information space, one must provide support to view and analyze the configuration of attributes in feature space. Interaction, therefore, plays a crucial role, not only to integrate the user into the automated analysis process but also to let the user explore the data. I employ interaction to identify patterns, foster interpretation, and navigate the information space.

This thesis follows a research methodology based on real-world use cases, (rapid) prototyping, and feedback elicitation through qualitative and quantitative evaluation. In particular, I develop, improve, prototype, and evaluate interactive visual analysis techniques to answer the research question. I focus on the generalizability and transferability of results. The data used is interchangeable, as long as it is encoded and formatted accordingly, thus can be transferred to different domains that deal with same challenges.

1.3 Thesis Outline & Contributions

The content of this thesis is bundled under the concept of *Interactive Visual Analysis*, which makes use of the power of data visualization paired with interactive methods to steer algorithms and interpret results to generate knowledge. Figure 1.3 depicts the structure of this work, which is divided into two parts. The first part is about the identification and interpretation of multivariate patterns. I describe new methods to identify and interpret patterns in multivariate projections in Chapter 3. Also, I investigate whether users untrained in DR can interpret the depiction of a projection. In Chapter 4, I propose a new method to

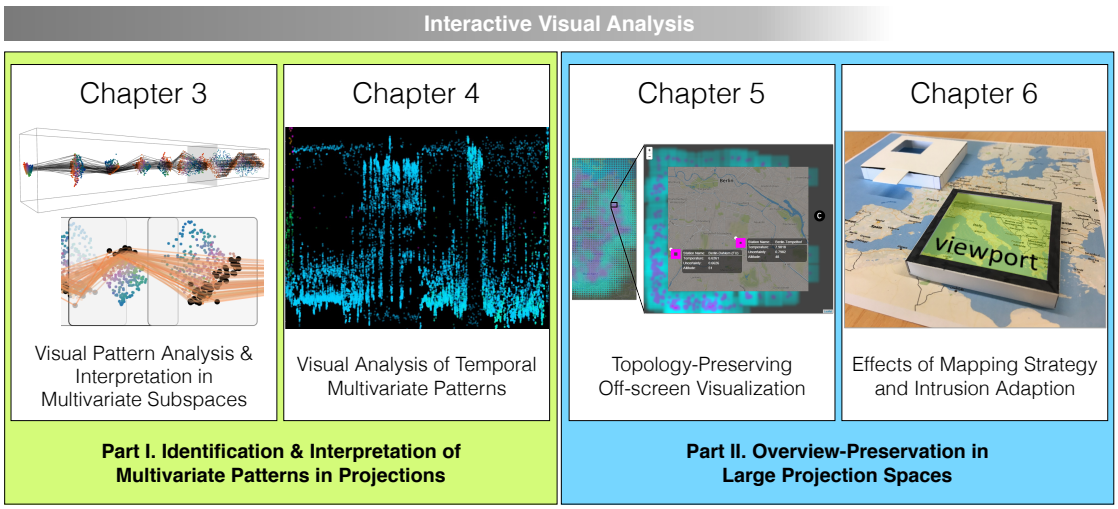


Figure 1.3: *Overview of the core chapters building the main contributions of this thesis. The chapters are assigned to two higher-level parts, each tackling one of the identified problems regarding projections for visual analysis of multivariate data. The interactive visual analysis is a key concept dominating this thesis, in particular Chapters 3 and 4, where the interpretation of multivariate projections is regarded. Chapter 5 presents a highly interactive approach. However, the focus is on preserving overview in 2D projections and only partially relates to interactive analysis. Chapter 6 analyzes the effects of the overview-preservation introduced in Chapter 5.*

generate patterns using sequential projections applied to data sequences. The second part deals with the navigation and context-preservation of the space, spanned by the projection, using off-screen visualization. Chapter 5 introduces the concept of off-screen visualization and contributes methods to handle large datasets. A core design decision to preserve the dimensions of the navigated space is to use an adaptive border intrusion. This design decision was evaluated together with the projection strategy, and the results are described in Chapter 6.

This dissertation claims the following two key *contributions*:

- *The development and evaluation of novel interactive visual analysis methods to foster identification and interpretation of patterns in multivariate data spaces using projections.*
- *The development and investigation of off-screen visualization techniques and strategies for context-aware navigation of information spaces spanned by the projection.*

These contributions distribute among the chapters as follows:

Chapter 3: This Chapter makes the following contributions towards the detection and interpretation of patterns in multivariate projections. First, a visual analytics system that integrates mixed data types into the projection. Since real-world datasets typically comprise different data types beyond numbers and categories, this system enables analysts to explore their domain-specific data. However, domain experts have diverse backgrounds and may not be used to such representations. To answer the question about interpretability, I then conducted a user study to investigate whether domain experts untrained in advanced statistics can interpret the results of a multivariate projection. The results show that they can do so,

given tasks that are particularly relevant in their domain. I observed that domain experts included attributes differently into the projection to verify their hypotheses in different subspaces. To tackle the question of how patterns change across different subspaces, I further contribute a method that includes subspaces into a small-multiple environment and enables users to inspect the pattern transitions among subspaces. Related to that, I developed a new similarity measure between multivariate projections to order the small multiples.

Chapter 4: In this Chapter, I contribute a visual method named Temporal Multidimensional Scaling (TMDS) that creates projections to identify patterns in multivariate data that may include sequential dependencies. A sliding window is applied to the data and a one-dimensional projection computed for each window. Aligning the projections one after another reveals not only patterns based on similarity but also patterns where sequences play a key role and contribute to the understanding. Based on the sequential projections, I furthermore contribute a method to find similar patterns in the resulting projection space based on a previously known pattern.

Chapter 5: This Chapter opens the design space of off-screen visualizations for context-preservation and contributes and discusses three interactive techniques that aim at different data characteristics. First, I propose an off-screen visualization, introducing a data-driven border region. Based on rasterization, points and shapes plus an additional data encoding can be preserved while navigating spatial datasets. Second and based on the rasterization, I propose to encode a second data value which I showcase using uncertainty information about the data. Third, I go one step further and propose to use multivariate star glyphs to encode more than two dimensions for off-screen information. All three approaches are based on aggregation. Since the aggregation using a dedicated border region represents the logical consequence compared to state-of-the-art techniques, I evaluated the usage against the latest off-screen technique making use of aggregation, namely HaloDot (instead of a border, HaloDot shows aggregated off-screen information using halos intersecting the viewport). Results are in favor of using a border region.

Chapter 6: In Chapter 5, I focus on visualizing off-screen objects. However, there are two unanswered questions: firstly, how can the dimensions of the navigated space be reflected and, secondly, which projection strategy (orthographic or radial) meets the users' intuition? Here, I introduce an adaptive border intrusion which I evaluated together with the projection strategy. There are two strategies: The orthographic strategy divides the off-screen space into eight different areas and projects the objects perpendicular to the viewport. In contrast, the radial strategy projects the off-screen located objects along a line towards the center of the viewport. The results show that there is no disadvantage in reflecting the dimensions of the navigated space in an additional encoding. Also, users perform significantly more accurate using the orthographic projection.

1.4 Publications

During the formation process of this thesis, I worked on different publications presenting my current research and intermediate results. The following list outlines all publications that contributed to this thesis as well as the work distribution among authors. The publications are ordered by chapters.

- **Interpretation of Dimensionally-Reduced Crime Data: A Study with Untrained Domain Experts.** D. Jäckle, F. Stoffel, S. Mittelstädt, D. A. Keim, and H. Reiterer. *Proc. Int. Conference on Information Visualization Theory and Applications, 2017.* **Best Student Paper Award**
  (Chapter 3)

  To tackle the research question "Can domain experts untrained in advanced statistics understand the depiction of a multivariate projection?", I conducted a qualitative user study together with F. Stoffel. I contributed: 1) A visual analytics system for fusing mixed data types that enables the exploration and steering of multivariate data projections. 2) A qualitative user study using the phenomenological methodology. I implemented the system, designed the study, and wrote all sections. F. Stoffel revised the paragraphs about the introduction of the domain experts in Sections 1 and 4.1, and reviewed paper drafts. Also, F. Stoffel helped to run the study with the domain experts. S. Mittelstädt contributed the multivariate color mapping strategy which he developed in [156]. H. Reiterer contributed to various discussions shaping the paper and commented together with D. Keim on paper drafts.

- **Pattern Trails: Visual Analysis of Pattern Transitions in Subspaces.** D. Jäckle, M. Hund, M. Behrisch, D. A. Keim, and T. Schreck. *IEEE Conference on Visual Analytics Science and Technology (VAST), 2017.*
  (Chapter 3)

  Together with M. Hund and T. Schreck, we identified the research question "How to identify and relate interesting patterns among multivariate subspaces, using interactive visual exploration?". I contributed: 1) Systematization and categorization of pattern transitions among subspaces of multivariate data. 2) A data-driven similarity measure for projections to group subspaces and overcome redundancy. I did the design and implementation of the prototype. M. Hund provided and improved a state-of-the-art subspace analysis algorithm. I wrote all sections of the paper. M. Hund, M. Behrisch, D. A. Keim, and T. Schreck actively reviewed and revised the paper and commented on paper drafts.

- **Temporal MDS Plots for Analysis of Multivariate Data.** D. Jäckle, F. Fischer, T. Schreck, and D. A. Keim. *IEEE Trans. Vis. Comput. Graph. 22(1): 141-150, 2016.*
  (Chapter 4)

  The research question "How to visually discover patterns in temporal multivariate data?" as well as a first idea using subsequent projections were identified in a discussion with

D. Keim. The follow-up research question "How to visually and automatically find similar patterns based on an already identified pattern?" was identified by me. I contributed: 1) A stable temporal multidimensional scaling algorithm using a sliding window approach. 2) A visual approach to identify patterns using a dimension-wise fingerprint matrix. 3) An algorithm to find similar patterns based on already known patterns. I implemented the used prototype, wrote the Sections 1, 2, 3, 4, 5, 7, 8 and revised Section 6. F. Fischer applied the technique to a real-world dataset as a case study, conducted a ground truth evaluation, and wrote Section 6. All co-authors (F. Fischer, T. Schreck, and D. Keim) actively reviewed and commented on paper drafts.

- **Star Glyph Insets for Overview Preservation of Multivariate Data.** D. Jäckle, J. Fuchs, and D. A. Keim. *IS&T Electronic Imaging Conference on Visualization and Data Analysis, 2016.*
(Chapter 5)

  The research question "How to preserve overview, in particular, the data-driven context, for spatial multivariate data?" was identified by me. I contributed: An effective integration of star glyphs as efficient visual insets for the representation of multivariate off-screen data objects. I did the design and implementation of the prototype. I wrote all sections of the paper and revised Section 3.1.2, which was initially written by J. Fuchs. D. Keim provided feedback on paper drafts.

- **Off-Screen Visualization Perspectives: Tasks and Challenges.** D. Jäckle, B. C. Kwon, and D. A. Keim. *Symposium on Visualization in Data Science (VDS) at IEEE VIS, 2015.*
(Chapter 5)

  The research questions "How can off-screen visualization techniques be integrated into visual data analysis?" and "What are the challenges of applying off-screen techniques in visual data analysis?" were both identified by me. I defined the contribution: A discussion of perspectives of potentials and challenges on off-screen visualization based on a thorough review of prior studies. I wrote all sections. B. C Kwon actively reviewed and revised the paper. D. Keim commented on paper drafts.

- **Integrated Spatial Uncertainty Visualization using Off-screen Aggregation.** D. Jäckle, H. Senaratne, J. Buchmüller, and D. A. Keim. *EuroVis Workshop on Visual Analytics (EuroVA), The Eurographics Association, 2015.*
(Chapter 5)

  To address the research questions "How to efficiently integrate spatial data and uncertainty?" and "How to preserve data-driven context thereupon?", I contributed: 1) An extrinsic uncertainty visualization using the Figure-Ground organization. 2) A topology-preserving off-screen visualization technique that incorporates the intrinsic uncertainty visualization. I wrote Sections 1, 2.2, 3, 5 and implemented the web-based prototype. H. Senaratne wrote the initial version of Section 2.1 and was involved in design decisions regarding the integration of uncertainty. J. Buchmüller implemented a server-based data storage and wrote Section 4. All authors commented on paper drafts.

- **Ambient Grids: Maintain Context-Awareness via Aggregated Off-Screen Visualization.** D. Jäckle, F. Stoffel, B. C. Kwon, D. Sacha, A. Stoffel, and D. A. Keim. *Eurographics Conference on Visualization (EuroVis) - Short Papers, The Eurographics Association, 2015.* (Chapter 5)

  To tackle the research question "How to preserve context and topology for vast amounts of shape and point data", I contributed: A data-driven off-screen visualization technique based on aggregation and rasterization. I wrote all Sections. F. Stoffel and me implemented the prototype. F. Stoffel further reviewed and revised parts of the paper and provided the data for the use case. B. C. Kwon, D. Sacha, A. Stoffel, and D. Keim commented on paper drafts.

- **Topology-Preserving Off-screen Visualization: Effects of Projection Strategy and Intrusion Adaption.** D. Jäckle, J. Fuchs, and H. Reiterer. *Technical Report, 2017.* (Chapter 6)

  To approach the research questions "How to properly reflect the dimensions of the navigated space?" and "Which projection strategy best preserves the data topology?" concerning off-screen visualization, I contributed: 1) A novel approach to reflect the space dimensions using an adaptive border intrusion. 2) A controlled experiment to research the effect of the adaptive border intrusion as well as the effect of the projection strategy. I implemented the prototype, conducted the experiment, and wrote all sections. J. Fuchs and H. Reiterer actively contributed to the design of the study and commented on paper drafts.

Also, I contributed to a number of publications that influenced my research trajectory, but are not included in this thesis:

- **Dynamite: Dynamic Monitoring Interface for Task Ensembles.** W. Jentner, M. El-Assady, D. Sacha, D. Jäckle, and F. Stoffel. *IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2016 MC1), 2016.* **Award: Notable Support for Streaming Analysis**
- **SpaceCuts: Making Room for Visualizations on Maps.** J. Buchmüller, D. Jäckle, F. Stoffel, and D. A. Keim. *Eurographics Conference on Visualization (EuroVis) - Short Papers, The Eurographics Association, 2016.*
- **Leaf Glyph - Visualizing Multi-Dimensional Data with Environmental Cues.** J. Fuchs, D. Jäckle, N. Weiler, and T. Schreck. *Proceedings of the 6th International Conference on Information Visualization Theory and Applications - Volume 1: IVAPP, (VISIGRAPP), pages 195–206, 2015* **Best Student Paper Award**
- **ColorCAT: Guided Design of Colormaps for Combined Analysis Tasks.** S. Mittelstädt, D. Jäckle, F. Stoffel, and D. A. Keim. *Eurographics Conference on Visualization (EuroVis) - Short Papers, The Eurographics Association, 2015.*
- **VisJockey: Enriching Data Stories through Orchestrated Visualization.** B. C. Kwon, F. Stoffel, D. Jäckle, B. Lee, and D. A. Keim. *Computation+Journalism, 2014.*

- **Geo-Temporal Visual Analysis of Customer Feedback Data Based on Self-Organizing Sentiment Maps.** H. Janetzko, D. Jäckle, and T. Schreck. *International Journal On Advances in Intelligent Systems, International Academy, Research, and Industry Association (IARIA), 7(1 and 2):237–246, 2014.*
- **State-of-the-Art Report of Visual Analysis for Event Detection in Text Data Streams.** F. Wanner, A. Stoffel, D. Jäckle, B. C. Kwon, A. Weiler, and D. A. Keim. *EuroVis - STARs, Eurographics Association, pages 125-139, 2014.*
- **Enhanced News-reading: Interactive and Visual Integration of Social Media Information.** F. Stoffel, D. Jäckle, and D. A. Keim. *LREC 2014 Workshop VisLR: Visualization as added value in the development, use and evaluation of Language Resources, 2014.*
- **Visual Abstraction of Complex Motion Patterns.** H. Janetzko, D. Jäckle, O. Deussen and D. A. Keim. *IS&T Electronic Imaging Conference on Visualization and Data Analysis, 2014.* **Best Paper Award**
- **Comparative visual analysis of large customer feedback based on self-organizing sentiment maps.** H. Janetzko, D. Jäckle, and T. Schreck. *Proc. International Conference on Advances in Information Mining and Management, 2013.* **Best Paper Award**

2
Background

Contents

THE common theme of this thesis is the comprehensive application of automatic analysis and interactive visualizations as means to generate insight and foster sensemaking [58, 177]. This chapter provides the necessary background for the present thesis regarding recent and ongoing research in the areas of interactive visual data analysis, multivariate data analysis and visualization, and overview-preservation in 2D information spaces.

2.1 Interactive Visual Data Analysis

Interactive Visual Data Analysis, also known as *Visual Analytics*, suggests involving the user into the automated analysis process using interactive data visualizations [126]. This thesis adheres to the notion of *Interactive Visual Data Analysis*, because it emphasizes the complex interplay between visualization, automated data analysis, and interaction, leading to the question: *Why use visualization?* To answer this question, we first need to understand what visualization is. Munzner defines *Visualization* as follows [159, p.1]:

> *"Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively. Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision making methods."*

The second part of this definition, furthermore, provides an explanation for *Why?* there is a clear need for visualization: Visualization leverages the human capabilities to detect and interpret trends and patterns as effectively as possible. An interesting but simple motivation for this statement is Anscombe's Quartet [8]. Anscombe motivated the value of statistical graphs (visualizations) via four different datasets. Each dataset consists of different values, however, have identical statistical characteristics (i.e., mean value, variance, correlation, and linear regression line). Although the datasets have identical statistical characteristics, they look very different when visually inspected. Matejka and Fitzmaurice further demonstrated this effect, and thus the need for visualization, by automating the process of generating distinct datasets that all share the same characteristics [153]. The visual representations reveal fundamental differences in the structures of the data. An interesting observation, even for such small datasets, is, that the statistical characteristics are calculated faster by the computer than manually by the human, hence motivating the field of automated data analysis.

Automated Data Analysis, also known under the term data mining, describes the process of discovering patterns in large amounts of data. Unlike humans, computers efficiently process large datasets, like for example, we encounter in large databases. This process is captured in the so-called Knowledge Discovery in Databases (KDD) process [61]. The KDD process comprises two main stages: First, the identification of the overall goal and transformation of the data to a representation suitable for data mining methods. Second, the search for patterns in the data using appropriate data mining methods (classification, clustering, regression), and their interpretation to generate knowledge. According to Fayyad et al. [61], the two main stages of the KDD process comprise the following nine consecutive steps: (1) identifying the goal of the KDD process, (2) selecting the target dataset, (3) data cleaning and preprocessing, (4) data reduction and projection to find useful attributes, (5) matching the goals of the process to a data mining method, (6) deciding on the data mining method, model, and parameters, (7) applying data mining to search for patterns, (8) interpreting the mined patterns that may involve visualization, (9) acting on discovered knowledge. The generated knowledge can be used to refine each step of the process iteratively. In summary, the KDD process extracts information from data and transforms it into a logical structure to foster knowledge generation [36].

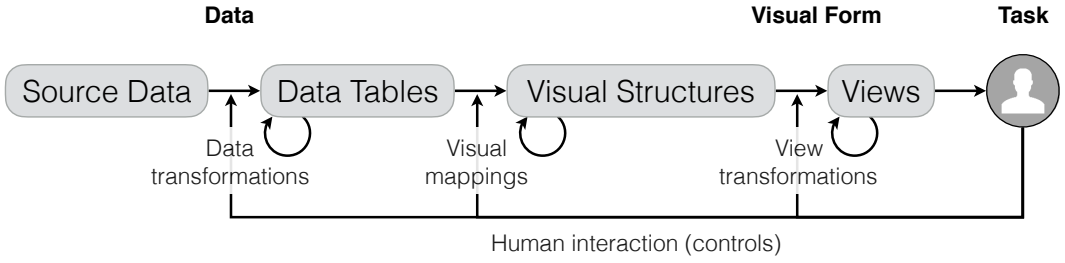


Figure 2.1: *The Information Visualization Reference Model [33] – InfoVis model, for short.*

The potential interaction with each step of the KDD process poses a key component, leading to the questions *What is interaction?* and *Why is it useful?*. Interaction describes the manipulation, either direct or indirect, with the aim to explore and generate different views

on the data. Automated data analysis typically creates one view on the data, for example, a visual view. Interaction with the remaining steps enables the user to efficiently create different views on the data based on the generated insight and knowledge. However, the KDD process has a major drawback. It does not earmark the direct manipulation with the visualization, which is key for exploratory data analysis [204]. This is where the *Information Visualization Reference Model* (InfoVis model) comes into play. In contrast to the KDD process, the InfoVis model suggests, among other things, the direct manipulation with the visualization. The InfoVis model is depicted in Figure 2.1 and comprises three main components: The *Data*, the *Visual Form*, and the *Task*. The main steps to iterate between these components are the following:

Data transformations transform the source data into data tables; this is an intermediate step to prepare the data for visualization. For example, a document vector can represent the raw text.

Visual mappings transform the data tables into visual structures. This means, the data is enriched with information, so that the data can be visualized, like, for example, spatial coordinates, color, among others.

View transformations represent transformations directly imposed on the visualization. Examples include scaling, illumination, or clipping.

Finally, based on the task, the user can interact with each of these steps. According to Card et al. [33], the general idea of this model on interaction is the following:

> *"Visualization can be described as the mapping of data to visual form that supports human interaction in a workplace for visual sense making."*

Interaction, in particular, is useful to handle complexity, because it enables the user to generate different views on the data [159]. For large datasets, it may not be enough to present only one view. Interaction fosters sensemaking and knowledge generation.

The commonality between the KDD process and the InfoVis model is the interaction. Interaction brings both sides, the automated data analysis, and the visual data exploration, together. This interplay was introduced as *Visual Analytics* [46], which enables the effective and efficient generation of knowledge. Later, visual analytics was expressed by different models, like, for example, by Van Wijk [208] or Keim et al. [126]. Keim et al. proposed the visual analytics process, depicted in Figure 2.2, that brings together the strengths of automated data analysis with the strengths of the human, that is the efficient detection of patterns and trends using interactive data visualization [125]. The visual analytics process starts with preprocessing the data from heterogeneous data sources to enable automated data analysis and build models. The visual analytics process reflects the KDD process in the lower part. Then, visualizations are used as means to display the underlying data models. With the aid of interaction, the analyst explores and gets insight into the data, which leads

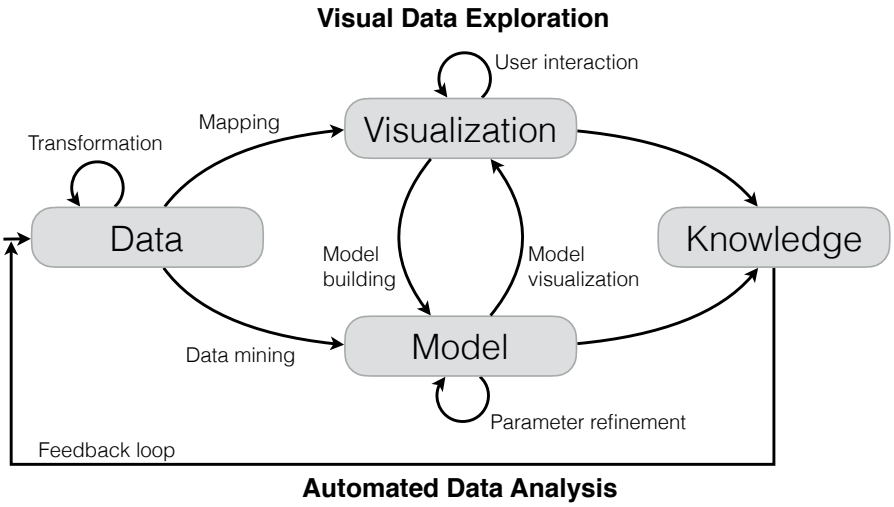


Figure 2.2: *The* Visual Analytics Process *by Keim et al. [126]. The visual analytics process combines the KDD process [61] and the InfoVis model [33] to foster sensemaking [58] and generate knowledge [177]. The pathway from* Data *to* Visualization *complies with the InfoVis model, and the pathway from* Data *to* Model *corresponds to the KDD process. While the InfoVis model enables* Visual Data Exploration *(top), the KDD process enables* Automated Data Analysis *(bottom). The goal of visual analytics is to enable effective and efficient knowledge generation by bringing the opposites together: The (fast) automated analysis of data with the human knowledge and capabilities to detect and interpret trends and patterns.*

to generating knowledge [58]. The visual analytics process reflects the InfoVis model in the upper part.

Sacha at al. [177] further elaborated the human side of this process: The generation of knowledge on the human side comprises three interactive concepts that build on top of each other: First, the exploration, which represents the basis for all knowledge generation. Through interaction with the system and the data, the analyst observes the feedback and summarizes the key features. Any action taken by the analyst is based on a particular finding or a concrete analytical goal. If no goal was defined, the actions serve to search for findings. Second, the verification. The verification is a direct result of any findings identified through exploration and represents the central part of knowledge generation [204]. The idea is to guide the exploration through confirmatory analysis, and create or confirm hypotheses about the data to get new insight. Finally, the knowledge generation, which is tightly integrated with the verification. Users generate new knowledge by formulating and verifying hypotheses.

The present thesis follows the idea of visual analytics, that is the combination of interactive visualization and data mining, and further applies the basic research methodology from human-computer interaction to evaluate developed techniques [137]. This includes to verify or reject certain hypotheses about the targeted users and/or tasks. By basic research methodology, I refer to methods suitable for evaluating developed techniques, such as task-based quantitative or qualitative evaluation. Depending on the research question derived for each of the following chapters, I decided on an appropriate evaluation method [55, 190].

2.2 Visual Analysis of Multivariate Data

Multivariate data analysis and visualization, in the context of visual analytics, aims at providing methods to help understanding relations in multivariate data with many attributes. Various techniques have been presented with the aim to make sense of multivariate data. In the following, I first provide an overview of advancements in recent years. Then, as the main focus of this thesis is on using projections, I give a rough overview of dimensionality reduction techniques. In particular, I describe a projection technique, called Multidimensional Scaling (MDS), which is used throughout this thesis as means to transform multivariate data to a lower-dimensional space. This section summarizes common techniques for multivariate analysis and visualization in view of the present thesis.

2.2.1 Multivariate Data Visualization

Multivariate data analysis methods consider several attributes simultaneously. Typically, attributes in multivariate datasets are related and cannot be regarded as independently [150]. This is different with respect to multi-dimensional data, where individual dimensions are orthogonal to each other and may be reduced, for example, by feature selection techniques [141]. In recent years, many techniques have been presented to make sense of interdependent variables in multivariate datasets visually. Chan [37] and Liu et al. [142] carried out comprehensive surveys for visualizing multivariate data. In the following, I give an overview of most common techniques. Generally speaking, the early works are characterized by the aim to visualize all the data in one display, without overlap, and with only little interaction possibilities.

Pixel-oriented Techniques The main idea of pixel-oriented techniques is to assign each data record to one pixel and to color the pixel with the attribute value. The visual structure, or arrangement, of the pixels makes global and local patterns salient. A well-known ordering strategy represents Recursive Patterns [122]. By arranging the data recursively and coloring the pixels, one can easily spot salient patterns. A similar technique is Circle Segments by Ankerst et al. [7]. Circle Segments segment a circular structure according to the number of attributes. Then, the pixels in each segment are colored with respect to the attribute values and arranged row by row from the outside to the inside of the circle. A similar technique is Pixel Bar Charts [123]. Based on a default bar chart, each bar is built by a tailored pixel-placement algorithm, which enables the visualization of large amounts of data.

Glyph-based Techniques Glyphs are typically applied to compress as much information as possible into space as small as possible. This enables the comparison of characteristics between different datasets or data subsets. There exist three different mapping categories of multivariate data glyphs [212]: The first category are many-to-many mappings, which support the intra-record comparison. A prominent example are profile glyphs [52]. Profile glyphs linearly arrange dimensions and use position or length encoding to represent respective attribute values. The second category are one-to-one mappings. Such designs encode data

values with different visual variables. An example are Chernoff Faces where face characteristics (such as the angle of the eyebrow or the size of the nose) are adjusted based on the underlying data values [42]. This category offers a nearly endless design space with the most flexible way of assigning data values to visual features. The third group are one-to-many mappings, which represent data values redundantly using at least two visual variables. Colored star glyphs [128, 185], for example, make use of length and color of data rays to encode the respective attributes. Important attributes can be visually boosted by assigning more than one visual variable.

Geometric Projections Geometric projections are typically axis-based, this means that each attribute is mapped to one axis. The general idea is to make patterns salient by comparing pairwise attribute relations. For example, Parallel Coordinate Plots (PCP) [98] map each attribute to one axis and arrange all axes side by side. Each data record is spread among all axes and represented as a single line connecting the attribute values. Diverse line patterns among axes leave room for interpretation [49]. However, patterns depend on the axes ordering, which still represents ongoing research. Similar to PCP are Star Coordinates [120]. Axes are aligned in a circular manner, and a single data point represents each data record. You can think of it as a force-directed layout, where each axis attracts the point according to its attribute value. An advantage of this approach is that each data point can be inspected at a glance about all its attribute values and all other data points. Another geometric projection strategy is the Scatterplot Matrix (SPLOM), which enables the pairwise comparison of attributes – each attribute is combined with all other attributes in pairwise scatterplots. Andrews Curve [6] follows a different approach and transform the data into frequency space using Fourier transformation. Similar to PCP, each line corresponds to one data record. This representation enables the identification of similar multivariate patterns.

Methods above give an overview of state-of-the-art, which has been followed in recent years. For further reading, there are of course many other intriguing techniques, which were reviewed, such as the surveys by Kehrer and Hauser [121], Chan [37], or Liu et al. [142]. The techniques, however, have one commonality: They provide an at-a-glance overview of the data but fall short integrating interaction to inspect details. One family of techniques has increased in popularity and includes interaction, namely multivariate projections. Similar to other techniques (e.g., Star Coordinates [120], Andrews Curve [6]), projections also layout the data in two dimensions, yet integrate interaction and statistics to inspect details and reveal patterns that are not obvious at first sight. Throughout this thesis, I apply projections as means to make multivariate data visually accessible to the user.

2.2.2 Using Dimensionality Reduction for Visual Analysis

DR techniques, also known as embedding methods, and in particular multivariate projections, transform the data to a lower-dimensional space, preserving the main structure. Speaking visually, the general idea is to layout the multivariate data typically in a two-dimensional space, in which similar data records are placed closer together than records not being considered

as similar. The notion of similarity, however, can be interpreted and derived differently, depending on projection techniques. More about that in the following sections.

So far, scatterplots are practically the first choice to depict the results of DR. Thereby, the proximity between points indicates how similar they are. Computing similarity or proximity between data, or reducing their dimensionality, is harder to do as more attributes are introduced. Typically, the more attributes are introduced, the less discriminative the projection result is. Thus, the structure of the data is not well-preserved, and patterns are either not visible or ambiguous. Bellman first described this effect as the *Curse of Dimensionality* [17]. Bellman described the curse of dimensionality as an exponential increase in volume when considering additional attributes. Kriegel et al. [133] discussed the curse of dimensionality in view of three problems, among others: First, adding additional attributes causes the range of values to increase; it becomes increasingly difficult to visualize the data. Second, the notion of similarity becomes blurred. DR techniques calculate the similarity between two data records by considering all attributes. The result is similar to an aggregation of the data because the more attributes are added, the less meaningful the computed similarity value becomes. Third, many attributes may be of no importance to the occurring of certain phenomena. However, they influence the resulting structure and possibly impair the formation of expected patterns.

With regard to this thesis, you, the reader, should bear in mind that it is still ongoing research to tackle the problems arising from the curse of dimensionality. This thesis does not provide any solution to this issue but applies two concepts. The first concept is to involve the user, who brings in domain-specific knowledge serving as a preselection of interesting attributes. The second concept is elaborated in Chapter 3, an investigation of meaningful attributes based on subspace analysis. In the following, I give a brief overview of traditional projection techniques for feature extraction, in particular, MDS, which is used throughout this work. Furthermore, I outline relevant visual interactive techniques, which incorporate common multivariate projection methods.

Feature Selection and Feature Extraction

DR can be divided into *Feature Selection* and *Feature Extraction*. Feature selection describes the process of selecting a subset of features (attributes) in a multivariate dataset, which is, for example, beneficial in case of many redundant or irrelevant features [141]. Feature selection can be performed by either including domain-specific knowledge or using computational methods such as correlation analysis or classification [84]. Feature extraction, in contrast, describes the transformation of any data into numerical feature vectors, as well as the transformation of existing features into new ones. Guyon and Elisseff [85] characterize feature extraction as a process that involves feature construction and feature subset generation.

Typically, we associate DR with feature extraction. Using DR, we derive attributes as linear or non-linear combinations of existing ones. The goal of non-linear DR techniques is to preserve the local neighborhood in the data. For example, the Swiss Roll dataset is well-known for pointing out the usefulness of non-linear DR techniques, also known as manifold techniques. The Swiss Roll is a three-dimensional dataset, which in 2D, should be a rolled out manifold.

Prominent non-linear techniques that achieve this result are Locally-linear embedding (LLE), Isomap [199], or t-distributed stochastic neighborhood embedding (t-sne) [145].

A linear technique, in contrast, produces a linear transformation of the input data in lower-dimensional space. Two well-known, traditional techniques are Principal Component Analysis (PCA) [117] and Multidimensional Scaling (MDS) [201]. PCA linearly transforms the data in such way that the variance is maximized. To derive a representation in lower-dimensional space, first, the covariance matrix of the data is computed and then the eigenvectors of the matrix, which reflect the variance. Due to the calculation of the covariances, PCA is typically applied to numerical attributes only. This is different for MDS.

Multidimensional Scaling

MDS represents the umbrella for several techniques that have the common objective to preserve pairwise distances between data records as best as possible in a planar layout [80]. In the following, I will refer to the very beginning of MDS and explain the *Classical MDS* presented by Torgerson [202] in 1958 and Gower [76] in 1966.

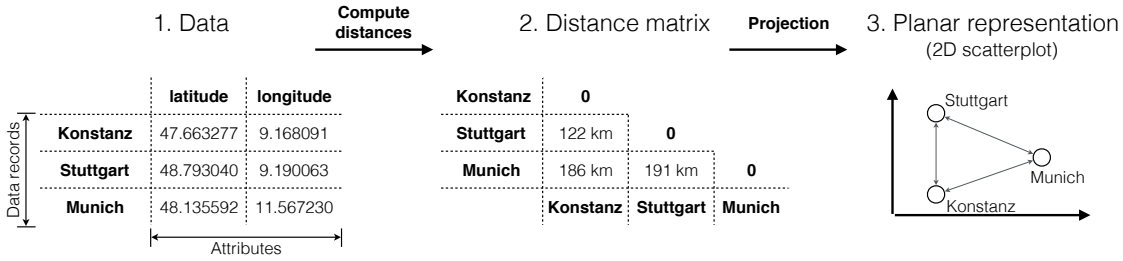


Figure 2.3: *MDS by the example of German cities. The data consists of (1) three cities and their respective geo-spatial position. Using MDS, we first calculate all (2) pairwise distance between cities. Using the classical MDS [202] approach, the data is projected to a plane. MDS aims to preserve the distances between the cities as best as possible.*

The MDS approaches follow the same workflow, depicted and exemplified in Figure 2.3. Consider different data records, each comprising multiple attributes. In this example, three different cities and their location. In the next step, the MDS approach computes the distance matrix of all pairwise distances. Based on the distances, the MDS then aims to preserve the distances, typically in a planar layout. Note, that in this example the source and the target number of attributes or dimensions are equal. MDS, however, is generally applied to transform data with many attributes to a lower-dimensional space; this can be from just a few up to several hundred attributes.

Classical MDS was first introduced by Torgerson [202] and transforms the data into a lower-dimensional space, so that $\mathbb{R}^n \rightarrow \mathbb{R}^m$ with $m < n$. MDS aims to provide a solution to the question:

> *Suppose the coordinate matrix* $\mathbf{X}$ *is not observed. Given the observed distance matrix* $\mathbf{D}$*, how to find* $\mathbf{X}$*?*

The following description of steps of the classical MDS is based on the work by Torgerson [202], Groenen and Borg [80], and the lecture notes by Cheng [1].

Let $\mathbf{D}$ be the *observed* distance (proximity) matrix of Euclidean distances derived from a $n \times p$ data matrix $\mathbf{X}$, which is *not observed*. Classical MDS is based on the fact that $\mathbf{X}$ can be derived by *eigenvalue decomposition* from $\mathbf{B} = \mathbf{X}\mathbf{X}^T$, i.e. the inner product of $\mathbf{X}$. Thus, the matrix of squared Euclidean distances

$$d_{ij}^2(\mathbf{X}) = \sum_{s=1}^{p}(x_{is} - x_{js})^2 \tag{2.1}$$

is given by

$$\mathbf{D}^{(2)} = \mathbf{1}\alpha^T + \alpha\mathbf{1}^T - 2\mathbf{X}\mathbf{X}^T \tag{2.2}$$

where α is a vector with diagonal elements of $\mathbf{B}$ and $\mathbf{1}$ is a vector of ones (length appropriate to α). We find $\mathbf{X}$ as follows. The centering matrix $\mathbf{J}$ is defined by

$$\mathbf{J} = \mathbf{I} - \mathbf{1}\mathbf{1}^T/\mathbf{1}^T\mathbf{1} \tag{2.3}$$

where $\mathbf{I}$ is the identity matrix. Multiplying the squared distance matrix $\mathbf{D}^{(2)}$ (Equation 2.2) with $\mathbf{J}$ and afterwards multiplying it by $\frac{1}{2}$, gives us

$$-\frac{1}{2}\mathbf{J}\mathbf{D}^{(2)}\mathbf{J} = \mathbf{X}\mathbf{X}^T \tag{2.4}$$

since $\mathbf{J}\mathbf{1} = 0$. The eigenvalue decomposition of $\mathbf{B}(= \mathbf{X}\mathbf{X}^T)$ is

$$\mathbf{B} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \tag{2.5}$$

where $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, ..., \lambda_n)$ is the diagonal matrix of eigenvalues of $\mathbf{B}$ and $\mathbf{Q} = [\mathbf{Q}_1, \mathbf{Q}_2, ...\mathbf{Q}_n]$ the matrix of normalized eigenvectors, this is, $\mathbf{Q}_i^T \cdot \mathbf{Q}_i = 1$. As a result, $\mathbf{B}$ can be chosen as

$$\mathbf{B} = \mathbf{Q}^*\mathbf{\Lambda}^*\mathbf{Q}^{*T} \tag{2.6}$$

where $\mathbf{Q}^*$ consists of the first p eigenvectors and $\mathbf{\Lambda}^*$ of the first p eigenvalues. A solution of the coordinate matrix $\mathbf{X} = \mathbf{Q}^*\mathbf{\Lambda}^{*1/2}$. The solution is derived based on the assumption that $\mathbf{X}$ is column centered.

Non-metric MDS Classical MDS can be considered under the general umbrella called *metric MDS*, which minimizes the stress (goodness of fit), that is the sum of squared distances. In contrast to metric MDS, the *non-metric MDS* is a ranked-based approach, which means that the ranks replace the original distances. Kruskal [134, 135], therefore, defined the stress as

[1] NTHU Lecture Notes on Applied Multivariate Analysis by Cheng, 2010:
`http://www.stat.nthu.edu.tw/~swcheng/Teaching/stat5191/`

$$Stress = \sqrt{\sum_{i,j} \left(d_{ij} - \hat{d}_{ij}\right)^2 / \sum_{i,j} d_{ij}^2} \tag{2.7}$$

where d_{ij} is the distance between samples i and j and the values of $\hat{d}_{ij}$ are the numbers which minimize the $Sress$ depending on that they have the same rank order as the distances δ_{ij} between the given objects. This means that $\hat{d}_{ij} \leq \hat{d}_{i'j'}$, whenever $\delta_{ij} < \delta_{i'j'}$ [135]. Further reading on MDS can be found in the well-known work by Cox and Cox [47].

Interactive Systems based on 2D Projections

Multivariate data consists of multiple attributes, posing a challenge to identify expressive ones that reveal interesting patterns. Results of DR are typically presented in a two-dimensional scatterplot where proximity between points indicates how similar they are. Several interactive systems have been presented based on DR methods such as MDS [47] or PCA [117], among others, which I briefly outline below.

According to Sacha et al., interaction enables exploratory data analysis of DR results and adapts to the "human needs and domain-specific problems" [178, p. 214]. The general idea is to understand either the structures in the data or the particular projection strategy. The commonality of recent works is that interaction is performed via direct manipulation with the visualization, or indirectly using control panels that show necessary parameters. First interactive approaches to DR include the pioneering work by Buja et al. [5] and Martin and Ward [214]. The authors introduced and integrated brushing capabilities into the two-dimensional depiction of multivariate data. Recent techniques that illustrate different interactive aspects to make sense of multivariate data include iPCA [111], Dimstiller [96], Brushing Dimensions [205], Data Context Map [41], and Probing Projections [189]. Let us consider one at a time. iPCA, which is similar to the work of Liz et al. [143], opens the "black box" of dimensionality reduction by the example of PCA. The authors presented a system that fosters interpretation of the PCA output by assisting the user with multiple coordinated views and a rich set of interactions. In contrast, Dimstiller, defines workflows guiding the user to analyze dimensionality reduced data. The work of Turkay et al. [206] goes one step further and provides iterative visual refinement of the results. The technique Brushing Dimensions proposes a visualization model that enables the interactive analysis of the data attributes and values in a joint, iterative fashion. The common theme of techniques is to generate more insight about the structures in the data, in particular by inspecting the data attributes and values. As a result, the Data Context Map proposes to layout the data in the context of their attributes. Going one step further, Probing Projections propose a comprehensive set of interactions to enable comparison between data points and clusters given the underlying attributes.

In contrast, so-called *magnet metaphor*-based approaches depict a two-dimensional projection but fix the positions of the attributes in a radial manner. Data records are then attracted by the attributes with different forces, according to the values. These techniques belong to the family of Star Coordinates [120] and have been continuously refined in recent years. Well-known techniques include the works by Yi et al. [219], Cheng and Mueller [40], and

Zanabria et al. [223], among others [88, 138]. All these techniques are based on the same core principle but differ by means of interaction and enhanced technical aspects. For example, the technique iStar [223] uses different length encodings for the attributes, compared to Dust & Magnet [219], where all "magnets" (attribute positions) can be positioned individually and freely. Magnet-based approaches pose a real alternative to projection techniques such as MDS, however, also need further investigation regarding readability and interpretability. One notable drawback is the effective representation of data records regarding the attribute emphasis. It becomes challenging to read the visual depiction when taking many data records and attributes into account. The global, as well as local structures, become blurred, impairing the ability to draw meaningful conclusions.

Additional interactive approaches can be found in the surveys of Liu et al. [142] or Sacha et al. [178]. Projections were also heavily applied for the visual analysis of data subspaces because they provide an effective at-a-glance overview of the most salient global structures. Chapter 3 provides a comprehensive review of related work and introduces a novel visual approach to interpret patterns among subspaces.

2.3 Overview-Preservation in Large 2D Spaces

Overview-preservation is the umbrella term for visualization techniques that enable to inspect an object of primary interest in full detail, and at the same time preserve the context, or overview, of objects that are not of primary concern. A single display combines overview and detail [33]. Such techniques are typically used in huge visual information spaces like they are found in projections of multivariate data. A requirement of overview-preserving techniques is the dynamic adaption to the user interactions. This means, the visual interface needs to continuously adapt to the performed user interactions, including zooming and panning operations, which shift the viewport.

According to Hornbæk and Hertzum [89, p 509], *"a key goal of many information visualizations is to provide a compact representation of the information space so as to assist users in thinking about and navigating the space"*. The authors note that the notion of overview can hold different meanings. For example, according to Spence [188], overview enables the rapid, pre-attentive qualitative awareness of one aspect of the data without cognitive effort. This refers to gain an overview of the information space. Others see the overview as an interface component [89]. In terms of this thesis, I refer to the first viewpoint, the conceptual overview-preservation in large information spaces.

For the purpose of overview-preservation, a variety of techniques have been presented in the past. However, the design space is huge, which is why it is still ongoing research. In this thesis, I present novel techniques based on the family of *off-screen visualizations*. Therefore, I briefly delineate the best-known and used overview-preserving techniques, and motivate the need for off-screen visualization.

There exist four well-known approaches to the overview problem: *Pan-Zoom + Overview*, the *Constrained Distortion lens*, *DragMap*, and *Classical Off-screen*. To delineate these approaches, I make use of the *space-scale diagram* representation which was introduced by

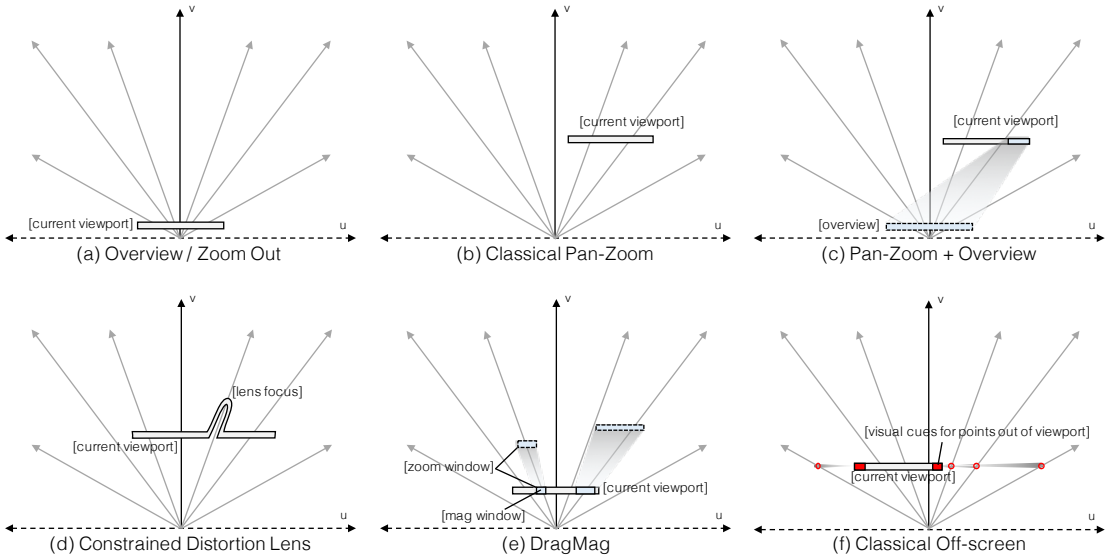


Figure 2.4: *Categorization of overview-preserving techniques based on space-scale diagrams [69] and the work of Pietriega et al. [170]. Space-scale diagrams are a conceptual representation of zooming and panning operations in the viewport. The u-axis and v-axis make up the space, in which the user can interact. Each ray (gray arrows) represents, for example, a data point and its relation to all others when zoomed. (a) is the classical overview, all data is presented in one view. When the user performs zooming and panning operations to inspect data in detail, (b) the viewport moves along the v-axis. All rays that do not intersect the [current viewport], are not visible in this view. While zoomed, the classical (c) overview interface creates a visual inset in the current viewport. The (c) distortion lens applies a selective lens in addition to zooming and panning. (e) DragMag interfaces show visual insets of different zoomed parts. In contrast to all techniques, the (f) classical off-screen visualization projects only the data points to the viewport via visual cues. Note that off-screen visualization is the only data-driven technique that changes the rendering of the data points when shown in the viewport, compared to the purely image-based techniques (c) to (e).*

Furnas [69] and used to explain the concept of multi-scale environments. Figure 2.4 (a) depicts an example of such a representation. The u-axis represents the 1D spatial dimension, and the v-axis is the scale dimension. The six gray arrows represent six data points. In (a), the viewport is placed in such manner, so that all data points are visible within the viewport. This means, no panning or zooming (scaling) is applied. Figure 2.4 (b) shows the result of zooming and panning operations; the viewport is zoomed and panned in $+u$ direction, and only two data points are visible. The higher the scale is, the more the gray arrows spread. In (b), four data points are not visible anymore, to which I refer to as the *overview-preservation problem*. Figure 2.4 (c) - (f) further delineate best known overview-preserving techniques.

Pan-Zoom + Overview This set of techniques is commonly known under the notion of *Overview-and-Detail*. The basic idea is to place a visual inset on the viewport that shows the overview (no scaling or panning) and additionally the viewport position when zoomed in (compare to Figure 2.4 (c)). Both representations, overview and detail, are displayed in distinct presentation spaces [44]. There come two shortcomings with this approach. First, it overlays with the visualization, covering potentially interesting information. Second, Javed et al. noted that "overview-and-detail divides user attention between viewports, and scales poorly to high magnification ratios" [110].

Constrained Distortion Lens Distortion lenses come in different variations but share the same principle. These approaches are known as *Focus-plus-Context* techniques (compare to Figure 2.4 (d)). The user picks a region (radial or rectangular) and magnifies this area in image space (focus), while the surrounding – the context – remains in place [34]. Despite improvements, the area between focus and context, also known as the drop-off area, suffers from extensive distortion. Cockburn et al. further state that "distortion-oriented displays are likely to impair the user's ability to make relative spatial judgments, and they can cause target acquisition problems" [44, p. 2:27].

DragMag Compared to overview-and-detail interfaces, DragMap operates vice versa. The user selects multiple regions she wants to see in full detail. These regions are displayed as visual overlays and can be freely positioned [215]. There is no distortion, like it is the case for Focus-plus-Context interfaces. However, it is likely that one loses the overview when dragging multiple focus areas around.

Classical Off-screen Off-screen visualization is data-driven and therefore differs from all other techniques. Cockburn et al. [44, p. 16] define this type of visualization as follows:

> *"Cue-based techniques [...] modify how objects are rendered and can introduce proxies for objects that might not be expected to appear in the display at all. They can be used in conjunction with any of the aforesaid schemes, and are typically applied in response to some search criteria."*

The main characteristic, and advantage, is that the overview can be adapted to the task and data at hand. This is of particular interest for multivariate data, because they comprise many attributes. Also, it poses a solution for the so-called *Desert Fog* problem, which are unknown empty regions in the information space [118]. Being unaware of such regions causes inefficient navigation through panning operations, in particular when zoomed in. However, the assumption that the user is only interested in the data overview effects the perception of the image-based overview, because the data is not embedded anymore into the information space. Thus, a main challenge is to preserve the overall data topology.

In the following, I briefly discuss scalable user interfaces, also known for semantic zooming, which are the foundation for overview-preserving interfaces. Then, I provide detail and related work for the two state-of-the-art techniques, Overview-and-Detail (Pan-Zoom + Overview) and Focus-plus-Context (Constrained Distortion Lens).

2.3.1 Scalable User-Interfaces

Scalable user-interfaces follow the idea of the *Visual Information-Seeking Mantra*, which was coined by Shneiderman [184, p. 336]: "overview first, zoom and filter, then details on demand". This process can be executed over and over again and is not restricted to a single

interaction. Visually, scalable user interfaces built a basis for the application of Shneiderman's mantra to large information spaces [69]. Figure 2.4 (b) depicts an example of zooming and panning operations. The general idea is to dive into a region of interest and focus on relevant details. Typically, the visual representation adapts to the zooming operation, either by geometrically scaling the representation or by presenting different levels of information. For example, when zooming, a previous rather small region is transformed to a comparatively large region, which allows presenting more information than before, namely *semantic zooming*.

One of the first multi-scale interfaces, Pad, was presented by Perlin and Fox [167]. Pad is based on the idea of spatial metaphors, the infinite two-dimensional plane (information space), where objects are organized geographically. With the use of so-called Portals, Pad enables an endless view into the display. Pad also introduced the idea of semantic zooming, which was further developed by Elmqvist and Fekete [56] using a hierarchical model. Semantic zooming organizes the data based on different levels of abstraction. The presented information is then dynamically exchanged depending on the zooming level, either refined when zooming in, or aggregated when zooming out. The successor of Pad, Pad++, introduced smooth zooming capabilities and generalized the idea of multi-scale interfaces [16]. The authors created a framework with the goal to facilitate the development of such interfaces. Pad++ was the basis for many techniques and visualizations based on multi-scale interfaces, which have become a standard in today's applications.

2.3.2 Overview-and-Detail

Maintaining orientation in large information spaces is a tedious necessity. This urge asks for a navigational help, a visual cue, which indicates the current location in space after having performed multiple zooming and panning operations. Building upon scalable user interfaces, an overview representation of the information spaces is depicted as a visual inset [14]. Consider an inset as added view that has the sole purpose to show the overview of the entire information space. Typically, the overview comes with an additional visual cue, such as a small rectangle, that indicates the area which is navigated in detail. For example, we can improve the navigation of a geographic map on street level, by simultaneously showing the overview map together with an indication of the position [90]. Jerding and Stasko further presented different application examples for Overview-and-Detail interfaces [112]. For instance, writing text or programming code may result in very long files. To quickly navigate to other regions, an overview is added, which depicts the densities in the entire file line by line. Jerding and Stasko also apply this method to large time-series data, so that one can inspect a certain moment in time in full detail, while a minified version of the entire dataset is additionally visualized. Plaisant et al. [172] go one step further and enhance the detailed exploration with multiple overview visualizations that show different zooming levels. However, one may be overwhelmed by the information load, which is why Javed et al. [110] presented PolyZoom, a technique that enables the user to build an interactive Overview-and-Detail hierarchy.

2.3.3 Focus-plus-Context

To overcome the drawbacks of Overview-and-Detail, Focus-plus-Context techniques integrate focus and context seamlessly into a single view. Hence, Focus-plus-Context is also referred to as distorted view, or constrained distortion lens. Apperley et al. [9] presented one of the first graphical Focus-plus-Context systems, the Bifocal Display, which is depicted in Figure 2.5. This visualization comprises three continuous viewports: The center corresponds to the focus, the left and right areas to the preceding and succeeding context.

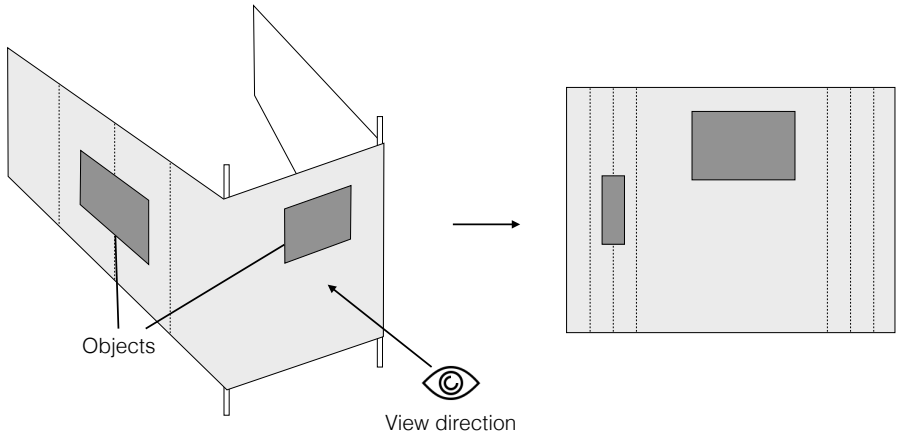


Figure 2.5: *Bifocal Display approach according to Apperley et al. [9]. The general idea is to wrap a wide continuous information space around two uprights, so that the entire information space is visible. Wrapping the space around two uprights has the effect similar to a magnifying glass: The user inspects the center part at full detail, while the left and right parts are distorted, but still visible.*

Figure 2.5 depicts the Bifocal Display principle. You can think of it as a wide information space, which is compressed into the display as a whole. To do so, Apperley uses a metaphor: The information space is wrapped around two uprights. When projected to 2D, the entire information space is squeezed into the display. However, the center area is undistorted, and the user inspects it at full detail, while the left and right area are distorted.

Building upon the Bifocal Display technique, many new techniques and applications have been built. For example, the Perspective Wall [149] maps the Bifocal Display concept, depicted in Figure 2.5 (a) into an interactive 3D space. The concept was also transferred to 2D distortions, like, for example, shown by the Table Lens [173], or the DateLens [15]. Other techniques embrace the idea of folding the information space in 3D space; this means that unimportant areas are folded towards the back of the 3D space. Representative techniques are Melange [57] and SpaceFold [31].

Furnas [68] later introduced the degree-of-interest (doi) function as the theoretical foundation for Focus-plus-Context systems. This function operates in data space and determines objects that are more interesting than others. Irrelevant objects are removed from the display. Based on the theoretical description by Furnas and the Polyfocal Projection approach by Kadmon and Shlomi [119], fish-eye distortion creates a seamlessly integrated hemispherical

image [179]. Results of this method are, for example, the Document Lens [174], which enables the user to read portions of text at full detail, while the overall page layout is preserved in a distorted manner. Additional examples include the Elastic Presentation Space by Carpendale and Montagnese [34] and Metro Maps [211]. The transition between focus and context (drop-off area), however, poses the main challenge to effectively interpreting spatial relations [44]. Despite recent improvements [35, 83, 169, 171], the switchover between focus and context remains. However, for discrete information spaces, such as system diagrams, SchemeLens [45] presents a promising approach. SchemeLens is a vector-based fisheye lens for network diagrams, which magnifies components and reduces the edge lengths given structurally-similar layouts.

Furthermore, Büring et al. [30] proposed a fish-eye distortion tailored to the navigation of scatterplots, whose result is similar to the Bifocal Display technique. The idea is to select and magnify a rectangular region in the viewport to such extent that the surrounding regions and the distances between data points are distorted. However, the overall topology of the data points is discernible, which is also due to the step-function that defines the drop-off area. This means, there is no gradient between focus and context. This approach goes beyond distorting the entire space to uncover the inherent, yet occluded structures [124], and enables the user to focus on areas of interest while preserving the structural overview.

2.4 Summary and Relevance

The present thesis integrates into two parts of the DR pipeline, which is depicted in Figure 1.1: the feature space and the visualization using novel interactive visualizations. The main contributions of this thesis are embedded into the family of pattern identification and interpretation in multivariate MDS projections, and the navigation of large spanned information spaces.

Interaction is the common theme in this thesis. It enables users to identify and navigate multivariate patterns that may not be visible at first sight. Also, interaction enables the user to interpret patterns by steering the underlying data model and, thus, causing results to adapt to the user's incentive dynamically. Interactive model steering acts as means to either verify hypotheses or to form new hypotheses about the data.

I briefly outlined well-known interactive approaches for visualizing multivariate data based on 2D projections. However, it is still ongoing research how to make sense of such multivariate projections, on which I follow up in this thesis. In particular, I investigate the interpretation of patterns in single and multiple views on the data in Chapter 3, as well as using projections to make sense of temporal multivariate data in Chapter 4.

Furthermore, I discussed the two most frequently used state-of-the-art techniques for overview-preservation in large information spaces: Overview-and-Detail and Focus-plus-Context. In particular, the approach by Büring et al. [30] is highly relevant for developing topology-preserving off-screen visualizations. In Chapter 5, I introduce a dedicated border region aiming at preserving the data topology. If only the data points are visualized like it is the case in standard scatterplots, the result is similar to the image-based approach by Büring et al. However, changing the rendering of off-screen objects characterizes off-screen

visualization. Visually, I build upon the approach of Büring et al. and extend it with respect to the visualization of off-screen information. This means, I adapt the rendering of off-screen content for a data-driven overview. Because focus-plus-context techniques are image-based, they are not suitable for data-driven tasks, where the visualization needs to adapt to certain data characteristics, such as the density or the multivariate nature. Therefore, I investigate the data-driven design space of off-screen visualizations, which are characterized by their ability to adapt the rendering based on the data and the task at hand.

My developed methods and findings are based on classical MDS or general spatial data but are generally valid and applicable to any other projection technique.

Part I

Identification & Interpretation of Multivariate Patterns in Projections

3

Visual Pattern Analysis and Interpretation in Multivariate Subspaces

Contents

MULTIVARIATE *projections are hard to interpret, in particular for domain experts not familiar with machine learning or advanced statistics*. This statement reflects the common understanding among many visualization researchers. Initially, this claim is held true, because the interpretation of a multivariate projection depends on whether a pattern can be identified and whether it reflects the mindset of the analyst. Patterns, however, typically occur in different subspaces of the data (attribute subsets of the data), posing a challenge to find them in general, and subsequently interpret them since they can mean something different in other subspaces. In recent years, interactive visual methods have been extensively researched for their ability to improve transparency and ease the interpretation. These methods have primarily been evaluated using case studies and interviews with experts trained in DR. This chapter, therefore, first explores whether it applies, that domain experts have difficulties

interpreting multivariate projections. I describe a phenomenological analysis investigating if researchers of a Law Enforcement Agency (LEA) with no or only limited training in machine learning or advanced statistics can interpret the depiction of dimensionally-reduced crime data, and what their incentives are during interaction.

The study, among other things, reveals that the domain experts manually explored different subspaces to seek for patterns. This behavior was expected, because some attributes are, simply put, not relevant for building meaningful patterns. Also, some attribute subsets may be more feasible than others on the domain knowledge of the experts. It is a tedious task to manually explore the subspaces for meaningful patterns without prior knowledge about the data. On these grounds, different subspace analysis methods have been proposed with the aim to ease this task. However, many of these analysis methods produce an abundant amount of patterns, which often remain redundant and are difficult to relate. Creating effective layouts for comparison of subspace pattern remains challenging. Therefore, I also introduce *Pattern Trails* in this chapter, which is a novel approach for visually ordering and comparing subspace patterns. Central to this approach is the notion of *pattern transition* as an interpretable structure imposed to order and compare patterns between subspaces. The basic idea is to visualize projections of subspace patterns side-by-side, and indicate changes in adjacent patterns by a linked representation (transitions). I demonstrate the usefulness of this approach by application to several use cases, indicating that data can be meaningfully ordered and interpreted in terms of pattern transitions.

This chapter is based on [109] and [105]:

Interpretation of Dimensionally-Reduced Crime Data: A Study with Untrained Domain Experts D. Jäckle, F. Stoffel, S. Mittelstädt, D. A. Keim, H. Reiterer. *Proc. Int. Conference on Information Visualization Theory and Applications, 2017.*

Pattern Trails: Visual Analysis of Pattern Transitions in Subspaces D. Jäckle, M. Hund, M. Behrisch, D. A. Keim, T. Schreck. *IEEE Conference on Visual Analytics Science and Technology (VAST), 2017.*

3.1 Introduction

Information is collected at large-scale in all areas of our day-to-day life: occurred crimes, statistical surveys of natural disasters or inhabitants, rankings of public institutions, or any tabular data that consists of multiple observations and attributes. The main task in understanding such multivariate data is to identify and interpret relevant patterns like dense groups (clusters), outliers, or correlations. Recent advances in machine learning propose DR to transform the data to a lower-dimensional space, preserving the main structure of the data. Results can be depicted in a 2D scatterplot, in which proximity between points indicates similarity. This abstract representation of DR results enables exploration of the structure but

brings in challenges about interpretability of the visualization and how the different attributes are reflected in the lower-dimensional representation. The common understanding among visualization researchers is that this is held true, in particular for domain experts not trained in advanced statistics or machine learning.

To this end, several interactive systems have been presented in support of domain experts. They typically build on top of a 2D depiction of results and enhance the interpretation via different additional interactive visualizations [214]. While the focus lies in improving the interpretability of DR results for domain-specific tasks, only little evidence is given that domain experts are indeed able to interpret the depiction of the data projection. State-of-the-art systems were evaluated in two different ways. Either by means of use cases and application examples or by a user study. The user studies, however, were carried out with domain experts specifically trained in DR [181] or with users unrelated to the field [189]. I argue that domain experts related to the data and tasks are differently motivated in pursuit of their goals compared to participants unrelated to the presented data and tasks. This effect is further amplified because untrained experts need first to learn how to read the depiction of DR results before they can interpret them. In conclusion and to the best of my knowledge, DR results have not been studied for domain-specific tasks including domain experts.

In the following, I report on a qualitative user study, driven by the question: *Can untrained domain experts use their domain knowledge to interpret and steer the visual depiction of a data projection?* I reached out to data analysts of a German LEA not trained in advanced statistics or machine learning. In research projects, the University of Konstanz has already gained great insight into their everyday work, typical tasks, and the challenges imposed by the huge amounts of data. The data analysts are eager to identify patterns among various data sources they have access to in order to leverage resources, identify suspects, relieve wrongly accused individuals, and more. So far, manual data analysis dominates their everyday work, for example, by creating tabular views of data, which enables them to compare different cases or data sources in the light of a specific information need. This is also held true for applications such as the comparative case analysis, where similarities and correlations among crimes are subject of work in a one-to-many comparison [1]. For example, a correlation between a crime *category* and *districts* in a subset of the data can be detected. However, a correlation among crime *category*, *district*, *time*, *description*, and *day of week* is demanding without any automated data analysis and visual support, even in small subsets of the data.

The study shows that the domain experts of a LEA effectively adapt to abstract representations of the data if they are familiar with the tasks and the type of data. Also, the study reveals that the experts performed a manual subspace analysis. This means that they searched for relevant patterns in specific attribute subsets, namely the subspaces. Particularly in multivariate data with many attributes, patterns may only be found in smaller subspaces and would get lost when considering all attributes at once [17]. However, one cannot assume that patterns will be similar across different subspaces. Rather, one can expect they may be structurally different in different subspaces, posing the challenge to identify, interpret, and compare them visually. The performed manual subspace analysis by the domain experts is a tedious task. This asks for Visual Analytics methods to support the effective visual and

automatic analysis of subspaces.

Automatic subspace analysis and clustering methods provide sets of possibly interesting patterns and subspaces. While such methods drastically reduce the amount of possible attribute configurations by ignoring subspaces with a high attribute and pattern overlap, they entirely leave out the analyst [131] from the initial search process. They typically provide no hints on an appropriate ordering or on relationships among the reported subspaces. Hence, the amount of considered subspaces can, in fact, be reduced, yet it is challenging to explore the data to find interesting patterns.

In recent years, several approaches have been presented to explore multivariate data visually, and in particular patterns in subspaces. Parallel Coordinate Plots (PCP) [99] present a key technique for multivariate data analysis. Besides researched challenges such as axis re-ordering, one axis merely corresponds to one attribute making it a difficult task to compare different attribute combinations with each other. A prominent means of making subspaces visually accessible is the application of scatterplots. Scatterplot Matrices (SPLOMs), for example, enable pairwise comparisons between attributes, effective for small to moderate sized data sets [181].

Techniques using DR typically present one view on the data (like, for example, in the study with the domain experts) or present all subspaces via small multiples [203], making it challenging to identify interesting patterns and trace their meaning in different subspaces. Even if applying subspace analysis before the visual exploration phase, subspaces can still be redundant and too many to identify relevant patterns. Automatic approaches require the user to adapt the analysis model based on the task at hand to retrieve relevant views on the data. Therefore, I introduce a technique called *Pattern Trails*, that aims to address the unresolved question: *How to identify and relate interesting patterns among multivariate subspaces, using interactive visual exploration?*

In this chapter, I first report on the user study and results perceived through the eyes of the domain experts. Then, I present Pattern Trails, a Visual Analytics approach to find and interpret patterns across subspaces of multivariate data. In the context of the present thesis, this chapter contributes to the identification and interpretation of patterns in multivariate projections.

3.2 Related Work

The common theme of this chapter is the interpretation of multivariate data using DR – in particular multivariate projections. Therefore, I first report on results of an interpretation study carried out with domain experts, and then present pattern trails for the automatic identification and interpretation of patterns among subspaces. Pattern trails supports the visual analysis of subspace patterns by ordering and relating patterns, hence helping users to explain structural changes of patterns among different subspaces. In order to motivate these two parts, I outline approaches for the visual analysis of mixed datasets and investigate how related work studied the interpretation of DR results. Furthermore, I discuss related work from subspace search and visualization, which are subtopics of high-dimensional data analysis and have recently gained research interest.

3.2.1 Visual Analysis of Mixed Datasets

Real-world datasets typically comprise different data types, such as nominal, ordinal, or any quantitative data type. Various approaches have been proposed considering so-called mixed datasets, which are datasets that comprise numerical and categorical data. Visual analysis of mixed datasets aims at unifying different data types and reflect their respective nature [19] and, thus, are related to the integration of diverse data types. For example, Parallel Sets [130] shows the frequencies of categories instead of individual data records with a restriction to the amount of categories that can be visualized at the same time.

For the combination of numerical and categorical data, Rosario et al. [175] and Johansson and Johansson [116] proposed to quantify categorical data; results can then be visualized using, for example, Scatterplots or Parallel Coordinates [98]. In contrast, Bernard et al. [19] identify relevant subgroups in mixed datasets by abstracting the data into bins. Another method, named the Contingency Wheel [3], allows to analyze associations of contingency tables using a radial representation interactively; associations are shown by connections between respective categorical segments.

A major issue is that these methods only consider numerical and categorical data. Real-world data, such as crime data, however, consists of numerical, textual, and categorical data. Making such data comparable is challenging, in particular for combinations of values. Towards incorporating multiple data types, DR techniques are worth a look. Similar to PCA [117], Correspondence Analysis [18] (CA) applies to categorical data and projects the values to 2D space using χ^2 statistic. Going one step further, MDS [47] attempts to preserve the distance between any two data entries as good as possible. MDS, therefore, requires a distance or dissimilarity matrix as input allowing to maintain the relations between any values, whose dissimilarities can be expressed numerically. Building on this, the Gower Metric [77] unifies similarity measures for different data types, based on which users can steer and explore the result.

3.2.2 Interpretation of Multivariate Projections

Relations between dimensions in multivariate data projections are complex and require the human in the loop to make sense of [178, 219]. State-of-the-art techniques assume that for users not trained in advanced statistics it is particularly difficult. Ward and Martin [214] and Buja et al. [5] presented interactive systems for multivariate data inspired by several issues and combinations of interactions. Recent work can be categorized into two evaluation types: (1) user studies and (2) case studies, application examples, or other. An overview is given in Figure 3.1. Various systems have been presented in the past claiming to improve the understanding for users or domain experts through interactive manipulation. While there is no doubt that these approaches improve the understanding of multivariate data, only a few approaches have conducted a user study to show the impact. They typically showcase their approach using application examples [131, 160, 182], use cases/case studies [62, 96, 115, 143, 205, 207, 222], or other [111]. In contrast, only few approaches conducted a user study. For example, researchers propose systems to help better understand DR results, in particular, the distance function [26, 219]. Sedlmair et al. [181] went one step further and provided guidance for

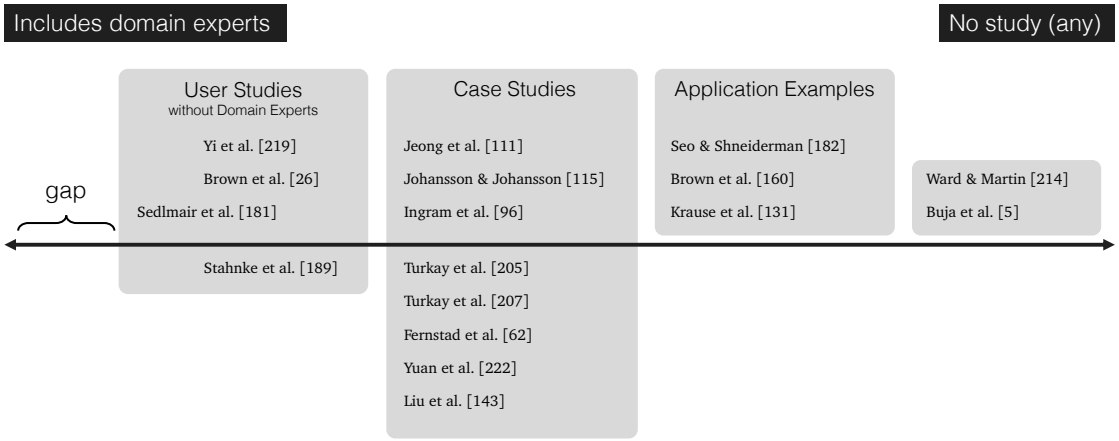


Figure 3.1: *Towards the evaluation of multivariate projections. This depiction gives an overview of well-known approaches that make use of projections as means to interpret multivariate data. The works are categorized by the type of evaluation they conducted. Only a few approaches were presented that carried out a user study, and only one included domain experts [181]. However, they were trained in machine learning. In this chapter, I address the following gap: A study with domain experts who are not trained in machine learning or advanced statistics.*

DR representation techniques, however, the study was carried out with two users not related to the data, but trained in machine learning and advanced statistics. Another evaluation was presented by Stahnke et al. [189]. The authors evaluated the interaction with DR results but did not involve domain experts having a certain incentive and mindset about the data.

Krause et al. [131] find very clear words for a situation that, to the best of our knowledge, has not been proven yet: they assume, that for domain experts not trained in advanced statistics and machine learning, it is tough to interpret DR results. This is a strong statement I pick up and investigate in this chapter with the help of a small group of data analysts of a LEA who analyze raw data tables for correlations and outliers on a daily basis.

3.2.3 Subspace Search and Visualization

So far, scatterplots are practically the first choice to depict the results of DR. Thereby, the proximity between points indicates how similar they are. Computing similarity or proximity between data, or reducing their dimensionality, is harder to do as more attributes are introduced. Typically, the more attributes are introduced, the less discriminative the projection result is. As a result, it is challenging to identify patterns, also known as the *curse of dimensionality* [17]. Kriegel et al. reason that as the number of attributes grows, the relevance of attributes can differ for different patterns [133]. In other words: Not all attributes contribute to the existence of a pattern, but relevant patterns exist in different combinations of attributes, namely the subspaces.

Recent advances in visualization build on top of automatic subspace analysis [133] or clustering approaches [165], and make the result accessible for exploration. In doing so, the visualization of subspaces is either applied to the attribute space of the data, the object space, or to both combined. The attribute space refers to the general statistical analysis of the attributes that comprise a given subspace. For instance, the approaches of Krause et al. either

provide a sorted frequency-based view on the attribute values [131], or enable finding relevant attributes (features) based on feature rankings [132]. In contrast, object views present the entities of the data set and allow, typically in combination with analysis possibilities including the attributes, to foster understanding of single subspaces or patterns. Liu et al. [144], for example, provide a comprehensive interactive view on the projection. Other approaches combine the object and the attribute view on the data [95, 196, 198, 222, 226].

Purely object-driven approaches are related to the field of projection pursuit [94], where the overall goal is to find significant projections of multivariate data; these are projections where points build unique patterns. Examples include the work by Anand et al. [4] and Lehmann and Theisel [139]. Related works on multivariate cluster visualization [32] and comparison [25] do not build on subspace analysis. They focus on the presentation of relevant features to build clusters and enable comparison. There are further related approaches, but in summary, they do not scale for many subspaces.

The commonality between approaches above is, that they either are overstrained by the number of subspaces, miss support for pattern ordering and comparison or produce an abundant amount of patterns, which often remain redundant and difficult to relate. Existing systems typically impose small-multiple [203] alike views on the data with some ordering of the views, but do not provide an ordering and linking dedicated to comparing subspaces about which data points change and which remain stable across the subspaces. In this chapter, the visual subspace analysis is improved by imposing meaningful ordering on subspaces, used to group similar subspaces and help interpreting pattern transitions. Also, I directly foster the comparison between subspaces by a linked representation and a systematization of occurring pattern transitions.

3.3 Interpretation of DR Data: Phenomenological Study

As aforementioned, the abstract representation of multivariate data by projections enables exploration of the structure but brings in challenges about interpretability of the visualization and how the different attributes are reflected. In this section, I report on a qualitative user study with personnel of a LEA not trained in advanced statistics. The study builds on top of the well-known *routine activity* [136], that models crimes by using dominating attributes for state-of-the-art intelligence data analysis: *place*, *time*, and *occasion*, whereby the *occasion* refers to the crime opportunity expressed by the description or category of a crime. In this study, I also made use of additional attributes to challenge the interpretation of the relevant depiction further. Based on the publicly available crime data collected in the San Francisco Bay Area [1], I created four coordinated tasks that include different aspects of the routine activity like, for example, the correlation between time and locations. Figure 3.2 illustrates the projection of the routine activity for the San Francisco Bay dataset.

Each task consists of one question intending to lead the domain expert to new insight in the data. Crime data comprises various data types, which is why we cannot rely on conventional interactive dimensionality reduction tools. In preparation for the study, I

[1] SF OpenData: `https://data.sfgov.org/`

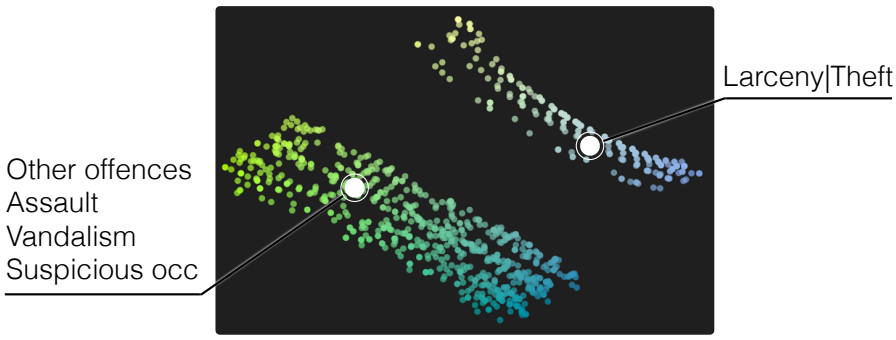


Figure 3.2: *Planar projection of 1000 crime reports collected over one week in the San Francisco Bay Area. This projection reflects the routine activity [136] and considers the attributes* place*,* time*, and* occasion *(category). The projection reveals two clusters. One cluster contains only crimes labeled as* Larceny/Theft*; these are visually separated from all other categories. One problem arising is, that even if considering only three attributes, we have yet problems interpreting the effect of* place *and* time*, because they do not cause any separation. This is the starting point for further exploration.*

developed a prototype which implements the Gower Metric [77] and carries the key data types throughout the entire analysis process. The Gower Metric computes the distance between two multivariate data entries by unifying the pairwise distances that may be tied to different similarity functions due to mixed data types. The prototype allows to steer projection parameters and further provides a minimum set of interactions meeting the visualization tasks *identification*, *comparison*, and *summarization* [23] of projected data objects.

3.3.1 Visualization Prototype: Integration of Mixed Data Types

Crime data comprises different data types making it challenging to interpret DR results; these are numerical, textual, and categorical data types. For this reason, I created a visualization prototype that fuses different data types and provides a minimum set of interactions to support the interpretation. One particular feature of this prototype is the close link between data type and interaction concepts. Each considered attribute and associated similarity function can be interactively changed at runtime with a direct effect on the depiction of the projection. Figure 3.3 outlines the structure of this section. First, I describe the integration of weight and similarity regarding the attribute and data type, and then I provide an overview of applied interaction concepts.

Weighting & Similarity

DR techniques preserve the relevant structure of the data, which is typically represented in lower-dimensional space using the concept of proximity or similarity between data objects. The application of DR techniques considers the similarity between objects based on all attributes unless told otherwise, not necessarily reflecting the incentive of the domain expert. Therefore, I include the known concept of attribute-wise weighting. This way, the domain expert can define the impact of each single attribute allowing to concentrate on relations and thus patterns that only occur in certain attribute combinations, namely the

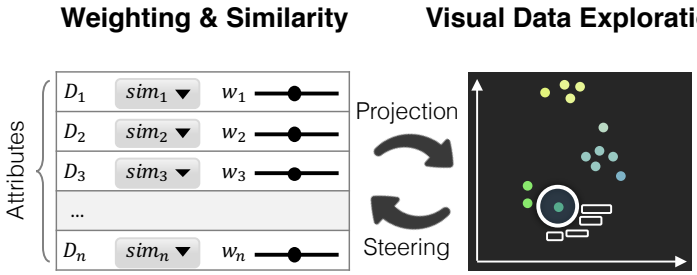


Figure 3.3: *A multivariate dataset comprises n different attributes. In the first step, the system automatically assigns a similarity function, based on the data type, and weight to each attribute. The weight is set to the value* 1 *in the possible range* $[0;1]$*. This means, every attribute is fully considered. The data is then projected to 2D space allowing the domain expert to explore the data, who can adapt the weights and similarities according to the findings.*

subspaces. Furthermore, multivariate crime data comprises different data types between which similarities are expressed differently. State-of-the-art DR techniques are typically based on similarities and distances between solely numerical or categorical values. However, crime data comprises different data types beyond merely numbers or categories. Gower's idea to address this issue is to use similarity functions in the range $[0;1]$ for each attribute D_i and then to aggregate the results. The system computes the pairwise distance between two multivariate data entries A and B based on the Gower Metric [77]:

$$dist(A,B) = \frac{\sum_{i=1}^{|attr|} sim_i(A_i,B_i) \cdot w_i}{|attr|} \tag{3.1}$$

The distance between A and B is computed by iterating all attributes (from $i = 1$ up to the number of attributes $|attr|$) and calculating the respective distance between two attributes A_i and B_i. Using the user-defined similarity function, the i-th similarity between the i-th attributes is computed. sim_i refers to the similarity function assigned to the i-th attribute. Finally, we multiply the result with the user-assigned weight w_i and build the average by dividing the overall result by the number of attributes $|attr|$.

The Gower Metric is applied together with MDS [47] that enables exploration of the global data structure. I include the Gower Metric in the prototype and enable to change the weight and similarity of each attribute at all times with direct impact on the result. Crime data consists of numerical, textual, and categorical data. **Numerical** values include any numerical data type: integers, floats, timestamps, etc. I compute the similarity between numerical values V_1 and V_2 using the Euclidean distance:

$$sim(V_1,V_2) = |V_1 - V_2| \tag{3.2}$$

Note that the range of computed similarity values between numerical values may vary. Therefore, numerical values need to be normalized using rescaling before computing the similarity.

Textual attributes comprise continuous text abstracted from sets of documents. The similarity between two documents is typically computed using the cosine similarity in vector space [187].

To do so, the documents are transformed into vector space according to a bag-of-words model and the resulting vectors v_1 and v_2 are then compared using the cosine similarity:

$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \cdot \|v_2\|} \tag{3.3}$$

Categorical attributes are typically characterized by unordered textual values that express a category. I apply Iverson Brackets [78] to compute the similarity between two categorical values V_1 and V_2:

$$sim(V_1, V_2) = [V_1 \neq V_2] \tag{3.4}$$

If two categories are the same, the similarity is 0 and 1 otherwise. The similarity can be seen as a synonym for distance since our aim is to compute a distance matrix as input for the MDS.

In the following, I describe that users can switch between a text label and histogram representation. In preparation for this concept, the data needs to be quantified. Numerical values are binned to value ranges, and categorical values are binned according to the categories. For textual values, I use the result of the bag-of-words-model and assign the frequency; the system shows a frequency distribution of extracted terms. To bin textual values, I compute the cosine distance between term vectors to the empty string and bin the results.

Visual Data Exploration

So far, the user can control the attribute-wise weighting and similarity function. To let the user interpret and make sense of the presented depiction of DR results, I provide a set of interaction techniques that consider the given attribute-wise information.

I propose to combine visualization and interaction with attribute-wise information allowing users to perform the low-level tasks in an explorative setup: identify, compare, and summarize projected data objects [23]. To ease the entry point to exploration, the system enables panning and zooming, and double encodes the implicit relations in the data using color [51]. Double encoding significantly helps to distinguish between patterns or point clusters, even if they seem to overlap; when overlapping, a color gradient reflects the separation. The color mapping is perceptually linear, thus supporting analysis of patterns in multivariate data spaces [156]. A remarkable result of this method is depicted in Figure 3.2.

To interactively tackle the progressive tasks from identification to comparison to summarization, I following describe three interaction concepts adapted to the exploration of multivariate data. For the identification and comparison of objects, I provide an adaptive tooltip as well as an interactive lens. For comparing and summarizing data objects, I provide a fingerprint matrix that encodes distributions on a per-attribute basis.

Tooltip and Content Lens I distinguish between two types of visual representations for the abstraction of different data types: histograms for quantitative data and weighted text labels otherwise. This decision results from the data itself. In multivariate crime data, we encounter text, different types of numbers, and categories. In general, we can use text labels to show

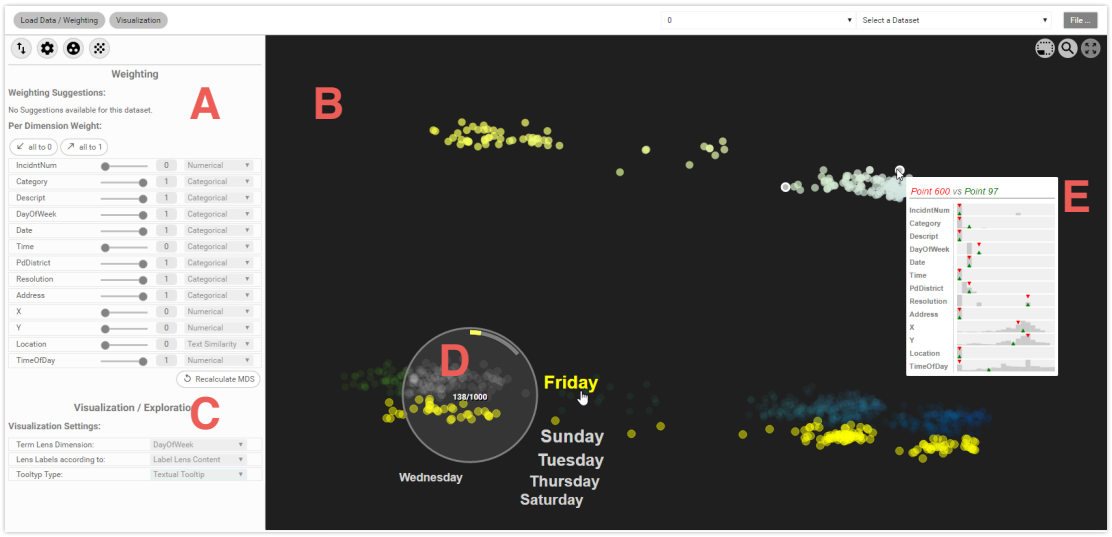


Figure 3.4: *Overview of the visualization prototype. The image shows the (B) visual result of a MDS projection of 1000 crime reports filed in San Francisco for (A) eight different attributes. This combination of attributes represents the starting point of the study with which each data analyst was confronted. To answer posed questions about the data, analysts used a minimal set of interaction techniques. Analysts could (A) steer the considered attributes, (D) investigate the data using a selection lens, or (E) clicking and hovering points to get detailed information of single crime reports. Also, analysts could (C) change the attribute considered by the lens and switch between a textual and histogram representation.*

any of this information. However, histograms are more effective for quantitative information or distributions within an attribute. The user can interactively change the representation from text labels to histograms and vice versa.

To identify and compare single data objects with others, Stahnke et al. [189] proposed to use a tooltip. The integration of attribute-wise histograms into the tooltip plus a visual cue indicating the position of the hovered data object allows bringing the object of interest into the relation of the overall data distribution. If the user clicks on one object and hovers another one, an additional cue is inserted into the tooltip allowing to bring both data objects into relation (see Figure 3.4(E)). In multivariate projections, for example, one is interested in the disjuncture of patterns. Telling in which attributes and how two points differ improves the understanding.

An interactive lens enables the selection and exploration of multiple data objects. A comprehensive survey about lenses has been carried out by Tominski et al. [200]. Figure 3.5 (a) depicts two lens approaches which can be interactively swapped during exploration. The lens consists of three additional parts: First, a textual hint of how many points are selected located in the center of the lens. Second, a visual representation reflecting the content of the lens; either as text label or histogram. Third, a radial bar indicating the amount of objects selected in relation to the entire amount of objects. If the user selects a value, all object occurrences are highlighted throughout the 2D data space. The left side of Figure 3.5 (a) depicts the representation of quantified values using a histogram visualization. The right side depicts the representation as text labels. The main issue with labels is that they try to

(a) Interactive lens (b) Fingerprint matrix

Figure 3.5: *(a) The interactive lens consists of three additional parts: a textual indication of selected points, a radial histogram, and the visualization of the values included in the selection. Left: visualization of the quantified content by a histogram. Right: visualization of the content by labels. (b) Excerpt from the fingerprint matrix for five attributes and six entries. The value of each attribute is binned based on the quantification. The size of the bin is then mapped to the color. This example shows that all* IncidentNum *are unique, because the color of all rows refers to the lowest possible value. In contrast, the attribute* DayOfWeek *indicates that the data entries happen on at least four different days: three rows are mapped to black (unique) and three have a very high binning value, meaning that these entries possibly share the same day.*

optimize the amount of labels as well as the proximity to the object within the lens. However, a label can refer to several selected objects. The aim is to stabilize the layout and maintain the order of information based on given frequencies. Therefore, I use a radial labeling algorithm which starts on the right hand side of the lens with the highest frequency and then adds labels counterclockwise in descending order until the starting point is reached. To prevent overlap, I check the position of the last label and move along the border of the lens to position the new label. Note that the user can exchange the considered attribute.

Fingerprint Matrix The lens also serves to select object groups for further analysis. Sets of objects can be compared attribute-wise using a so-called fingerprint matrix. Figure 3.5 (b) shows an excerpt. On the top, each column is labeled according to its attribute name. On the bottom, the data type is shown based on the quantification: number (N), category (C), or text (T) with respect to the crime datasets. Each attribute is colored according to the value scale from low (black) to high (yellow) data values. The matrix is linked to the projected data view using brushing and linking. Users can drill down to full detail by clicking on a row. A new window opens showing the raw multivariate data presented as a table. To be able to compare different patterns, the user can store and merge multiple selections.

3.3.2 Interpretation Study

This study investigates if domain experts, who work with raw multivariate data tables on a daily basis, are able to interpret the abstract 2D representation of DR results given their inexperience in advanced statistics. Ellis and Dix carved out problems that come along with evaluating visualizations such as complexity, diversity, and measurement which can be reduced to two major issues: the generative nature of visualizations and the lack of clarity of the purpose [55]. Results of DR techniques, in particular, aggregate the information to such

extent that it is challenging to interpret what the similarities or distances are made of; which attribute contributes in which way to the final layout or structure presented to the user. I argue that domain experts approach a complex visualization differently, which is why we [2] conducted a guided explorative study – a phenomenological analysis, to be precise.

Data

The main issue about real-world crime data is that it is very delicate. However, for this study we confronted the analysts with data that reflects real data as realistic as possible. I found that among others, the cities San Francisco, Chicago, and New York host an open data clearinghouse. We asked the LEA data analysts to align their data structure with the structure of the available open data with the result that a thorough description of the occurred crime is missing. However, the data analysts asserted that the open data reflects the main contents by means of attributes and thus suits this study. There was no need to preprocess the data. In order to prepare the study and define the tasks, I chose the San Francisco Bay Area [3] as a data source. The data consists of 13 attributes, among them 6 categorical attributes (*Category*, *Day of Week*, *Date*, *PdDistrict*, *Resolution*, *Address*), 5 numerical attributes (*IncidentNo*, *Time*, *X*, *Y*, *Time of Day*), and 2 textual attributes (*Description*, *Location*). Thereby, the *Category* consists of 36 different crime categories, the *Resolution* indicates if and how a crime was solved, and *X* and *Y* correspond to longitude and latitude. I am familiar with the city due to several visits and know about specific characteristics of districts as well as no-go areas. Because LEA data analysts typically analyze the data in weekly intervals and due to a seven day week this is also the shortest possible period to identify patterns, I chose the data for the week from Monday, July 25, 2016 to Monday, August 1, 2016. Note that this week includes two Mondays, a design decision to force a moment of Ah-hah!.

Tasks

The overall aim is to investigate whether untrained data analysts can interpret the 2D depiction of DR results given a minimum set of interactions. I created four consecutive tasks that force the analyst to gain a deeper understanding of the data by means of how data objects are grouped and how they differ from others. Figure 3.6 outlines all four tasks and their ordering. Following, I describe each task, its structure, and what the model solution looks like.

Task 1: *Is there a pattern among attributes between days?*
The first task introduces the analyst to the data. Figure 3.6 and Figure 3.4 show the starting point. The starting point consists of a pre-calculated result for the attributes *Category*, *Description*, *Day of Week*, *Date*, *PdDistrict*, *Resolution*, *Address*, and *Time of Day*. The analyst can change this setup at all times; we would not interrupt the process. The sheer amount of attributes that build up the four big clusters forces the analyst to focus on one single attribute and to see whether this attribute impacts the pattern. In the model solution, we can see

[2] Hereinafter, *"we"* refers to me and Florian Stoffel, who carried out interviews with the domain experts.
[3] SF OpenData: `https://data.sfgov.org/`

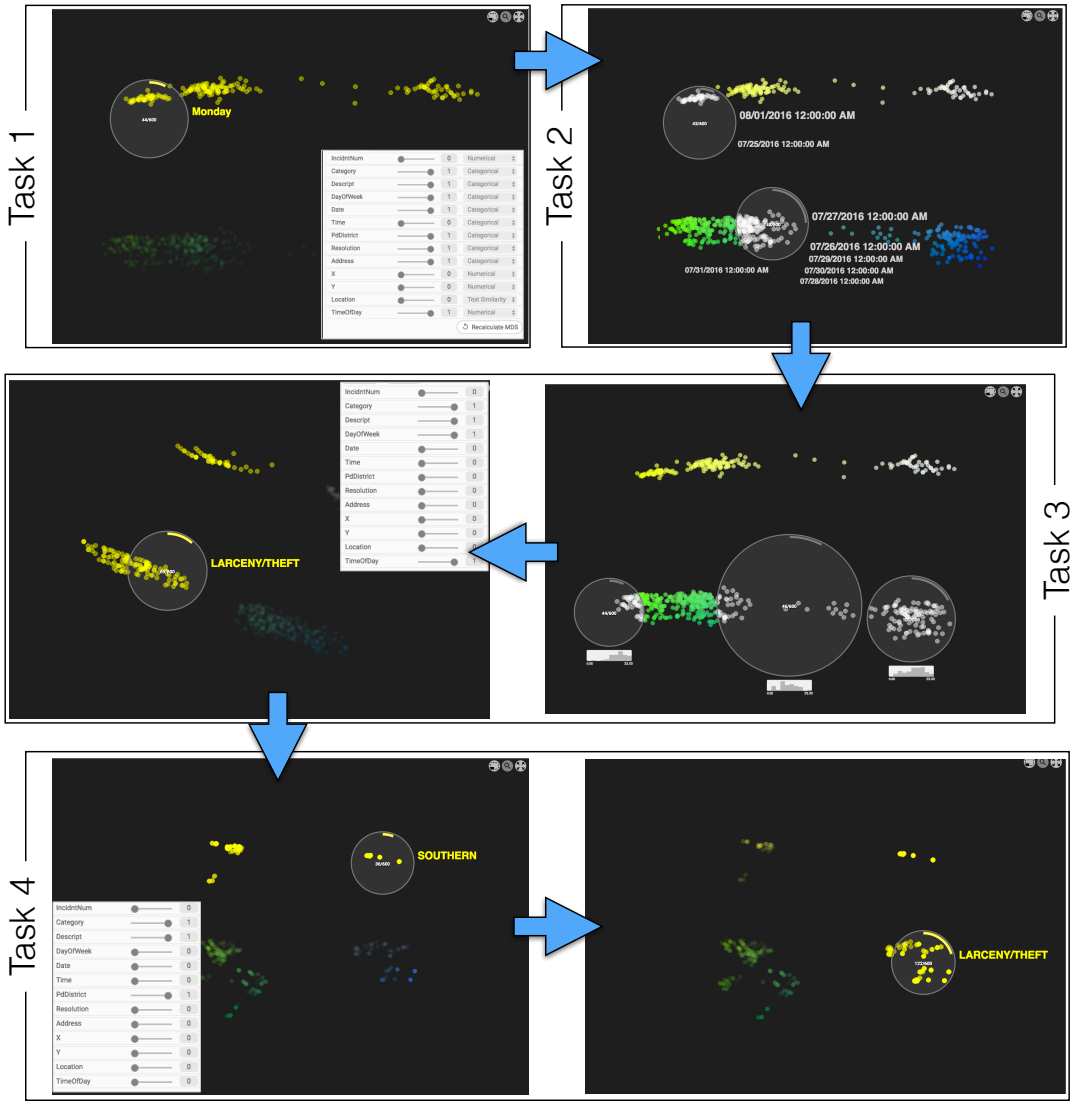


Figure 3.6: *Subsequent workflow of interpretation tasks. Each task corresponds to one question posed to the analyst. The DR results of the Tasks 1 to 4 can be interpreted as follows: In Task 1, the DR result splits the data into four clusters. Using the lens, one knows that the top left entries occurred on a Monday. This is because of the selection option: When hovering data objects with the lens, one can click on a label, and all occurrences are highlighted. In this case, the upper two clusters are highlighted when clicking on Monday. In Task 2, we can assume that the two dates on the top left lens correspond to two Mondays since these dates appear where the Monday cluster was found. As a result, the bottom clusters correspond to all remaining days of the week. In Task 3, the upper two clusters still correspond to the two Mondays. Changing the lens labels to* Category *reveals a huge cluster of* Larceny/Theft*. Building the intersection between the Monday and* Larceny/Theft *clusters means that the upper left cluster contains* Larceny/Theft *that occurred on a Monday. Changing the attribute-wise weighting in Task 4 reveals a similar phenomenon: Out of 10 police districts, the upper two clusters correspond to the* Southern *district. The two clusters on the right are categorized as* Larceny/Theft*. In conclusion, the top right cluster contains* Larceny/Theft *that only occurred in the* Southern *part of San Francisco.*

that two out of four clusters contain crimes that solely occurred on a Monday. The lens is placed on the top left cluster, a click on the only label Monday highlights all occurrences: the upper two clusters. This task can be solved by either using the tooltip, the content lens, or

the fingerprint matrix. For the sake of clarity, the images in Figure 3.6 primarily make use of the content lens. Once the analyst has identified this pattern, we proceed to Task 2.

Task 2: *Why is the day* Monday *separated from all other days of the week? What is special about the* Date *distribution?*
In the second task, we ask for the reason of this pattern – two out of four clusters occurred on a Monday. Switching one's focus to the attribute *Date* reveals that Monday, in contrast to all other days of the week, is assigned to two different days. Since the two dates appear at the same position, where the day Monday was determined, one can assume that there are two Mondays distributed among the two clusters at the top. One can conclude that all other days of the week are distributed among the two bottom clusters. Also, the Monday clusters cover approximately one-third of the overall data. This is the first Ah-hah! moment of the study, where the analyst is supposed to obtain new insight.

Task 3: *Which distribution of attribute values can you find for the rest of the week?*
The histogram attached to the lens reveals that there is a trend of crimes towards night time. The bottom left, and bottom right lens contain increased crimes at nighttime while the crimes in between tend to happen on daytime. Because of this temporal trend, the analyst adapts the multivariate projection and narrows the attributes down to *Category*, *Description*, *Day of Week*, and *Time of Day*. The result is again four clusters, two of them separated because of the double entry Monday. The two upper clusters again correspond to Monday, which can be observed via animation when one changes the weightings. To explain this phenomenon, the analyst analyzes the attribute *Category* that reveals a second pattern. Two out of four clusters deal a lot with *Larceny/Theft*, which can be identified by clicking on the lens label. Changing to the attribute *Description* shows that the category *Larceny/Theft* consists mainly of *grand auto theft*, *petty*, and *lock*.

Task 4: *Leaving the temporal aspect behind, is there a pattern based on places or crime types?*
For this task, the analyst has to change the projection and neglect the temporal aspect. The selection of the attributes *Category*, *Description*, and *PdDistrict*, however, shows four huge clusters again. Investigating these clusters by *Category* and *PdDistrict* reveals that there is one cluster that builds the intersection between the *Southern* part of San Francisco and the category *Larceny/Theft*. To locals this may be of no surprise, but most likely for the data analyst.

Design and Procedure

The study was carried out in a quiet room at the premises of a LEA. Each data analyst was placed in front of the notebook and received an introduction to the data dimensions and the interaction techniques. Each interaction technique was shown separately with a different dataset to not influence the actual study. The data analyst and the interviewer (experimenter) were the only persons present in the room.

Each data analyst was confronted with the same task order. However, we always started with the first task and then introduced the following task as an analysis question we posed to the analyst. We provided verbal clues if the analyst was not able to accomplish the given task. We further asked each data analyst to think aloud [21] and give insight not only in which interaction he or she is physically executing next, but also what the incentive and approach was. This way, we get an idea whether the analyst understands the results and can draw conclusions. All interactions were recorded using screen capturing, and the voice was recorded using the built-in notebook microphone.

After the study, we showed the analyst a labeled screenshot of the system and let him/her fill out a questionnaire regarding the basic understanding, the interaction concepts, and the extraction of knowledge. Furthermore, analysts filled out a form providing additional positive and negative feedback about the analysis of DR results.

Apparatus The studies were conducted using a 15" notebook monitor, one QWERTY keyboard, and one cord mouse. The display has a resolution of 1920x1080 pixels. The prototype was presented in full screen to the LEA researchers. For later analysis, we captured the screen as well as the voice of the participant.

Participants We reached out to the research department of a Law Enforcement Agency and recruited 3 data analysts (1 female) not trained in DR techniques or advanced statistics. One participant was trained in basic statistics but not in DR techniques. All participants had normal or corrected to normal vision. All participants work with multivariate data tables on a daily basis, however, are not used to working with abstract data representations such as planar projections.

3.3.3 Findings

We started this study with one question we posed to the data analysts. During the analysis, we encountered four interesting situations which I will elaborate in this section as findings (*F*). Note that we introduced the interaction techniques but did not provide any further conceptual explanation of the meaning of the 2D multivariate projection.

F1: The analysis starts with an already known hypothesis.
To launch the study, we posed one specific question (Task 1) to the analysts, yet we could not influence the mindset and thus the taken approach. Each analyst first tried to verify his or her hypothesis of the unknown data before tackling the task asked. All three analysts started by changing the depiction to the setup of the routine activity. This means, they first tried to approve a temporal and occasional pattern.

F2: Analysts always consider to add/remove attributes to the depiction to explain a cluster separation.
The tasks consisted of two examples of cluster separations among one attribute: two clusters

for *Monday* and two for the category *Larceny/Theft*. For both cases, all three analysts added or removed attributes to track visual changes in the multivariate depiction. One participant even used attribute-wise weights in 0.2 steps to track minor changes. This finding is particularly interesting because we did not provide any conceptual explanation of cluster separation factors to the analysts.

F3: Analysts do not add/remove attributes to explain an anomaly they are insecure about.
We observed that analysts subsequently add or remove attributes to explain a cluster separation. However, they do not follow this routine if they cannot explain something unexpected in the data. For example, before they started to add other attributes, they first sought for an explanation using the content lens, the tooltip, or the fingerprint matrix.

F4: Analysts untrained in DR have a great understanding of a multivariate depiction given a use case relating to their domain.
All three data analysts had a great understanding of the multivariate data depiction. Following their procedure and interviewing them afterward showed, that despite initial difficulties, they easily built a deeper understanding of the data and the correlations in it. It took the analyst in average 30 minutes to solve all given tasks. The first half of this time was used to confirm their hypothesis and to solve Task 1. After that, the analysis speed increased drastically. One analyst noted that he is searching for variances among one attribute, but a DR technique is different, "there is a big pot and you can combine lots of different things together". All analysts recognized that they need to adapt their way of thinking but at the same time acknowledged that using DR is an excellent way to check correlations quickly.

The fact that DR techniques state an entirely different approach to analyze crime data also hinders the integration into the standard workflow of the analysts. To efficiently use DR in their workflow, the analysts wish for additional statistics helping to interpret the configuration of clusters and outliers.

The observation of the participants also revealed the extensive use of the lens in combination with the manual steering of the attribute-wise weights. Steering the weights represents an essential interaction concept to understand which dimensions correlate. Then, applying the lens enables the straightforward exploration of the configuration without overwhelming the analyst with too much information. While the analysts appreciated the simplicity of the lens, they also wished for more elaborated selection techniques since a lens restricts the selection to a circular extent as well as additional statistics as mentioned before.

The evaluation of the questionnaire furthermore showed that all analysts found it very easy to understand dependencies in the data and that clusters corresponded their expectations. However, one analyst found it difficult to draw conclusions based on depicted correlations among dimensions.

The findings *F1* and *F2* lead to the interpretation that the performed interactions represent a manual subspace analysis. The data analysts played with different combinations of attributes with the aim to verify their hypotheses, confirm a finding, or find a relevant pattern. Although

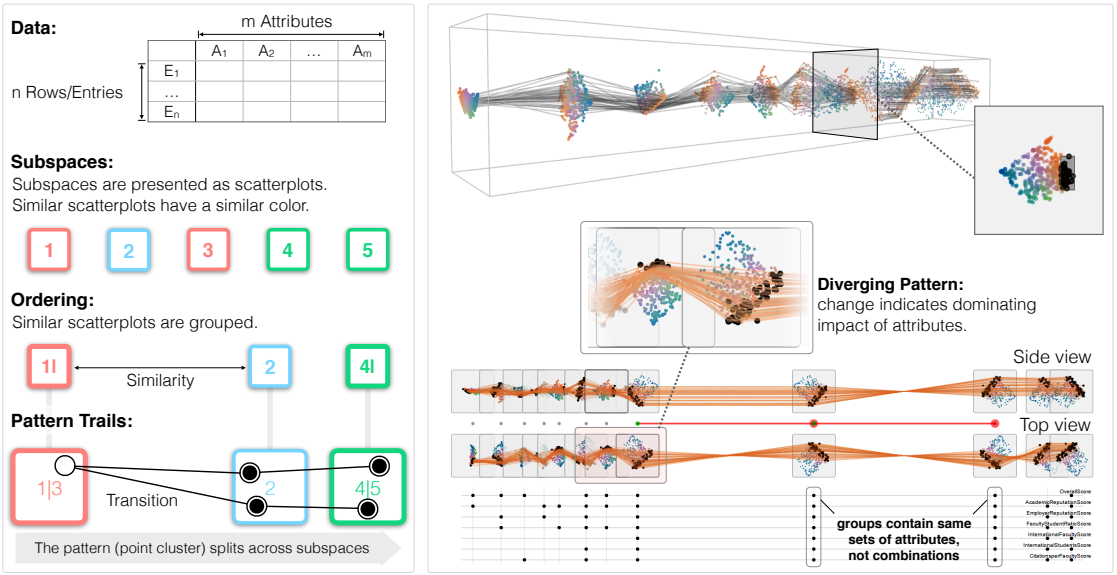


Figure 3.7: *Visual analysis of pattern transitions by a series of consecutive pattern transitions between scatterplots. (a) Scatterplots depict subspaces and are grouped and sorted based on similarity. (b) This example shows the pattern transitions in the University data set. Based on a space-time cube like visualization, one can trace sorted patterns in a side and top view on the cube (see Section 3.4.5).*

it was not relevant for this study, patterns may occur differently and structurally change among different subspaces of the data. Because it is a tedious task to search subspaces and patterns manually, there is a clear need for a system that supports the user. In the following section, I build upon this result and propose Pattern Trails, a novel approach to identify and interpret patterns across subspaces of multivariate data.

3.4 Pattern Trails: Pattern Transitions in Subspaces

Pattern Trails is an interactive visual approach to find and explain pattern transitions among subspaces of multivariate data. I first apply automatic subspace analysis and then DR, in particular, the distance-preserving projection MDS, to visualize the results in a small-multiple [203] environment using scatterplots. Furthermore, I highlight transitions between the subspace depictions for tracing structural changes of patterns. This section makes two contributions towards understanding multivariate patterns and pattern transitions: First, a *systematization and categorization of pattern transitions among subspaces of multivariate data*. I introduce a *trail* as a set of pairwise transitions between subspaces along which patterns can be grouped and meaningfully compared. Thereby, various transition types occur, each having a different meaning. Second, a *data-driven similarity measure for projections to group subspaces and overcome redundancy*. Because data sizes and tasks differ, I tightly couple this process and the user who steers the parameters to obtain an effective aggregation of scatterplots, and thus subspaces.

The systematization of pattern transitions and the user-steered similarity between subspaces

are integrated into a visual approach. In summary, the visual approach consists of a horizontal view on the objects and a vertical view on the contributing attributes. The view on both spaces supports the interpretation and understanding of patterns.

3.4.1 Basic Idea

Pattern Trails is a visual interactive approach for expert users, enabling the exploration and understanding of patterns across subspaces of multivariate data. The main idea is to order a set of subspaces in meaningful sequences, such that groups of patterns can be distinguished and their changes effectively traced and interpreted across the subspaces. Pattern Trails follows a three-step procedure depicted in Figure 3.7 (a). First, I derive interesting subspaces of the multivariate data using a state-of-the-art method for feature selection called SURFING [13]. SURFING searches for subspaces in which data objects form a (hierarchical) clustering structure. The algorithm measures the interestingness of a subspace based on the variance among the k^{th} nearest neighbor of every object. To make the subspaces visually accessible to the user, I then apply DR and project each subspace to 2D space. The results are visualized as scatterplots and arranged side by side in a small-multiple environment [203]. Tatu et al. [198], for example, also visualize all subspace projections, but compare them in a 2D MDS layout without tracing the pattern changes across the subspaces.

Second, I group the subspaces based on their similarity. Subspace search algorithms like SURFING often yield subspaces similar in terms of involved dimensions and/or data relationships, hence producing redundant results. I enable the user to group the projections based on a data-driven similarity measure interactively. Grouping in my experience is essential in subspace analysis, to reduce the abundance of subspaces to a smaller number of representative ones for practical exploration. The projections are also reordered using the data-driven similarity measure which allows the user to set a threshold regarding the level of aggregation.

Finally, I highlight the change of a pattern based on a user-defined selection, which can also be applied automatically using subspace clustering methods. I connect the data points that belong to a selected pattern across all subspaces using lines – I refer to this connected view as pattern *transitions* between subspaces. The transitions describe the structural change of a pattern across subspaces. Changes in subspaces can be modeled as operations to insert, delete or replace subspace dimensions. For example, consider a dense cluster of points spreading over different clusters in the subspace succeeding it in the determined order. This is a significant change in the pattern structure which needs to be further examined in terms of the change of the respective subspace dimensions. This representation may remind of the highly interactive PCPs [99]. However, this approach represents further development regarding the comparison of subspace patterns; it enables to draw conclusions based on links between subspaces of multiple attributes, instead of single attribute axes. Using Pattern Trails, one can find and explain pattern transitions in multivariate data.

Compared to automatic approaches for feature selection (for example, optimizing classification accuracy), Pattern Trails does not search for an optimal set of attributes that form a pattern but rather analyzes which attributes cause a change in the structure of a pattern.

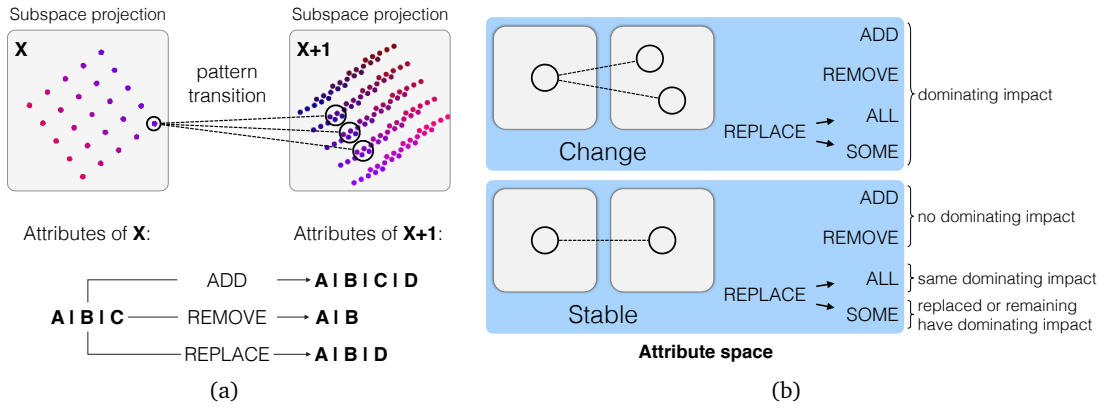


Figure 3.8: *Overview of add, remove, and replace operations in the attribute space.*
(a) depicts the impact of attributes to a single pattern transition. Left: A pattern transition always migrates from one subspace to the subsequent and is influenced by the attributes building up the subspace. Right: By looking at the attribute space instead of the visual space, we can distinguish between three cases: A pattern remains unchanged or alters based on (1) adding*, (2)* removing*, or (3)* replacing *attributes that build the subsequent subspace. Each of these operations can be interpreted differently regarding the transition type.*
(b) outlines a high level categorization. Pattern transition between two adjacent subspace projections can show one of two different states: changing or stable. Depending on the operation in the attribute space, the affected attributes imply a different meaning. If a pattern changes, the affected attributes, which contribute to the subsequent subspace, dominate the change. For a stable pattern transition, the affected attributes have either no dominating impact or the same dominating impact as the attributes of the preceding subspace.

Pattern Trails enables the user to analyze any selected pattern in one subspace projection using visual interaction and is not limited by required class-labels or given cluster structures.

3.4.2 Subspace Pattern Transitions and Its Interpretation

Patter Trails highlights the change of patterns in multivariate subspaces. As pattern change, I refer to clusters, outliers, or correlations that vary *structurally* across the projections of different subspaces. To make these structural changes visible and accessible to the user, I project the data using DR into the two-dimensional space and represent the results as a scatterplot. An example is illustrated in Figure 3.8(a). In the first projection X, the pattern corresponds to a cluster that divides into three clusters in the second projection $X + 1$. Connecting lines visualize the transition. I conceive there exist two fundamental transition types, as depicted in Figure 3.8(b): a *changing* and a *stable* transition, leading to the question: *What is the meaning of these transitions?* To enable the interpretation of such transitions, it is of high importance to consider the attribute space, because it provides information about which attributes are dominant and influential to the structure change of subspace patterns, and which have less influence.

We can distinguish between three basic operations which lead to an either changing or stable transition: Attribute(s) can be *added*, *removed*, or *replaced*. Each operation impacts the interpretation of patterns on the dominance of the attribute(s). An attribute is considered as *dominant* if it significantly controls the structure of the subspace, non-dominant and probably redundant, otherwise. As a result of adding, removing, or replacing attributes, a pattern

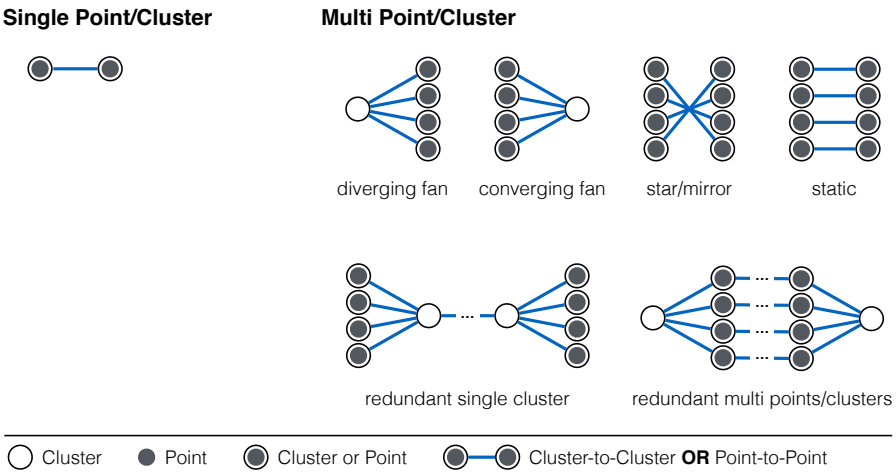


Figure 3.9: *Taxonomy of pattern transitions grouped into transitions between single and transitions between multiple points/clusters.*

alters or remains unchanged in the subsequent subspace. The combination of domain-specific knowledge and the visual representation is paramount to explain pattern transitions. Figure 3.8(a) and Figure 3.8(b) provide an overview of operations and interpretation regarding the attribute space. Based on the distinction between transition states (changing or stable), I next derive a taxonomy of occurring pattern transitions. The combination of all transitions among all subspaces builds the pattern trail. Following, I discuss the interpretation of pattern transitions about the attribute space and the transition classes (see Figure 3.9).

Single Point or Cluster Pattern

The first class of transition patterns refers to single point/cluster transitions.

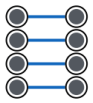
Static Single Pattern (P1): Within the transition to another subspace, a single point or cluster remains in its consistent/static state. In the case of a single point, the point does not become a member of another pattern, and in the case of a cluster, the cluster does not split or merge with other clusters. The interpretation concerning the attribute space is as follows: *Added* or *removed* attributes are non-dominant and have no impact on the subspace structure. *Replaced* attributes, however, can be dominant if the pattern remains stable. But it is also possible that the untouched attributes are the dominant ones, which impacts the structure.

Multi Points or Clusters Pattern

The second class of pattern transitions considers all transitions that involve more than one point or cluster. Furthermore, I introduce pattern developments among several subspaces leading to the identification of possibly redundant attributes and subspaces.

Converging Fan Pattern (P2): Points and/or clusters merge into a single cluster. This transition type indicates a major change regarding the subspace structure. As a matter of fact, the subsequent subspace contains information that reinforces the similarity between patterns and causes them to merge. That is, in attribute space *added* attributes contain information that reinforces the similarity and thus dominate the creation of a cluster. *Removed* attributes take away information. The remaining attributes share more similar information causing the points/clusters to merge. *Replaced* attributes can be interpreted as either dominating or non-dominating, depending on whether they take off information or add information causing the patterns to merge.

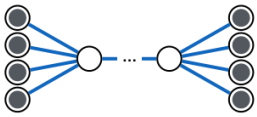
Diverging Fan Pattern (P3): The diverging fan corresponds to the inverted converging fan **P2**. This means, information is added or removed causing a cluster to split; similar patterns reorganize in different groups. *Added* and *Replaced* attributes dominate the subsequent subspace structure and append information so that the cluster content regroups. *Removed* attributes take away information that caused the formation of a cluster at first. Without this information, the overall similarity within the cluster decreases.



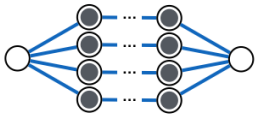
Static Pattern (P4): Similar to **P1**, the static pattern describes a stable transition without changes between sets of patterns. *Added* or *removed* attributes are non-dominant and have no impact on the subspace structure. *Replaced* attributes can be either dominant or non-dominant, depending on whether the untouched attributes dominate the subspace structure.

Star/Mirror Pattern (P5): The interpretation of this transition pattern is identical to the static pattern **P4**. The effect of mirroring can be traced back to the creation of the two-dimensional plots. They are created using a planar projection strategy, such as MDS or PCA, that are known for not being mirror/rotation invariant. Whenever this pattern transition appears, the projection technique mirrored the underlying data.

Understanding the impact of attributes is key to interpret the subspace structure as well as the meaning of a pattern. Consider one pattern (e.g., cluster or outlier) that occurs in an arbitrary subspace. The attribute space provides details in three general scenarios: First, the same pattern takes place in various subspaces. Second, the pattern only occurs in one subspace. Third, the pattern takes place in different structures in different subspaces. For each scenario, one needs to draw conclusions in the attribute space to determine dominating, non-dominating, and redundant attributes. This way, one can determine expressive attributes that steer the structure of subspaces. The following two patterns describe combinations of pattern transitions among several subspaces and open the space for attributes that can be considered as being redundant; that is without significant impact to the subspace structure.



Redundant Single Cluster (P6): First, points/clusters converge to a single cluster. Then, the single cluster remains stable among subspaces and finally diverges again to different points/clusters which are not necessarily identical to the initial ones. The interpretation is as follows: The attributes causing patterns to merge and split impact the subspace structure so that patterns regroup. Since these attributes have an impact on the structure, they are likely to hold information that is interesting for further analysis. For all transitions between the subspaces 2^{nd} and $(n-1)^{th}$, the added, removed, or replaced attributes do not show a specific impact on the structure of the subspace and the formation of patterns. Therefore, I consider these attributes as being *redundant* regarding their impact or expressiveness.



Redundant Multi Points/Clusters (P7): This pattern represents the inverse situation to **P6**. First, a cluster diverges. Then, multiple points/clusters remain stable among several subspaces, before they finally converge to a single cluster. As for **P6**, added, removed, or replaced attributes that do not show a specific impact on the subspace structure between subspaces $2nd$ and $(n-1)^{th}$, can be considered as being redundant. The information they take away or bring in is not expressive enough to cause the structure to change.

Different pattern transitions can occur within one transition between two subspaces. However, it becomes challenging to interpret the visual depiction. For example, an attribute can be dominant for one pattern transition but redundant for another one such as the combination of the patterns **P1** and **P2** within one single subspace transition. Furthermore, many combinations are possible.

Automated Support for Interpreting Patterns

One major challenge of Visual Analytics is the automatic support of users in interpreting patterns [177]. In this approach, I consider transitions between subspaces and aim to identify structural changes in patterns based on operations in the attribute space. This leads me to the question: *How can a visual analysis system support the user in understanding pattern transitions?*

Based on the two abstract transition types, depicted in Figure 3.8(b), I can provide interpretation aid. Generally speaking, if a pattern changes with the transition, the affected attributes dominate the subsequent subspace structure, whether they are added, removed, or replaced. This is different for a stable pattern. Either the affected attributes have no dominating impact (add, remove) or they may have the same dominating impact as the attributes of the preceding subspace (replace). This gives us a powerful tool. In combination with the automatic detection of pattern transitions (see Section 3.4.4), the visualization system can suggest a valid interpretation. Even for combinations of transitions, the system can provide a compound of possible interpretations, yet, it is up to the user to employ this information and to gain new insight.

This approach is applicable for verifying hypotheses about the data, and also for explorative tasks such as identifying interesting subgroups and changes. However, the identification and interpretation of a pattern transition depend directly on a meaningful ordering of the subspace representations. In the following Section 3.4.3, I discuss the similarity-based ordering of the subspace representations.

3.4.3 Similarity-based Ordering of Subspace Views

I make use of distance-preserving projections as a means to visualize subspaces of multivariate data. A visual representation makes subspaces accessible and enables the efficient identification of patterns, such as clusters and outliers. To understand how a pattern changes among different subspaces, I consider their pattern transitions. While a pattern transition is unique between a pair of subspaces, it is challenging to find relevant transitions beyond multiple subspaces, in particular using a visual representation. Simply lining up all subspaces one after another raises the question for a meaningful ordering, which enables the efficient identification of relevant pattern transitions, so that one can draw conclusions regarding relevant and redundant attributes (consider, e.g., the patterns **P6** and **P7**). A meaningful ordering for multiple subspaces is key for this task.

The issue of finding a meaningful ordering is a known NP-Hard problem and well-known in the parallel coordinate plots [99] domain. It is still subject of ongoing research on how to re-order the axes to obtain expressive results. Examples of ordering goals include an ordering based on maximum pairwise correlations or image-based metrics like reducing the number of line crossings [49]. The main problem of finding a meaningful ordering is that each axis has at most two neighboring axes, as it is the case for our representation; each subspace can be visually connected to at most two neighboring subspaces. However, the problem is different. In contrast to parallel coordinate plots, I handle transitions between two-dimensional projections of multivariate data that are built by more than one attribute. To find an optimal predecessor and successor, I like to consider the notion of similarity between subspaces, in particular, the similarity between their visual representations. This is due to the visual representation of transition types that enables us to interpret on which level patterns are similar or different among subspaces. Expressing the similarity between subspaces is possible in many ways, but two natural ones are based on the similarity between the attribute sets that make up the subspaces, as well as based on the visual similarity between subspaces. I discuss in this Section the application of an attribute-based similarity and introduced a new similarity measure based on the multivariate projection.

Attribute-based Similarity

It seems apparent that the similarity between two subspaces can be expressed by the similarity of the attribute sets. Two prominent examples used by state-of-the-art approaches are the Jaccard similarity [101] and the Szymkiewicz-Simpson [194] coefficient (compare, e.g., Tatu et al. [198] and Hund et al. [95]). The Jaccard similarity between two attribute sets is defined as the ratio between the size of the intersection and the size of the union of sets. In

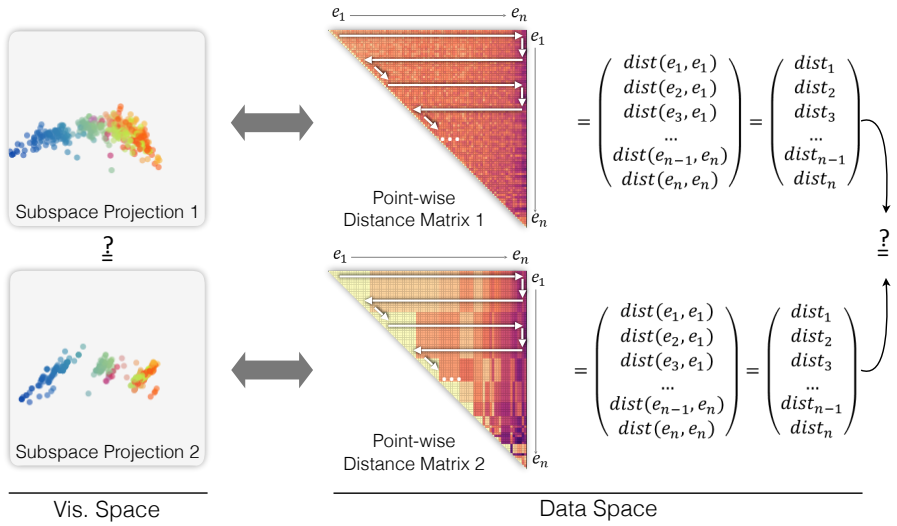


Figure 3.10: *Similarity computation between two subspace projections. A possible solution to the question* how to compute the similarity between two subspace projections? *using the example of MDS. I transform the point-wise distance matrices (input for the subspace projections) to 1D feature vectors. Compared to the projection results, the input matrix is invariant to rotation, scaling, and translation. Then, a standard distance measure can be applied to vectors to derive the similarity.*

contrast, the Szymkiewicz-Simpson coefficient expresses the similarity as the ratio between the size of the intersection and the minimum size of sets. Both coefficients are based on the set of attributes rather than the subspace structure. While the Jaccard similarity provides a ratio of common attributes, the Szymkiewicz-Simpson coefficient considers the exact amount of overlapping attributes among subspaces. However, the attribute-based similarity poses a suboptimal solution. Just because the same attributes are used to some extent does not provide any information about the similarity of the data and the projection.

Projection-based Similarity

A common approach to computing the similarity between pairs of subspace projections is to apply state-of-the-art image-based similarity measures to the visual depiction of each projection. For example, Lehmann and Theisel [139] investigate the affine transformations between pairs of projections with the goal to find the most expressive, discriminative projections. However, projections transform the data to a lower two-dimensional space and thus inevitably introduce bias, the projection error, interfering with the notion of similarity. Furthermore, projections are not invariant to rotation, scaling, and translation, known as the Procrustes problem.

In this section, I provide a solution to the question: *How to compute the similarity between two visual projections of subspaces, namely the scatterplots?* Even though there exist approaches to overcome the named issues, I aim for a solution that is invariant to rotation, scaling, and translation. To do so, we have to look at how a data projection is computed. Pattern trails applies MDS to the data to derive a visual representation of the underlying subspace. Thereby, the MDS derives the final layout by computing a distance matrix and then preserving the distances in a two-dimensional manner. The commonality between projection techniques

such as MDS [47], PCA [117], or t-SNE [145] among others is that they derive the final layout based on an input matrix: either a distance matrix, a covariance matrix or a probability distribution matrix. I propose to consider the matrices rather than the visual representation, because no projection error is yet introduced, and matrices are known to be invariant to rotation, scaling, and translation. Furthermore, the ordering of rows and columns is not of major concern when computing distances.

I elaborate the question of how to compute the similarity between two projections of subspaces in Figure 3.10 with application to MDS. Each depiction of a subspace projection is based on a distance matrix that consists of the aggregated distances between pairs of data records. In order to linearize the matrix, the matrix is traversed row-by-row. This way I build an n-dimensional feature vector, whereas n describes the number of data records. Based on the feature vectors, I can compute the distance between two projections using distance functions like Euclidean or Manhattan distance. Visually, the similarity approach orders the subspace projections in terms of the spread between data points in the projection, which is due to the content of the distance matrices. Consider two distance matrices with very different data variances. Naturally, the distance between both matrices is significantly large, which is also reflected by the visual representation. This distance computation is based on the data rather than the sets of contributing attributes and overcomes projection errors introduced by the visual representation.

Ordering Computation. Based on the derived similarity, the system computes all pairwise distances between subspaces to determine an ordering. This means, it computes a distance matrix of distances/similarities of all pairwise subspaces. The distance matrix can serve as input to any technique that linearizes the distances, or in other words, preserves the proximities between subspaces in a linear manner. This way, the system cannot only provide an ordering of subspaces but also visually point out how close subspaces are to each other. I present results for the well-known iris data set [140] in Figure 3.11. In this depiction, I compare the (2) Jaccard similarity and the (3) data-driven similarity based on the input matrix. In comparison, the attribute-driven (2) Jaccard similarity performs worse because it ignores the underlying data. The (3) similarity based on the input matrices clearly separates the *Petal Width* and *Height*, which are known for steering the clusters in this data set. The bottom row shows the results after applying user-steered Agglomerative clustering between subspaces. For each cluster, the projections are replaced with a new projection taking into account all clustered attributes. An interesting observation is that the similarity between projections also reflects the spread in the data. From left to right, the point clusters move closer while the transitions remains static. In this example, no prior subspace analysis is applied, yet we can find relevant subspaces and explain their meaning regarding the similarity and the contributing attributes. The displayed pattern transitions in the bottom row furthermore reveals a static development (pattern **P4**) suggesting that in combination with the similarity ordering, the attributes *Petal Width* and *Height* control the subspace structure.

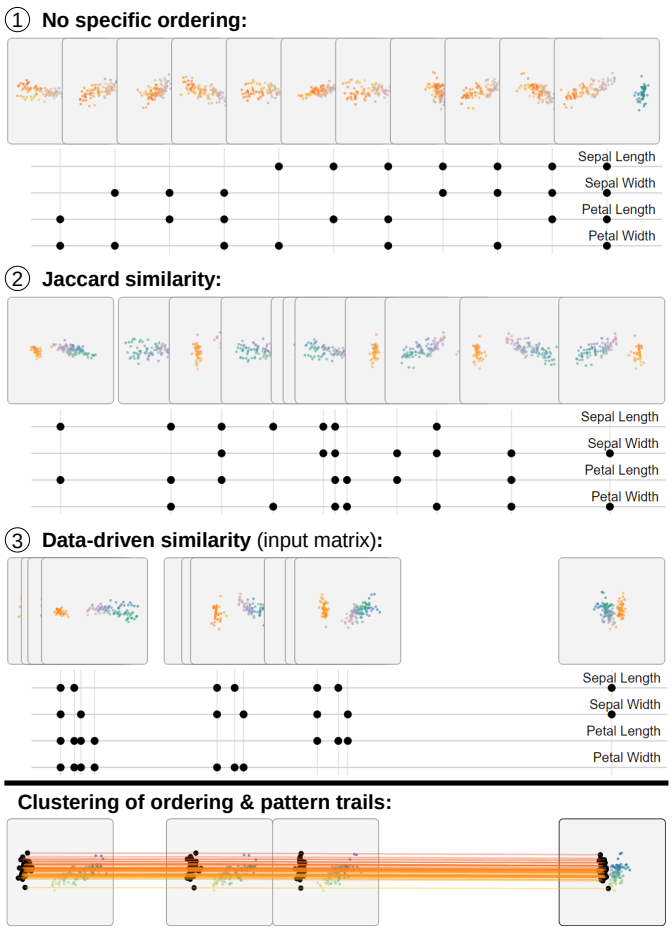


Figure 3.11: *Ordering of the iris data [140]. (1) All subspaces without specific ordering. (2) Application of the Jaccard similarity. (3) Application of our data-driven similarity for projections based on the input matrix. Distances between projections encode similarity, which is linearized using MDS. Very similar projections overlap.* Petal Length *and* Petal Width *steer the clustering and are clearly separated to the rest, compared to the Jaccard similarity. For each ordering, the color encoding is re-computed in respect of the first projection. To reduce overlap, I cluster the projections and select a pattern in the visualization, resulting a static pattern transitions.*

3.4.4 Visual Identification of Patterns

A key task in multivariate data analysis is the identification of relevant subspaces and patterns. Patterns, however, can change their overall structure among subspaces, and thus express a different meaning. To explore subspaces and the pattern transitions, I introduce a visual approach based on Sections 3.4.2 and 3.4.3. The general goal is to integrate the categorization, as well as the similarity-based ordering and clustering of subspaces. The visual approach comprises a horizontal perspective on the objects and a vertical perspective on the contributing attributes. That is, the visualization of the subspace projections in a horizontal manner side-by-side, and below each projection (vertically), an overview of contributing attributes (compare to the visualization of projections in Figure 3.11). Thereupon, I employ interaction as means to explore pattern transitions.

Subspace Cube

Pattern transitions represent an interpretable structure imposed to order and compare patterns between subspaces. The basic idea is to visualize subspaces side-by-side, enabling the identification of changes between adjacent patterns. To visualize the subspaces, I project each subspace to two-dimensional space using the distance-reserving projection MDS [47]. A 3D cube, similar to space-time cubes [11], is a clear choice for visualizing pattern changes across 2D subspace projections, leading to the name of this representation: the *Subspace Cube*. Figure 3.7 depicts the application to the 2012 University Ranking data set. From a visualization point of view, Pattern Trails looks similar to PCPs. Johansson et al. [113] argue that a 2D representation of PCPs is more effective than a 3D representation, which is why I transform the Subspace Cube to 2D in such way that the main structural pattern changes are preserved. Tufte [203], therefore, introduced Small Multiples as an efficient way to visualize discrete changes in the data. Using small multiples, I depict the side and top view of the cube by rotating the small multiples by 0.5π. These two views on the cube differ the most from one another, thus preserving the main structural changes.

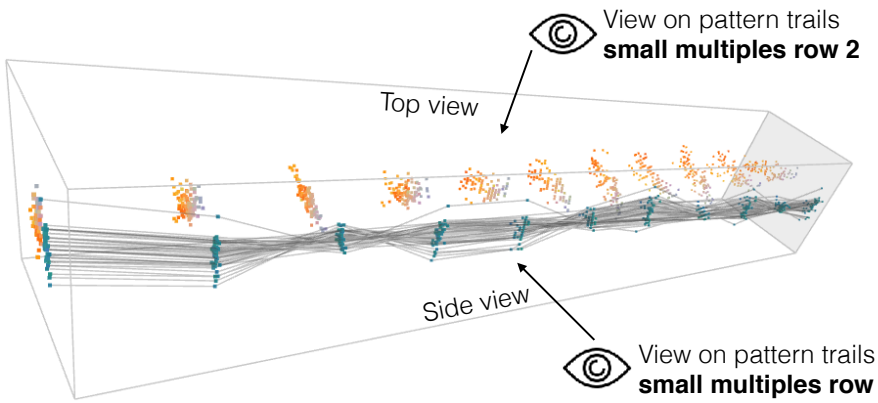


Figure 3.12: *Subspace Cube and how it incorporates the small multiple view on the pattern transitions.*

Compared to the approaches of Fanea et al. [60] or Yu et al. [220], the Subspace Cube visualizes projections and pattern transitions disregarding advanced interaction concepts. Rotating, as well as zooming and panning, are the only available interactions in the hope that the Subspace Cube provides data contexts, which are lost using the small multiple representation. For example, the Subspace Cube provides different angles, which are not available in the small-multiple representation. The projections are equally ordered in the cube as well as the small multiple representation.

Linked Multiple View

I combine two visual methods to point out the structural changes of a pattern. The first method color-encodes the projected data points based on a 2D colormap. The idea is to lay all projected points out on a predefined 2D color plane and assign each point the color with identical $x-$ and $y-$coordinates [191]. To point out pattern changes, I color-encode the data

of the first subspace projection and reuse this encoding for all succeeding projections. If the colors mix in a projection, one can draw conclusions regarding the similarity between points – different points are considered to be more or less similar than before.

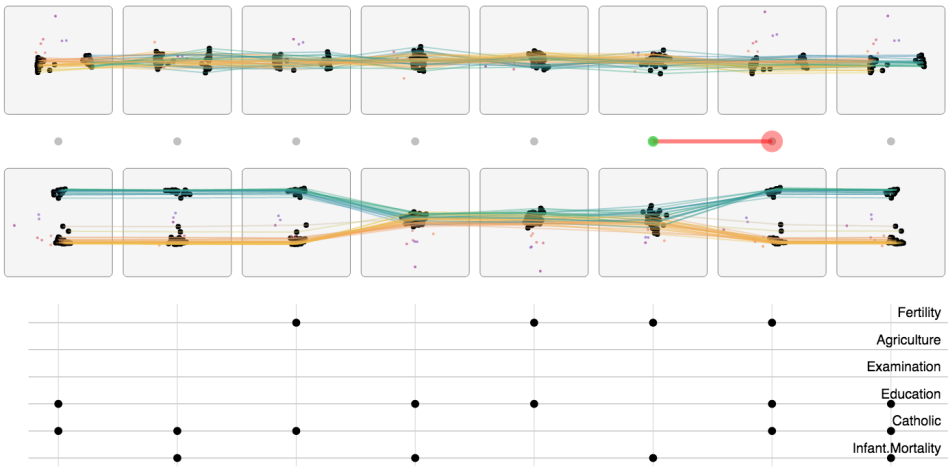


Figure 3.13: *Visualization of the Standardized fertility measure and socio-economic indicators for French-speaking provinces of Switzerland at about 1888 after the application of the subspace analysis algorithm SURFING [13]. The data consists of 6 attributes (*Fertility, Agriculture, Examination, Education, Catholic, Infant.Mortality*), of which the combinations of Fertility, Education, and Infant. Mortality causes the patterns to merge. The pattern is static for multiple clusters with adding the attribute Catholic. This attribute is numeric. However, the content is discrete, which causes the pattern to split into two separate clusters: People who are catholic, and people who are not. If this attribute is not considered, then the pattern merges (compare Pattern **P6**). Top: static pattern development. Bottom: view rotated by* 0.5π*, we see converging, diverging and static patterns. The black points indicate the selection and the lines are colored with respect to the point colors. The diverging fan pattern is highlighted via a red-green connector.*

Color-encoding, however, also introduces problems regarding the identification of particular pattern transitions as introduced in Figure 3.9. For example, it is challenging to differentiate between pattern transitions if many points overlay. Therefore, I additionally visualize the transition between adjacent subspace projections via lines; same points are connected in adjacent projections by single lines. Figure 3.13 depicts the transitions from a static to a converging, and from a static to a diverging pattern, which is challenging to identify without visual links. The Figure 3.13, furthermore, visualizes an additional small multiple row of projections. In the top row, the converging and diverging pattern are not visible. Rotating all projections by 0.5π enables the identification of such patterns, which is why I provide both views on the data.

The representation using lines follows the work of Dasgupta and Kosara [49], in which they classified lined-based patterns between axis in parallel coordinates [99]. The interpretation of patterns is different for transitions between adjacent projections, because each subspace projection consists of multiple attributes and is visualized in a way that pattern translations, or mirror images, do not affect the structure of a pattern. This is different for parallel coordinates, where each axis reflects exactly one attribute. Yuan et al. [221] developed a visually similar approach to Pattern Trails which also enables the comparison of multiple attributes. The authors propose to seamlessly integrate scatterplots into parallel coordinates.

Thereby, the scatterplot reflects the relations between manually selected attributes (axes) thus allowing a visual inspection of complex patterns. This approach, however, requires the manual selection of subspaces and presents a continuous switch between the PCP and scatterplot representation, posing a challenge to find interesting subspaces and patterns. Pattern Trails relies on small multiples that enable the effective comparison of patterns across many subspaces at a glance.

Similarity-based Ordering and Clustering. The effectiveness of the small multiple representation is highly influenced by a meaningful ordering to make sense of patterns efficiently. In addition, small multiples are limited by the horizontal display dimensions, causing inevitable overlap between multiples, like e.g. depicted in Figure 3.11. Therefore, the system orders the small multiples regarding visual similarity and applies a user-steered Agglomerative clustering: the user is given control over the distance parameter and can interactively change the size of clusters.

In accordance with the ordering computation described in Section 3.4.3, I use the input matrices of all subspace projections and compute all pairwise similarities. Using a 1D MDS [47], I horizontally reassign the positions of all projections. Distance indicates how similar or dissimilar respective projections are. To reduce overplotting and visual redundancy between projections, and thus subspaces, I apply clustering. Agglomerative clustering has the advantage of only one operating parameter, which reflects the notion of distance between clusters. The user can interactively cluster overlapping subspace projections by steering the distance parameter. A prototype representation then replaces all clusters, this is a new projection including all attributes of the cluster.

Supporting Pattern Detection

So far, pattern changes are visualized and found interactively. That is, the user first selects an interesting projection from the small multiples. Then she selects a pattern in the selected projection, for which the full of transitions is displayed across all subspace projections. Identifying interesting subspaces and patterns is challenging for increasing amounts of data and thus subspaces. I provide *automated support* for finding interesting patterns and transitions by narrowing the representation down to pairwise transitions between adjacent subspace projections. I provide an overview of possible transition patterns in Section 3.4.2, Figure 3.9. The general idea is to point the user to these transition occurrences between subspaces as a starting point for further exploration.

The automatic identification of possibly interesting transitions is based on three criteria: the transition type, the number of affected points that form the transition type, and the minimum distance between affected points. Based on the taxonomy in Figure 3.9, I distinguish between five transition types. For each type, I automatically detect its occurrence between adjacent pairs of subspaces using *heuristics* based on the Density-based spatial clustering of applications with noise (DBSCAN) algorithm [59]. The core understanding of DBSCAN suits the exploration workflow because it depends on two parameters: the minimum amount of points, and a minimum distance between points that form a cluster. The user can interactively

set the size and density of the expected patterns. Based on the user input, I compute the clusters for both subspace projections (following, let us call them projections A and B) separately, and identify the transition type as follows:

Single Point/Cluster: I iterate all clusters in A and check whether a significant amount of points remains in the same cluster in B.

Converging Fan: I iterate all clusters in B and check whether a significant amount of points ends up in at least two different clusters in A.

Diverging Fan: Similar to the *Converging Fan*, I reverse the direction and check whether the points emerge from A to B.

Star/Mirror: I iterate all clusters in A and check if a significant amount of points remains in the same cluster. Then, I check if the cluster centroids are mirrored concerning their $y-$ order.

Static: I iterate all clusters in A and check whether a significant amount of points remains in the same cluster in B. Then I check if the clusters are not mirrored.

Based on my experiments, I consider a significant amount of points as 95%. If the relevant condition is met, the transition between two subspaces is detected and highlighted. I include the visualization of visual cues to detected pattern changes, for example, in Figure 3.7 and Figure 3.13: A red line with a green starting and a red ending circle indicates an interesting transition. Clicking on this line opens a detail view containing only the affected subspace projections in a scrollable list with all transition occurrences.

3.4.5 Use Case: University Rankings

In the following, I demonstrate the usefulness of Pattern Trails by application to the 2012 World University Rankings [4] data. The data suits the general idea of Pattern Trails and comprises seven numerical attributes (scores): *Overall, Academic Reputation, Employer Reputation, Faculty Student Ratio, International Faculty, International Students, Citations per Faculty*. The data was studied by Gratzl et al. [79], who presented a comprehensive system to rank the data including customized preferences. In this work, I seek for interesting attributes in terms of subspaces of the data. One can investigate interesting attribute combinations (i.e. subspaces that contain salient patterns) before applying statistic-driven analysis of patterns.

I analyze the data in two different ways: First I examine the data without the use of automatic subspace methods. Then, I perform the analysis again, using an automatic approach before the visual analysis. I show that I come to the same conclusions with Pattern Trails. This also underlines the strength of my approach, as Pattern Trails can also be applied without prior automatic subspace analysis.

[4] US News/QS 2012 World University Rankings: https://goo.gl/t8zzMe

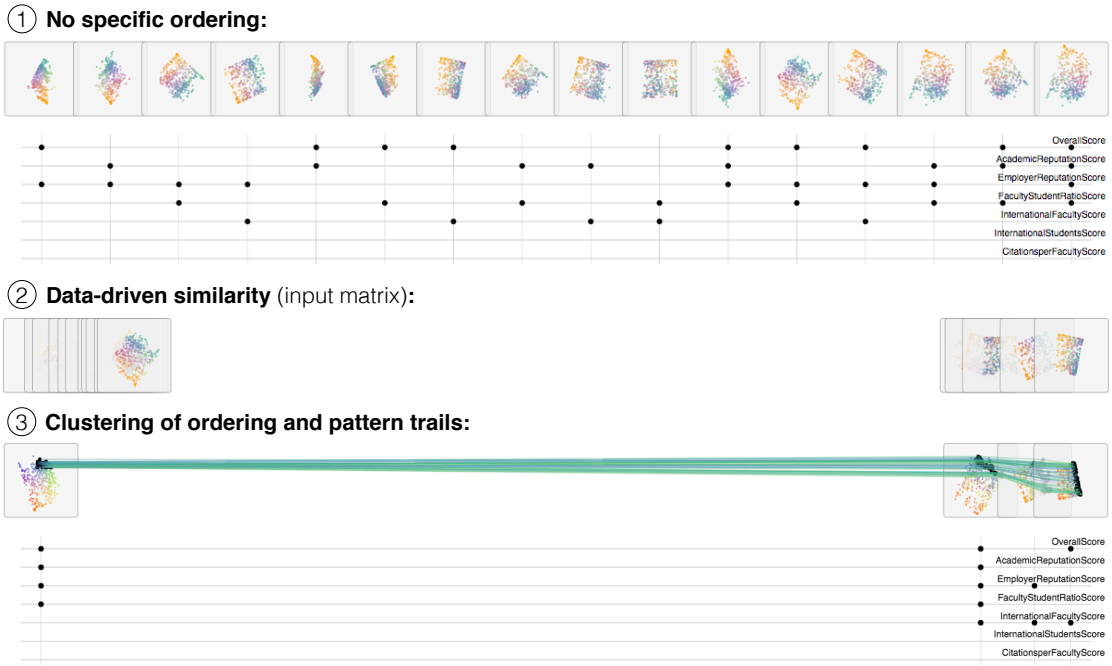


Figure 3.14: *Results of the US News/QS 2012 World University Rankings after application of the SURFING [13] algorithm. (1) The subspace analysis results are shown without any specific ordering. One observation is that the* InternationalStudentsScore *and the* CitationsperFacultyScore *attributes do not contribute to any interesting subspaces. Applying the (2) data-driven similarity measure as well as (3) clustering reveals a pattern transitions that consists of only one split caused by the attribute* InternationalFacultyScore*. The results are consistent with my analysis without the application of prior subspace analysis.*

Manual Analysis of Pattern Transitions

In manual analysis, I load the data, calculate all possible combinations of subspaces, and project them into the 2D space. The data comprises seven attributes. This means, the system has to compute 120 (calculation: $(2^n - 1) - n$, with n attributes) combinations if it ignores all subspaces that consist of exactly one attribute. I then apply the data-driven similarity measure, which is based on the input matrices of the projection. To reduce the 120 subspace projections, I apply agglomerative clustering in addition to the similarity calculation. The result is depicted in Figure 3.7. A known side effect of clustering is the inherent, continuous information loss because subspaces are combined and replaced by a prototype representation to reduce overlap, and thus cognitive load. Figure 3.7 points out an example: the same attribute combinations are projected multiple times. Although the subspaces are different in terms of the attribute space, they can result in the same sets of attributes when clustering. However, the visual representation indicates that these prototype representations are dissimilar to each other asking for looking into detail. Clustering is a trade-off between massive overlap at full detail and partial information loss. Interaction is key to overcome this limitation. I enable the user to go back and forth during clustering to alter between overview and detail. Also, one can apply subspace analysis to reduce the sheer amount of subspaces as described in the next section.

We start to read the subspaces depicted in Figure 3.7 from right to left because the subspaces

on the right are clearly separated based on their similarity. The last two subspaces differ by the attribute *Overall Score* and show a static pattern (**P4**). This means, this attribute has no impact on the pattern structure. This behavior repeats between the subspaces 10 and 11, thus not adding new insight. Note that the same attribute sets generate the next three subspaces 8, 9, and 10 and show static (**P4**) or mirror (**P5**) patterns. Again, we cannot reason regarding the attribute space. Between the subspaces 7 and 8 (highlighted in Figure 3.7), the selected pattern diverges (**P3**) indicating a structural change caused by dominating attributes. To find about which attributes cause this transition, we need to consider the transition between the subspaces 6 and 7. The transition is static (**P4**), meaning that no dominating attribute exists between them. This allows us to compare subspaces 6 and 8 directly. They differ by only one attribute, which is the *International Faculty Score*. We can consider this attribute as being dominant. Furthermore, the attributes *International Students* and *Citations per Faculty* are removed between the subspaces 6 and 7 and have no impact on the transition, which is why we can consider them as being redundant. The remaining subspaces 1 to 5 show static patterns (**P4**), hence their transitions do not reveal additional insight.

Automatically Aided Analysis of Pattern Transitions

In this use case, I aim to confirm the results from the previous manual analysis and show how the automatic support helps to improve the analysis. I load the data and then apply the SURFING [13] algorithm. The resulting 16 subspaces are shown in Figure 3.14. (1) First, we see the result without a specific order. It is striking that the attributes *International Students* and *Citations per Faculty* play no role, which is in line with the results of the purely manual analysis. (2) I apply again the data-driven similarity and (3) cluster the results. Also, I select the data points which are separated from the rest by color (blueish color). From left to right, the diverging (**P3**) pattern develops into a series of static (**P4**) patterns due to the influence of the attribute *International Faculty*. This observation finally confirms the purely manual analysis. Note that we get the same result, but faster, with less effort, and more clearly.

3.4.6 Use Case: Forest Fires

Based on the visual analysis of the Forest Fires [140] dataset, I further elaborate the strengths of Pattern Trails approach. The data comprises seven attributes: *Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), Initial Spread Index (ISI), Temperature (temp), Relative Humidity (RH), Wind, Rain*. To begin with, I load the data and compute interesting subspaces using the SURFING [13] algorithm. Then, I project and visualize the resulting 24 suggested subspaces, as depicted in Figure 3.15. We see the subspaces with no specific ordering and the pattern transitions for a cluster in (1), which I selected in the very last projection. We can see several static patterns that develop into each other by means of diverging and converging patterns.

From left to right, we identify 5 interesting transition patterns. The first 9 subspace projections are combinations of static (**P4**) and mirror (**P5**) patterns, converging (**P2**) into a dense line bundle (**P1**). The converging pattern is caused by omitting the attributes *wind,*

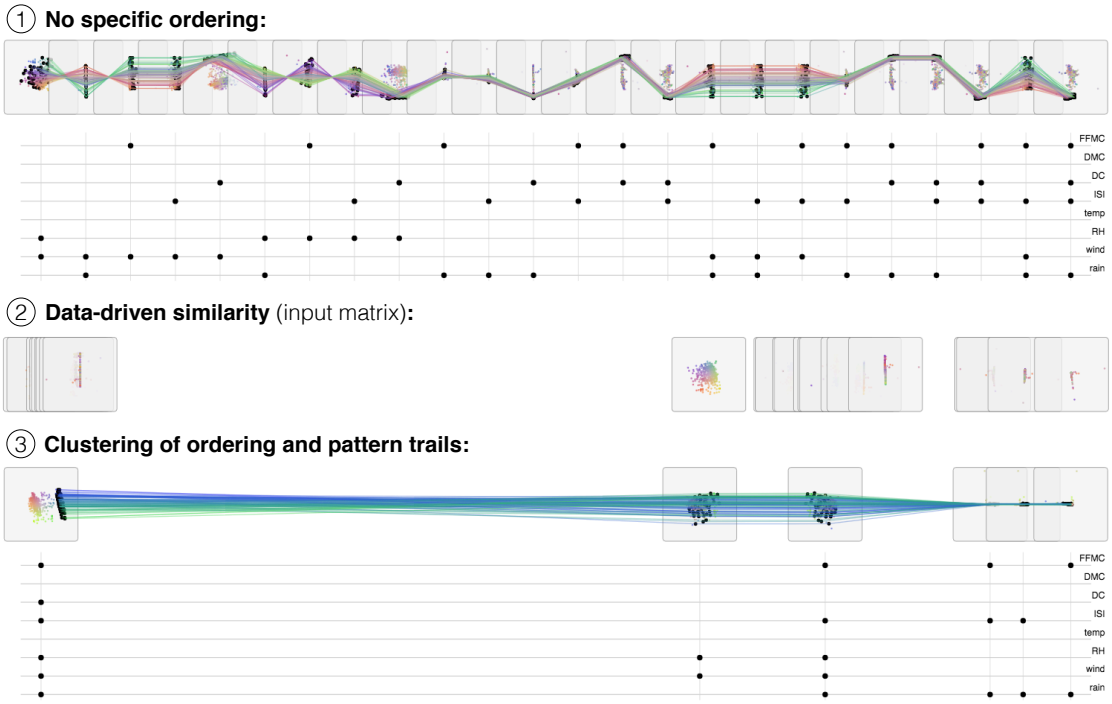


Figure 3.15: *Results of the Forest Fire data set [140] after application of SURFING [13]. (1) Without specific ordering, we identify alternating static and mirror pattern transitions. To expose the role of the attributes in producing patterns, I apply (2) the data-driven similarity and then (3) cluster the subspace projections. The pattern transitions reveals a dominating impact of the attributes* Drought Code (DC), Relative Humidity (RH), *and* wind.

RH. The line bundle diverges when again considering the attribute *wind*. This behavior is repeated two more times. Also, we can identify the attribute *DC* as being present, except for one time, in the pattern **P1**.

Identifying these relations is time-consuming. To speed this process up, I can also apply the (2) similarity-based ordering and then (3) cluster the data to avoid artifacts regarding the pattern transitions. (3) shows the result and the transitions for a selected cluster in the very left subspace cluster. We can efficiently identify that *DMC* and *temp* do not play a major role. Also, we see 3 static (**P4**) patterns that converge (**P2**) by means of omitting the attributes *RH* and *wind*. In addition, the subspace cluster containing *DC* is clearly separated from the rest.

On the interpretation, the attributes *RH* and *wind* cause the projected data to spread, meaning there are more interesting parts in the data that can be analyzed. Adding the attribute *DC* separates the data clearly into multiple clusters. The interpretation follows a logical structure because the relative humidity (RH) and the wind most definitely have an effect on the forest fires. The drought code (DC) has most likely the highest impact on the fires. This is held true for the real world, but also for the data.

3.5 Discussion & Future Directions

In this chapter, I report on a qualitative study that aims to prove whether data analysts of a LEA untrained in advanced statistics can interpret the 2D depiction of DR results. Overall,

the study was well-received by the data analysts. While the study seems very easy for someone trained in advanced statistics, I like to highlight that solving the described tasks by only analyzing the raw data table represents a real challenge. One result was that the data analysts performed a manual multivariate data subspace analysis, which is a tedious task. This motivated Pattern Trails, a visual interactive approach that includes the user to identify and explain interesting attributes, and thus subspaces, in possibly large multivariate datasets. Introducing pattern transitions, I enable the user to analyze the structural changes in subspaces and also provide interpretations for these changes. In the following, I discuss the study as well as the Pattern Trails approach and propose future directions.

Interpretation Study

The study was structured and guided which can be in conflict with the idea of a purely explorative study. However, we posed a question and observed the analyst tackling this question. There were no restrictions regarding time or analysis; as a matter of fact, the analysts started the study by confirming their hypothesis, which can be considered as explorative, however, they had to tackle specific tasks using a minimal set of interaction techniques.

The interaction techniques are a means to an end to solve the tasks. The study also gives insight in the application of the interaction techniques which, however, was not the focus of this study. The findings of this study regarding the interpretation of multivariate patterns were not possible without any interaction possibilities given. A data point in 2D space has a x- and y-value, we cannot assume if this point corresponds to *Monday* or belongs to a category such as *Larceny/Theft*. It becomes even more difficult for correlations among multiple dimensions. The position of each data point conceals multivariate dependencies. There is a need for interaction to draw conclusions about similarities and distances between data points. Also, I applied state-of-the-art interaction techniques allowing us to transfer the results to other systems that use similar interaction approaches.

Study Limitations and Perspectives The study has some limitations I aim to cope with in future work. The study showed that domain experts can understand a multivariate projection if they are familiar with the task and data type. This raises two questions. Are three experts enough to prove this point? Do experts, who are not analyzing data on a daily basis, perform similarly? I specifically aimed for domain experts who analyze data on a daily basis, but are untrained in DR. One can imagine that it is difficult to find participants who qualify for this study, given the recent rise of machine learning and data analytics among industries. Domain experts often apply concepts but cannot look into the so-called black box, where the algorithms are computed. This study represents the first attempt to investigate whether domain experts can interpret the results by steering high-level parameters. The domain experts showed us that one does not have to be trained in DR or advanced statistics to understand DR results. Even though we conducted this study with only three participants, I consider it as representative. In future work, I plan to extend the prototype with more advanced interaction concepts such as touch, the selection of non-circular regions, and the

integration of different data sources. We encountered that LEAs adapt rapidly to and bring forward current research. I plan to extend the study to domains such as finance or health care, also considering different DR approaches. Still, it is difficult to identify experts who work with the data and analyze it, but have not applied machine learning or advanced statistics.

Pattern Trails

Pattern Trails can be considered as an extension to the statistic-driven analysis of clusters, outliers, and correlations that occur in a subspace. The design space of the Pattern Trails approach yet opens questions regarding different analysis steps that I aim to discuss at this point.

Pattern Selection Given a number of subspace projections, there exist various strategies to identify a pattern (e.g., a cluster) as a starting point to analyze its structural changes. An effective approach is to select all points within a subspace. Because the data is encoded with a 2D colormap, one can identify major structural changes in the data transitions. While this approach works fine for small data sets, it may result in a cluttered representation for larger data sets which impedes the identification of interesting subspaces. Using my data-driven similarity and applying Agglomerative clustering reduces the number of subspace projections. Aggregation reduces the displayed data, yet at the expense of information. The more attributes are combined, the less expressive the subspace projection can get. Therefore, it is up to the user to strike a balance between aggregation and aspired information.

Another strategy is to provide automatic support for the application of data-driven clustering algorithms that point out the salience of patterns in subspaces. However, such algorithms do not point the user to structural changes across subspaces, which is why I go one step further and provide automatic support for identifying meaningful transition pattern. By clicking on a transition, the user can explore all contributing patterns and refine the query.

Attribute Redundancy The notion of redundant and dominant (or expressive) attributes raises the question for: Should users omit redundant attributes? That may depend. On the one hand, we seek for attributes which determine the structure of the subspaces, which is why we can argue that all other attributes may not sufficiently contribute in terms of information. On the other hand, attributes considered to be possibly redundant can still be classified by domain experts as important with respect to their task at hand. Consider for example 10 attributes, of which 3 attributes determine the structure of the subspace. The remaining 7 attributes are not of major concern regarding the subspace structure, however, contain information that can be relevant to get insight and bring attributes into context. For this reason, I included Pattern Trails into the Visual Analytics process, so that it is up to the user to decide whether certain attributes and subspace structures are of interest.

This observation is also related to the interestingness of a subspace, which can be considered as interesting based on the set of involved attributes, or its structure. With my data-driven similarity approach, I make a first step towards including a visual structure-based analysis, which I aim to extend in the future.

Expressiveness of Subspace Projections The visual representation of subspace projections brings in challenges regarding the expressiveness and uniqueness of the subspaces. First, multidimensional scaling is known to be not invariant to rotation, which is why I introduced the mirror transition patterns. This transition is salient (e.g. in Figure 3.7), but holds the same interpretation as the static pattern. In addition, I rotated all small multiples to provide an extensive view on the subspace projections and not to leave the user with ambiguities. Second, the curse of dimensionality [17] impedes the interpretation of the structure of a subspace projection, because many attributes can contribute to the structure but not all can be fully taken into consideration when projecting to 2D space. I tackle this issue by providing a full list of all involved attributes per projection, yet do not overcome the curse of dimensionality in principle.

I applied the SURFING [13] algorithm in this approach because it is parameter-free and proposes interesting subspaces based on their structure. I am aware of a variety of other algorithms but like to leave their integration for possible future work since they are not an essential part of my claimed contributions.

Subspace Ordering Using Pattern Trails, I layout all subspace projections side by side and encode similarity between projections via distance. This approach introduces a meaningful ordering regarding groups of subspaces that are clearly separated. Incorporating different distance functions, however, can introduce diverse orderings that may affect the interpretation of patterns. Although my approach is invariant to orderings, we may come to different conclusions when applying clustering to similar subspace projections. This is due to different groups that comprise different attributes. Therefore, it is of highest interest to confirm findings using state-of-the-art statistical analysis methods or to also investigate attribute combinations without clustering.

Different orderings also introduce possibly arbitrary combinations of transition patterns. I consider the taxonomy presented in Section 3.4.2 as complete in terms of single transformations and the identification of redundant attributes, but plan to extend the taxonomy for arbitrary combinations and their interpretation in the future.

Scalability I encounter major scalability challenges concerning the amount of visualized subspace projections and data records. For the amount of visualized subspaces, I propose to apply my data-driven similarity followed by Agglomerative clustering. This proceeding presents a trade-off between a very detailed view on the data and a condensed view on possibly interesting parts. To tackle the number of data records, I apply a 2D colormap to identify overall structural changes but am aware that this is not an optimal solution. For future work, I imagine including scatterplot-based aggregation techniques such as heatmaps, and bundling techniques for the transition connectors. I further aim to apply my approach to high-dimensional datasets beyond 20 dimensions and to incorporate different subspace analysis methods including a comparison with other subspace analysis tools such as the work of Tatu et al. [198].

I implemented the prototype using a client-server architecture, which allows handling large amounts of data. However, the computation and results of subspaces and projections are affected by the data characteristics (e.g. amount of attributes), traced to the curse of dimensionality [17], the data types, and the data size.

4

Visual Analysis of Temporal Multivariate Patterns

Contents

In the previous chapter, I considered the structural change of patterns among different subspaces of the data. I intentionally left out a possible temporal aspect in the data; a temporal evolution so to speak. Multivariate data, however, can evolve over time. Examples include data from computer networks, healthcare, social networks, or financial markets. Often, patterns in such data evolve among multiple attributes and are hard to detect and interpret. Even though multivariate projections enable analysis and visualization of multivariate data, they per se do not provide means to explore patterns over time. I propose Temporal Multidimensional Scaling (TMDS), a novel visualization technique that computes sequential one-dimensional MDS plots for evolving multivariate data. Using a sliding window approach, MDS is computed for each data window separately, and the results are plotted sequentially along the time axis, taking care of plot alignment. TMDS plots enable visual identification of patterns based on multidimensional similarity of the data evolving over time. I demonstrate

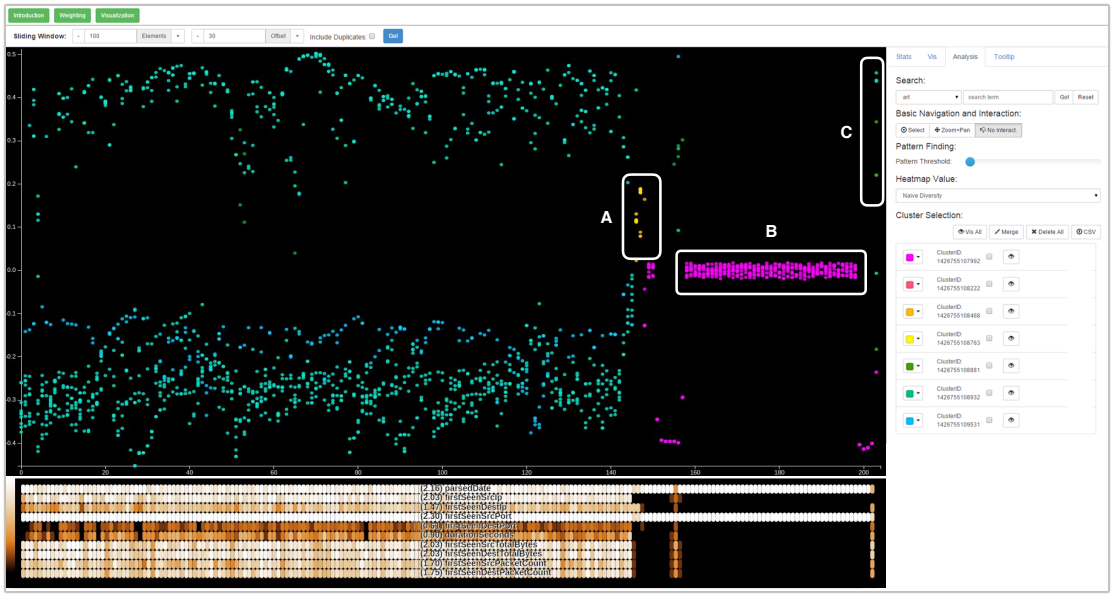


Figure 4.1: *Temporal MDS plots (top) applied to network traffic data, which is part of the VAST Challenge 2013 MC3. The sequentially aligned diversity matrix (bottom) provides an overview of correlations among attributes based on the Shannon Entropy. The matrix is aligned with the temporal MDS plot so that each MDS window is represented by one column in the matrix. The visualization shows the application of TMDS to the first day on 2013-04-01 00:00 to 23:59 and reveals three different patterns. Pattern A and B point out a subtle event that corresponds to port scans. The blue and green patterns on the top and the bottom correspond to normal legitimate incoming network traffic. The pattern B is followed by port scans, indicated by pattern C.*

the usefulness of this approach in the field of network security and show how users can iteratively explore the data to identify previously unknown temporal evolving patterns.

This chapter is based on [102]:

> **Temporal MDS Plots for Analysis of Multivariate Data.** D. Jäckle, F. Fischer, T. Schreck, and D. A. Keim. *IEEE Trans. Vis. Comput. Graph. 22(1): 141-150, 2016.*

4.1 Introduction

Especially in the exploration phase, a key task in understanding multivariate data is to group it into a set of discernible areas, aforementioned introduced as patterns. Building on Chapter 3, real-world data is often not only multivariate, but also evolves over time, posing a challenge to detecting such patterns visually.

For example in the field of network security, threats show different and often very dynamic behaviors. In so-called port scans, attackers explicitly search for open ports to receive access to a given system. However, such computer threats constantly evolve over time and change their behavior. Significant amounts of data need to be analyzed without any prior knowledge

about which threats to expect, and when. Existing approaches such as supervised machine learning, typically rely on classifiers and thus prior knowledge to detect or predict patterns. Visual data analysis is promising in helping explore and analyze data in an unsupervised way. Prominent examples of successful multivariate visualization techniques include, e.g., parallel coordinates and glyph-based techniques. However, many visualization techniques cannot directly consider the temporal aspect, or only for selected single attributes. The so-called time series path techniques and others (see Section 4.2) consider multivariate temporal data but lack presenting temporal multivariate data with multiple events at a time.

In this chapter, I propose *Temporal Multidimensional Scaling (TMDS)*, a novel approach that sequentially aligns multivariate projections to make temporal patterns salient. TMDS visually presents the data enabling analysts to identify patterns and explore the data space, following the visual analytics process. This approach is based on two driving questions: *Firstly, how to process and visualize temporal multivariate data to allow analysts explore patterns? Secondly, once a pattern has been identified, how can we automatically find similar patterns?* TMDS applies a sliding window approach on the data and computes a one-dimensional (1D) MDS for each window. The resulting sequence of 1D MDS mappings are then organized along the temporal axis: The x-axis represents the time, and the y-axis represents the MDS similarity value. Similar events are grouped over time and can efficiently be identified. To analyze the multivariate nature, I augment the visualization with a sequenced diversity matrix aligned with the MDS plot revealing the different temporal behaviors of single variables. The diversity matrix provides additional insight into the data and thus fosters interpretation. Furthermore, I introduce a new algorithm to find similar patterns based on the user selection and the behavior along attributes. TMDS enables the efficient detection of recurring patterns and further allows to identify the evolution of patterns, being based on varying scales and intervals.

This chapter builds upon Chapter 3 and introduces sequential projections to identify, and a diversity matrix to interpret temporal multivariate patterns. This chapter is organized as follows. First, I discuss related work in Section 4.2. In Section 4.3, I give a brief example and provide a three-step pipeline to derive TMDS. In Section 4.4, I propose two extensions, which enable the user to visually and automatically identify patterns. I further show the usefulness of my approach in a case study with application to network security in Section 4.5 and provide a discussion of results in Section 4.6.

4.2 Related Work

The following overview builds upon the work outlined in Chapter 2, and brings the discussed techniques into the context of multivariate temporal data visualization. Typically, these visualization approaches do not explicitly consider the temporal aspect of multivariate data. Either the temporal behavior is visualized only for a single attribute, or a statistical aggregate of the multivariate variables. A common approach to combine both, the temporal as well as the multivariate aspect, is the application of small multiples [203]: This way, multivariate visualizations can be tracked over time. However, visualizations are sequenced forcing the

user to split perception attention which can impede accurate identification of temporally evolving patterns. An extensive survey of other visualization techniques for time-oriented data is found in [2].

4.2.1 Time-Dependent Dimensionality Reduction

Analysis and visualization of multivariate data are in general a difficult problem. The so-called curse of dimensionality impedes the ability to compactly visualize and identify patterns in multivariate data [17, 87]. DR techniques target to detect and consider only interesting attributes and their relation to each other for analysis. For example, SOMs [129] can be utilized to group data based on their similarity to each other into a predefined layout. PCA [117] and MDS [47] among others, are not tied to producing 2D layouts but project the data into a predefined n-dimensional space. Results represent a linear or non-linear combination of the original attributes [150]. DR techniques are widely used for the analysis and visualization of multivariate data. Multivariate data is yet often temporally evolving and the dimension of time needs to be considered.

Dwyer et al. [54] take first steps and propose to map time to the third dimension of a 2D baseline MDS plot. Conceptually similar to a space-time cube, changes over time can be tracked in a 2.5D view. Several approaches have been presented [20, 93, 151, 213], which project multivariate data to two attributes. The idea is that a single data entry is tracked over time, and the path of such single entry is visualized in the resulting projection. While these techniques may allow detecting e.g., cyclic patterns, the displays may also quickly lead to cluttered structures, as the data items may be plotted to arbitrary (x,y) coordinates and following them in 2D is a perceptually arduous task. Crnovrsanin et al. [48] therefore discuss the usage of 1D plots for temporal analysis of movement data using PCA.

4.2.2 Delineation to Temporal MDS

The methods above track time for single data records (entries in a data table) that give rise to a temporal behavior in a 2D plot with continuous coordinates. Existing approaches typically consider a discrete projection for every time point, which may include abrupt changes in the (x,y) position of the data item location. In this approach, I propose a temporally smoothed version of time-dependent multivariate plots, by considering a sliding window approach with overlap to make the projection. Thereby, the resulting temporal patterns show higher smoothness, which acts as a filter to suppress abrupt changes which, however, may only have very limited local support. TMDS can emphasize temporal evolving patterns of multivariate data, whose entries are not associated with each other but possibly share a common behavior across several attributes. By using a 1D layout, I also introduce a simpler linear structure (as opposed to 2D trajectories), which analysts can easily follow and link to data details. As the case study in Section 4.5 shows, this approach allows identifying patterns at various resolutions and interval sizes, which can be effectively tracked by the analyst.

4.3 Temporal Multidimensional Scaling

The Temporal Multidimensional Scaling (TMDS) approach computes aligned temporal 1D MDS plots for multivariate data. TMDS apply a sliding window to the temporal sequence of multivariate data and compute the MDS for each slice separately. Note that this approach applies to sequential as well as temporal data and therefore requires the data to be loaded in ascending temporal order, making it also suitable for real-time applications. The results are then plotted sequentially along the time axis.

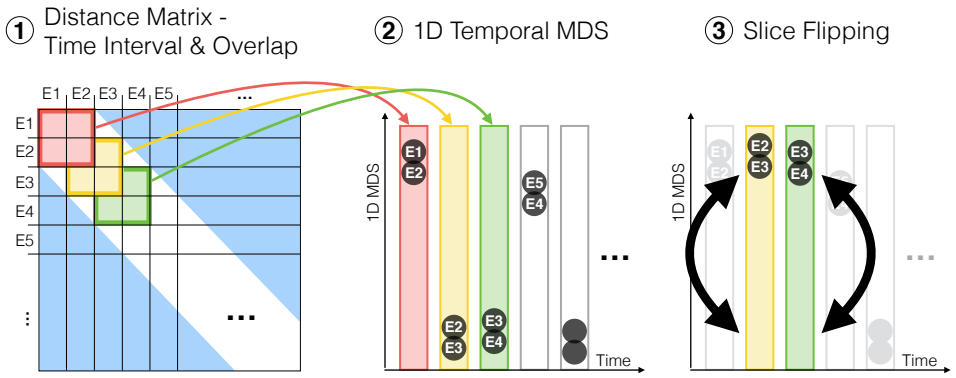


Figure 4.2: *Three-step pipeline for the TMDS computation. A dataset comprises the entries E_1 to E_N (each entry holding multiple attributes), which are processed in temporal ascending order. (1) Based on the weighted distance matrix, a sliding window with overlap is applied and (2) 1D MDS computed for each window separately. The result is sequentially aligned on the time axis. (3) Because MDS is not invariant to rotation, I apply a slice flipping heuristic.*

To implement TMDS, I define the following three-step algorithm (Figure 4.2):

1. **Sliding Window:** Run sliding window with overlap and user-defined parameters along the sequence of data items, and compute the distance matrix for all entries in the given window.
2. **Temporal 1D MDS:** Apply MDS to the distance matrix of each window step. The outcome of each computation is a one-dimensional ordering of the multivariate records and the basis for the sequential visualization.
3. **MDS Slice Flipping:** MDS is not invariant to rotation. Similarity values upon multiple TMDS computed slices may not share same similarity positions but be rotated by 180 degrees. Thus, temporally evolving patterns may not be identifiable. A canonical orientation of the 1D MDS orderings needs to be obtained. I propose to use a heuristic based on the orientation of the entries included in the overlap. Salient patterns are marked in Figure 4.3 (3), which were not equally clear without the flipping test.

Following, I outline the basic idea in Section 4.3.1 and give a detailed overview of these steps. Starting in Section 4.3.3, I discuss the sliding window, followed by 4.3.4, where I introduce TMDS and slice flipping.

4.3.1 Basic Idea

The TMDS approach follows a visual analytics process and allows the user to browse the data and to refine parameters and inputs. TMDS is suitable for the analysis of any temporal evolving multivariate data, either numerical or categorical. This approach first allows the analyst to select and weight single attributes of a multivariate input data set, according to their importance. Then, TMDS applies a sliding window of given size and overlap to the time-dependent data. For each window, a 1D MDS analysis is performed. This way, similar entries are grouped accordingly over time, based on the correlation of attributes. The analyst may also adjust the weighting of the attributes so that attributes of interest can be prioritized and the task requirements are met. The re-computation of the MDS may reveal several temporal aligned patterns, as depicted in Figure 4.1. They are spatially separated along the y-axis (similarity) but still evolve over time (x-axis). The patterns are presented as groups of entries that evolve over time and are aligned on the same similarity level; they can be validated in combination with the diversity matrix plotted below the MDS. The diversity matrix is aligned with the window and quantitatively presents computed diversity indices among attributes through color. It helps the analyst to understand correlations between attributes and thus to draw conclusions efficiently. The analyst further selects a salient pattern and runs the algorithm to find similar patterns. Similar patterns are then visually highlighted and separated through color, and listed next to the visualization. Using visual analytics, the analyst re-configures TMDS utilizing identified similar patterns as new input. This way, data can be explored based on new insight. An example result is presented in Figure 4.1. The selection is highlighted in magenta. The prototype provides details on demand and thus allows the analyst to inspect the raw data of the selection, which reveals a distributed brute-force attack to different servers.

4.3.2 Similarity Computation

MDS is a well-known DR technique, used to preserve similarities across multivariate data. Compared to e.g., PCA which requires co-variances, MDS requires a distance matrix as input, which defines the similarity between entries. Distances can be computed for various data types, including the numerical distance between numerical values, the binary distance between strings, the cosine distance between documents in vector space, among others. Hence, MDS suites our needs of handling categorical data, which we encounter for instance in network security data (Section 4.5), healthcare data, and finance data. The distances for categorical data are computed as follows:

$$distance(A,B) = \frac{\sum_{i=1}^{|attr|} [A_i \neq B_i] \cdot w_i}{|attr|} \tag{4.1}$$

The distance or dissimilarity between two entries A and B is computed by first iterating all attributes from $i = 1$ to the total amount of attributes $|attr|$. Using Iverson Brackets [78], the i-th attribute of the entries are compared with each other, and the result is multiplied

by the attribute weight w_i. I then compute the average by dividing by the total amount of attributes to derive the final distance value.

Weighting single attributes has direct impact on the similarity calculation and thus the 1D MDS plot. TMDS allows the user to define the weights w_i for each attribute individually. Especially domain experts typically know attributes that are of interest for certain tasks at hand and can make use of their domain knowledge to influence the computation and the exploration of temporal patterns. The step of weighting becomes tedious if the data includes loads of attributes. Hence, our approach supports predefined weightings, which are presented as suggestions. Suggestions are predefined but facilitate and speed up the analysis if combinations of attributes reoccur among different datasets. Weighting attributes is possible in the range of $[0,1]$. The lower the weight is, the less impact the weighted attribute has in the MDS analysis. For example, an attribute weighted with 0 is excluded from the computation of the distance matrix.

4.3.3 Sliding Window

To find temporally evolving patterns within multivariate data, I apply the sliding window approach to the data and compute the distance matrix for each window separately, serving as input to the 1D MDS computation. The sliding window approach is suitable because it sequentially processes the data taking its temporal behavior into account. This way, patterns are connected step-wise (window-wise) and do not require additional cognitive load as it is the case using for example small multiples [203], among other spatially disconnected visualizations.

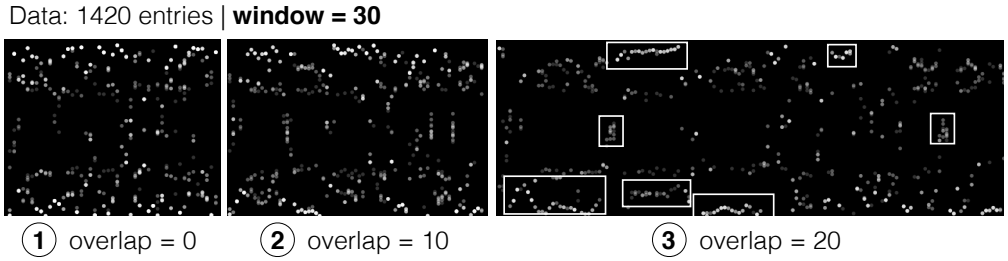


Figure 4.3: *Sliding window approach applied to a multivariate dataset with 15 attributes and 1420 entries. For a window size of 30 entries, I applied an overlap of (1) 0 entries, (2) 10 entries, and (3) 20 entries. The bigger the defined overlap, the more slices are computed, resulting in a visualization whose layout becomes stable and reveals salient patterns.*

Suppose we compute the MDS for several windows without any overlap. The result is 1D, which means that the attribute with the highest variance has the highest impact on the result. In this scenario, the windows do not share any relation by means of reused entries. This means, for each window the attribute with the highest variance can be a different attribute, which results in an unstable layout. Hence, it would be harder to identify possible patterns. Figure 4.3 shows the effect of the sliding window approach applied to a dataset that contains 1420 entries with a window size of 30 entries. (1) shows the result of TMDS without any overlap. As expected, the sequence of 1D projections is not smooth but fluctuates significantly.

With increasing overlap, more windows are computed. (2) shows the same result, but with an overlap of 10 entries. Patterns do already stand out prominently. In (3), the overlap is 20 entries, even further increasing the stability, clearly showing evolving temporal patterns. Figure 4.5 (3) depicts the path of reused entries that are included in the chosen overlap. Reused entries evolve on the same similarity level not causing a distorted perception of patterns.

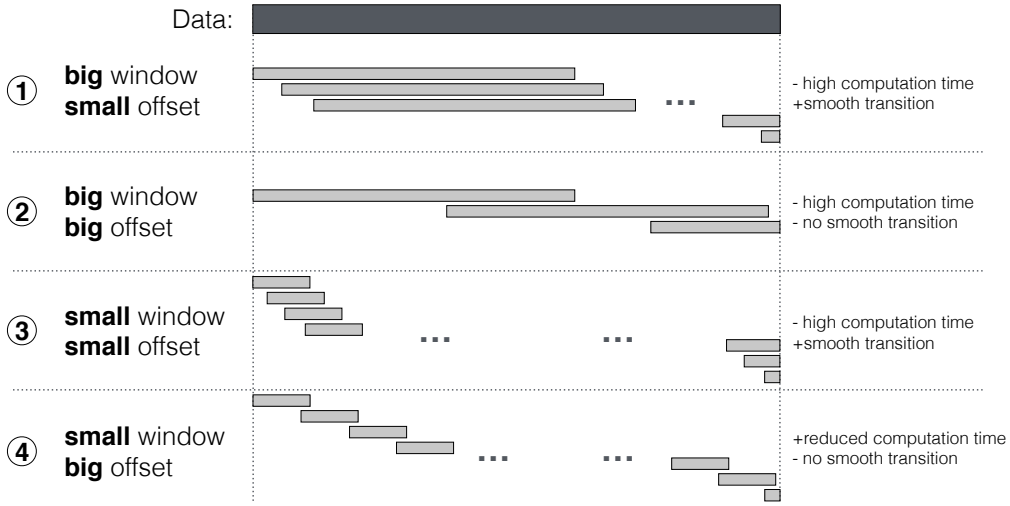


Figure 4.4: *Distinction of cases on window size and offset.*

Adding overlap to the sliding window results in a smooth transition in the patterns. In the final visualization, reused entries are discarded and each presented slice contains the same amount of data entries according to the window size. The complexity of computing the distance matrix for n entries and m dimensions lies in $\mathcal{O}(n^2 \cdot m)$ in the worst case, meaning that the chosen window covers all entries. The application of a classical MDS per window results in $\mathcal{O}(n^3)$ due to the expensive step of performing the eigendecomposition [218]. Thus, the overall complexity lies in $\mathcal{O}(n^3)$. However, in practice, the window size is not n and thus the computation time differs on window size and offset. Figure 4.4 depicts the distinction of cases on window size and offset. Offset refers to the number of entries a window is moved – small offset implies high overlap. Following, I outline the four derived cases relating to Figure 4.4:

1. A *big window* size and a *small offset* results in high computation time but provides a smooth transition in the patterns.
2. A *big window* size combined with a *big offset* requires high computation time and possibly does not provide smooth transitions.
3. Applying a *small window* size and a *small offset* still requires high computation time due to the number of computed windows. However, transitions in the patterns are likely to be even smoother compared to case (1), because of the small window size and the reduced amount of considered entries per slice.

4. In contrast, the application of a *small window* size and a *big offset* reduces the computation time but does not provide smooth transitions in the patterns.

I assume it is desired to have low computation time and smooth temporal slices. A big window size results in increased computation time since both distance matrix and MDS are computed for a larger amount of entries. Then choosing a small offset increases the computation time even more. I argue that regarding the distinction of cases, cases (3) and (4) are most promising on computation time and offset. For now, I use pragmatic rule-of-thumbs to set the offset, e.g., using 10% of the window size with good results. Automatically finding appropriate overlap sizes is left to future work, with one idea to use a visual salience function to search automatically for parameters that show interesting visual results. Techniques based on Hough transform analysis, or other interest measures may be a starting point to this end [197]. A drawback of smoothing is the possibly hiding of outliers. By increasing the window size and thus adding data entries to a slice, former outliers are likely to join a cluster if the discrepancy decreases among entries. However, outliers that show great discrepancy to all other entries are preserved.

4.3.4 Temporal 1D MDS and Slice Flipping

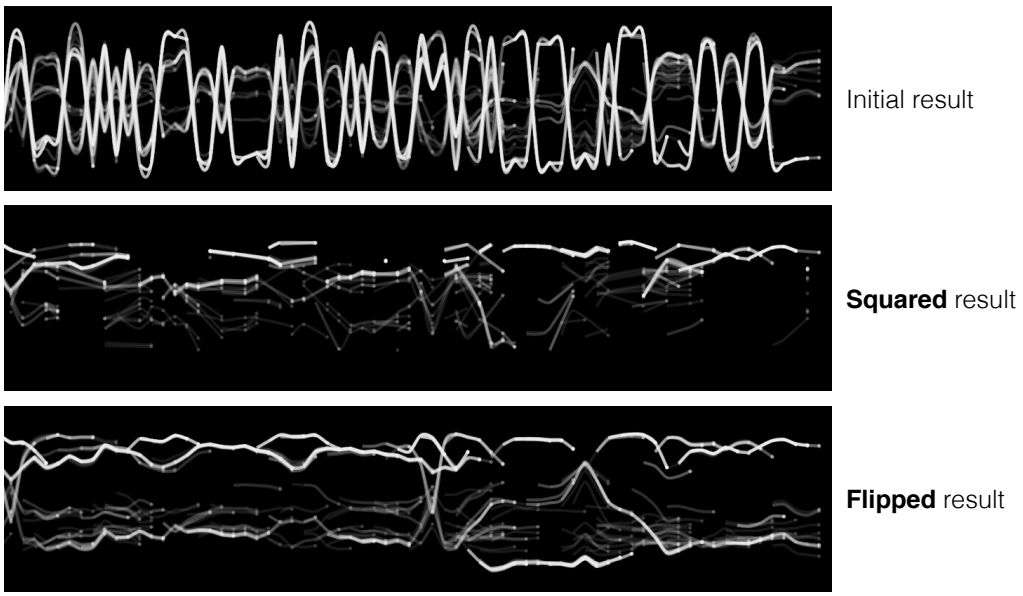


Figure 4.5: *The visualization shows the paths of reused entries of subsequent MDS slices. (1) shows the initial temporal result. MDS is not invariant to rotation. I therefore propose two heuristics to the result: (2) Square all results to level them. (3) Flip slices if more than half of reused entries change their leading sign in the subsequent slice.*

Based on the sliding window, the MDS is computed for each window separately and then sequentially aligned in the Cartesian coordinate system. The x-axis is the time, and the y-axis is the 1D similarity value derived from the MDS computation. The numerical range of similarity values along the y-axis is always restricted by the bounds $[-0.5, +0.5]$. This is because the computation of distances (see Equation 4.1), including weighting, is normalized

to the range of $[0, 1]$. I created a pathline visualization to test how good temporal patterns evolve; it visualizes the paths of data entries (overlap) separately. Figure 4.5 (1) shows the initial result. As illustrated, the pathline alternates between positive and negative values within the bounds $[-0.5, +0.5]$. An alternating leading sign for a majority of reused entries (indicated by alpha-blending) in subsequent slices indicates that also co-located points are likely to alternate their leading sign, meaning that possible temporal patterns are interrupted. Patterns evolve on opposite sites instead next to each other. A reasonable solution is presented in Figure 4.5 (2): all results are squared leading to a leveling of the results and thus patterns. Squaring the results has the side effect that adjacent mingled patterns are additionally spatially separated. However, squaring the same negative as well as positive value leads to an overplot of patterns, because they are leveled to the same positive position. This is why we do not want to discard negative values. Therefore, I decided to check subsequent slices, if the leading sign of reused entries changes. If more than half of the reused entries switch their leading sign, I change the leading sign of all entries contained in the subsequent slice (slice flipping). Thus, I compare the next but one slice to the current subsequent slice, and so on. Using a lower or higher threshold for flipping might not improve the visualization in quality because either too few or many slices are likely to flip. The result of this heuristic is presented in Figure 4.5 (3), with the effect that additional temporal patterns become visible.

4.4 Visual and Automatic Pattern Detection

The TMDS visualization provides an aggregate view to a time series of multivariate data. Specifically, the sequence of 1D MDS plots shows gradual changes of 1D distributions of entries, reflecting on the similarity relationships between the data entries and their change over time. However, the 1D plots do not show the behavior of the underlying attributes. This detail information is needed in two cases: Firstly, if patterns are visually separated, the user needs details on the attributes to find possible explanations for the MDS pattern. Secondly, increasing the number of attributes can result in less prominent visual patterns; having information on correlations of attributes helps to interactively select (reduce) individual attributes and facilitate the analysis by excluding potentially irrelevant attributes. Therefore, I provide facilities to manually brush a data region and browse corresponding entries to find correlations among attributes. However, this task is challenging even for a small set of attributes, because various attribute permutations need to be taken into account. Following, I introduce two techniques: A visual approach using a matrix that visualizes correlations among attributes and windows, and an automatic approach to finding similar patterns, based on a user selection.

4.4.1 Visually Identifying Patterns

The visual identification of multivariate patterns is mapped to identifying salient patterns within the TMDS visualization. To support the process of finding patterns also attribute-wise, I introduce a diversity matrix as a heatmap, which is displayed below the TMDS plot, aligned

with the sliding window. In this heatmap, each column corresponds to one window and each row to one attribute.

Diversity quantitatively reflects the amount of differing types or values within the attribute and can be computed in various ways. It helps the user to draw conclusions based on the diversity correlations which describe the attributes. I implemented two information theoretic measures to assess the diversity of attribute values per window. Considering that, I determine the different categories (following referred to as i) of values per attribute by binning. The *Shannon Entropy H* is computed as follows [183]:

$$H = -\sum_{i}^{n} p_i \cdot log_2(p_i) \tag{4.2}$$

According to the definition of the Shannon Entropy, p_i describes the proportion of a character i occurring in a string. Applied to our scenario of having multivariate categorical data, p_i describes the probability of category i occurring within an attribute. In contrast, the *Simpson Index D* is computed as follows [186]:

$$D = 1 - \sum_{i}^{n} \frac{m_i \cdot (m_i - 1)}{m \cdot (m - 1)} \tag{4.3}$$

m_i describes the number of occurrences of category i and m the total amount of the categories per attributes. I apply *min-max* normalization (feature scaling) to color code single diversity values on all attributes and windows. The used colormap maps low diversity to black and high diversity to white along with a brown gradient. Figure 4.6 shows the use of both the Shannon Entropy and the Simpson Index diversity measures for a test dataset of the domain of network security, containing approximately 25.000 entries and 16 attributes. We observe a high correlation between the Entropy and the Simpson index in the applications. While I implemented these two in the prototype, additional information theoretic measures can be chosen, depending on task and data. The diversity heatmaps provide an overview of the diversity of attributes and their changing over time. It is useful for identifying correlated attributes for attribute filtering. At the same time, it is useful to compare changes in the MDS plot with changes in the diversity across the data attributes, for exploring a) interesting time slides, and b) obtaining starting points for explaining the MDS patterns by properties of the underlying multivariate data.

4.4.2 Automatic Detection of Similar Patterns

In the previous Section 4.4.1, I describe how patterns can be visually identified using the proposed TMDS in combination with the diversity matrix. Salient patterns that can be identified using the TMDS, typically consist of several data entries building a temporal cohort. This means the entries share alike similarity values over a certain period. Such patterns can be salient during one period, but hidden during another period. Patterns are declared as hidden if they correlate with other patterns and/or are distributed within other patterns.

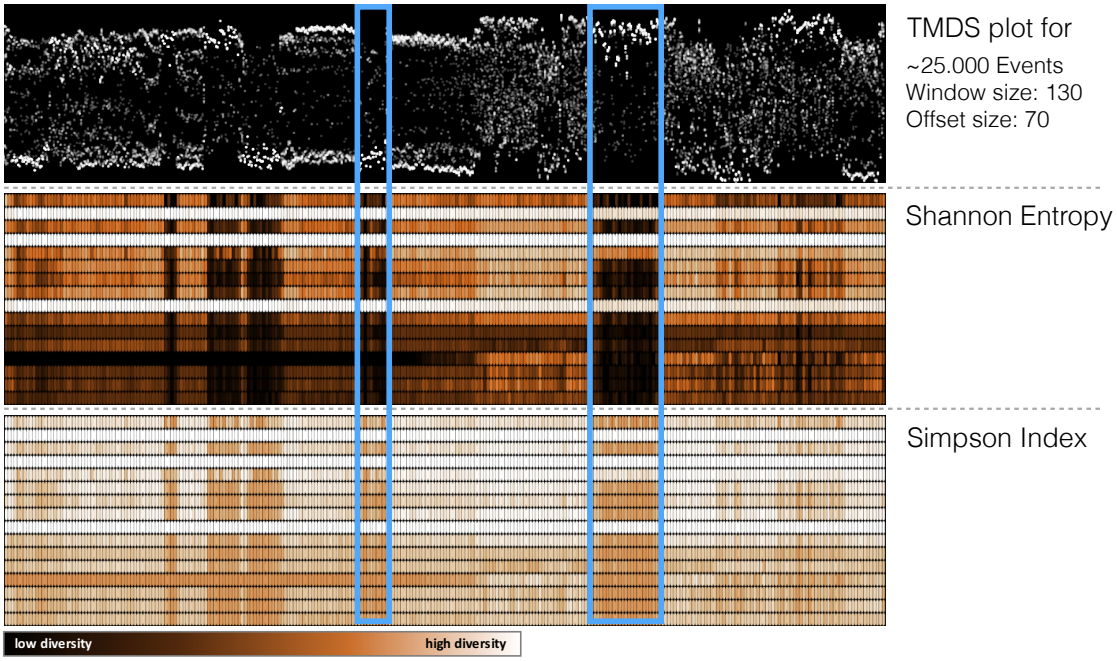


Figure 4.6: *Diversity Matrix. Application of Shannon Entropy (middle) and Simpson Index (bottom). Columns are temporally aligned with the TMDS (top). Diversity is mapped to color (black is low diversity and white indicates high diversity) and reveals correlations between attributes (rows).*

This effect can occur because of the sheer amount of other data entries which can influence the projection. To reveal other related patterns, I offer the user the option to manually select corresponding data entries, based on which related cluster of entries are automatically computed and highlighted. I argue that using an approach similar to the computation of the distance matrix (proposed in Section 4.3) will reveal similar patterns aligned with the layout of the TMDS.

Function *findSimilarPatterns*$(\mathbb{D}, \mathbb{S}, threshold)$
  $d \leftarrow 0$
  $similarities \leftarrow [\,]$
  foreach *entry* E *in* $\mathbb{D}$ **do**
    $d \leftarrow distance(E, \mathbb{S})$
    $similarities \leftarrow sortedInsert(d, similarites)$
  end
  return $1D\_DBSCAN(similarities, threshold)$
End

Algorithm 1: Find similar patterns.

Algorithm 1 provides a short overview of how similar patterns are found. Input are the overall data $\mathbb{D}$, the selection $\mathbb{S}$ and a user-defined $threshold$. I follow two consecutive steps: Firstly, calculate a distance value per entry and user-defined selection. Secondly, sort all distance values and cluster them. I compute the per entry distance d between each entry and

the selection as follows:

$$distance(E,\mathbb{S}) = \frac{\sum_{i}^{|attr|}\left(\sum_{j}^{rows(\mathbb{S})}[E_i \neq \mathbb{S}_{i,j}]\cdot w_i\right)\cdot\frac{1}{rows(\mathbb{S})}}{|attr|} \tag{4.4}$$

In contrast to Equation 4.1, I transpose the input and calculate the distance per attribute instead of per entry. The distance between one entry E and the user-defined selection of entries $\mathbb{S}$ is computed by first iterating all attributes from $i = 1$ to the total amount of attributes $|attr|$. Then, all rows for the *i-th* attribute are iterated from $j = 1$ to the total amount of rows (entries) $rows(\mathbb{S})$. Using Iverson Brackets [78], the *i-th* dimension of the entry is compared to all *i-th* attributes of the selection $\mathbb{S}$ and then the average is computed. To determine the final distance value d, I compute the average value of all attributes. This distance value, determined for each data entry individually, is inserted into a list which is sorted in descending order on the distance value. Clusters of similar entries, compared

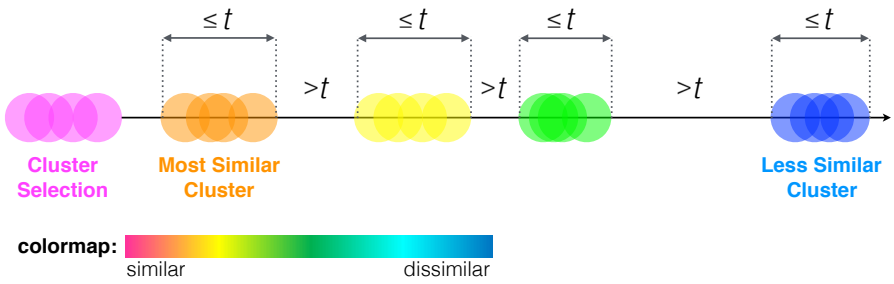


Figure 4.7: *One-dimensional DBSCAN algorithm for similarity values using a user-defined threshold t.*

to the user-selection, are further derived by performing a one-dimensional density-based clustering (using the DBSCAN algorithm [59], see Figure 4.7). I set the threshold to 0.01 per default (applied to all examples in this chapter). The threshold depends on how granular the user wants to find similar patterns. The higher the threshold, the higher the distance between found clusters is, allowing a high discrepancy within the clusters. The density-based algorithm performs as follows: Considering the threshold, the algorithm starts at the first entry of the similarity list and successively compares the similarity value n (currently visited) to the value $n+1$. I distinguish between three cases:

- If the difference between those two values is smaller than the threshold, the entries are combined to a pattern cluster.
- If the entry of n already belongs to a pattern cluster, the entry for $n+1$ is added to the cluster.
- If the distance is greater than the defined threshold, the entry for $n+1$ starts a new pattern cluster.

Using this algorithm, I adapt to the creation of the distance matrix, which influences the outcome of the TMDS. This way, the algorithm finds patterns that already have been considered

by the MDS computation. To derive the complexity of the proposed algorithm for finding similar patterns, I split the algorithm into the three elementary parts: Firstly, the complexity of computing the distance of all points to the selection lies in $\mathcal{O}(n^2)$. Secondly, the complexity of sorting all similarity values is $\mathcal{O}(n \cdot log(n))$. Thirdly, the 1D DBSCAN has the complexity $\mathcal{O}(n)$. Hence, the overall complexity of this algorithm lies in $\mathcal{O}(n^2)$ in the worst case.

4.5 Case Study: Network Security

In this section, I describe the application of TMDS in the field of network security. Recent work regarding the analysis of network security data focuses either on temporal independent patterns like, for example, shown by Fischer et al. [63], or on large-scale temporal patterns that thus cannot be analyzed promptly, as addressed by Goodall et al. [75]. Because, this case study discusses, among others, identified port scans in the data, I particularly delineate the TMDS approach from the PortVis approach [155]. At first sight, TMDS looks similar to PortVis. However, PortVis is tailored to the identification of port scans and uses a two-dimensional representation, where the axes are time and port range. TMDS can find port scans but is a general approach that also finds any other patterns based on user-defined weighted attributes.

In the following, I first apply the technique to an artificially generated dataset to showcase the validity of results. Then, I report on a conducted analysis together with a domain expert, which further motivates the need for a technique such as TMDS in a real world application. Finally, I describe a ground truth based evaluation of my approach. The ground truth based evaluation builds on the analysis of a network security dataset from the VAST Challenge 2013 [1].

Data Attributes The network data in this case study comprises 10 (Section 4.5.3) and 15 (Sections 4.5.1 and 4.5.2) attributes, of which the most interesting and meaningful attributes are: *time stamp, source IP, source port, destination IP, destination port, protocol*. Moreover, for the data in Sections 4.5.1 and 4.5.2: *event name, source geo location, target geo location, device vendor*.

4.5.1 Proof of Concept: Artificial Network Dataset

The TMDS approach generates patterns, or in other words, makes existing patterns visible, uncovers them. One of the main challenges regarding TMDS is the reliability of the multivariate projection technique. The demonstrated approach and heuristics to overcome rotation issues seem legit. However, we do not know yet whether the visualization correctly reflects the patterns included in the data using the continuous similarity values over multiple sequences, as well as the separation of patterns from the rest of the data. What if the data contains significant patterns that are not reflected in the visualization at all? Because this case study is tailored to the network security domain, I following report on the analysis of an artificial dataset, that shows the same structure as any network security dataset.

[1] http://vacommunity.org/VAST+Challenge+2013

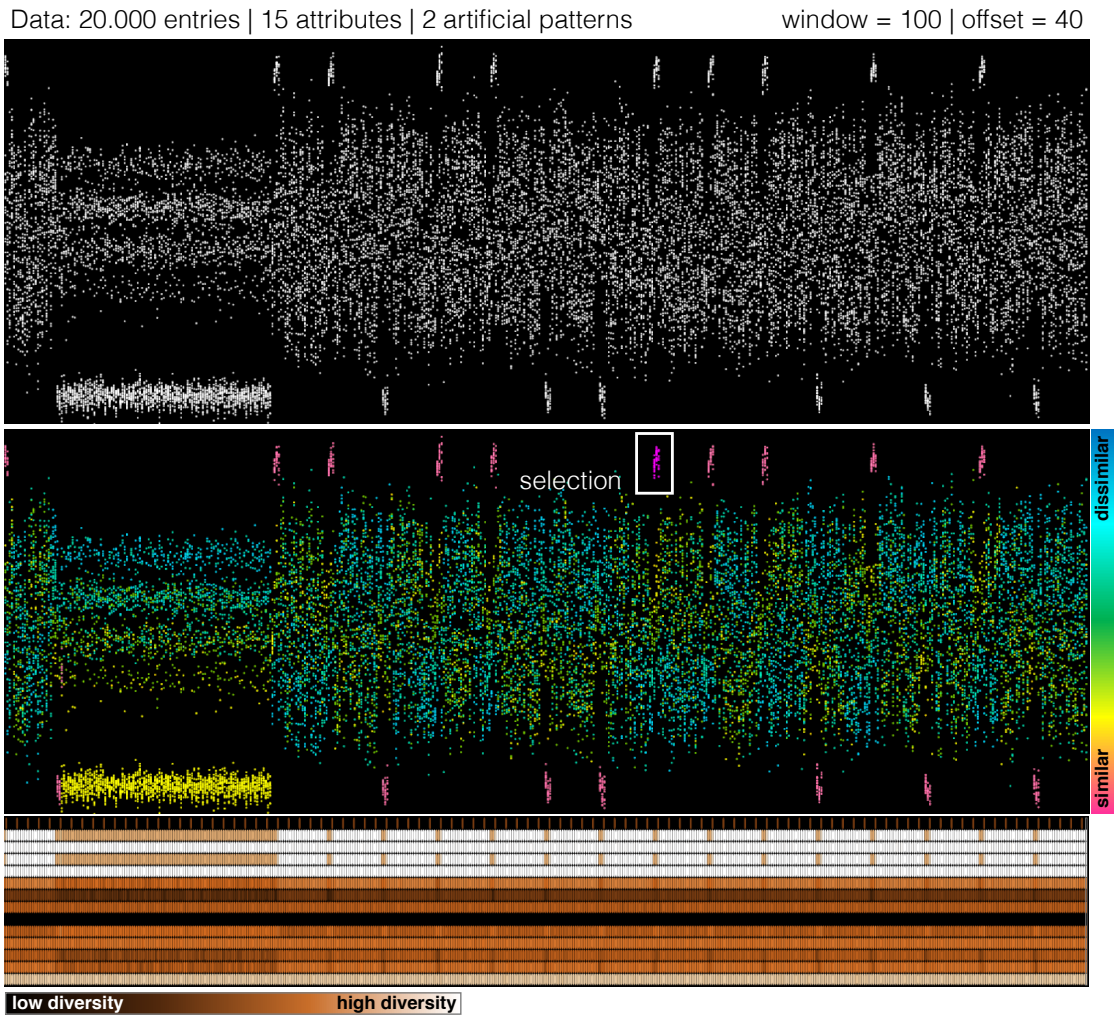


Figure 4.8: *Two port scan patterns encoded in 20k data entries. The visualization clearly separates one continuous and one repeating pattern from the rest of the data.*

Similar to the datasets used in Sections 4.5.2 and 4.5.3, I artificially created a temporal network security dataset that consists of 20k entries and 15 attributes. The data contains two encoded patterns; this means two patterns where attributes share the same behavior. To obfuscate the two patterns, I integrated the patterns in noise data. The noise data is carried forward through the entire dataset, so that any pattern is intentionally interrupted. This constraint further test if the sliding window approach is effective. The two patterns are defined as follows:

- **Continuous pattern:** To generate this pattern, I encoded an attribute pattern in the data that evolves over a fixed period. Note that this pattern does not appear as one single sequence, but is mingled with the noise data. For this pattern, 8 out of the 15 attributes share a commonality; this means the attributes *source IP, destination IP, event name, protocol, source geo location, target geo location* encode information to be considered as similar. Furthermore, the *source port* and *destination port* are alternating.

This pattern corresponds a typical port scan: a source computer attacks a destination computer and first searches for open ports.

- **Repeating pattern:** To generate this pattern, I encoded a small attribute pattern in the data that repeats after a certain period. The pattern is also mingled with the noise data and, similar to the *continuous pattern*, the following attributes are considered to be similar: *source IP, destination IP, event name, protocol, source geo location, target geo location*. Also, the *source port* and *destination port* are alternating. The resulting pattern again corresponds to a typical port scan, but now distributed over time.

I applied TMDS to the data with a sliding window size of 100 and an offset of 40 entries. The result is depicted in Figure 4.8. The Figure shows three visualizations. Starting with the top visualization, we see the result after applying TMDS. We can identify one continuous pattern on the bottom left, as well as a repeating pattern shown as an indentation in equidistant intervals at the top and bottom. After inspecting the raw data, I can confirm that the encoded patterns are also visually separated from the rest. Interestingly, one observation is that the noise data distributes around the center of the visualization while the patterns are clearly separated.

To test the validity of the automatic pattern finding algorithm, I selected one of the repeating patterns on the top, shown in the image in the middle. As a result, the repeating pattern is highlighted in red, which means *very similar*. Also, the continuous pattern is highlighted as *similar* in yellow, which is understandable because the same attributes show a different port scan.

4.5.2 Uncovering Patterns with a Domain Expert

Especially in network security applications, the identification of port scans plays a crucial role and is key to prevent data theft. Attackers use port scans as prominent means to identify possibly open ports in a network, to intrude the systems, place Trojans, and spy out confidential information. One may argue that domain experts have already gained knowledge and that tools are sophisticated enough to detect targeted port scans. To put TMDS to the test, the approach was tested with a domain expert, who works as a network analyst in a company. The data used in this example is obfuscated.

Port scans typically show inherent similarities or commonalities, which may not always be visible or identifiable at first sight. So far, many domain experts either directly inspect the raw log files or use simple graphical interfaces like, for example, port graphs, these are scatterplots with one axis representing time and the other one the queried port number. The domain expert confronted me with a port scan pattern, that was not salient neither in the raw data nor in the port graph. Because this pattern could not be identified at the moment it occurred, the domain expert asked for the streaming capability of TMDS and whether this approach can find this pattern. Because port scans show similarities, TMDS may be a good approach to find even *hidden patterns*.

The used network data consists of approximately 800 entries and the aforementioned 15 attributes. The domain expert claimed a port scan for port 5000. I assigned the highest weights to the destination IP address and port because these attributes are likely to reveal an intended port scan. Figure 4.9 on the left, shows the result after applying TMDS with a

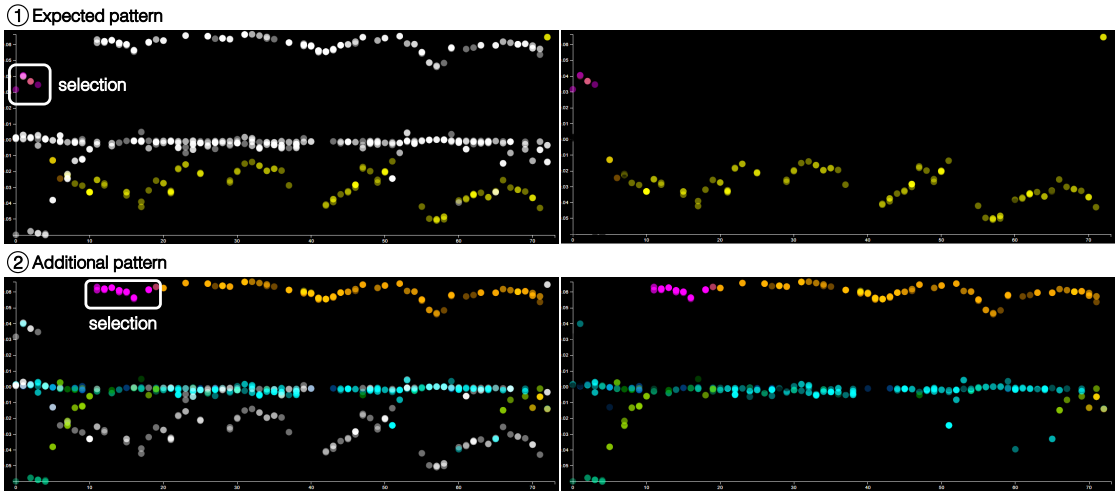


Figure 4.9: *TMDS applied to real network security data. The domain expert put TMDS to the test and was interested in whether this approach can reveal a pattern which is challenging to detect. (1) shows the pattern that is hidden in common analysis techniques. A port scan from changing source IP addresses and ports as well as a changing internal destination IP. Only the port 5000 remains stable. (2) TMDS reveals an additional pattern. This pattern corresponds to a normal port scan, meaning source IP and port, as well as the destination IP remain stable. Only the destination port is changing.*

window size of 100 and an offset of 10 entries. We immediately see one continuous pattern on top, one in the middle, and one on the bottom. This result is plausible because the attributes destination IP and port have the highest weights. This means, in best case three groups are to be expected in the TMDS plot. The first group consisting of coherent destination ports, the second group of coherent destination IPs, and the third group of a mix of both. The TMDS plot reflects this expected behavior. To start with, (1) The expert selected a portion of the data entries that are separated from the rest and are plotted right at the beginning. After looking into the raw data, it turns out that these entries already have one commonality: all selected entries share the destination port 5000. The follow-up question was: How does this pattern develop over time? To answer this question, the expert searched for similar patterns leading to the highlighting in Figure 4.9 (1). The patterns is challenging to detect with common methods, because it consists of a changing source IP, a changing source port, a changing internal destination IP, but of a fixed destination port. Distributed sources tried to find an unprotected computer in the network system listening on port 5000.

Interestingly, the TMDS plot revealed a second separated pattern at the top of the visualization. Figure 4.9 (2) shows the visualization after having selected a part of it and searched for similar patterns. The pattern highlighted in orange corresponds to the continuation of the selection. The pattern represents a common port scan, meaning the destination port changes, but the source IP, source port, and destination IP remain stable. I note that the domain expert was very skeptical about the TMDS approach and its application to the network security domain. However, in particular, the identification of the described “hidden” pattern solely by looking into similarities over time convinced him that TMDS is a powerful tool.

4.5.3 VAST Challenge Dataset: Identification of Events

In the previous sections, TMDS was successfully applied to real network security data, network traffic data to be more specific. To further demonstrate the effectiveness and validate this approach, I following describe the application of TMDS to the VAST Challenge 2013 Mini-Challenge 3 (MC3). The VAST Challenge 2013 is based on an artificial dataset of a large company network that involves a ground truth of included patterns and interesting observations. Overall, the provided network data comprises various suspicious events over a period of two weeks. In this validation, we [2] aim to identify and verify the known patterns visually.

Data Processing

After loading the data into the prototype, we weight the attributes regarding increasing impact of the destination port and the IP addresses. With respect to the NetFlow dataset, we aim to investigate how possible attackers access services within the network. The operating system or router assigns the respective source port from an ephemeral port range, making it less meaningful for our analysis, which is why we neglect it. The destination port, on the other hand, is crucial to associate attacks with similar attack vectors. Assigning destination ports a higher weighting results in visual attack clusters to the same service like, for example, to port TCP/80, which is default HTTP traffic. Then, we filter the NetFlow data for incoming traffic only and focus on uni-directional data flows. To further narrow down the data, we only consider source addresses within 10.0.0.0/8 (in the data, this corresponds to the entire Internet) and destination addresses within 172.0.0.0/8 (this reflects the internal company network). Furthermore, we remove any response to a low port number, because these are likely to be outgoing connections that were initiated from the company network. The visualization of the processed data using TMDS reveals that DoS attacks lead to vast amounts of network flow, which is why we apply adjusted stratified sampling based on destination ports to reduce the vast amounts of data further. Compared to global sampling techniques, adjusted stratified sampling enables to expose other subtle patterns, which are missed otherwise. We assign all attributes the standard weight of 1.0, except for the attribute *time stamp*, which we intentionally exclude by applying a weight of 0.0. We apply TMDS using a window size of 100 and an offset of 10 entries.

Ground Truth Validation

We present an overview of all ground truth events in Table 4.1. The TMDS patterns cover a time span of multiple days. Certain *Event Types* do not meet the data requirements of TMDS, which is why we neglect them. These events are intentionally left blank within the column *TMDS*. We compare our findings with the official ground truth and check each successfully identified event (green background). Events we are not able to identify, are crossed (yellow background). In total, we successfully identify 16 events, which corresponds to 84%. Following, we present

[2] Hereinafter, "we" refers to me and Fabian Fischer, who prepared and processed the data and carried out the ground truth validation as described in [102]. In this section, I refer to this validation to demonstrate the effectiveness of TMDS, and include two convincing examples.

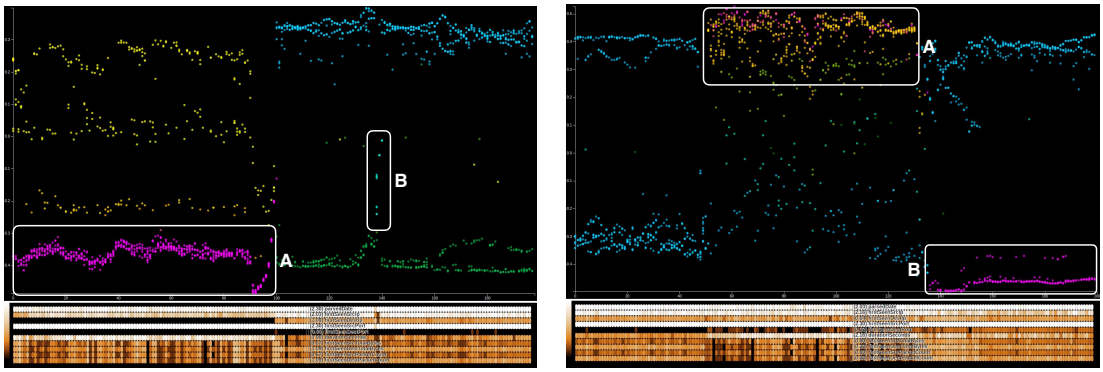
Event ID	Subtlety	Event Type	Data Source	TMDS	Pattern
(1)	Questions only	Videoconference	-	-	-
(2)	Questions only	Threatening Letter	-	-	-
(3)	Subtle	Port Scans	NetFlow/BB	✓	Fig. 4.1
(4)	Subtle	Port Scans	NetFlow	✓	Fig. 4.1
(5)	Obvious	DoS	NetFlow	✓	Fig. 4.10(a)
(6a)	Subtle	Server Crash	NetFlow/BB	×	-
(6b)	Subtle	Server Return	NetFlow	(×)	-
(7)	Subtle	Port Scans	NetFlow	✓	Fig. 4.10(a)
(8a)	Obvious	DoS	NetFlow/BB	✓	Fig. 4.10(b)
(8b)	Obvious	DoS	NetFlow	(✓)	Fig. 4.10(b)
(9a)	Subtle	Server Crash	NetFlow/BB	×	-
(9b)	Subtle	Server Return	NetFlow	(×)	-
(10)	Subtle	Malicious Redirects	NetFlow	×	-
(11)	Obvious	Exfiltration	NetFlow	-	-
(12)	Obvious	Port Scans	NetFlow	✓	-
(13)	Obvious	Port Scans	NetFlow	✓	-
(14)	Obvious	Exfiltration	NetFlow	-	-
(15)	Questions only	Threatening Letter	-	-	-
(16)	Obvious	Network Down	NetFlow	✓	-
(17)	Obvious	Port Scans	NetFlow/IPS	✓	-
(18)	Obvious	Port Scans	NetFlow/IPS	✓	-
(19)	Obvious	Failed DoS	NetFlow/IPS	✓	-
(20)	Obvious	Failed Exfiltration	IPS	-	-
(21)	Obvious	Port Scans	NetFlow/IPS	✓	-
(22)	Subtle	Botnet Infection	NetFlow	-	-
(23)	Obvious	Botnet Communication	NetFlow	-	-
(24)	Obvious	Port Scans	NetFlow/IPS	✓	-
(25)	Obvious	Port Scans	NetFlow/IPS	✓	-
(26)	Obvious	Botnet DoS Attacks	NetFlow/IPS	-	-
(27)	Obvious	Botnet DoS Attacks	NetFlow/IPS	-	-
(28)	Obvious	Port Scans	NetFlow/IPS	✓	-
(29)	Obvious	Port Scans	NetFlow/IPS	✓	-

Table 4.1: *The ground truth for the VAST Challenge 2013 MC3 consists of 29 official events. After analyzing the data with default weightings, we compared our findings with the official ground truth and used a check mark to highlight successfully identified event patterns using TMDS.*

an extract of screen captures that outline some of the most salient visual TMDS patterns. For further reading, I refer to the published work by Jäckle et al. [102].

The Events (1) and (2) in Table 4.1 are not identifiable using TMDS because they are not visible in the data in general. These events were linked to specific questions that the organizers of the VAST challenge could be asked. The data provider classifies the first identifiable Event (3) as a *subtle* event. Figure 4.1 point this event out as pattern A and B. The continuous blue and green patterns on the top and bottom correspond to normal legitimate incoming network traffic. Applying details-on-demand, we identify pattern A as an attack from source IP 10.6.6.6 to 172.30.0.x machines. According to the ground truth, this qualifies *"as subtle because firewall allows mainly ports 25 and 80"*. Pattern B is described as *"high volume web browsing traffic"*. These patterns are not only visible in the TMDS plot, but also in the diversity matrix (bottom) based on the Shannon Entropy – the patterns leave out a salient black area in the matrix indicating a low entropy, thus a low diversity. The main correlating attributes are the *source IP*, *destination IP*, and *destination port*. The low entropy furthermore indicates that the attacker continuously generated almost identical requests. The pattern B is *"followed by port scans"*, indicated by pattern C on 2013-04-01 22:18.

Next, we apply TMDS to the days 2 and 3 in the data. Figure 4.10(a) shows the result for



(a) TMDS applied to the 2nd day on 2013-04-02 00:00 to 23:59. Pattern A relates to a DoS attack, and pattern B to a subtle port scan.

(b) TMDS applied to the 3rd day on 2013-04-03 00:00 to 23:59. The visualization shows a sudden pattern change. Pattern A relates to an ongoing distributed DoS attack and pattern B corresponds to a different attacker who primarily attacks another webserver.

Figure 4.10: *TMDS application to the VAST Challenge 2013 MC3 data, days 2 and 3.*

the second day. We select a portion of pattern A leading to the shown highlighting. Pattern A (magenta) is salient and corresponds to a DoS attack between 05:22 and 07:22. The pattern, furthermore, originates from 10 different attackers to webserver 172.30.0.2. For Event (6a), the ground truth provides a webserver that becomes temporarily unresponsive. We cannot find such events using TMDS on this case study, because we analyze incoming traffic only and, furthermore, do not consider missing data. Pattern B (lime green) is related to Event (7) and corresponds to subtle port scans attacking port TCP/25 from 10.6.6.6 and 107.7.7.10.

TMDS applied to the data of the third day reveals sudden changes in patterns. We come to the result depicted in Figure 4.10(b) by selecting a portion of pattern B and then searching for similar patterns. Pattern A corresponds to a major pattern change from 9:30 until approximately 11:48, which is another ongoing distributed DoS attack. This pattern is not as dense as others, such as pattern B, because it originates from several attackers and not a single one. Pattern A relates to Event (8a) in the ground truth. Pattern B (magenta) corresponds to Event (8b) and reveals an attacker with IP 10.15.7.85, who attacks a different webserver 172.20.0.15.

In summary, we focused on incoming NetFlow data only. However, six events (11, 14, 22, 23, 26, 27) relate to outgoing data which, as a consequence, are not marked as identified in Table 4.1. The ground truth validation by application to the VAST Challenge 2013 data shows the general applicability of the TMDS approach. The integration of TMDS into security applications seems further promising and could significantly improve state-of-the-art systems.

4.6 Discussion & Future Directions

TMDS is geared to the visual analytics process [126], and enables a novel analysis of temporal multivariate data. To be more specific, TMDS enables the identification of sequential or temporal patterns and supports the interpretation with additional visual representations of

the plotted data. The design space of TMDS opens questions regarding parameter settings, scalability, and possible extensions and alternatives, which I discuss in the following.

Window and Offset Size As pointed out, the application of TMDS to network data works fine, and we retrieve plausible results using rules of thumbs. I cannot provide fixed parameters for the window and step size, because it depends on the data characteristics and size. One way to suggest plausible parameters is to generate multiple plots, taking into account the window and overlap size discussion of Section 4.3.3. Then, apply visual quality metrics to the plots such as Hough Transformation [53] or contour tracking.

For pragmatic reasons and as a first step, I chose a rectangular windowing function for all data entries contained in a sliding window. As the sliding window in practice spans a larger number of entries, the changes introduced by the unweighted exit and entry of entries on each sliding step do each not have a huge impact on the projection result. However, I expect that for smaller window sizes and/or larger offsets, TMDS would require a non-uniform weighting scheme to provide sufficient stability of the projections. For example, Gaussian or triangular weighting schemes centered on the sliding window may be useful [50]. I tested with different parameters, finding that with an offset of circa 10% of the window size, and a window size of at least tens of entries, I achieve sufficiently stable results for unit weighting.

Another possibility is to include the experience and knowledge of the user using interaction. TMDS provides an initial size for first results but then includes the user to refine the parameters and make patterns salient. I leave the assessment of the effect of alternative weighting schemes with respect to window size, offset, and data and analysis tasks as an important subject for future work.

Scalability For now, the visualization of TMDS is based on vector graphics, meaning it does not scale to vast amounts of data. Moving this part to the graphics card, for example, by using WebGL, can significantly increase the scalability on the visualization side. About the computation of the TMDS plots, I refer to the discussion of window and overlap size of Section 4.3.3. However, when applying TMDS to a large window, or even data sizes, one should consider the applied multivariate projection technique. In this work, I apply classical MDS, but other techniques may be more feasible for larger data sizes, such as Glimmer [97]. The choice of the right projection technique is key not only for the runtime but also for the data. If the analyst is interested in local rather than global structures, she should apply fast non-linear techniques like t-SNE [145]. Also, the application to categorical data is only preliminary and can be extended in various ways. For instance, involving the user [114] and adding semantic information like hierarchies in the categories, can improve results and enhance the analysis process.

Streaming Capabilities TMDS operates on parallel threads, meaning the windows are computed each separately and in parallel. This property opens possibilities for streaming applications, such as real-time network or financial data analysis. One of the arising issues is the limited space. This means we run into space issues when more and more windows are

added to the visual information space. Sequential or temporal patterns may also appear in windows far apart using computed windows in between. One way to handle the limited screen real estate is presented in Chapter 5. Using off-screen visualization, the data characteristics can be preserved, such as the pattern membership. However, for this task also common Focus-plus-Context approaches based on image-space distortion are applicable.

Relation to Pattern Trails At first sight, the technique Patter Trails, described in Chapter 3, seems applicable to sequential or temporal data. A hard requirement of Pattern Trails is that points repeat among small multiples and subspaces, which is not given in temporal data; the data evolves and does not repeat. So far Pattern Trails can not be applied to such type of data. An eligible extension to TMDS represents a sequential 3D visualization using 2D MDS plots to provide additional spatial separations of patterns, similar to the Subspace Cube. However, using the 1D MDS approach, patterns are presented in a clearer way, because a sequential 2D MDS inevitably leads to overplotting. Additionally, a new heuristic needs to be derived to handle the possible rotation between 0° and 360° due to the fact, that MDS is not invariant to rotation. The benefits and drawbacks of a temporal 2D MDS need to be compared systematically to the present approach.

Part II

Overview-preservation in Large Projection Spaces

5

Topology-Preserving Off-screen Visualization

Contents

In the Chapters 3 and 4, I introduced novel methods for the detection and interpretation of multivariate patterns based on DR, in particular, multivariate projections. The created information space – depicted as a 2D scatterplot – can thereby become vast and often requires zooming and panning operations to obtain details. However, drilling down to see details

results in the loss of contextual overview. This problem is held true for any spatial representation, including geo-spatial maps and graphs. Existing overview-preserving approaches typically operate in image space and provide context while the user examines details, but suffer from distortion or overplotting (see Chapter 2). Two-dimensional multivariate scatterplots, however, demand tailored solutions to preserve attribute-dependent information as well as the overall topology, because the distance between data points comes with the meaning of similarity attached to it.

I propose to apply *Off-screen Visualization*, a family of techniques which provide data-driven context with the aid of visual proxies. Visual proxies can be visually encoded and adapted to the necessary data context with respect to scalability and visualization of multivariate data. In this chapter, I open the design space for off-screen visualization tailored to the overview-preservation of multivariate data characteristics. I propose three novel off-screen visualization techniques based on aggregation that build on top of each other: First, the visualization of aggregated off-screen points or shapes based on rasterization. This technique introduces a data-driven border region as means to preserve the data topology, based on which I then introduce an intrinsic, glyph-based technique for the visualization of an additional data value besides the spatial dimensions. Third, I propose the use of star glyph insets to encode more than one attribute. The use of a dedicated border region together with aggregation represents the core idea of this thesis regarding overview-preservation. Therefore, I conducted a user study against HaloDot [74], a state-of-the-art off-screen visualization technique, showing that users perform significantly better when given aggregation and the data topology. Furthermore, I provide a comprehensive discussion regarding tasks and derived challenges for off-screen visualization in general. Given the DR pipeline depicted in Figure 1.1, this chapter aims at improving navigation on the visualization side of the pipeline.

This chapter is based on [106], [107], [108], and [103]:

Off-Screen Visualization Perspectives: Tasks and Challenges. D. Jäckle, B. C. Kwon, and D. A. Keim. *Symposium on Visualization in Data Science (VDS) at IEEE VIS 2015, 2015.*

Integrated Spatial Uncertainty Visualization using Off-screen Aggregation. D. Jäckle, H. Senaratne, J. Buchmüller, and D. A. Keim. *EuroVis Workshop on Visual Analytics (EuroVA), The Eurographics Association, 2015.*

Ambient Grids: Maintain Context-Awareness via Aggregated Off-Screen Visualization. D. Jäckle, F. Stoffel, B. C. Kwon, D. Sacha, A. Stoffel, and D. A. Keim. *Eurographics Conference on Visualization (EuroVis) - Short Papers, The Eurographics Association, 2015.*

Star Glyph Insets for Overview Preservation of Multivariate Data. D. Jäckle, J. Fuchs, and D. A. Keim. *IS&T Electronic Imaging Conference on Visualization and Data Analysis, 2016.*

5.1 Introduction

Multivariate projections transform the data to a lower-dimensional space, preserving its main structure. Note, that the number of data records remains the same, before, and after the transformation. The transformation to a lower-dimensional space, such as a 2D space, enables the visual exploration of the data, but makes great demands on visualizations; they need to scale to the vast amounts while remaining interactive so that users can explore patterns and gain meaningful insight at the same time. However, visualizing ever-increasing amounts of data is often challenging due to the limited screen real estate. Instead, within the limited space, users perform effective interaction techniques to aggregate information for an overview and to focus on areas of interest back and forth. In the event users apply zooming or panning operations to explore local data patterns, the operations have one important commonality: both zooming and panning imply that the user is only analyzing and/or looking at one specific area in detail. In such situations, relations to potentially interesting patterns are lost. As a result, users face the inherent trade-off between overview and detail.

It is still an ongoing, unsolved research how to providing overview and the context while showing a particular area in detail. Consider the visualization of dimensionality-reduced data in a scatterplot after the application of MDS. While the axes hold no specific meaning, the proximity between points indicates their similarity. A key challenge is to preserve relative positions and proximities between data points. As the detail view magnifies, the space for the overview shrinks. In this case, aggregation is key, but also represents a trade-off to the preservation of the data topology [1]. Many prior studies provide inspiring solutions, which also show inspiration and areas for improvement at the same time. This thesis gives an overview of state-of-the-art approaches for overview-preservation in Chapter 2. While classical overview-preserving approaches, such as Overview-and-Detail or Focus-plus-Context, operate in image space, I aim for a data-driven solution that preserves the data characteristics rather than the overview in image space. I propose to use *Off-screen Visualization*, a pioneering approach, which shows lots of potential for visual data analysis. The main idea is to project data points that move off-screen, due to panning and zooming operations, back to the border region of the viewport in terms of visual proxies. This principle is depicted in Figure 5.1. Cockburn et al. [44, p. 16] defined the main characteristic, yet advantage, of visual proxies as the possibility to *"modify how objects are rendered"*. Adapting the rendering of data points introduces new opportunities regarding preserving a data-driven and task-dependent overview.

The driving question of this chapter is: *"How to preserve the (multivariate) data characteristics in an overview-preserving environment based on off-screen visualization?"* This includes the overall data topology, as well as the multivariate aspects of the data and tasks. To tackle this question, I following assess relevant design considerations, based on which I contribute and discuss three interactive techniques that aim at different data characteristics and tasks. First, I contribute an off-screen visualization that introduces a data-driven border region to preserve the overall data topology. Based on *rasterization* of the off-screen space, points and shapes and an additional data encoding are *aggregated* and preserved in an overview-preserving

[1] Spatial relations and properties are unaffected by the change of shape or size of objects (according to [192])

manner. I build upon the findings of HaloDot [74], which suggest making use of aggregation as a means to overcome the limited available space and overplotting to preserve overview. I showcase this technique based on scatterplots and also demonstrate the usefulness in a real-world monitoring task of geo-spatial data. Based on rasterization and aggregation, I then propose to integrate glyphs to encode a second data value, which I showcase using uncertainty information; examples include recorded errors of measurement stations or spatial errors introduced by the multivariate projection. For the third technique, I go one step further and propose to use multivariate star glyph insets to encode more than two data attributes. The application of star glyphs points out the limitations regarding the size of the border region, as well as the size of the glyph. I showcase the use of star glyph insets based on multivariate projections, as well as the investigation of crime incidents in San Francisco.

The commonality between the named three approaches is that they apply aggregation to provide an overlap-free overview. Furthermore, the dedicated border region aims at preserving the data topology, which is why I conducted a user study against HaloDot [74], a state-of-the-art technique that introduced aggregation for off-screen environments but does not preserve the data topology. The study shows that users can improve their performance in context-aware analysis tasks using a dedicated border region.

For each of the three off-screen approaches, I consider the use case scenarios in the broader context of the high-level task description by Brehmer and Munzner [24]. The authors provide abstract rather than domain-specific task descriptions, which makes it suitable to express the general application of off-screen techniques. As per the authors, a task description is formed by a combination of elements of the questions *Why?*, *How?*, and *What?*. *Why* a task is performed is described by a three level, bottom-up hierarchy: I classify the use case scenarios regarding the highest level comprising three different intents:

- **Present** refers to the visualization of information and thus is used for decision making or instructional processes.
- **Discover** is tailored to the visual analytics process and helps to generate and verify hypotheses. An example represents the exploration of multivariate data.
- **Enjoy** refers to visual representations that are novel, interesting and consider attention, but are not used by experts for specific analyses.

Each intent requires the user to find interesting elements in the visualization, characterized by the second level: *lookup*, *locate*, *browse*, and *explore*. The lowest level provides means to *identify*, *compare*, and *summarize* targets found in the mid-level. However, none of the intents can be addressed without interaction, which is described by *How* a task is performed. In this chapter, I solely concentrate on the node *manipulate*, which introduces the basic interaction techniques: *select*, *navigate*, *arrange*, *change*, *filter*, and *aggregate*. Finally, *What* defines the task inputs and outputs. In the present thesis, I consider the input as being multivariate.

In the scope of this chapter, I do not consider the *produce* node with all possible interactions falling into the category of *introduce*, because they refer to tasks where users generate new artifacts. This part is not covered in the present thesis leaving it as a future direction.

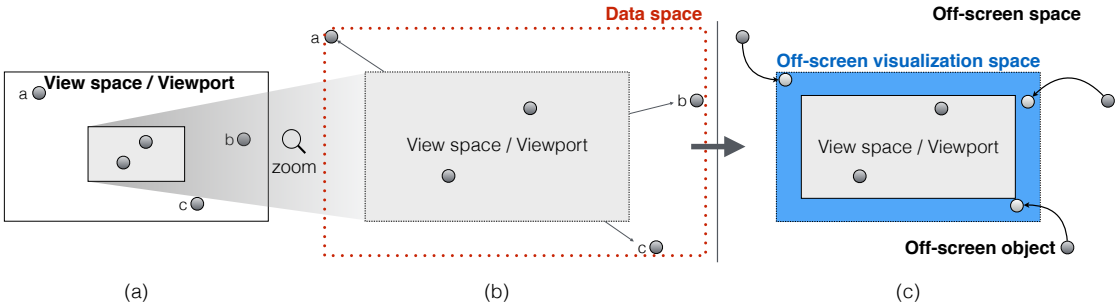


Figure 5.1: *Off-screen visualization pipeline and terminology. (a) The user applies zooming and panning operations on the visible space. (b) After performing such operations, some points move out of sight, meaning off-screen (in this case, the points a, b, and c). (c) To provide an overview, the viewport is surrounded by a dedicated border region and the off-screen objects are mapped to this border in a topology-preserving manner. I refer to the visible part, typically represented as the display content, as the* view space*, or* viewport*. I refer to the bounding box that contains the data as* data space*. Furthermore, I refer to the invisible space as the* off-screen space*, and the border region surrounding the viewport as* off-screen visualization space*. The objects that are mapped to this border regions, are named* off-screen objects*.*

In following sections, I introduce three novel off-screen techniques and multiple related use case scenarios. The scenarios result from the highest level of *why* a task is performed and are therefore aligned with the nodes above.

Following, I briefly outline the used terminology with respect to off-screen visualization:

Terminology I define a terminology that describes different spaces used throughout this thesis. Figure 5.1 outlines four different spaces: the *data space*, the *view space*, the *off-screen space*, and the *off-screen visualization space*.
The *data space* D is defined as follows:

$$D := \{(x, y) | (x, y) \in \mathbb{R}^2\} \tag{5.1}$$

In the context of this thesis, the data space is considered to be the bounding box of all depicted data points.
The *view space* V is defined as the set of points, where x and y are restricted to the area enclosed by a rectangular frame with the coordinates $x = v_x, y = v_y$ and $width = \hat{v}_x - v_x$, $height = \hat{v}_y - v_y$. That is, V contains all visible data points from the data space.

$$V := \{(x, y) | (v_x, v_y) \leq (x, y) \leq (\hat{v}_x, \hat{v}_y) \wedge v, \hat{v} \in \mathbb{R}^2\} \tag{5.2}$$

In large data spaces, the view space typically represents a subset of the data space. The view space is further restricted in size by the screen resolution. For example, if the whole data space is presented on screen, it is smaller than or equal to the view space. The user can manipulate the view space by translating the x and y coordinates or changing the scale. In this thesis, I also denote the visual representation of the view space as *viewport*.

The section of the data space, which is not part of the view space is formally referred to as *off-screen space*.

$$O := D \backslash V \tag{5.3}$$

I refer to objects contained in this space also as *off-screen objects*.
The *off-screen visualization space*, often denoted as *border* region or area, refers to the pictorial space; that is the area of the viewport, which is used to display the off-screen objects. Actual size and appearance are dependent on the visualization technique and the users.

5.2 Related Work

Off-screen visualization preserves the data-driven overview when performing zooming or panning operations in large information spaces, such as scatterplots or any spatial information. Following, I outline existing off-screen visualization techniques given the data they preserve (point versus graph data) and interaction.

5.2.1 Off-screen for Point Data

Off-screen visualization supports navigation and makes use of visual proxies located at the display border, which indicate the direction to the direction to off-screen objects. The straightforward solution to indicate the position of off-screen objects is to apply aligned arrows. The tip of the arrow points towards the off-screen object. Apart from arrows, Zellweger et al. [225] proposed City Lights as a family of off-screen techniques: Halos and City Light cues. While City Light cues only visualize existence and orientation of off-screen objects, Halos [12] show distance and location by intersecting the display with an arc. The arc is derived by creating a circle around each off-screen object. The extent of the circle is chosen in such way that it intrudes the viewport to a certain user-defined degree. Burigat et al. [29] conducted a study showing that Halos and arrows do not significantly differ in general: Arrows allows users to order off-screen objects faster and more accurately, while Halo enables the faster and more accurately identification of off-screen locations.

Because Halo introduced clutter issues, Gustafson et al. [81] proposed to replace halos with isosceles triangles that point towards the off-screen objects. To overcome clutter, the authors improved the layout of the Wedges in a post-processing step. The study by Burigat and Chittaro [27], furthermore, showed that Wedge outperforms arrows in complex tasks, such as ordering off-screen objects by their distance. Another improvement over Halo is HaloDot [74]. HaloDot is the first technique to be known that applies aggregation to off-screen objects to significantly reduce clutter. To do so, HaloDot lays out a grid in off-screen space and bins all off-screen objects that fall into the same grid cell. A grid cell that contains at least one object is represented via a Halo. A user study further showed that using HaloDot searching and pointing relevant off-screen objects is achieved faster than with Halo [73].

Going one step further, EdgeRadar [82] introduced a dedicated border region to preserve the overall topology. EdgeRadar was studied for moving objects and outperformed the use of Halo. So far, off-screen visualization has typically been used for navigation tasks in geo-spatial

environments. Games and Joshi [70], therefore, applied the idea of off-screen visualization to scatterplots and histograms, however, not in a topology-preserving manner.

Another interesting area is the hardware-based application of off-screen visualization. For example, Sparkle [158] or Glowworms and Fireflies [168] are two techniques that augment the display with LEDs. The LEDs can be encoded with different colors and glow whenever an off-screen object is located in their direction. A distinct advantage represents the additional hardware that does not interfere with the visible dimensions of the display.

5.2.2 Off-screen for Graphs

Graphs, also referred to as node-link diagrams, can become vast, demanding tailored solutions to maintain overview and be able to navigate the space effectively. Moscovich et al. [157], therefore, proposed two techniques, Bring&Go and Link Sliding, that enable the user first to retrieve an overview and then to navigate to off-screen nodes. Frisch and Dachselt [64, 65], furthermore, integrated graph-based off-screen visualization into the application of UML (Unified Modeling Language) class diagrams. They make use of a dedicated border region, where they introduce different types of visual proxies, tailored to the application of navigating large class diagrams.

Another important technique is Dynamic Insets by Ghani et al. [72]. The general idea is to present a snapshot of the off-screen located node as an inset in the viewport. Furthermore, to concentrate only on connected nodes, the authors apply a degree-of-interest function that determines the nodes which impact the overall overview.

Off-screen visualization can also bring an overview into the organization of information on our desktop. This is what Geymayer et al. [71] proposed. They created a graph-based off-screen visualization that links windows, text, and other information on the desktop, thus providing an efficient overview.

In this thesis, however, I focus on unconnected point data, as it is the case for scatterplots as result of multivariate projections.

5.2.3 Interaction in Off-screen Environments

As yet, navigation is the main task of off-screen visualization. Interaction has already been introduced together with new techniques. The main idea is to reduce the amount of necessary zooming and panning interactions when navigating to an off-screen location. For example, the graph-based approach by Moscovich et al. [157] enables users to automatically zoom and pan to an off-screen location by clicking on the respective visual proxy. This idea was further developed for point data. EdgeSplit [91] is based on EdgeRadar and also applies a dedicated border region for the visualization of off-screen points. EdgeSplit applies a Voronoi tessellation to the border region, so that there is enough reserved space to click or tap on an off-screen object. However, this technique is only applicable for the visualization of few data points. Otherwise there is no real benefit.

Hopping [100] introduces additional proxies linked to the Halo representation that appear when the user intersects the halos with a laser beam. By doing this the authors aim to

overcome clutter issues. When clicking on the visual proxy, the visualization automatically hops (teleports) to the intended off-screen location. An improvement to Hopping represents WinHop [166]: Instead of hopping to the off-screen location it introduces a preview of the location without having the user to leave her current location.

In contrast, Predictive Jumping [195] applies a prediction algorithm that integrates the natural movement in space with a goal-directed movement, meaning the system predicts where the user aims to navigate.

I partially adapt named interaction techniques in this thesis. However, one should be aware that the off-screen interaction techniques can be integrated with any off-screen visualization technique.

5.3 Design Considerations

The design space for off-screen visualizations is rather huge. Typically, the visual proxies are designed subject to the task, but other design decisions have only partially, or not at all, been considered and discussed, yet. In the following, I will introduce and briefly discuss design decisions taken towards *topology-preserving off-screen visualization in large information spaces*. The most important design decisions need to be taken with regard to the viewport shape, the choice of an adequate intrusion design, the use of aggregation, and the selection of suitable interaction techniques. The remainder of this chapter is built upon the justified design decisions. Note, that not all decisions have been formally evaluated yet, and partially are open to future research.

Viewport

Topology-awareness and analysis tasks such as comparison or identification of data values, among others, play a crucial role. To provide a maximum focus region as well as topology-awareness of surrounding data, I augment the viewport with a dedicated border region which incorporates off-screen information. For the shape of the viewport, rectangular or circular shaped viewports are possible. In the present thesis, I decide on a rectangular shaped viewport to meet the dimensions of the display, and also not to waste screen real estate. The idea of providing a rectangular shaped maximum focus is furthermore inspired by bifocal displays [10] and possible misinterpretations caused by radial viewports. Zanella et al. [224, p. 119] claim that radial distortions lead to misinterpretations and there is a “lack [...] of adequate support that allows people to comprehend the manner in which the information is being presented”. The issue with radial viewports is that the direction towards off-screen objects is always mapped towards the center of the viewport, which is not optimal if one aims to relate different data points to each other.

Dedicated Border Region for Topology-Preservation

In the context of this thesis, topology-preservation refers to the unaffected relations between visualized off-screen objects when projected back to the viewport. This poses the key challenge

because the available display space for visualizing off-screen information is restricted and not big enough to preserve relations in a 1 : 1 ratio. Thus, it is not possible to maintain the topology to a full extent, yet an approximation. The goodness of the approximation also depends on how the off-screen visualization is integrated into the viewport.

The usage of a border region implies a dedicated region attached to the viewport, which is exclusively reserved for including off-screen information. A border region also implies a solution to the *Desert Fog* problem defined by Jul and Furnas [118]: Empty regions are efficiently identifiable, and hence, the user is able to navigate to regions of interest efficiently. This way, we are able to save zoom and pan iterations and augment the Visual Information-Seeking Mantra [184]: Overview first, zoom and filter, then overview and details-on-demand. The dimensions of the dedicated border region depend on various factors: the zooming level, the position of the viewport in the data space, the data distribution, and the analysis task. There is no unique definition of dimensions since there exist many different combinations of factors influencing this design decision.

I, therefore, propose the concept of an off-screen border, whose size adapts to the zooming level:

$$B_{size} = B_{maxSize} - \left(\alpha \cdot \frac{(B_{maxSize})^2}{d(P_{vc}, P_{oMax})} \right) \tag{5.4}$$

A constraint for the calculation of the border width B_{size} is, that the zooming level is at least that high so that the data space covers the viewport. Otherwise, off-screen data does not exist, and the border region is void. In each interaction step – zooming or panning – B_{size} is adapted. Therefore, I first derive the relative scale from the relation between maximal possible border size $B_{maxSize}$ and the distance between viewport center P_{vc} and maximal outer bounds of the data space P_{oMax}. The value $B_{maxSize}$ is limited to half of the display dimensions (use half of the display height if the display is widescreen, width otherwise). Visual Analytics argues that the user plays a crucial role in the exploration process. I therefore introduce the parameter α. The user can select the α value between $[0, 1]$ and thus determine the relative maximal size of the border region with direct impact on $B_{maxSize}$. For now, $B_{maxSize}$ is initially set to 35 pixels pursuant to EdgeRadar [82]. Depending on the position of the viewport and the zooming level, the border region gets more space assigned the higher the zooming level is and the more data needs to be represented within the border region.

Besides the use of a dedicated border region, the strategy for projecting the off-screen objects to the border region additionally influences the preservation of the topology. In general, there exist two common projection strategies, that are also covered and used in related work: the *orthographic* and the *radial* projection strategy. The orthographic strategy projects off-screen objects perpendicular to the viewport, whereas the radial strategy projects off-screen objects along a line originating from the center of the viewport. I take up the projection strategy, among other things, in Chapter 6, and show that the orthographic projection strategy does not only preserve the topology more effective than the radial strategy but also meets the users' intuition.

Aggregation

Aggregation in visualization can be applied for reducing the amount of clutter for the sake of a high-level overview. Different aggregation techniques exist, grid-based binning presumably one of the simplest and well-known ones. For example, the technique HaloDot [74] applies a grid-based aggregation for off-screen content. This represents a simple but effective method for aggregating data. Hereinafter, I also apply grid-based aggregation, yet being well-aware of alternative techniques.

Interaction

Interaction is key to the navigation of large information spaces, as it is the case for multivariate projections. In the present thesis, I focus on the visualization of off-screen information rather than the development of novel interaction methods, as being introduced especially in regard to mobile interfaces. However, I integrate state-of-the-art interaction possibilities for navigating the information space.

Two very necessary and inevitable interactions are *zooming* and *panning*. These techniques enable the basic navigation through the depicted space. Furthermore, I instantiate *clicking* for details on demand, as well as *brushing & linking* to put data objects into relation. For example, in the depiction of a multivariate projection, a certain attribute can play a major role. To relate data objects based on attributes, brushing enables the selection of the attribute and linking highlights all relevant, or similar, data objects.

Also, I build upon the work of Moscovich et al. [157] and Irani et al. [100] and enable the automatic navigation to a user-defined off-screen destination by clicking on the projected off-screen object. For the automatic navigation, I employ an automatic version of zooming and panning proposed by van Wijk and Nuij [209]. The basic idea is to provide a smooth transition from one location to another with the help of simultaneous zooming and panning. Generally speaking, the method implements an animation that continuously pans and simultaneously first zooms out and then zooms in when reaching the destination. This method helps to keep the context while automatically moving through space. The off-screen visualization continuously updates to the panned and zoomed position.

5.4 Density-based Visualization of Points and Shapes

In this section, I take first steps towards topology-preserving off-screen visualization based on aggregation. I present a data-driven off-screen visualization technique, which enhances context-awareness while providing maximum focus. The viewport is augmented by a border, which is used to represent off-screen objects in a clutter-free way, while the viewport can be utilized to show detailed visual data representations. Topology within the border is preserved by mapping off-screen objects to the border region and thus preserve relative distances. In contrast to distortion-oriented techniques, this approach is able to handle data values and characteristics separately. This technique is suitable for displaying point data as well as shape

data, which is a design decision based on the application requirements and analysis task at hand. I further show the usefulness in two use cases.

5.4.1 Technique: Topology-preserving Aggregation

This technique surrounds the *viewport* with a *border area*, which is used to display data objects located outside the viewport (*off-screen* objects). I visualize off-screen located objects and shapes using grid-based aggregation. Let me note, that grid-based aggregation is commonly used, for example for the creation of heat maps, however, any other aggregation technique is applicable. The idea of grid-based binning is adapted from HaloDot [74], a technique which also applies a grid to aggregate off-screen located points, but then lacks of presenting the topology. The main idea of using aggregation is to reduce the amount of displayed data, so that the structure is still perceptible in view of a limited screen estate that is reserved for the overview.

In the case of points, the general idea is first to map the off-screen located points to the border region, and then aggregate them by applying a 2D grid on the border with user-defined dimensions, and then bin the data points. The usage of a grid-based approach yields various benefits regarding point-based, but also regarding shape-based data. For example, grid-based aggregation overcomes possible overplotting issues while the overall structure is still conceivable, depending on the grid dimensions.

This technique is also applicable to visualize off-screen located shapes. Shapes are rasterized in off-screen space using the grid and are then projected to the border region. The design decision to rasterize shapes is based on two reasons: (1) Rasterization speeds up the rendering process and (2) provides the possibility to select a level-of-detail-based on the cell size. Rasterization is achieved by intersecting a shape with the grid cells. All intersected and included cells form the rasterized shape. The level-of-detail can be steered by increasing or decreasing the cell size respectively with direct impact on the rasterization. I am aware of the fact that the level-of-detail depends on the user and the task at hand.

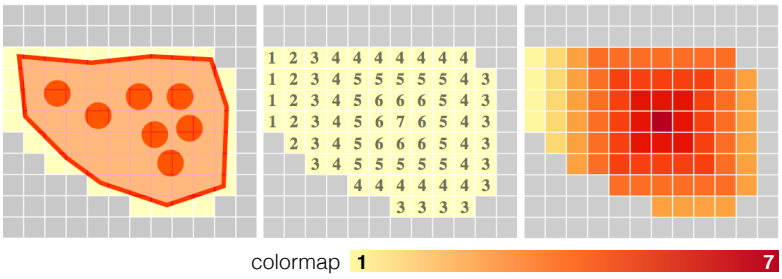


Figure 5.2: *The mapping of a shape (left) to numbers (middle) to colors (right) determining the off-screen grid cell color.*

To carve out the structure of the off-screen data, I apply a sequential color scheme to the aggregated data. Based on the sequential color schemes proposed by Harrower and Brewer [86], I follow the convention *"dark equals more"*, which means that a cluster containing lots of points is colored darker compared to the one containing fewer points. Based on the application, the grid coloring can be adapted to support the requirements and tasks

accordingly. I propose to align the colors with the number of binned points or to apply a color gradient in the case of shapes. For the following use cases, I derive the color of the shape by the number of points which were combined into a cluster using DBSCAN [59]. To assign each cell a color, I use a radial cell-based color gradient, which is illustrated in Figure 5.2. Starting at the cell which contains the shape centroid, I radially assign the colors to the adjacent cells, depending on the distance to the centroid cell. The color of the starting cell is determined by the number of points in the cluster. In case more than one shape intersects a cell, it is unclear which color to assign. A possible solution to this problem is first to compute the corresponding cell color derived from each shape separately, and then to interpolate between the computed colors respectively.

The present technique preserves the topology of off-screen located data objects with grid precision. This technique, however, represents a starting point and can be adapted to task requirements or user preferences.

5.4.2 Use Case: Epidemic Monitoring

In this use case, I apply this technique in the context of geo-spatial monitoring. The overview-preservation problem poses a serious challenge for analysts who monitor data in a focal area and concurrently keep track of surrounding changes. In the following, I describe an epidemic outbreak. The task of analysts is not only to trace the spread of a disease in a local district but also to understand precise situations about surrounding areas to make informed decisions.

I use the VAST Challenge 2011 MC1 (Mini Challenge 1) dataset, which contains microblog entries for a period of 21 full days. All messages are geo-referenced in the extent of the fictional city *Vastopolis*. The given scenario is an epidemic outbreak in Vastopolis, and participants were asked to determine the epidemic spread as well as the outbreak, location-based on the provided message data. To narrow down the vast amounts of microblog messages, I filter the messages for general symptoms, such as *pneumonia*, *fever*, and *flu*. Any message that does not match at least one of the symptoms is excluded from the dataset. Based on the symptom-based filtering, the whole dataset is split into 21 subsets, each containing the data for a single day. I use the data subsets to simulate a time context by applying the proposed off-screen visualization to each dataset separately. Before visualizing the data, I run a density-based clustering algorithm (DBSCAN) [59] to generate areas that contain many disease-related microblog messages. I choose a low amount of MinPts to capture also small clusters and the epsilon sufficiently small to capture the different hospitals that are located in Vastopolis. In this scenario, the off-screen visualization shows the resulting convex hulls of the clusters, which are treated as shapes.

My goal for the visualization, depicted in Figure 5.3, is to focus on Downtown Vastopolis while the epidemic spread in the outer area of the city can still be observed in the border region. Focusing on Downtown is primarily motivated by the fact, that many events take place in that area, for example, an antique convention or a basketball game. These are work observing because emergencies or other events that require immediate action of law enforcement do often happen on those occasions. This scenario illustrates that off-screen

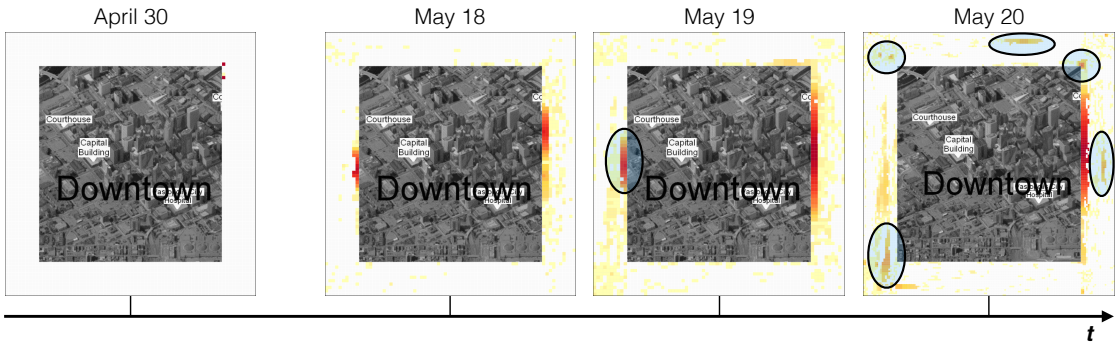


Figure 5.3: *Application of topology-preserving off-screen visualization to the VAST Challenge 2011 microblog data subset. While an analyst follows the epidemic outbreak in the downtown area, she can keep track of the temporal development of the outbreak outside the viewport (to demonstrate the usefulness of this technique, I intentionally omit data within the viewport). On the left, the situation before the epidemic is shown. The three images on the right show the fast development of the epidemic over three days. Black encircled off-screen clusters depict temporal changes of the epidemic spread.*

visualization enables to focus on particular type of data, while it is still possible to be informed about other aspects of the data, thanks to the visualization in the border region.

On the leftmost visualization of Figure 5.3, the off-screen visualization area is almost empty. This means that we have very few cases of the disease on April 30. More than two weeks later on May 18, the visualization shows the beginning of the epidemic. Yellow grid cells are popping up over the entire surrounding areas. In particular, the red areas emerge on the west and the east sides. This shows that the disease spreads out from the downtown toward both sides. On May 19, a majority of grid cells is colored, which indicates the disease was spread out to the entire city by this time. On May 20, the epidemic is on its peak. Several additional areas show darker yellow than the day before (see black circles in Figure 5.3). These areas have been identified to be the hospitals of Vastopolis. Now we can understand that many people start visiting the hospital to treat the disease.

Regarding the task typology by Brehmer and Munzner [23], epidemic monitoring supports decision making. Thus, visual information is *presented* to the analyst or operator. One needs to keep track and make decisions based on rapidly updating information on a detailed level, but also at a relating global scale. The aggregated off-screen visualization enables to *lookup* and *locate* epidemic developments when zoomed in. This way, one can *identify* new outbreaks that relate to hospitals, *compare* the epidemic impact, or *summarize* areas with similar distributions, while retrieving detailed information in one particular location.

5.4.3 Use Case: Scatterplot Navigation

In this example, I stress the generalizability of the technique. Figure 5.4 presents a scatterplot consisting of five thousand data points. With application to multivariate data, such a scatterplot can be the result of the DR technique MDS. Proximities between points indicate similarity, which is why patterns, such as clusters or outliers, are of particular interest. In this example, the points are aggregated using the grid, and no clustering is applied before exploration. To maintain a coarse overview while inspecting patterns in detail, I choose a relatively narrow

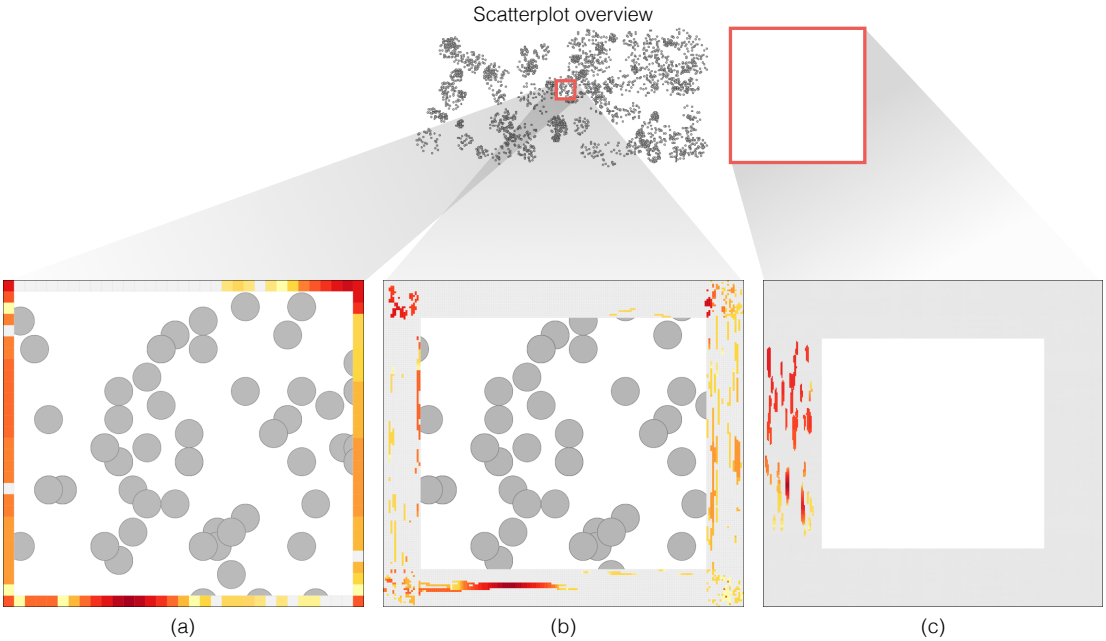


Figure 5.4: *Exploration of a scatterplot consisting of five thousand data points. (a) I perform zooming and panning operations on the scatterplot (indicated by the red rectangle). The context information is presented in the border region of the display. In this case, I choose a single row visualization with relatively large cell dimensions. (b) I increase the number of rows while decreasing the cell dimensions. As a result, we retrieve a more detailed view on the dataset. For instance, the bottom right corner reveals three hotspots that can also be seen in the overview. (c) To show that context, as well as overview, are preserved, this image outlines the visualization if the viewport lays outside the data space. On the left border, we can still perceive a reproduction of the data space.*

border region and only one grid row. The result of the zooming interaction is depicted in Figure 5.4 (a). We can see that the direction to dense point areas is salient. However, the off-screen data topology is not visible impairing the ability to assess spatial similarities.

To visualize the data topology, I reduce the cell size and increase the number of grid rows. As a result, more detail is provided due to the small cell size, and the topology is apparent. Note, that the cell size and the amount of grid rows can be interactively adapted according to the task at hand or the user preferences. The result is depicted in Figure 5.4 (b). Despite the high zooming factor, the context-aware off-screen visualization reveals spots of interest as well as dense and sparse areas. On the bottom right, three very dense spots can be identified. Also, the differences between the top and the bottom are striking: Whereas the bottom area exposes very dense point clusters, the top area is sparse by means of the number of points in total. Also, sparse areas can contain dense spots, as it can be seen in the top left corner.

In Figure 5.4 (c), I particularly emphasize the context-awareness of the technique by moving the viewport entirely outside the data space. The left area of the border visualizes a reproduction of the data space. These characteristics are helpful in getting a detailed idea of the data which is not displayed in the current viewport while inspecting parts in detail.

Scatterplot navigation can be categorized as *discover* intent [23]. In the case of multivariate projections, it is of interest to learn about relations and patterns in the data. When drilling

down to certain patterns of interest, one can apply off-screen visualization to show the multivariate overview. Using the presented approach, I can *explore* or *locate* certain patterns based on their density or correlation. However, this approach is limited with regard to providing additional information. It poses a challenge to *identify*, *compare*, or *summarize* patterns based on attribute values. So far, it is possible based on the relations to each other in 2D space. I take up this issue with the following described off-screen techniques.

5.5 Extrinsic Visualization of Integrated Spatial Uncertainty

In this section, I take first steps towards integrating two attribute values into the visualization of off-screen information. To answer the question *How to integrate two attribute values?*, I consider the field of spatial uncertainty visualization. Generally speaking, spatial uncertainty visualization aims at presenting data and its inherent uncertainty simultaneously, which qualifies as motivation for the aspired solution. The visualization of uncertainty in combination with the actual data values is essential for informed decision making where the quality of the underlying data plays a crucial role. The data exploration implies two challenges: Firstly, data values and corresponding uncertainties need to be integrated into a meaningful visualization, and secondly, analysis tasks often require to focus certain regions. For example, when inspecting temperature distributions and their quantified uncertainty (see Figure 5.6), analysts will need to examine locations in detail but also draw conclusions regarding their comparison to the surrounding. Again, one has to compromise between *Overview and Detail*. This section builds upon the previous Section 5.4, and introduces a glyph-based visualization as means to combine two attribute values.

Recent works motivate this approach in the field of uncertainty visualization. MacEachren et al. [147] asserted that importance should not only be given to the visual-syntactic with which uncertainty measures are matched with visual variables, but also to the way data and uncertainties are linked and represented. As such, Kinkeldey et al. [127] mention three prominent dichotomous categories for uncertainty visualization, considering work of Howard and MacEachren [92] and MacEachren [146], among others:

- **intrinsic/extrinsic** with respect to situating data and uncertainty. *Intrinsic* refers to the manipulation of the visual variable to represent uncertainty, and *extrinsic* to the integration of additional glyph-based representations.
- **coincident/adjacent** with respect to view organization. *Coincident* means that data and uncertainty are integrated into one single view. *Adjacent* is the typically juxtaposed representation of data and uncertainty.
- **static/dynamic** with respect to the interactive nature of the display. *Static* refers to a static representation, where interaction is disabled. *Dynamic* visualizations enable interaction.

Existing uncertainty visualizations typically focused on intrinsic, coincident, and static techniques, while dynamic extrinsic and adjacent techniques are seldom being used [127]. This

is understandable, because it poses a real challenge to visualize data and uncertainty in one view and at the same time visually separate data and uncertainty. However, this is a hard requirement for off-screen visualization. Glyphs have become most popular among extrinsic visualizations due to their multivariate nature, and are utilized to represent variables through various parameters such as location, shape, size, color, orientation, aspect ratio, or curvature [22]. Works by Pang [164] and Cliburn et al. [43] have demonstrated the use of glyphs for uncertainty visualization in geo-spatial data under various settings.

This glyph-based representation can be applied to any data that requires visualizing two integrated attribute values. For example, results of MDS introduce a pairwise location error, which can be integrated into the visualization using this glyph-based approach.

Following, I contribute an extrinsic uncertainty visualization using the Figure-Ground organization that is integrated into the border region of the viewport for overview-preservation. The extrinsic glyph design allows the user to discretely perceive data and uncertainty values using the occlusion metaphor: occluded data values appear less certain than non-occluded values. Furthermore, I showcase the usefulness in a use case.

5.5.1 Technique: Extrinsic Uncertainty Visualization

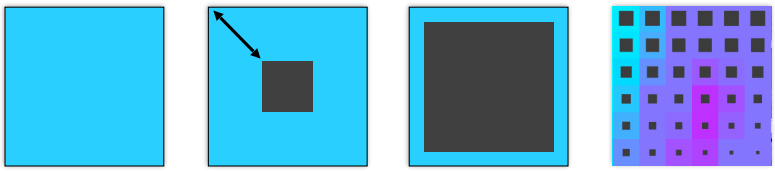


Figure 5.5: *Occlusion metaphor: the less occluded a data value is, the less uncertain it is. From left to right: (1) Minimal uncertainty. The grid cell is not occluded. (2) Partial uncertainty is presented as the distance from the cell's boundary to the occluding rectangle's boundary. (3) Maximum uncertainty leaves a whit of the data value. (4) Visualization in a grid-based environment.*

In many cases, uncertainty data is determined and presented as a grid, which suggests the choice of a grid-based visualization. This choice entails different advantages: The visualization is overlap-free and also generalizable because any glyph representation can be integrated into grid cells. When squeezing information into the border region, there might not be enough space to visualize each data value and its uncertainty, derived from the initial grid, separately. Hence, I first map each data point to the border region and then overlay all mapped entries with the grid-based glyph.

Regarding the glyph design and the integrated visualization of data and uncertainty, I encode two visual variables in addition to location [148]: color reflects the data value, and size of the occluding rectangle its uncertainty value. I make use of the occlusion metaphor to encode uncertainty and add a black rectangle on top of the grid cell. The size of the black rectangle corresponds the uncertainty value. The less occluded the data representation is, the less uncertain it is (see Figure 5.5). The choice of the rectangle size to indicate uncertainty results from the grid-based approach. However, uncertainty is not represented precisely this way, but provides an effective overview of the data; the accurate perception of uncertainty is

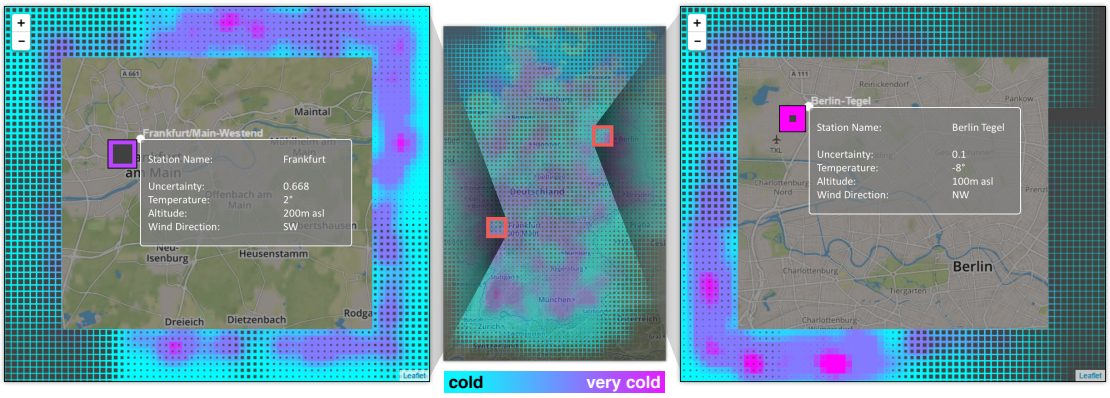


Figure 5.6: *The uncertainty visualization applied as topology-preserving off-screen technique. The center image shows the temperature distribution in Germany. The extrinsic uncertainty visualization allows an easy identification of (a) uncertain regions, (b) certain regions, and (c) regions without any values. Left and right image show different levels of detail within the border.*

not a strict requirement of this application. In this application scenario, I use the cold color map from cyan to magenta to visualize the varying cold to very cold temperature values. These colors show a high saturation and can be perceived and identified, even if they are partly occluded by a dark rectangle since the contrast is very high [180]. This approach is adopted from Oelke et al. [163], who used a similar glyph representation for visual opinion analysis. Furthermore, Stoffel et al. [193] visualized electoral results and used the distance from outer to inner shape to visualize first and second placed parties.

I additively derive this design decision to use the occlusion metaphor from Edgar Rubin – data and uncertainty are grouped according to the *Figure-Ground Organization* [176]. Depending on what the user focuses, she either perceives the actual data values (background) or the uncertainty values (foreground).

As depicted in Figure 5.6, I use the glyph in two variations. Either the glyph is integrated into a grid that visualizes off-screen content (left and right images) or simply overlays the map (center image), or the glyph is used to provide discrete information on the map. The left and right images show the glyph as discrete representation and the application to the German temperature measuring stations. A more detailed description follows in the use case.

5.5.2 Use Case: Urban Planning

In this use case, I showcase the integration of glyphs into the border region by application to temperature measurements collected all over Germany. The data was collected by the German Weather Service. Besides possible measurement errors caused by the weather stations, the temperature between weather stations is uncertain, because not every square inch across Germany is measured. This means, the temperature measurements introduce an error – an uncertainty so to say – between the locations where the temperatures were measured. We [2] used the measured data to create an artificial, grid-based scalar dataset that uses the standard

[2]Hereinafter, “we” refers to me and Juri Buchmüller, who preprocessed and provided the dataset.

deviation to the weather stations per grid cell to derive the uncertainty values.

This use case is based on the perspective of a climate researcher, who works for an environmental agency, and is responsible for the planning and implementation of a national scientific measurement infrastructure. A typical task includes the observation of climate changes. Therefore, it is fundamental to enable the researcher to view the overall temperature distribution, meaning the uncertainty distribution for a region, as it indicates the possibly poor quality of measurements in the area. However, for information on the city, or even street level, the researcher requires drill-down capabilities, so that she can find out about uncovered regions where the sensor network can be improved. Also, she can see information about other data attributes, such as station altitude, average precipitation, or sunshine, among others. The integration of many different attributes is typical for multivariate datasets. Figure 5.6 depicts the temperature uncertainty distribution across Germany, before and after the application of zooming and panning operations.

The topology-preserving off-screen visualization combined with the extrinsic glyph representation enable the researcher to compare temperature and uncertainty between off-screen located areas of arbitrary detail and the rest of the dataset. The overview-preservation of uncertainty further enables the researcher to efficiently observe the uncertainty distribution and development in each direction, originating from the currently specified viewport position. In case, a region is of interest because of its uncertainty values, the researcher can directly navigate there by one click, as introduced in the design decisions of this chapter. This interaction saves time-consuming panning and zooming operations.

This use case is also applicable to the visualization and following navigation of multivariate data derived by application of DR techniques, such as MDS. In such case, uncertainty corresponds to the caused projection error. In some cases, it is important to know the distribution of projection errors, even when zoomed in, like for example when assessing the quality of relationships between data records. Concerning the task typology [23], this glyph-based representation qualifies as *discover*, but also as *present* intent, and enables to *lookup* and *locate* attribute values of interest while zoomed in. Furthermore, the exploratory analysis is enabled to *explore* and *browse* the entire depicted information space. Analysts can also *identify* values of interest, *compare* values to each other, and *summarize* value distributions.

5.6 Star Glyph Insets for Visualization of Multivariate Data

An additional challenge, compared to Section 5.5, arises if the visualized data is of multivariate nature, which we encounter in many different domains. In this section, I dynamically integrate *star glyphs* as insets into the spatial representation of multivariate data, thus providing an overview while inspecting details. Star glyphs pose an efficient and space saving method to visualize multivariate data, which qualifies them as integrated data representative.

The integration of a complex glyph representation, however, also points out the limitations of the topology-preserving off-screen approach. It is a trade-off between the readability of a glyph and the topology preservation of the data. In the following, I derive the glyph design and discuss design decisions taken. Furthermore, I demonstrate the usefulness of this

approach in two use cases: The spatial exploration of multivariate crime data collected in San Francisco and the exploration of multivariate whiskey data.

5.6.1 Technique: Star Glyph Insets

This section is inspired by the approach of combining several visualization techniques [217] to a system that addresses the overview-preservation problem. I adapt the idea of aggregation and integration of glyphs, which was introduced in the previous Section 5.5, and the use of insets to provide a data-driven snapshot of off-screen objects proposed by Ghani et al. [72].

Various techniques exist to represent multivariate data visually. Besides scatterplot matrices, parallel coordinate plots, or pixel-visualizations, glyph-based visualizations are well-established alternatives (compare to Chapter 2). The main advantages of glyphs are the compact design and flexibility regarding their layout on the screen. This makes glyph-based visualizations a perfect fit for off-screen visualizations. Glyphs can be integrated into different basic visualization techniques and positioned independently because their reading performance is not influenced by the amount of context information displayed in the background [152].

Glyph Layout

Several data glyph designs exist, hence making it crucial to distinguish between three different mapping categories [212] as discussed in the Background Chapter 2.2.1. Firstly, *many-to-one mappings*, which support the intra-record comparison by mapping all data values to the same visual variable. Then, *one-to-one mappings*, which encode data values with different visual variables. Finally, *one-to-many mappings*, which represent data values redundantly using at least two visual variables.

Based on the mapping strategy, different analysis tasks are supported. Since I do not want to restrict myself in the analysis process, I aim for a design that supports both intra-record and inter-record comparisons. Besides detecting similar data objects, I also aim to enable the comparison between single attributes. Hence, elementary tasks and synoptic tasks should be supported by the data glyph design. This makes many-to-one mappings an appropriate choice, although their design space is limited.

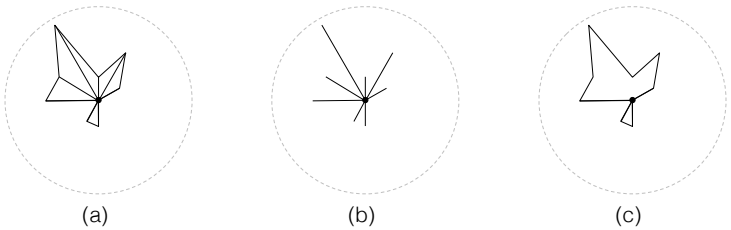


Figure 5.7: *The three considered star glyph variations [185]: (a) The common star glyph uses data lines radiating from the center; maximum values of the data lines are connected and form a contour line. (b) In contrast, the whisker glyph only shows data lines. (c) The sensitivity star glyph only shows the contour line. Quantitative experiments suggest to use the whisker glyph to improve similarity judgments [67].*

Before deriving the final glyph design, I first discuss the layout of attributes. Results from quantitative experiments suggest using circular designs rather than linear ones because

they facilitate the detection of single attributes [66]. The ranking of visual variables by Cleveland and McGill suggests using position or length for displaying the data value rather than color. In combination, radial length encodings are more efficient compared to circular color encodings [161]. The most prominent design with length encoding and radial layout are star glyphs [185]. Figure 5.7 outlines the three different existing variations to choose from: (a) First, the common star glyph that uses data lines radiating from the center of the glyph and are connected via a surrounding contour line to create a closed shape. (b) Second, the whisker glyph using the same encoding, however, the surrounding contour line is removed. (c) Third, the polygon or sensitivity star glyph [38] that displays the contour line only, and thus hides the single data rays.

Results from quantitative experiments suggest using the whisker glyph without the contour line to improve similarity judgments on multivariate data [67]. Additionally, the data rays can be colored to avoid wrong data similarity judgments based on salient shapes [128]. However, enormous amounts of attributes impede efficient color mappings.

Furthermore, I keep the size of data glyphs as small as possible. Labels are removed from the design to allow compact representation. However, little research has yet been carried out on the minimal size of glyphs. Because of this lack of guidance, I decide to use a minimum size of 30x30 pixels for our quadratic aspect ratio, which is considered a convenient size based on previous comments from expert users.

Data Aggregation

As for the grid-based approach presented in this chapter, star glyph insets also require additional space. I build upon the grid-based aggregation of off-screen objects [74] and replace a bin with a star glyph, respectively. Again, the grid dimensions depend on the aspired level of detail; the bigger the grid cell, the less detail is shown, which in some cases is desired.

```
Function aggregate(D, isScale)
    result ← Map<Attribute, Value>
    foreach data object O in D do
        foreach attribute a in O do
            result[a] ← result[a] + O[a]
        end
    end
    return isScale ? buildAVG(result, D.size) : result
End
```

Algorithm 2: Attribute-wise aggregation.

I implemented two aggregation methods for the application to star glyphs: the sum of values as well as the average of values per attribute. Calculating either the average or the sum depends on the data. If the data attributes are assigned to a particular scale, the average value per attribute expresses which taste category is likely to occur in certain areas. In contrast, if the data attributes state countable numerical amounts, the sum of values per dimension is the

superior choice. For example in crime analysis, attributes such as theft or attack are absolute numbers that describe an amount of committed crimes. Thus, we require total crime numbers that occur in certain areas. Algorithm 2 shows the implementation. Parameters are: the set of data objects D which will be aggregated and a boolean value $isScale$ which determines if the values are assigned to a scale. I then initialize an empty map $result$ that contains the attribute as key and the sum or average as value. This map is used later to build the glyph. I iterate all data objects, and for each object O, I iterate the attributes. For each attribute a, I add the value to the corresponding attribute contained in the result map $result[a]$. Finally, if the data is assigned to a scale, the average value per attribute is returned, the already built sum per attribute otherwise.

I provide the user the extent of the aggregation through the background color of the star glyph. I apply $min-max$ normalization (feature scaling) to the amount of aggregated data objects among all bins and then derive the color value for the star glyph background, respectively. I use a linear colormap from black to light blue, whereas black means low and light blue high aggregation.

The star glyph is used in two variations: Either directly plotted on the map, or the glyph is used as an inset for off-screen objects which is described in the following. I motivate the use of star glyph insets, which is why I intentionally and solely apply the glyph to objects located off-screen.

Layout and Interaction

Additionally, I adapt the idea of HaloDot [74], and use transparency as the indicator for the distance between off-screen object and viewport, namely the relevance; objects located near to the viewport are considered to be of higher interest than objects located far apart. This is because it is likely that the user is interested in the surrounding of the area she is currently exploring – once she drills down from overview to detail.

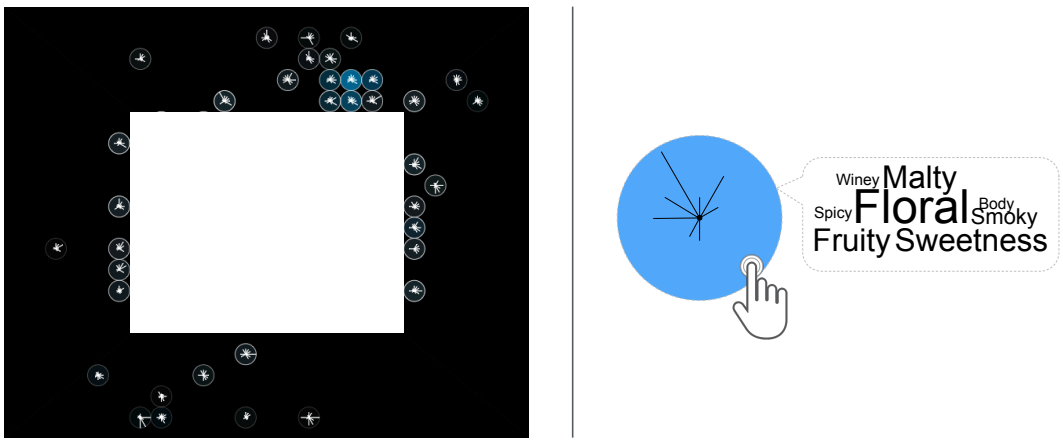


Figure 5.8: *Towards the preservation of data topology. For the attempt to preserve data topology, I reserved a border region of 200 pixels and reduced the size of glyphs to 20x20 pixels. Furthermore, only 86 data records are considered. As we can see, data objects still need to be aggregated (indicated by blueish color), and the sheer amount of small glyphs distributed among the border region complicates efficient perception of dimensions.*

Star glyph insets are space consuming and cannot be perceived within the glimpse of an eye, compared to the mere visualization of data points only. However, they provide an efficient overview. Therefore, and contrary to the approach of EdgeRadar [82], I argue that for the use of multidimensional insets the preservation of the full data topology is hardly possible and it also does not adequately supports navigation. Preservation of full topology results in potentially small glyph sizes, so that attributes are not efficiently perceivable, as depicted in Figure 5.8. The Figure highlights the drawback of combining topology preservation and multivariate data – additional cognitive load is given by the fact that small glyphs are distributed according to the position of off-screen objects among the border region of the viewport. The Figure shows a small dataset consisting of 86 data objects. This effect intensifies with an increasing amount of data objects. This is why I following rather increase the size of the insets to showcase the technique for the sake of readability. However, it has yet not been proven up to which minimum glyph size a star glyph is fully interpretable. Therefore, I leave the choice to the user up to which level the topology should be preserved.

In particular, when showing multivariate data, interaction is key. To help users to read the exact data values, I implement an enlarged version of the glyph in a detailed view (see Figure 5.9). This visualization offers a closer look at data values and adds labels to the attributes. Since data glyph designs are used more often for synoptic tasks in overview visualizations, such a close up helps to read data values more accurately and, therefore, supports elementary tasks like direct lookups as well. Once the user clicks on a star glyph inset, the detail view is updated.

When hovering an inset, a word cloud is visualized next to the glyph. The visualization of a word cloud facilitates overview because important terms have bigger font sizes than unimportant terms. Each term in the word cloud corresponds to one data attribute. Like in Wordle [210], each term is assigned a weight. I derive the weight from the value in the glyph, which is assigned to the term.

5.6.2 Use Case: Crime Analysis

The city of San Francisco offers collected crime data through its open data portal [3]. Data from this data portal was already used in Chapter 3. Figure 5.9 shows all locations of committed crimes visualized in a geo-spatial context. We can clearly identify crime accumulations in the northeastern area of San Francisco. This area is also known as a tourist attraction. The dataset comprises 36 different crime categories ranging from *larceny/theft*, to *vehicle theft*, up to *assault and kidnapping*. Figure 5.9 provides an overview of all crime categories I make use of as attributes within the star glyph inset representation. In this example, a lens is positioned on top of the map, and the content is aggregated into a detailed star glyph view, juxtaposed to the visualization. This detailed view is also shown whenever the user clicks on an inset that represents aggregated off-screen visualization. As discussed, topology-preservation is possible within the visualization, however, for the sake of the glyph readability. In these use cases, I showcase the technique using only one single row of visual proxies but highlight that it is possible to decrease the glyph size and increase the border region so that the topology is present.

[3] SF OpenData: `https://data.sfgov.org/`

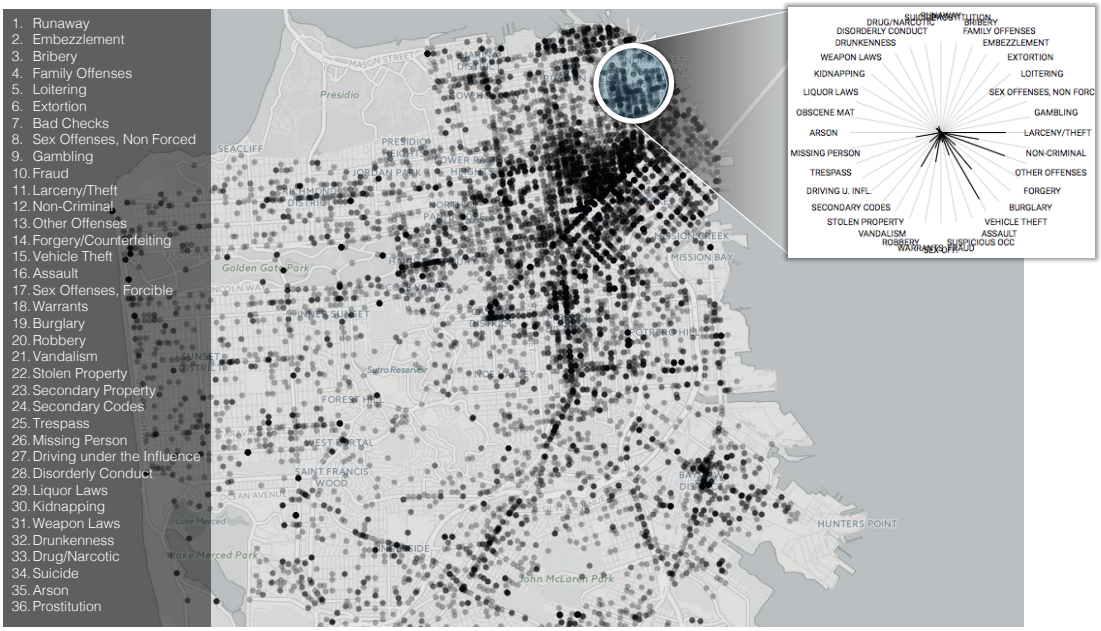


Figure 5.9: *Mapping of all committed crimes to the map of San Francisco for the period of June 1st, 2015 to June 30, 2015. In total, the dataset consists of 12480 committed and registered crimes. Left: a list of all considered crime categories. The concept of an interactive lens, or the click on a star glyph inset, opens an additional view where the user can investigate the distribution of attributes on a detailed level.*

This use case involves a user who wants to explore the crime situation in San Francisco before planning her holiday. To do so, she drills down to the downtown area of San Francisco, near Union Square. In order to highlight the results of the star glyph insets, I intentionally do not show any visualizations related to data objects contained in the focus region.

Figure 5.10 depicts the data for the entire month of June 2015. While exploring the downtown area of San Francisco, the user is interested in surrounding areas to navigate to. The combination of color and transparency provides a great overview of the surrounding areas. The user can clearly see, that crimes in the north are in total closer to her current position than crimes committed in the south of the city. Also, the blueish colored background of the star glyph insets in the south indicates a higher aggregation level, thus revealing more committed crimes. In fact, the southern area (e.g. South Market) of San Francisco is well known to have high crime rates. This is also revealed by the inset word cloud generated from the glyphs. In all directions, the crime category *larceny/theft* dominates the number of incidents. It cannot only be clearly seen with the help of the word clouds, but also the insets highlight this very prominent dimension. However, the categories differ among the other crime categories. For example, in the southern area of San Francisco, *other offences* and *vehicle theft* also stick out, compared to the other regions. Furthermore, in the eastern area of San Francisco, *burglary* appears to be prominent.

Note, that the word clouds support the user to distinguish between the frequencies of different crime categories at a glance. This use case can be considered as a *present* or *discover* intent, depending on whether it is used for decision making in the visual analysis of the data.

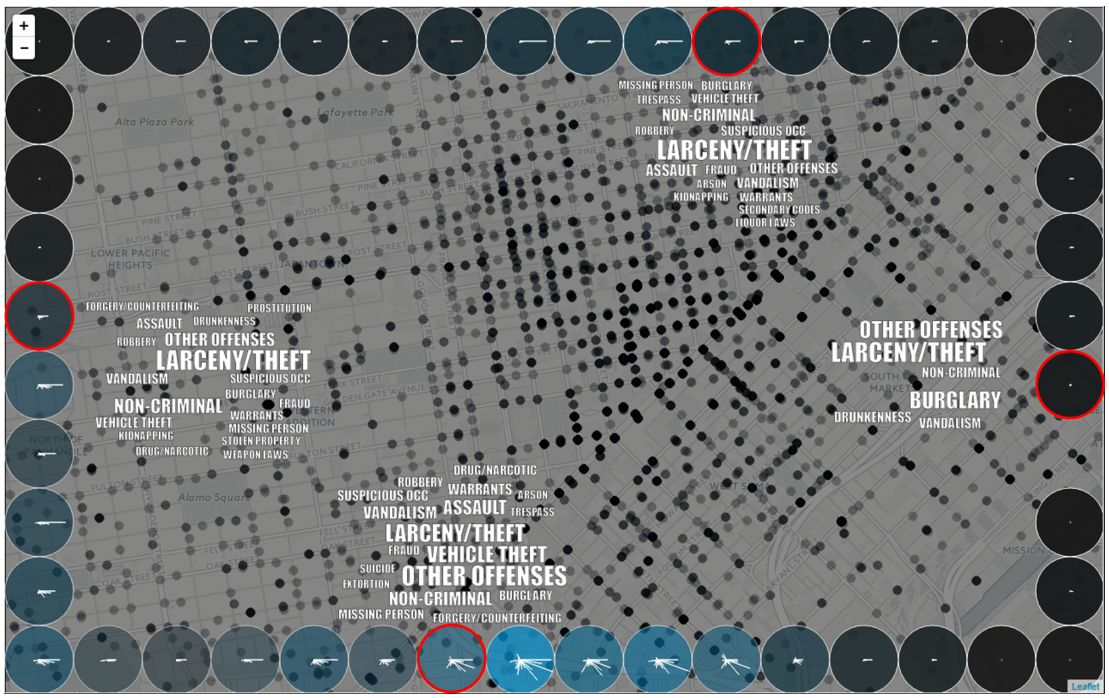


Figure 5.10: *Overview-preservation of committed crimes in San Francisco over a period of one month (June 1st, 2015 to June 30, 2015). The visualization shows a cutout of the city center of San Francisco, near Union Square. At the same time, by inspecting and hovering the insets, the user can see that in the northern as well as in the western directions, the category larceny/theft is very prominent. This is indicated on the one hand by the word cloud and on the other hand by the star glyph inset, which clearly points out this single dimension. However, the eastern and southern areas are different. While the eastern area seems balanced, the southern area shows more other offenses, assaults, and vehicle thefts.*

The glyph-based off-screen representation not only enables to *lookup* and *locate* patterns and relations among data records but also attribute distributions while inspecting details. In an exploratory environment, one also can *browse* and *explore* the entire space. Using this glyph-based representation, the user can *identify*, *compare*, and *summarize* interesting patterns and attributes. Regarding the described use case, areas showing a certain attribute peak can be identified and then put into relation to other areas in order to find the safest place. Also, entire no-go boroughs can be identified by summarizing neighboring areas.

5.6.3 Use Case: Scottish Whiskey Data

The second use case includes the context-preservation and navigation of geo-spatial whiskey data [162], as well as the spatial layout derived by the multivariate projection MDS. The multivariate whiskey dataset comprises 86 Scottish distilleries and 12 taste categories, respectively. In this use case, I aim to explore the given variety of whiskeys without a particular analysis task at hand. While one might have certain preferences regarding whiskey compositions, such as being *sweet*, *fruity*, and a little bit *smoky*, a key task for whiskey connoisseurs is to find new, exceptional types. Figure 5.11 shows the overview of all distilleries on the top right, distributed across Scotland. In particular, in the area of Dufftown, many distilleries seem to

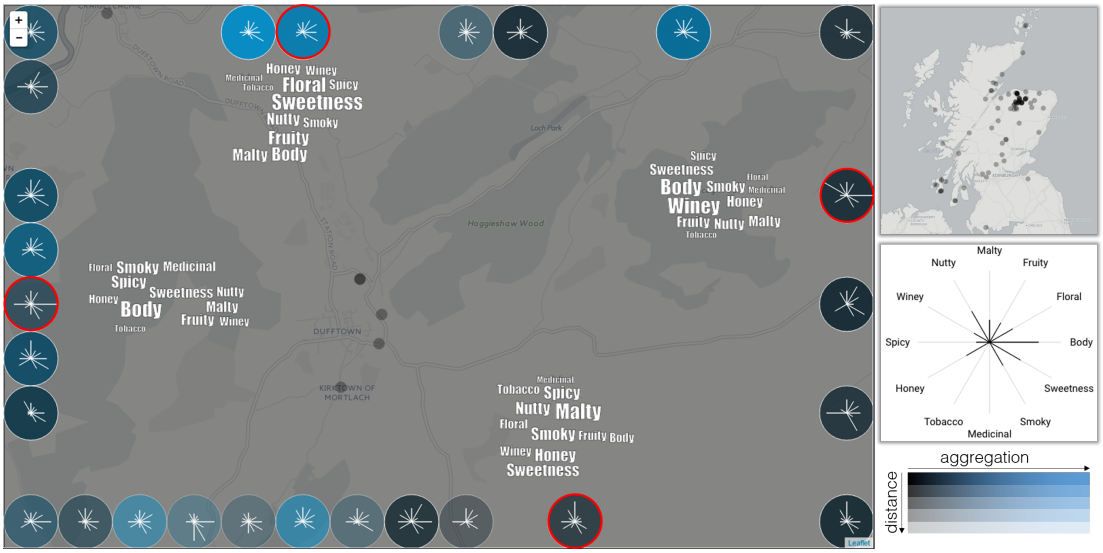


Figure 5.11: *Visualization of multivariate whiskey data. I drill-down to town-level to see details in the area of Dufftown. Through interactions, I can analyze details and explore the surrounding of the focus area. In the northern part, the I select sherry-intensive whiskeys – this can be identified by looking at the star glyph which reveals whiskeys are mostly* malty, fruity, floral, *and* sweet. *This is also shown in the detail view. With the help of star glyph insets, I can seek for taste distributions that I enjoy most, or can even efficiently compare them.*

be located, which is indicated by the high saturation of the black color. Figure 5.11 gives an overview of all included taste categories on the right-hand side. First, I explore the whiskeys on the map, and then I apply MDS to the data and explore the corresponding scatterplot for similar whiskey varieties.

In the first exploration phase, I explore the map of Scotland for different whiskey taste categories. Therefore, I zoom the most promising area: Dufftown. Figure 5.11 shows the result of the performed zooming interaction. While Dufftown is in the focus region, I can still explore the surrounding areas. The blueish colored background of the star glyph insets indicates the level of aggregation. In the northern part, I select the very blue glyph. Both detail view and word cloud immediately confirm my assumption based on the star glyph inset and reveal that 12 different distilleries have been aggregated leading to an overall result of mostly sherry-intensive taste categories: *sweetness, floral, body, fruity, malty*.

At the same time, I detect a whiskey in the south that seems very *malty*. It is interesting that the neighbor star glyph on the right has nearly the same shape. Also, I detect in the western area a star glyph that reveals more *body*-intensive whiskeys and in the eastern part a star glyph that reveals *winey*-intensive whiskeys.

However, I decide to seek for similar whiskeys, and therefore I apply MDS to the data. Figure 5.12 depicts the result on the top right. The result is a scatterplot that contains all distilleries mapped according to their similarity value to each other. I zoom to a very distinct cluster as can be seen in the Figure. Star glyph insets are generated and visualized for the northern and eastern areas of the scatterplot. It is protruding that the star glyphs in the north as well as the star glyphs in the east look somehow similar. By hovering the corresponding

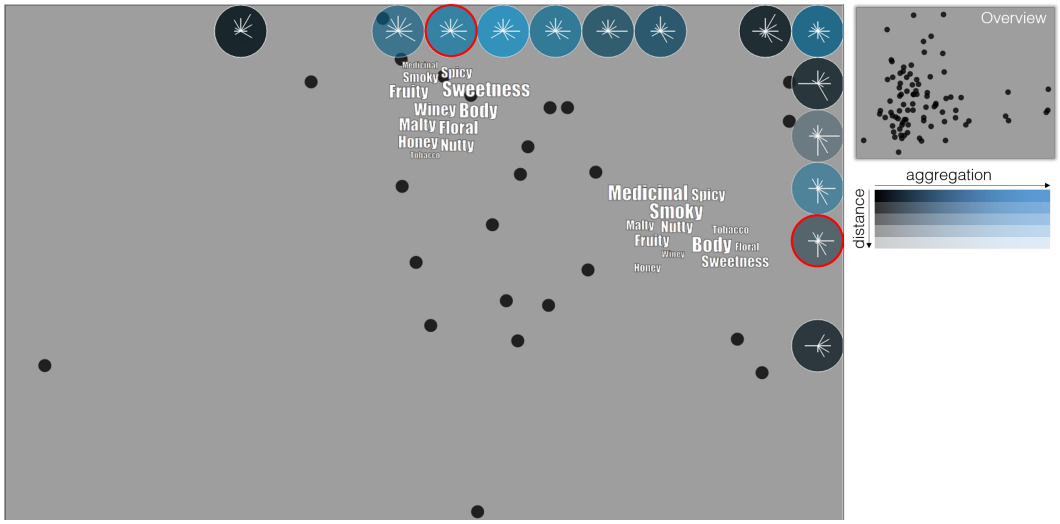


Figure 5.12: *Application of MDS to the whiskey data. While examining a cluster of diverse whiskeys in the focus region, the star glyph insets clearly reveal that in the northern part of the MDS scatterplot the taste categories* winey *and* body *dominate, whereas in the east the taste categories* body, smoky, *and* medicinal *protrude.*

glyphs, the word clouds reveal that the northern ones are dominated by the taste categories *body* and *sweetness*. The glyphs in the eastern area are dominated by the taste categories *body*, *smoky*, and *medicinal*.

This use case depicts the traditional exploratory analysis; the *discover* intent in terms of the task typology [23]. The off-screen visualization integrated with the multivariate projection of whiskey data allows connoisseurs to *lookup* and *locate* interesting whiskeys while zoomed in. Also, they can simply *explore* the space and *browse* different taste categories in order to find related, new whiskeys. To do so, the glyph-based representation enables to *identify* interesting taste categories, *compare* whiskeys based on taste specifications, or *summarize* whiskeys based on taste commonalities, such as being floral or winey.

5.7 User Study: Topology-preserving Aggregation against HaloDot

I presented a topology-preserving off-screen visualization based on grid-based aggregation in Section 5.4. Based on this concept, I proposed two additional glyph-based techniques towards the off-screen visualization of multivariate data. In this section, we [4] compare the concept of aggregated topology-preserving off-screen visualization (in short TopOff) to HaloDot [74], a state-of-the-art off-screen visualization technique. HaloDot is based on the well-known off-screen technique Halo [12]. The general idea is to use arcs (halos) to represent off-screen objects. Therefore, each off-screen object is surrounded by a ring, whereby the extent of the ring is chosen in such manner that it intersects the viewport. The resulting visible arc should indicate distance and direction of the off-screen located object. Because this technique is prone for overplotting, HaloDot goes one step further and applies a grid-based aggregation;

[4]Hereinafter, “we” refers to me and Bum Chul Kwon, who helped in designing and conducting this user study.

an arc represents each off-screen grid cell that contains at least one object. The techniques presented in this Chapter are based on this type of aggregation, but further introduce topology preservation. To investigate whether there is a real benefit attached to the preservation of the overall topology of off-screen content, we conducted a user study against HaloDot. I, furthermore, considered rather large datasets (more than 1000 data points) to meet the requirements of real-world applications. For a more fair comparison, we improved HaloDot by applying arcs to the calculated clusters instead of grid cells, which significantly reduces the amount of visualized arcs. Furthermore, we mapped the number of points that are contained in each cluster to the thickness of the respective arc.

We compared TopOff with HaloDot on three different tasks, which are described in the following. Also, as prior studies (e.g., [73]) indicate, the size of data influences the performance of off-screen visualization techniques. Thus, we also tested whether the size of the data can influence TopOff.

5.7.1 Tasks

We compared TopOff with HaloDot on three different tasks: (1) context-interpretation comparison task, (2) context-interpretation overview task, and (3) target selection task. These three tasks are highly relevant to monitoring and navigation tasks, where off-screen visualization techniques are highly applicable.

Task 1: Context-Interpretation – Comparison Task

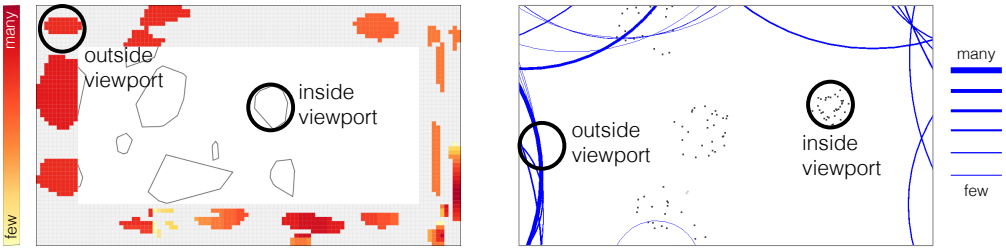


Figure 5.13: *Schematic representation of the context-interpretation comparison task. Participants were asked to decide whether the biggest aggregated area lies inside or outside the viewport.*

The *comparison task* was designed to test whether the asked point of interest (POI) is located in-screen or off-screen (see Figure 5.13). In monitoring, this task reflects situations where analysts need to quickly estimate the relative differences of measurements inside and outside the screen area. To make the evaluation results comparable, we used the nine different datasets in a randomly chosen order for TopOff as well as for HaloDot. In this task, participants were asked to determine whether the biggest cluster, with regard to contained points, is visualized in-screen or via the corresponding off-screen technique. Therefore, a static image was presented to participants using TopOff or HaloDot. After each trial, participants responded to a survey question on how confident they were with their decision via a 4-level scale: *1 = unsure, 2 = somewhat unsure, 3 = somewhat confident* to *4 = totally confident*.

Note, that this scale does not follow a Likert scale so that participants were forced to decide whether they were confident or not.

Task 2: Context-Interpretation – Overview Task

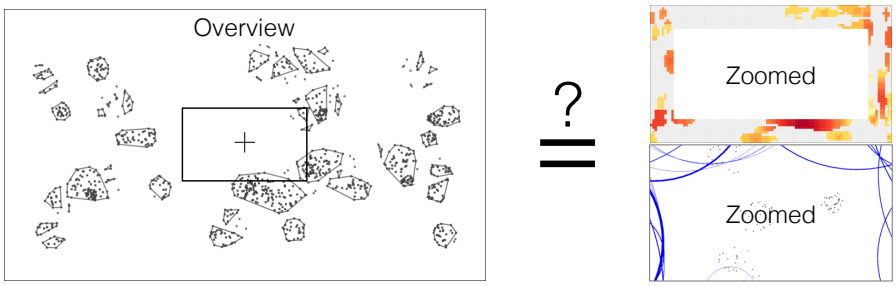


Figure 5.14: *Schematic representation of the context-interpretation overview task. Participants were asked to decide whether one zoomed image – out of a selection – corresponds to the overview scatterplot image.*

The *overview task* aimed to test the performances of techniques on establishment of overview of the whole datasets (see Figure 5.14). In monitoring, this task reflects how analysts can be aware of the overall picture of entire data points without changing the zoom level. For this task, we presented a static overview image of a scatterplot. Within the given image, we marked the area which will be zoomed. Below the scatterplot, we offer a selection of three images showing the zoomed area using a given technique: HaloDot or TopOff. One out of the three options corresponds to the correct scatterplot, so the other two are for distraction. After each trial, the participants specified the confidence level of their choice via the 4-level scale.

Task 3: Target Selection Task

In the *target selection task*, participants were asked to select a specific point of interest interactively, a cluster with the largest number of points in our case, among given data points randomly spread out in a map space outside the focal viewport. This task intends to test the effectiveness of given off-screen visualization techniques for navigation. We collected execution time (i.e. the time elapsed until selection) and accuracy, which is defined as the correctness of selection.

5.7.2 Hypotheses

Considering the topology preservation, aggregation, and different data sizes, we derive the following hypotheses.

Context-Interpretation – Comparison

H1.1: *The task performance time will be less with TopOff than with HaloDot.* Given the topology preservation of off-screen data, we expect participants to be faster.

H1.2: *The task performance accuracy will be higher with TopOff than with HaloDot.* Because the topology is preserved, we expect participants to be more accurate.

H1.3: *The confidence level will be higher with TopOff than with HaloDot.* We expect participants to be more confident about their decisions because points/clusters can be put into relation to each other effectively.

Context-Interpretation – Overview

H2.1: *The task performance time will be less with TopOff than with HaloDot.* Given the topology preservation of off-screen data, we expect participants to be faster.

H2.2: *The task performance accuracy will be higher with TopOff than with HaloDot.* Because the topology is preserved, we expect participants to be more accurate.

H2.3: *The confidence level will be higher with TopOff than with HaloDot.* We expect participants to be more confident about their decisions because points/clusters can be put into relation to each other effectively.

Target Selection Task

H3.1: *The task performance time of TopOff will be comparable to HaloDot.*

H3.2: *The task performance accuracy of TopOff will be comparable to HaloDot.*

For the hypotheses **H3.1** and **H3.2**, we expect that the performance time and accuracy will be comparable between both techniques because the preservation of the topology is not decisive for navigating to a specific point. This is also due to the improvements applied to HaloDot. TopOff provides the overview without interaction, and the participant has to look for the corresponding target and then navigate there. In contrast, HaloDot is cluttered and does not provide the topology overview. However, when we were using HaloDot, we encountered that through panning interactions certain aggregations became salient. We are aware that using panning interactions is not wanted when demanding the overview.

5.7.3 Design & Procedure

To test the hypotheses, we designed a within-subject study. The dependent variables were accuracy, time, and confidence level.

The experiment was carried out in a quiet room at our university. Each participant was placed in front of the monitor and received an introduction to the topic: Before each task, the participant was presented a detailed description exemplifying the used technique within the task. Furthermore, for each technique within a task, participants performed a training phase. A training phase consisted of three examples for respective tasks and per technique. After each example, the correct solution was presented.

All tasks were performed for both techniques, TopOff and HaloDot and applied to three different data sizes: one thousand, three thousand, and five thousand data points of interest. We conducted a two-stage user study. First, we evaluated the effectiveness of both techniques on context-awareness (**H1** and **H2**). Second, we evaluate the performance on target selection (**H3**). Using the first stage, we wanted to show whether TopOff helps users preserve overview

while analyzing parts in detail. It is expected that the context-awareness tasks are in favor of TopOff because it particularly aims at topology-preservation. At the same time, we wanted to show TopOff's comparable performance in target selection. In order not to influence mutual results, we decided to conduct a two-stage user study with two user groups.

The first user study addressed context-awareness. To test **H1** and **H2**, each participant accomplished 36 trials, which means 18 trials per task: *context-interpretation comparison* and *context-interpretation overview*. For the experiment, the tasks followed a random order. The second user study addressed target selection. To test **H3**, each participant accomplished 18 trials. For each task, the 18 trials are characterized as follows: 9 trials per technique and within these, 3 trials per data sizes (i.e., 1000, 3000, and 5000 data points), respectively. For the experiment, the trials followed a random order. This means, within a task, we randomized the order of the used technique; and within a technique, we randomized the data size. After all 54 trials, participants were asked to respond to a demographic survey.

Apparatus The experiment was conducted on a computer with the fixed window height of the prototype of 1390 pixels, and fixed width of 710 pixels. A keyboard and a mouse were provided for user input and interaction.

Participants In total, 23 participants were recruited (mean age = 27.9, min age = 20, max age = 51) for the first-stage user study. Among them, 15 participants majored in computer science, 6 in engineering, and the rest of 1 in economics and civil services, respectively. For the second-stage user study, 22 participants were recruited (mean age = 32.7, min age = 20, max age = 71). Among them, 14 participants majored in computer science, 3 in engineering, 2 in administrative studies, and 1 in each, economics, social science, and teaching. All participants had normal or corrected-to-normal vision (20/20). All participants used computers on a regular basis (30 minutes to 8 hours a day) and had a basic understanding of visualizations.

5.7.4 Results

We used ANOVA for the task completion time and a non-parametric Friedman's test for accuracy as well as for the confidence score. Our test results failed to confirm **H1.1**, but successfully confirm **H1.2** and **H1.3**: *in context-interpretation comparison task, TopOff showed higher accuracy than HaloDot with equal task performance time.* There was no significant difference between techniques ($p = 0.809$, $F(1,388) = 0.06$) and between datasets ($p = 0.318$, $F(2,388) = 1.15$) on task completion time. On the other hand, there was significant difference between techniques ($p < 0.001$, $\chi^2(1, N = 396) = 31.179$) and dataset sizes ($p = 0.002$, $\chi^2(2, N = 396) = 12.138$) on accuracy. In addition, there was significant difference between techniques ($p < 0.001$, $\chi^2(1, N = 396) = 25.372$) and between dataset sizes ($p < 0.001$, $\chi^2(2, N = 396) = 26.452$) on confidence. Figure 5.15 shows that participants performed more accurately with TopOff. Additionally, Figure 5.16 (a) shows that participants felt more confident as post hoc analysis showed ($1000 > 3000 > 5000$); Figure 5.16 (c)

achieved a higher accuracy even across different data sizes as post hoc analysis showed (1000 > 5000). It also shows a tendency that the error rate increases as the dataset size increases.

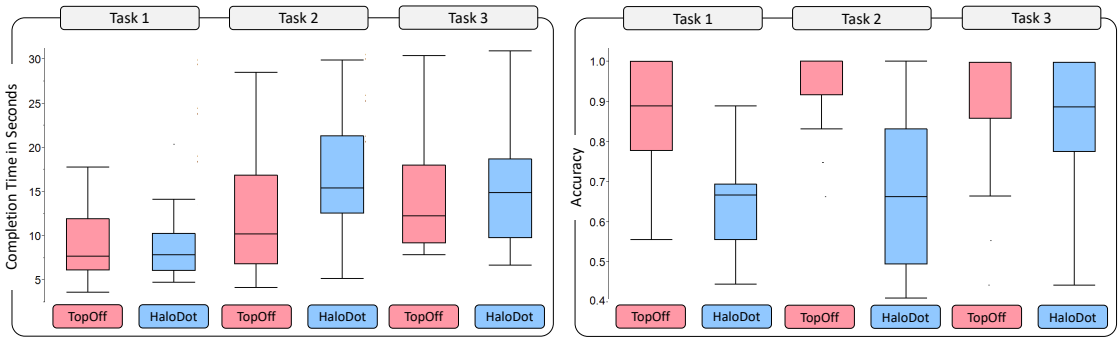


Figure 5.15: *Experiment results for the context interpretation comparison (Task 1) and overview (Task 2), as well as the target selection (Task 3) task. We could prove that for Tasks 1 and 2, TopOff performs significantly better than HaloDot regarding accuracy with a tendency to a faster completion time.*

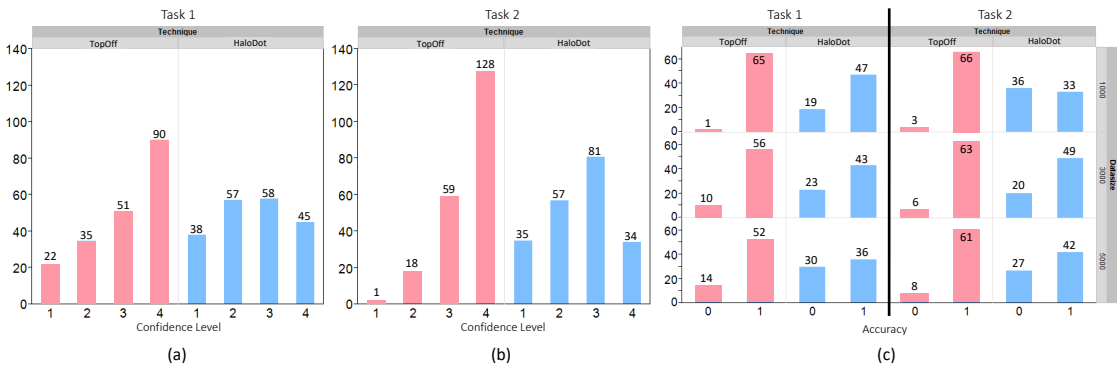


Figure 5.16: *The histograms show the distribution for TopOff and HaloDot depending on confidence level and accuracy regarding data size. (a) and (b) show for both, context-interpretation comparison task and context-interpretation overview task, the distribution of the confidence level (1- unsure, 2 - somewhat unsure, 3 - somewhat confident, 4 - totally confident). (c) shows the distribution of accuracy (0 - wrong, 1- correct) according to technique and data size.*

Our test results successfully confirmed **H2.1**, **H2.2**, and **H2.3**: *in context-interpretation overview task, TopOff showed higher accuracy, less task completion time, and higher confidence than HaloDot.* There was significant difference between techniques ($p < 0.001$, $F(1,388) = 26.48$) but no difference between datasets ($p = 0.298$, $F(2,388) = 1.15$) on task completion time. In addition, there was significant difference between techniques ($p < 0.001$, $\chi^2(1, N = 396) = 61.429$) but no difference between dataset sizes ($p = 0.002$, $\chi^2(2, N = 396) = 3.583$) on accuracy. In addition, there was significant difference between techniques ($p < 0.001$, $\chi^2(1, N = 396) = 118.624$) and between dataset sizes ($p = 0.012$, $\chi^2(2, N = 396) = 16.382$) on confidence. Figure 5.15 shows that participants performed more accurately with TopOff. Furthermore, Figure 5.16 (b) shows that participants felt more confident as post hoc analysis showed (1000 > 5000); Figure 5.16 (c) achieved a higher accuracy even across different data sizes as post hoc analysis showed (1000 > 5000). It also shows tendency that the error rate increases as the dataset size increases.

The test results confirm our hypothesis **H3** (Figure 5.15): *in target selection task, the two techniques showed comparable performances in terms of task performance time and accuracy*. Our test results did not show significant difference between techniques ($p = 0.264$, $F(1,257) = 1.25$) and between datasets ($p = 0.344$, $F(2,257) = 1.07$) on time. In addition, we did not find significant difference on accuracy between techniques ($p = 0.140$, $\chi^2(1, N = 264) = 2.178$) as well as data sizes ($p = 0.211$, $\chi^2(2, N = 264) = 3.109$).

Validity of results We showed the benefit of TopOff on context-interpretation tasks compared with HaloDot. The results are in favor of the topology-preserving off-screen visualization using aggregation. One may argue that particularly Task 1 and Task 2 were designed unfairly because topology-preservation is always in favor for static context-aware tasks. However, we argue that the benefits of topology preservation have not been shown in the context of off-screen visualization, so far. Also, we evaluated TopOff against an improved version of HaloDot, which is the most relevant and related technique. Compared to all other off-screen techniques, we identified HaloDot as the most advanced one regarding clutter reduction with the help of aggregation.

Another concern may raise when considering other overview-preserving families, such as the classical overview-and-detail or focus-plus-context interfaces. In Chapter 2, I show that to the best of my knowledge these techniques are mainly operating in image space. Off-screen visualizations, however, are data-driven posing a challenge to compare them to conceptually different techniques. To be able to preserve data characteristics, overview-and-detail or focus-plus-context techniques require improvements to such extent that they depict a new technique. Beyond that, comparing two different families of techniques, here image-based versus data-driven, is also not fair; results will not express whether the design decisions taken for off-screen visualization indeed improve this set of techniques.

5.8 Discussion & Future Directions

I presented three topology-preserving off-screen techniques that build upon each other and apply aggregation to overcome the vast amounts of data. So far, off-screen visualization has mainly been motivated in view of geo-spatial navigation. I outlined that these techniques cannot only be applied to geo-spatial data, but also to projections of multivariate data. Due to the data-driven nature, off-screen visualization is effective in real-world tasks and scenarios including monitoring or uncertainty analysis. The evaluation showed that I significantly improved off-screen visualization over state of the art regarding the preservation of the data overview. Still, there is room for improvement to achieve effective off-screen visualizations. Therefore, we need to tackle certain challenges. Following, I discuss topology-preserving off-screen visualization regarding derived challenges and point out future directions.

Computational Efficiency and Scalability One of the main technical challenges represents scalability of off-screen visualizations regarding large datasets and high dimensional data. Since off-screen visualizations aim to provide overview and details at the same time while users

are performing interactions, it is highly desirable to process the computation of aggregation and update of cues promptly. Due to the limited computational resources, one needs to find a balance between an accurate representation of data and fast processing to ensure seamless interaction. Designers will encounter numerous questions to define scalable approaches to resolve issues, such as: How do we aggregate data up to several dimensions in visualizations? How do we simplify representations for efficient overview? How do we make sure users maintain accurate awareness of data objects while performing interactions?

I presented first steps towards data aggregation. However, the proposed techniques require improvement with regard to scalability. Furthermore, I am not aware of techniques that can handle the possibly sheer amounts of streaming data. Streaming data holds additional challenges such as fluctuations, or the context of incoming data to each other. At many points in time, it is not clear if new incoming data is connected to already visualized data.

Context-Preservation Several considerations come together for the design space to preserve context. This challenge is primarily related to how is context provided and which methods are used. Following, I list according to my opinion the most important design considerations. Depending on each point the overall context can be significantly improved.

- **Projection Method** refers to how off-screen objects are projected to the viewport. For example, a radial projection may be better than an orthographic projection. I take up this point in Chapter 6 and show that users are more accurate using the orthographic projection strategy in topology-preserving environments.
- **Topology Preservation** refers to the capability of the off-screen visualization technique to maintain the overall topology of objects even when projected back to the viewport. This partially addresses the desert fog problem [118] – the user is aware of empty areas and saves zooming and panning operations. However, the need for topology preservation depends on the task at hand. I showcased that it can also be a trade-off between the readability of a complex data representation and topology preservation.
- **Visual Proxy Design** refers to the appropriate design of visual proxies. Depending on the design, the context may be preserved in a better or worse way. Also, the quality of the topology preservation is reflected by the design.

Interaction is crucial for off-screen visualizations because users are supposed to have full control of the viewport but also of objects located outside the viewport. Besides existing interaction techniques that have been proposed so far [72, 91, 100, 157, 195], there is still a clear need for improvement. Interaction with off-screen located objects requires tailored solutions. Users can adjust the granularity of abstraction in their off-screen visualizations depending on their needs. Not only those, users can be given numerous parameters and specifications of viewports to maximize the value of off-screen visualizations. Then, the question is whether users will benefit from such interaction, if so, how we can support their interaction through automation or feedback. In more detail, I also have numerous questions about how to let users interact with the main viewport and the off-screen viewport at the

same time. Furthermore, there will be challenges of scaling users' interaction between the main viewport and the off-screen viewport and vice versa.

High Dimensional Data Different datasets provide new challenges for off-screen visualization techniques. To the best of my knowledge, there is a lack of techniques taking into account multiple dimensions of the presented data. I presented a starting point: the visualization of uncertainty data, which integrates two dimensions into a glyph representation. Proceeding this idea, I then presented a higher dimensional glyph to encode high-dimensional off-screen information. However, this is just a first concept. The main challenges on how to aggregate high-dimensional off-screen data and how to present it remain.

Evaluation Evaluations are task dependent. Most evaluations have so far been carried out for the well-known techniques, namely halos, arrows, and wedges [27, 28, 82]. Within these studies, they have also been partially compared to Overview-and-Detail systems (application of a second viewport). To the best of my knowledge, existing evaluations have only considered up to 124 off-screen objects, which were presented in an aggregated manner [73]. This evaluation was also carried out against the usage of a second viewport. However, I argue that a minified map does not meet the requirements of being scalable to several thousands of off-screen objects. Techniques like Dynamic Insets [72] used bigger datasets, but at the same time applied a degree-of-interest function making the amount of to be presented off-screen objects shrink significantly. Evaluation of off-screen techniques inherently presents a challenge. Almost every presented off-screen technique provides an evaluation. However, we need to ask ourselves: How do you evaluate design decisions that are not comparable to other off-screen techniques? If for example somebody comes up with a new way of visualizing multivariate data, it is not clear to which off-screen technique to compare to. The same applies to off-screen techniques applied to different visualizations than maps or scatterplots. Furthermore, a comparison to focus-plus-context systems seems justified at first sight but remains questionable – focus-plus-context systems primarily are used to distort the image space not taking data characteristics into account.

Application to Mobile Devices Off-screen visualization has mainly been applied in mobile and touch environments. I want to highlight that all methods presented in this chapter are also applicable to mobile environments. However, in particular, the visual analysis of multivariate data is typically carried out on a standard computer or large wall displays, which can be due to all the additional information one needs to make sense of. In the present thesis, I focus on the visualization capabilities in terms of the visual proxy design rather than the interaction concepts. This is why I carried out all use cases and evaluations on standard computer displays, yet being well-aware of the application possibilities on mobile devices that I leave to future work. In particular, an integrated approach can open the design space for the multivariate data analysis on mobile devices, because of the display size limitations.

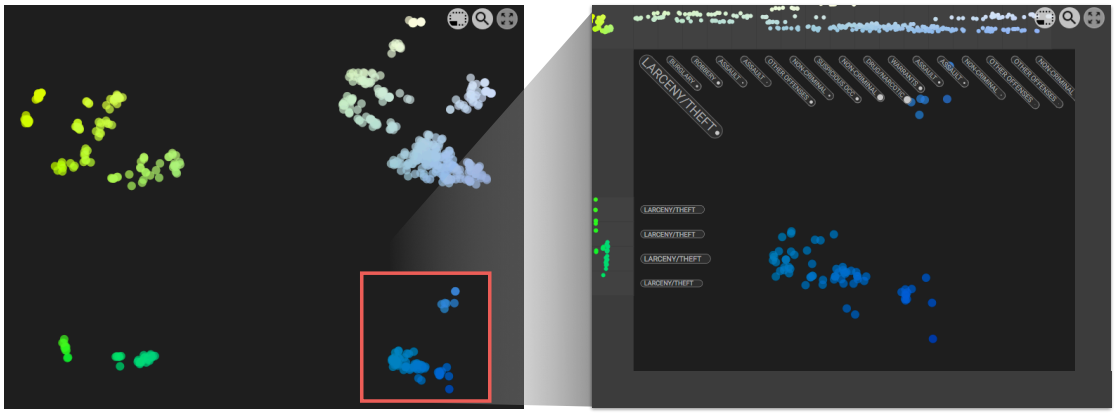


Figure 5.17: *Towards composites of off-screen visualization techniques. Left: Visualization of the crime dataset used in Chapter 3 after the application of MDS. In this case, I want to preserve the overview of crime categories in addition to the topology. Right: After zooming a region of interest, the topology of the data is preserved in the border region, and the categories are displayed as labels that augment the border region.*

Towards Composites of Off-screen Techniques I outlined the trade-off one must accept between topology-preservation and readability of complex visual proxy designs. Because the design space of off-screen visualization is huge, there is still room for improvements to provide effective techniques that enable topology-preservation as well as the integration of complex visual proxies to a full extent. One possible solution to this problem are composites of off-screen techniques, meaning we combine different techniques subject to the tasks and requirements. To showcase a possible solution, consider the San Francisco crime data used in Chapter 3. In this case, we want to preserve the overall topology, as well as the textual information regarding crime categories. So far, the attribute-based information was layouted around an interactive lens; now I want to use them as off-screen visualization technique. The solution is depicted in Figure 5.17. Left: the 2D scatterplot after the application of MDS. Right: I zoom in the lower right part of the scatterplot. The dedicated border region serves to preserve the overall topology (in this example without aggregation). Then, the border is divided into regions (bins), and for each region, a label is visualized. The size of the label corresponds to the occurrences in the bin. The label with the highest frequency is chosen to be visualized. Clicking on a label opens a list of all included labels. Figure 5.17 shows, for example, that in the top left corner region, the crime category *larceny/theft* appears prominent, even more prominent than all other categories across all border regions. The apparent drawback is the additional space that is occupied. This is a restriction which needs to be further investigated in future work.

6

Effects of Mapping Strategy and Intrusion Adaption

Contents

THE general idea of off-screen visualization is to overcome the trade-off between overview and detail by projecting off-screen objects back to the available screen real estate. In Chapter 5, I presented novel methods to visualize off-screen information, but also highlighted the need to further investigate the projection strategy regarding topology-preservation and the users' intuition. Detached visual cues, such as halos or arrows, encode information on position and distance, but fall short showing the topology of off-screen objects. For that reason, I build upon state-of-the-art techniques and integrate visual cues into a dedicated border region. Although the border region adapts to the zooming level, the dimensions of the navigated space are not reflected properly, which is why I propose to adapt the intrusion of the border pursuant to the position in space. I further aim to derive a decision on whether the *radial* or *orthographic* projection strategy should be applied in topology-preserving environments using on a border region. I following describe a controlled experiment to investigate the effect of

the adaptive border intrusion to the topology as well as the users' intuition regarding the projection strategy. The results of the experiment suggest to use the orthographic projection strategy for point data in an adaptive border design. I further discuss the results including the given informal feedback of participants as well as the observations.

This chapter is based on [104]:

Topology-Preserving Off-screen Visualization: Effects of Projection Strategy and Intrusion Adaption D. Jäckle, J.Fuchs, H. Reiterer. *Technical Report, 2017.*

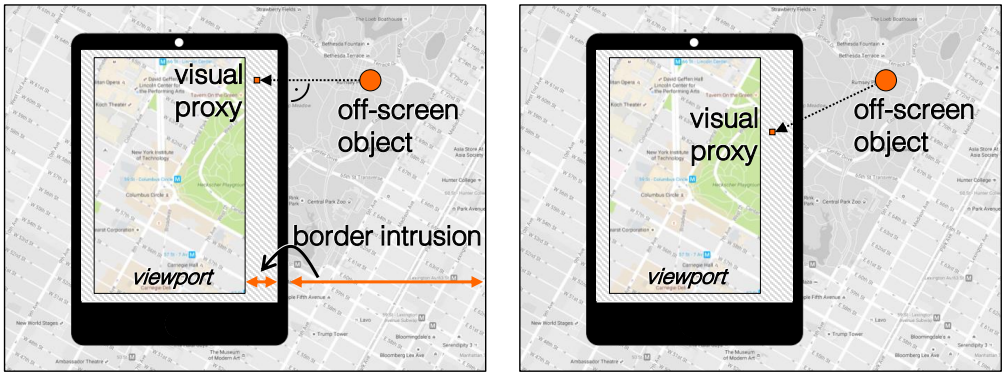
6.1 Introduction

Off-screen visualization techniques have been extensively researched to overcome the trade-off between overview and detail. As described in Chapter 5, they are characterized by the idea of projecting off-screen located objects back to the available screen real estate. Detached visual cues overlay the visible space along the display edge and can encode spatial properties like direction, distance, up to full topology. Techniques, such as EdgeRadar [82], as well as the techniques presented in Chapter 5, encode the topology of off-screen objects in a dedicated border enclosing the visible space. Distances are squeezed proportionally into the border allowing to put off-screen objects in relation with each other efficiently. However, the dimensions of the navigated space are not reflected due to the uniform intrusion of the border on each side of the display.

Another decision to take refers to the projection strategy, which indicates the direction to off-screen objects and enables efficient navigation. Existing techniques choose one out of two strategies to project off-screen objects back to the border: radial or orthographic. Both strategies are illustrated in Figure 6.1. Off-screen objects are projected perpendicular to the viewport using the (a) orthographic projection strategy, whereas the (b) radial strategy projects off-screen objects along a line originating from the center of the viewport. While well-known techniques [12, 81] use orthographic projection, there also exist graph-based [72] techniques that apply the radial projection strategy. The usage of a border region, in particular, opens up the question which projection to use since both strategies are applicable. So far, only few approaches have considered the projection strategy. Despite comprehensive discussions [65, 225] and conducted quantitative evaluations [65, 158], it has yet not been shown which projection strategy to choose for topology-preserving off-screen visualization.

The driving questions of this chapter are: *"How to properly reflect the dimensions of the navigated space?"*, moreover, *"Which projection strategy preserves the data topology, and meets the users' intuition?"* I build upon the border region defined in Chapter 5 and propose to adapt the intrusion of the border pursuant to the dimensions of its adjacent off-screen space, thus improving the awareness of the navigated space. Figure 6.1 depicts an example: the border's intrusion on the right is proportional to the off-screen space on the right. The right border is the widest compared to the top, left, and bottom side. One problem arising is that

off-screen points are also positioned proportionally to the border size, which can be different on each side of the display. This requires first to count back the intrusion of the border to its actual size, namely the adjacent off-screen space, and only then being able to relate the objects to each other correctly.



(a) Orthographic Projection (b) Radial Projection

Figure 6.1: *Off-screen projection strategies integrated into an adaptive intrusion border environment. The (a) orthographic strategy projects off-screen objects along a line perpendicular to the viewport. In contrast, (b) the radial strategy projects off-screen objects along a line towards a point of interest lying inside the viewport, which in this case is the center of the viewport. In addition to Chapter 5, the adaptive border intrusion adapts to the position of the viewport in space. The intrusion of the border is adapted to the considered space in off-screen: The border's dimension on the right is relative to the off-screen space and thus the widest compared to the other sides.*

I conducted a controlled experiment to research the effect of the adaptive border intrusion to the data topology as well as the users' intuition regarding the projection strategy. The experiment consists of three consecutive tasks build upon a comprehensive derivation of the design. The results indicate that an adaptive intrusion does not affect perceived relations of and between off-screen objects and that significantly more users apply the orthographic projection strategy. In preparation for the experiment, I reviewed existing off-screen techniques regarding encoded spatial characteristics, the usage of a border, and the projection strategy. Also, I conducted a pre-study using paper prototyping. The pre-study aimed at confirming that the adaptive border intrusion design is understandable and it is ruled out to harm the experiment. Users were asked to place a viewport made of paper, with an adaptive border attached to it within a specified, delimited space. Participants had no difficulties understanding the adaptive intrusion design.

The following systematic evaluation in consideration of different data characteristics confirms the assumption in Chapter 5, that the orthographic projection preserves data characteristics over the radial projection. However, it is unclear whether a correct data representation also corresponds to the users' intuition, leading to the key contribution of this chapter. Based on the results, I furthermore contribute design considerations regarding topology-preserving off-screen visualization.

Paper/Technique	Direction	Distance	Topology	Border	Projection
City Lights [225]	●	◓			orthographic, radial
Halo [12]	●	●			(orthographic)
Hopping [100]	●	●	◓		orthographic, radial
EdgeRadar [82]	●	●	●	✓	orthographic
Wedge [81]	●	●			orthographic (adapted)
Predictive Jumping [195]	●	●			orthographic
HaloDot [74]	●	◓			(orthographic)
EdgeSplit [91]	●	●	●	✓	orthographic
Off-screen data on tablets [70]	●	●		✓	orthographic, radial
Sparkle [158]	●	●			orthographic
Techniques Chapter 5	●	◓	●	✓	orthographic
Glowworms and Fireflies [168]	●	●			orthographic

Table 6.1: *Overview of surveyed papers that respectively introduce a novel off-screen visualization technique, ordered by year. Columns outline the relevant characteristics of each paper: the encoding of the technique (direction, distance, and topology), whether the visual cues are integrated into a border and the used projection strategy. The ● indicates if a characteristic is fulfilled, the ◓ if partly fulfilled, and the ✓ whether a dedicated border is used.*

6.2 Related Work

As mentioned in Chapter 5, off-screen techniques have been widely designed for navigation in networks or graphs [71, 72, 154, 157]. Connected off-screen objects are projected to the screen real estate using along the line projection for efficient edge routing, which is comparable to the radial projection strategy except that it originates from multiple sources (the graph's nodes) within the viewport. An exception represents the approach by Frisch and Dachselt [65], who discuss the projection strategy for navigating class diagrams. Objects in such diagrams are not connected via the shortest path, demanding a special solution. However, users expected a radial projection. I argue that it is not apparent which projection strategy to implement, explicitly for unconnected spatial objects in a topology-preserving environment. Table 6.1 provides an overview of reviewed papers that respectively introduce a novel off-screen visualization technique based on unconnected objects. In the previous Chapter 5, I introduced the area of off-screen visualization and outlined related work in general. Following, I discuss off-screen visualization, thereby focusing on the encoding (direction, distance, and topology), the usage of a dedicated border region, and the applied projection strategy.

6.2.1 Encoding Direction, Distance, and Topology

Each visual cue representing an off-screen object encodes at least one of the encodings: direction and distance. The direction indicates the route and the distance provides information on how far to pan to reach an off-screen object. While all papers in Table 6.1 encode the direction, not all of them clearly indicate the distance (marked with a ◓). Techniques using aggregation [74, 107, 108] provide an approximation of the distance because aggregated points are visualized via a representative. This is also held true for the techniques presented in Chapter 5. The topology builds upon direction and distance and adds information on

adjacent objects; the topology should show the object of interest as well as the spatial relations to other objects, either nearby or far off. Techniques like Bring&Go (graph-based) [157] or Hopping [100] therefore use a radar-like representation and project off-screen located objects to the viewport in a radial manner. Other techniques such as EdgeRadar [82] and the ones presented in the previous Chapter 5 use a border region to compress the topological information of the respective adjacent off-screen space.

6.2.2 Dedicated Border Region

According to Table 6.1, the visualization of topology goes hand in hand with the application of a dedicated border region. In the first place, the pioneering work of Apperley et al. [9] – the bifocal display technique – inspired the usage of a dedicated border region. The surrounding is distorted while the focus region is maximized. Applied to off-screen visualization, the off-screen space is compressed into the adjacent border, but in data-space and not in image-space. Due to the uniform border size at each side of the display, present off-screen techniques (compare to [82, 91] and the techniques presented in Chapter 5) fall short to reflect the off-screen space dimensions. As a result, one loses awareness of the position in space, particularly where the focus was set to. For example, if one focuses the very left side of the data space, the border

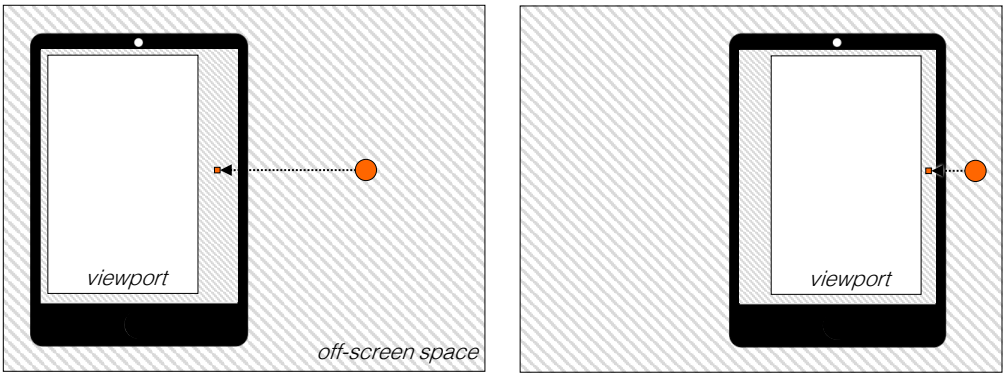


Figure 6.2: *The two settings illustrate the effect of focusing different parts of the data space. Left: The focus is placed to the very left in the data space. Right: The focus is set to the right in the data space. In each case, an off-screen point is positioned in the middle of the adjacent off-screen space. Depending on the focus position, the border adapts to the data space bounds. Independent of how big the off-screen space is, the centered off-screen point is projected to the respective middle of the border.*

should indicate that only little data space remains on the left and that there is a bigger space to navigate on the right. To the best of my knowledge, the unawareness of the dimensions of the navigated space has yet not been addressed in the context of off-screen visualization. The solution I propose is to adapt the border intrusion on each side of the display individually. Figure 6.2 illustrates this concept: The image on the left shows the border intrusions in case the focus is positioned on the very left in data space. The right border intrusion is larger compared to the left border intrusion because more space to navigate remains on the right-hand side.

6.2.3 Projection Strategy

The projection strategy is one of the most important design decisions to make; it determines the perceived direction and therefore affects the navigation route. CityLights [225] (not topology-preserving) discusses the projections, but yet no evaluation, hence, no evidence is given for choosing one projection strategy over the other in topology-preserving environments. Müller et al. [158] evaluated the projections for a non-topology-preserving environment using LEDs, which is why results are not applicable. In this chapter, I aim to present empirical evidence for the right projection choice in topology-preserving off-screen environments. [1]

6.3 Design Space

Based on identified encodings of off-screen visualization techniques in Table 6.1, I discuss the design space of the experiment. This includes the visual abstraction of off-screen objects, the calculation of the border intrusion, and the projection. An adaptive border intrusion provides awareness of the data space dimensions and, at the same time, preserves the topology of off-screen objects. So far, no evidence has been given which projection strategy meets the users' intuition in a topology-preserving environment.

6.3.1 Visual Abstraction

According to Cockburn et al. [44], off-screen visualization modifies the representation of off-screen objects. However, to indicate the presence of off-screen objects the visualization does not have to appear necessarily on-screen, as it is the case for techniques based on ambient light [158, 168]. Apart from that, we can distinguish between three classes of modifications: encoding of direction, distance, and data-specific characteristics. Various techniques exceed the encoding of direction via arrows and also encode distance, as depicted in Table 6.1. Also, few techniques encode data-specific characteristics, such as a unique colored reference [158, 168], the amount or density of clusters [81], among others.

Beyond that, topology-preserving techniques also reveal relations of off-screen objects to each other using direction and distance within a border. The representation of objects within the border has yet not exceeded simple rectangles [82] or grid-based representations as in HaloDot or presented in Chapter 5. For this experiment, I use rectangles and apply a simple color coding to be able to distinguish between objects, but without any meaning attached.

6.3.2 Adaptive Border Intrusion

The border is placed along the display edge surrounding the viewport. To reflect the dimensions of the navigated space, the size of the border needs to adapt to the adjacent off-screen

[1] In Chapter 5, I applied the orthographic projection strategy based on the results of this study. Note, that the projection strategy does not contribute to the basic understanding of how off-screen visualization operate in general and also does not influence the presented use cases. With regard to the user study described in Chapter 5.7, HaloDot also applies the orthographic projection strategy, thus, not adversely affecting the results of the conducted study.

space. This means, the border grows or shrinks proportionally in a linear manner to the off-screen space on the respective side. Furthermore, the higher the zoom level is, the more space needs to be provided as it is the case in multi-scale interfaces [69]. A solution to adapt to the zooming level is discussed and provided in Chapter 5. In this case, all four sides are computed separately giving an impression of the current position in the navigated space. I dynamically calculate the intrusion of each side of the border while zooming and panning as follows:

$$size_x = \alpha \cdot \underbrace{\frac{zoom}{maxZoom}}_{1:\ zoom} \cdot \underbrace{min\left(1.0, \frac{d(vp, d\_bounds_x)}{d\_dimension_x}\right)}_{2:\ position\ in\ data\ space}$$

The dimensions of the border depend on two factors: (1) the zoom level, and (2) the distance to the outer bounds of the navigated space. α determines the maximum possible size of the border, which I set to $\alpha = 35$ pixels pursuant to EdgeRadar [82] (I take up the impact of the border size in the discussion section). α is multiplied by the first factor, the zoom level: the bigger the zoom level, the more space is assigned to the border region. For the experiment, I set this factor to 1.0, because I do not want the zoom level to affect the decisions of how off-screen objects relate to each other. The second factor determines the size of the border region according to the viewport position in the navigated space and is calculated for each side of the viewport vp separately, which is why $x \in \{top, left, bottom, right\}$. The restriction to 1.0 ensures that the size of the border does not increase if the navigated space is located entirely outside the viewport. Otherwise, the second factor is computed by dividing the distance of the viewport vp to the outer bounds of the navigated space with the viewport dimension. For top and bottom, the $d\_dimension$ corresponds to the height of the viewport, the width otherwise.

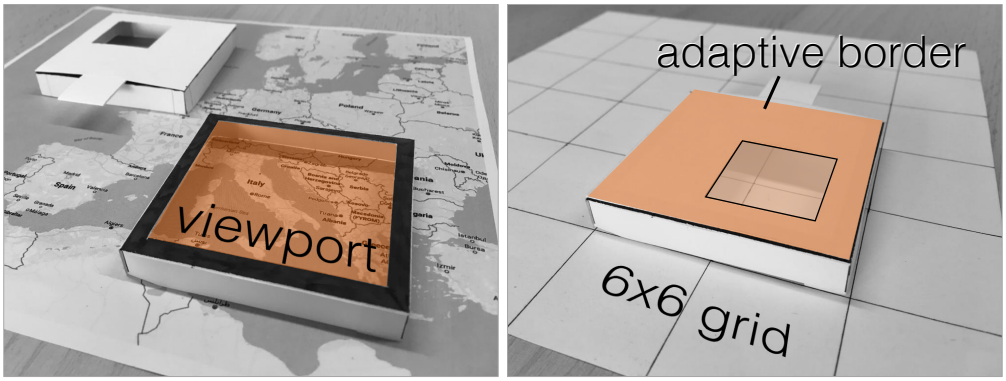


Figure 6.3: *Paper prototype. Any position of the viewport on the grid can be simulated using adaptive border cutouts.*

Paper Prototyping

I carried out a preliminary study to investigate if the adaptive intrusion design of the border is well-received and does not mislead in any manner. Therefore, I created a paper prototype. I prepared a rectangular surface that consists of a grid of 36 equal sized cells (6x6). For

the representation of the viewport, I built an elevated rectangle made of paper and cut out the inner area. The viewport covers four cells and can be freely placed on the grid surface similar to peephole interfaces. Additionally, I set up five different border regions made of paper, which can be placed on the viewport. They are designed proportionally correct to the surface dimensions and allow to simulate any position of the viewport on the grid by rotating the border region: One for the center position. One for all positions where the viewport is moved horizontally or vertically by one cell. One for moving the viewport two cells, either horizontally or vertically. One for moving the viewport diagonally by one cell. One for moving the viewport diagonally by two cells.

I recruited 5 participants (1 female) with normal or corrected-to-normal vision and did not report on color blindness. I tested with two elderly persons (69 and 73 years old) and 3 participants of age 25, 28, and 30. Among all participants, only one participant has worked with visualization as well as off-screen visualization before.

The setup is depicted in Figure 6.3: I first introduced participants to the setup using a map. I placed the viewport on the map and demonstrated in one scenario how the intrusion of the border changes when navigating to the top left corner region. Then, I presented sequentially and in arbitrary order five different scenarios with no scenario being redundant. Participants were asked to place the viewport on the grid for each scenario separately. I recorded all actions on video.

I measured the positioning error of the viewport in half and full grid cells. In total, participants placed the viewport with an average error of 0.12 grid cells. Two out of the five participants did not perform any error. The two oldest participants mentioned that participating in this study “gave them a reality check”. They used websites like Google Maps before but did not fully understand the underlying principle. The paper prototype simply explained them how such applications operate. One interesting observation was that none of the participants asked for further explanations, yet well-performing the task. In consequence, I claim that the principle of an adaptive intrusion is easy to understand.

6.3.3 Projecting Off-screen Objects to the Border

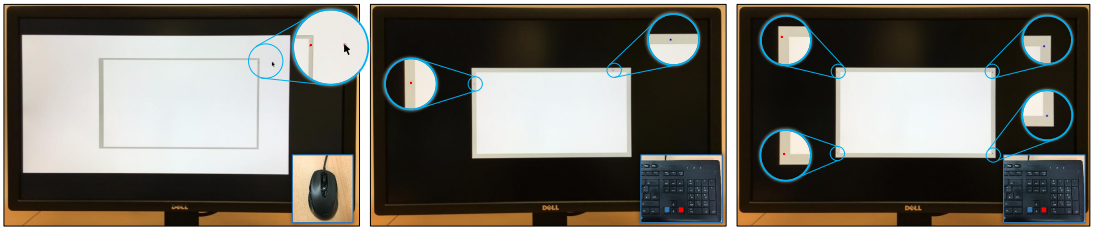
As introduced in the previous Section 6.3.2, the adaptive border intrusion is calculated for each of the sides individually. The two main advantages are the preservation of the overall topology and the awareness of the surrounding data space dimensions.

One unanswered question refers to the choice of the distance encoding. The degree-of-interest function by Furnas [68] proposes to consider nearby objects as more interesting than others. One can argue to adapt the distance function, especially in a topology-preserving environment, to this requirement. For example, a logarithmic-alike function grants nearby off-screen objects more space. While such a function increases the awareness of nearby objects, it interferes with the idea of putting objects into relation to each other plus deciding on the projection strategy. The primary goal of this study is to investigate the effect of the adaptive border and the projection strategy, which is why I choose a linear distance encoding.

6.4 Experiment

We [2] conducted a controlled experiment to investigate the effect of the adaptive intrusion of the border as well as how users perceive off-screen objects, which are projected back to the screen real estate. All tasks of the experiment were carried out using the same apparatus.

For the description of this experiment, the data space is the space to be navigated. Furthermore, off-screen objects are denoted as off-screen points. This is because I aim at investigating perceived spatial information and not any other characteristics, which may be assumed when using the term *object*.



(a) Task 1 – Localization (b) Task 2 – Relation to the viewport (c) Task 3 – Relation between points

Figure 6.4: *Taken pictures of all three tasks. (a) Task 1: one point was positioned in the top right border region, and the user clicks in the pointing hub (slightly shifted to the right) where she assumed the position. (b) Task 2: one red point was placed on the top left, and one blue point was placed on the top right. (c) Task 3: two red points were positioned on the left and two blue points were positioned on the right of the viewport.*

6.4.1 Tasks

I reviewed existing off-screen techniques (see Table 6.1) regarding the evaluation type and tasks. Based on the related work, we derived three consecutive tasks for our experiment: First, we tested how users localize the position of off-screen points in the data space. Second, we tested how users judge relations of off-screen points to the viewport and third, we tested how users judge relations between off-screen points. Following, I describe the three tasks.

Task 1: Localization of Off-screen Points

The first task aimed to investigate the users' intuition regarding the projection strategy. We tested for the orthographic and the radial projection strategy.

Based on the idea of Sparkle [158], the display was divided into two areas: The center area imitated the viewport including the off-screen visualization using a dedicated border. The surrounding area represented the maximum possible data space which did not exceed the display bounds. We successively presented a projected point in the border region and asked participants to click in the off-screen space where they expected the initial position of the data point.

The off-screen area that allows participants to mark the assumed position is called *pointing hub*. All presented cases are outlined in Figure 6.5. First, we distinguish between the fixed and the adaptive border design, depicted left and right, respectively. The fixed border design

[2] Hereinafter, "we" refers to me and Johannes Fuchs, who helped to recruit and carry out the experiment.

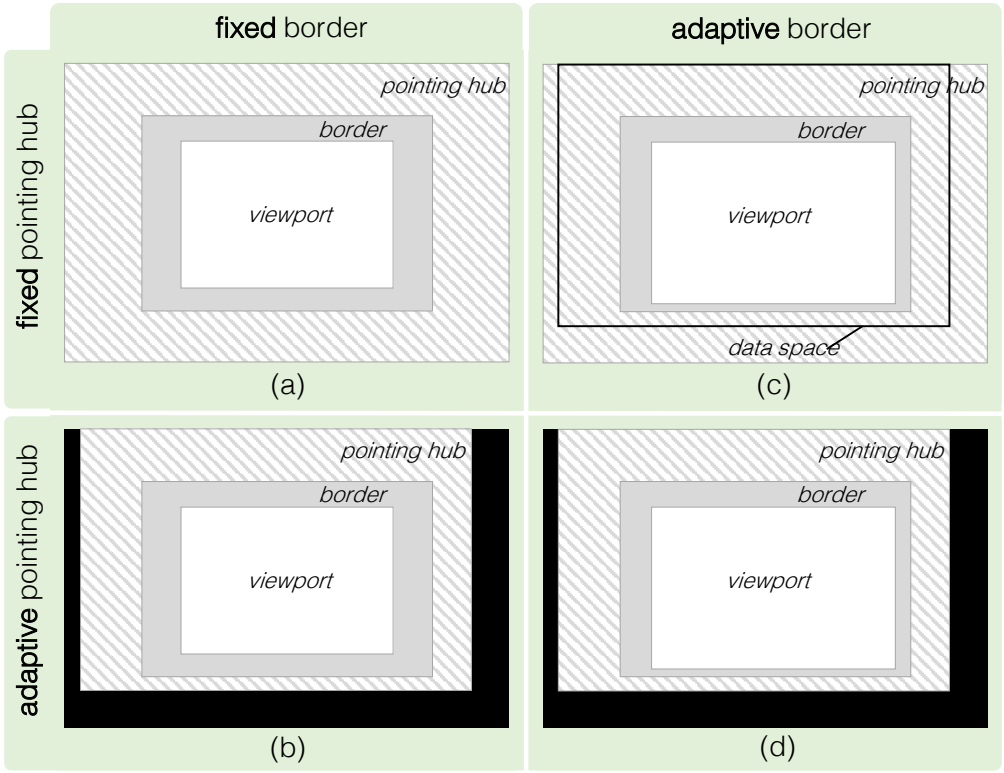


Figure 6.5: *Differentiation of cases for Task 1. Four combinations of fixed border, adaptive border, fixed pointing hub, and adaptive pointing hub were presented to the participant to test the projection strategy and the adaptive border design.*

corresponds to a uniform border that does not adapt to the data space dimensions, whereas the adaptive border design reflects the dimensions. Then, we distinguish between a fixed and an adapted pointing hub. A fixed pointing hub (top row in Figure 6.5) stands for a fully extended area surrounding the off-screen visualization. An adapted hub (bottom row in Figure 6.5) represents a variable extent surrounding the off-screen visualization. Interaction is only allowed within the crosshatched pointing hub. We distinguish between a total of four cases. Figure 6.5 (a) depicts the case of the fixed border design and a full pointing hub extent. Here, the off-screen space dimensions correspond to the pointing hub dimensions. Figure 6.5 (b) shows the same case, but for an adaptive border design. The pointing hub is also fully extended. However, the off-screen space dimensions do not correspond to the pointing hub dimensions; participants were asked to count back the presented adaptive border. The bottom row in Figure 6.5 depicts the cases for an adaptive pointing hub. This means, the pointing hub is adapted to the dimensions of the off-screen space. Figure 6.5 (c) shows the fixed border design, however, the off-screen space dimensions and thus the pointing hub dimensions, are not fully extended. Figure 6.5 (d) shows the same off-screen space design, but the border is adapted to the surrounding off-screen space.

Figure 6.4 (a) depicts the setup and one example: In this example, a red point is placed on the top right, and the participant uses the mouse to click on the assumed position, which

is slightly shifted to the right. The depicted case corresponds to the adaptive border design combined with a variable off-screen space and positioning hub (compare to Figure 6.5 (d)).

Task 2: Relation to the Viewport

In this second task, we intended to investigate the influence of the adaptive border design to relations between off-screen points and the viewport. Therefore, we tested the fixed border design versus the adaptive border design.

We successively presented to participants an off-screen visualization that contained two projected points; one was filled blue and the other one red. Participants were then asked to decide which projected off-screen point was farther away from the viewport. Figure 6.4 (b) depicts the setup and an example: In the example, one red point is placed left and one blue point at the top right. Two keys on the keyboard were marked in the respective colors so that participants could efficiently decide on the distances. In contrast to Task 1, the surrounding area of the off-screen visualization was colored in black. However, the full extent to the display size still corresponded to the possibly full extent of the off-screen space, providing participants an area of reference when counting back the border intrusion as well as the projection.

Task 3: Relation between Points

This task tested relations between points integrated into the adaptive border design. We tested the fixed border design against the adaptive border design.

Successively, two blue and two red projected off-screen points were presented to the participant, who had to decide which points are closer to each other: the blue ones or the red ones. Two keys were marked in the corresponding colors on the keyboard, as depicted in Figure 6.4 (c). The Figure illustrates an example, were two red points are located on the left and two blue points on the right. As described in Task 2, the surrounding area of the off-screen visualization was colored in black, and the full extent to the display size reflected the possibly maximal dimensions of the off-screen space.

6.4.2 Data Generation

In this section, I focus on the generation of data points for the described tasks. Therefore, I first derived cases that show the biggest difference between projection strategies and then generated the data points. I chose cases showing the biggest difference because I aimed at confronting participants with the biggest possible discrepancy between the orthographic and radial projection strategy. This way, I can make a clear statement and provide guidelines which projection strategy to use in a topology-preserving environment. The biggest difference can be defined as the maximum Euclidean distance between projected points, which represent the same off-screen point, but use a different projection strategy. Consider a point p located in off-screen space. The orthographic strategy projects point p to a location p_O and the radial strategy projects the point to a location p_R within the border. The aim was to identify cases, for which the distance between p_O and p_R is maximal. This way, results are interpretable without being biased by the intended projection strategy.

Following, I first derive aforementioned extreme cases and then derive the data generation on a per-task basis.

Derivation of Extreme Cases

A good positioning of points minimizes the randomization of locations. Furthermore, it makes results interpretable, because results are not in favor of a specific projection strategy; this means, the presented projected point does not pretend a specific projection strategy. I derived various areas in off-screen space, which meet the requirement of providing the biggest difference regarding the projection result. I call these: *extreme cases*. To derive the extreme cases, I first focused on the commonalities between both projection strategies.

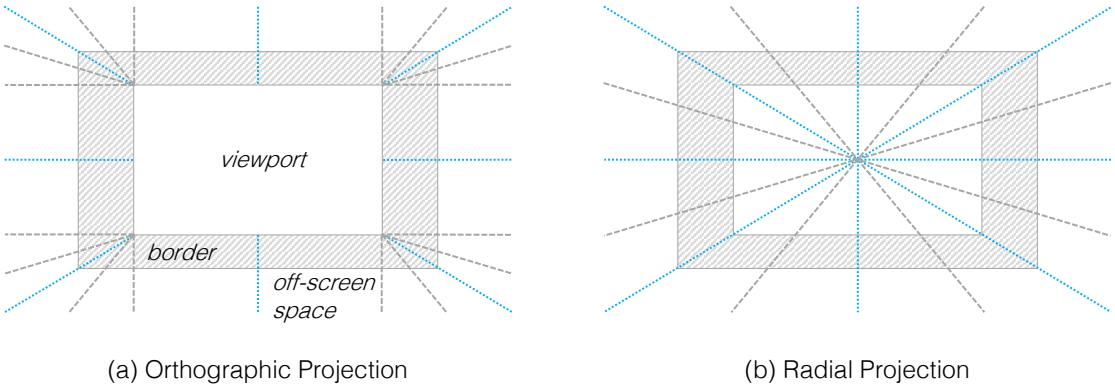


Figure 6.6: *Distinction of projection strategies with respect to identical results. The dotted lines highlight differences and commonalities between projection strategies. (a) shows the orthographic strategy: off-screen points are projected perpendicular to the viewport. In the corner areas, points are projected evenly in x- and y-direction. Off-screen points are projected along the line towards the center of the viewport using the (b) radial projection strategy. The blue colored lines represent the projection lines along which the results of the orthographic and radial projection are identical.*

I sketched both projection strategies in Figure 6.6. The dotted lines illustrate the concepts of how the corresponding projection strategies project off-screen located points back to the screen real estate. On the left-hand side, the (a) orthographic projection strategy is characterized by the uniform projection perpendicular to the viewport, either in x-direction for left/right sides or in y-direction for top/bottom sides. The corner areas call for special treatment, which is an even projection in x- and y-direction similar to the bifocal display technique [9]. This means, off-screen points located in the corner areas are projected, similar to the radial strategy, towards the respective corner of the viewport. For the (b) radial projection strategy, depicted on the right-hand side, off-screen points are projected along the line towards the center of the viewport. The blue colored lines represent the projection lines, along which both projection strategies yield equal results. This is, all off-screen points placed along the blue lines produce the same result when being projected, either orthographically or radially.

For the remainder of this description, I reference to all positions along the blue diagonal lines as function f_d and along the blue median lines as function f_o. The functions f_d and f_o define all points which are identically projected to the border, whether following the

orthographic or the radial projection strategy. I assume that the extreme cases lie in the area enclosed by the functions f_d and f_o.

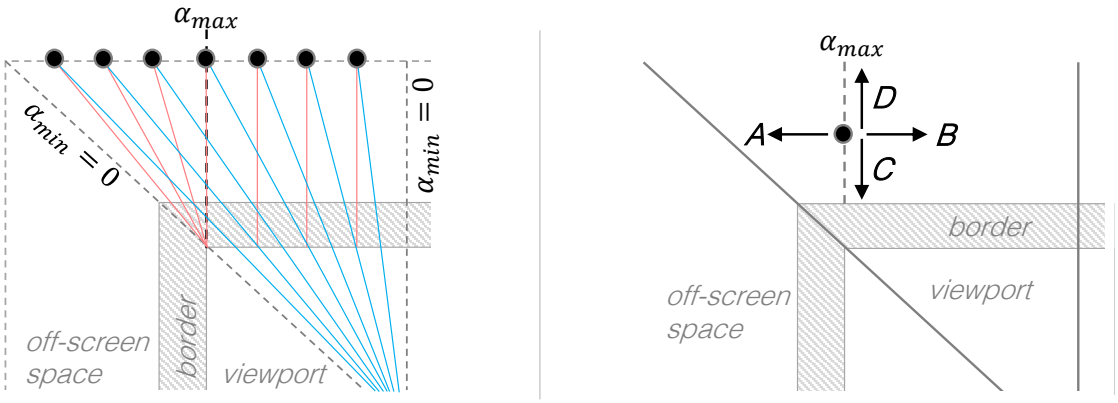


Figure 6.7: *Left: Explanation of the difference between projection strategies by inspecting the angle α. α describes the angle between lines, along which the* ***point*** *(black) is projected, either* ***orthographically*** *(red lines) or* ***radially*** *(blue lines). Right: Degrees of freedom an off-screen point can be moved. α_{max} is defined along the line, which represents the transition between the side and corner area. Along this line, α_{max} determines that the radial and orthographic projection strategy differ the most.*

To inspect the behavior of projections in this area, I investigated the angle α between projection strategies. Consider one point being projected to the border region. The point is projected along a line perpendicular to the viewport using the orthographic strategy. The point is projected along a line towards the center of the viewport using the radial strategy. The angle α is the angle between these two projection lines. Figure 6.7 illustrates an example, based on which we can visually derive the extreme cases. The functions f_d and f_o define all cases, in which projections are identical; i.e. $\alpha = 0$. Figure 6.7 depicts on the left-hand side what happens to α when moving the off-screen point (colored in black) between the median f_o and the diagonal line f_d. For example, when moving from the median towards the diagonal line, α first increases and after having reached its maximum value α_{max}, it decreases again. For a quadratic viewport, α_{max} is reached at the transition between the side and corner area. For a non-quadratic viewport α_{max} moves towards f_d or f_o, respectively.

Based on the data point being placed on the transition between the side and corner area, I distinguished between four cases how α changes when moving the point. Figure 6.7 illustrates these cases on the right-hand side: Moving towards **A** or **B**, α reaches the value 0, as already discussed. Moving the point towards **C**, $\alpha \to \frac{1}{2}\pi$. For **D**, $\alpha \to 0$, which also decreases the projection error. For the sake of simplicity, I assume a quadratic viewport for the data generation, because the shift of α_{max} does not drastically affect the projection result. I call the line, along which α_{max} is placed, in the following: axis. Because of symmetry, there are eight extreme case axes, on which I positioned the data points.

Data for Task 1

I presented the design of the first task in Figure 6.5. For cases (b), (c), and (d), in which the off-screen space is not fully extended, I chose the extent on a per-side basis. One side was

fully extended to provide participants an area of reference. For the remaining three sides, I randomly chose the extent to be within the inner 50% of the possibly full extent. To choose the extent within the inner 50% implies that we do not test for border cases, which refer to special cases such as positioning the data point just right at the transition between off-screen space and viewport, or at the very outer bounds. Therefore, I ensured a fair distribution of points. The allocation of all sides was randomly chosen. The presented data points were positioned along the 8 axes, which were identified as extreme cases.

Data for Task 2

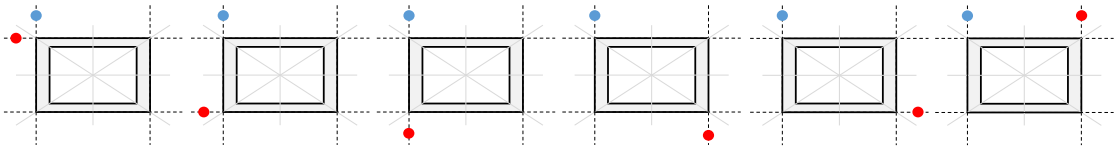


Figure 6.8: *Tested cases of the second task. Due to the requirement that two points shall not be placed on the same axis, and symmetry of cases, six cases remain. Each case can be randomized by means of rotation.*

In the second task, we presented two points at a time. One point was filled blue; the other point was filled red. For the generation of the data space, same as in Task 1, I chose one side to be fully extended as the area of reference. For the remaining three sides, I chose the extent to be within the inner 50% of the possibly full extent. Data points were positioned within the extent of derived axes that describe the extreme cases in data space, respectively. Overall, there exist eight axes (two on each side of the viewport) along one can place the points. In order not to allow comparing distances between points being placed on the very same axes, I only used cases where points were placed on different axes. Also, I considered the symmetry of cases. Figure 6.8 depicts all considered cases. In total, there exist six cases.

Data for Task 3

In the third task, four points were presented to participants at a time. Two points were filled blue; and two points were filled red. For the generation of the data space, same as in Task 1, I chose one side to be fully extended as the area of reference and the remaining sides to be extended within the inner 50% of the possibly full extent. Data points were positioned within the extent of derived axes that describe the extreme cases in data space, respectively. Based on the same requirements of Task 2, that red and blue off-screen points shall not be placed on the same axis and having regard to symmetry; I derived 36 cases. Out of this 36 cases, I randomly picked six cases for each participant but made sure that all cases were evenly covered among all participants.

6.4.3 Hypotheses

Considering the carried out paper prototyping evaluation as well as the systematic evaluation of projection strategies, we derive the following hypotheses on a per task basis:

Localization of Off-screen Points

H1.1: *The orthographic projection will be chosen over the radial projection strategy.* Given related work and our conducted systematic evaluation, the orthographic projection is preferred over the radial projection by participants.

H1.2 *The task performance time will be higher with the fixed border design than with the adaptive border design.* In cases, in which participants need to count back the off-screen space dimensions, participants are expected to be slower.

Relation of Off-screen Points to the Viewport

H2.1: *The adaptive intrusion will not negatively influence the perception of relations of off-screen points.* The task performance accuracy using the adaptive border design will be comparable to the fixed border design. The adaptive border does not provide an enhanced perception of off-screen objects, but an overview of the data space dimensions.

H2.2: *The task performance time will be higher for the fixed border design than for the adaptive border design.* This is due to the additional step of counting back the off-screen space for the adaptive border design.

Relation of Off-screen points to Each Other

H3.1: *The adaptive intrusion will not negatively influence the perception of relations between off-screen points.* The task performance accuracy using the adaptive border design will be comparable to the fixed border design. The adaptive border does not provide an enhanced perception of off-screen objects, as described in **H2.1**.

H3.2: *The task performance time will be higher for the fixed border design than for the adaptive border design.* This is due to the additional step of counting back the off-screen space for the adaptive border design.

6.4.4 Design & Procedure

This study used a within-subject design. The independent variables were for Task 1: border and pointing hub. For Task 2 and Task 3, the independent variable was: border. There were three dependent variables for Task 1: error distance, time, and projection strategy. For Task 2 and Task 3, the dependent variables were accuracy and time. In the following, I discuss the derivation of the data and thus the trials.

Participants We recruited 18 participants (2 female) mainly from the local student population. All participants had normal or corrected to normal vision. The age ranged from 21 to 68 years (Median age 28) with 3 participants reporting previous experience with off-screen visualizations. However, all participants were familiar with basic visualization techniques (i.e. line and pie charts, among others).

Apparatus The studies were conducted using a 27" monitor, one QWERTY keyboard, as well as a corded mouse. The display has a resolution of 2580x1440 pixels and was divided into two areas. The off-screen visualization was positioned in the middle of the display with the dimensions 1920x1080 simulating common 24" screens. The surrounding area represented the maximum possible data space and was communicated as this to participants.

Procedure The experiment was carried out in a quiet room at our university. Each participant was placed in front of the monitor and received an introduction to the topic of off-screen visualization using examples. In order not to prime participants for a specific projection strategy, we did not mention the projection strategy at all and only explained examples which are same for the orthographic as well as radial projection strategy. Also, we provided a comprehensive explanation of the adaptive border intrusion. During the study, the experimenter and the participant were the only persons present.

The tasks were ordered from easy to difficult starting with Task 1, then Task 2, and finally Task 3. Before each task, the experimenter explained the task as well as necessary interactions. For each task, the participant stepped through a short training session using the default projection cases that did not suggest one of the two projection strategies. After each task, the participant had a short break. Following, I provide an overview of the amount of performed trials on a per task basis. As aforementioned, experimental factors were randomized and did not follow any defined order.

For Task 1, I collected the task performance time as well as the distances between the marked position and the retraced orthographically and radially projected case. This means, I can argue if the participants assumed the position of the initial point to be nearer the orthographic or the radial projection.

2	border properties (F, A)	$\times$
2	pointing hub properties (F, A)	$\times$
8	repetitions	$=$
32	trials per participant	$\times$
18	participants	$=$
576	**trials in total for Task 1**	

For Task 2 and 3, I collected the task performance accuracy and time. For all tasks, the overall order of trials was randomized.

2	border properties (F, A)	$\times$
12	repetitions	$=$
24	trials per participant	$\times$
18	participants	$=$
432	**trials**	$\times$
2	tasks (**T2**, **T3**)	$=$
864	**trials in total for Task 2 and Task 3**	

6.5 Results

We report statistically significant results ($p < .05$) from our quantitative analysis and refer to qualitative feedback in the discussion section.

Task 1 - Localization

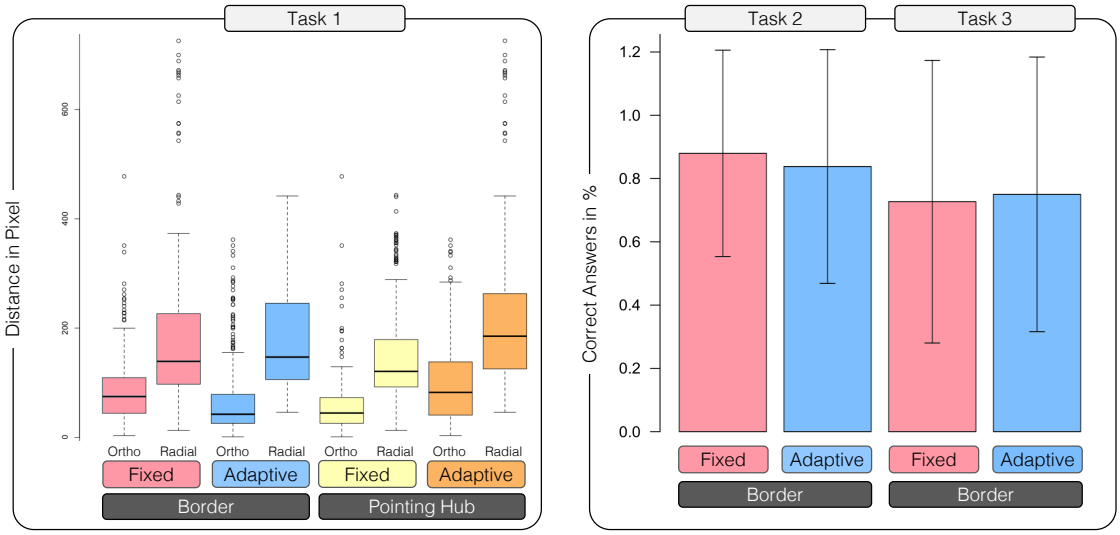


Figure 6.9: *Error Distance: the box plots represent the Euclidean distance (in pixels) between the selected position and the location of both projections (orthographic and radial), respectively. Independent of the design, participants tend to project data points orthographically. Accuracy: Bar charts with mean and standard deviation showing the percentage of correct answers for task 2 and 3.*

Because of the non-normal nature of the data we used a non-parametric Friedman's test to compare the error distance and participants' projection strategy. For analyzing the completion time, we applied a two-way repeated measures ANOVA.

Error Distance

There was an overall significant effect of *projection strategy* on *error distance* ($\chi^2(1, N = 576) = 18, p < .001$).
Post-hoc tests revealed that error distance was significantly lower for the orthographic projection strategy ($78.3px$) compared to the radial projection strategy ($183.4px, p < .001$).

Border: There was an overall significant effect of *border* on *error distance* for both the orthographic projection strategy ($\chi^2(1, N = 288) = 5.6, p < .05$) and the radial projection strategy ($\chi^2(1, N = 288) = 18, p < .001$).
Post-hoc tests revealed that error distance was significantly lower for the adaptive border (orthographic: $69.5px$, radial: $177.9px$) compared to the fixed border (orthographic: $87.0px$, radial: $188.9px; p < .05$).

Pointing Hub: There was an overall significant effect of *pointing hub* on *error distance* for both the orthographic projection strategy ($\chi^2(1, N = 288) = 18, p < .001$) and the radial

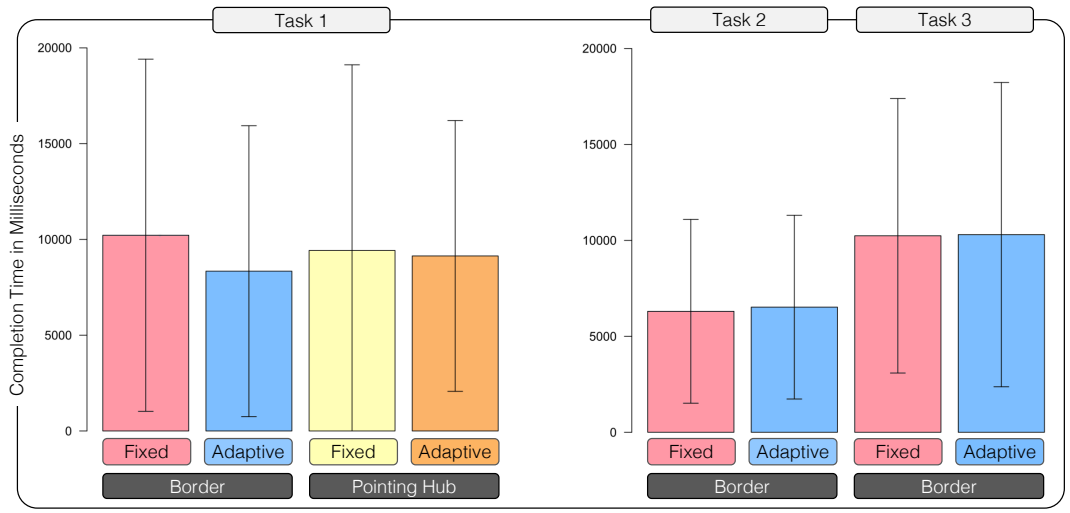


Figure 6.10: *Completion Time: Bar charts with mean and standard deviation showing the completion time in milliseconds for all three tasks. The adaptive border has a positive effect on the completion time for Task 1 and does not perform significantly worse for the other two tasks.*

projection strategy ($\chi^2(1, N = 288) = 18, p < .001$).
Post-hoc tests revealed that error distance was significantly lower for the fixed pointing hub (orthographic: $56.1px$, radial: $152px$) compared to the adapted pointing hub (orthographic: $100.4px$, radial: $214.8px; p < .001$).

Projection Strategy

Figure 6.9 (left) illustrates high-level results. Overall, participants preferred an orthographic projection strategy (94.4%).

Border: There was a significant effect of *border* on *projection strategy* ($\chi^2(1, N = 288) = 7.36, p < .01$).
Post-hoc tests revealed that participants used more often an orthographic projection strategy when working with the adaptive border (98.6%) compared to the fixed border (90.3%, $p < .005$).

Pointing Hub: There was a significant effect of *pointing hub* on *projection strategy* ($\chi^2(1, N = 288) = 4.5, p < .05$).
Post-hoc tests revealed that participants used more often an orthographic projection strategy when working with the adapted pointing hub (96.2%) compared to the fixed pointing hub (92.7%, $p < .05$).

Completion Time

There was an overall effect of *border* on *completion time* ($F_{1,17} = 9.3, p < .01$).
Post-hoc tests revealed that participants were faster when working with the adaptive border ($8.3sec$) compared to the fixed border ($10.2, p < .01$).

Task 2 - Relation to the Viewport

Error Rate

No significant results can be reported. Participants tend to answer equally correct for both the fixed border (88%) and the adaptive border (83.8).

Completion Time

There is no significant effect between *border* and *completion time*. Participants were slightly faster when working with the fixed border (6.3sec) compared to the adaptive border (6.5sec)

Task 3 - Relation Between Points

Error Rate

No significant results can be reported. Participants tend to answer equally correct for both the adaptive border (75%) and the fixed border (72.7).

Completion Time

There is no significant effect between *border* and *completion time*. Participants were slightly faster when working with the fixed border (10.2sec) compared to the adaptive border (10.3sec)

6.6 Discussion & Future Directions

Overall, the adaptive border design was well received as already assumed by the preliminary study using paper prototyping. Furthermore, I can show that even in a topology-preserving off-screen environment users tend to apply the orthographic projection strategy.

Findings & Derived Design Considerations

Following, I discuss the result of our experiment as well as the individual feedback and derive Design Considerations (DC).

Projection Strategy The results indicate that participants selected an orthographic projection strategy significantly more often confirming H1.1. The error distance was significantly lower for the orthographic projection strategy than for the radial projection strategy. Additionally, 17 out of 18 participants reported that the orthographic projection strategy seems more intuitive. In consideration of related work (see Table 6.1), I can give evidence that the orthographic strategy is preferred over the radial strategy for topology-preserving off-screen visualization. This finding is of particular interest compared to state-of-the-art off-screen techniques, which explicitly do not apply a border, because points that are integrated into a border do not present a certain direction. One example represents Wedge [81], which encodes the direction already within the visual cue.

Further intriguing feedback was collected from one participant, who argued that even though the orthographic projection strategy seems intuitive, he would have used the radial projection strategy for implementation. This means the participant finds the radial strategy correct from an implementation point of view. We can learn from his feedback to first focus on the consumers and their needs before creating a solution.

DC1: *The orthographic projection strategy is the best choice for topology-preserving off-screen visualizations of unconnected point data.*

Adaptive vs. Fixed Border The results indicate that the adaptive border design does not negatively influence the perception of relations of and between off-screen points, confirming H2.1 and H3.1. Furthermore, the results of Task 1 indicate that participants were significantly more accurate counting back the position of projected off-screen points using the adaptive border design. However, participants were not significantly faster with the fixed border design, thus rejecting H1.2, H2.2, and H3.2. In other words, the task performance time was not significantly influenced by a different, more demanding border design.

Based on the preliminary study using paper prototyping, we expected participants to adopt the adaptive border design efficiently. We were surprised by how well participants adopted the technique, which is also reflected by the measured task performance time. In summary, there is no clear disadvantage of the adaptive border design, but the advantage of increased awareness of the data space dimensions. The additional step of first counting back the off-screen space and only then being able to decide on the position of off-screen points seems effortless.

The adaptive border design preserves the surrounding data space similar to focus-plus-context systems. One well-known focus-plus-context technique is the distortion lens: It allows the user to pick a specific part of the overall information landscape and then to magnify this part without losing the context of the surrounding. Besides its merits, it maximizes the focus region in image space making judgments based on the data difficult, because the data representation is distorted together with the surrounding. I argue to use off-screen visualization with an adaptive border design if the data is paramount.

DC2: *Use the adaptive border design for increased awareness of the data space.*

Task Dependency The experiment was not based on a certain task. After the study, 5 out of 18 participants reported on having difficulties at the beginning of the study to decide on a projection strategy. This is because we solely presented projections integrated into the border, which could not be associated with a specific projection strategy. However, the encountered difficulties raise the question for task-dependency when considering the projection strategy: Can a projection strategy be associated with a certain task and thus eliminate ambiguities? Consider a navigational task such as using the TomTom [3] device in your car. Typically, your

[3] https://www.tomtom.com/

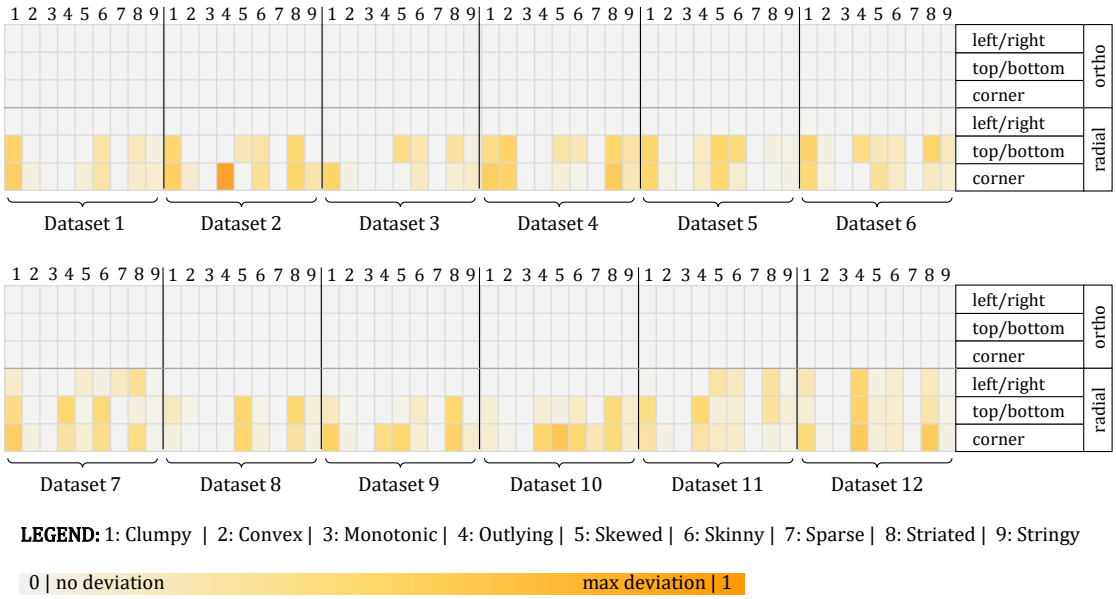


Figure 6.11: *Systematic comparison between the orthographic (ortho) and radial projection strategy. We tested twelve datasets using Scagnostics regarding the distortion of the sides left/right, top/bottom, as well as the corner region in a standard* 16 : 9 *viewport. The table shows the deviation to the undistorted dataset and suggests to use the orthographic projection.*

position marks the point of interest on the map, and a visual cue (normally an arrow) indicates the direction to drive to. A radial projection is used originating from your position, and we all are used to it. However, then, consider a monitoring task such as monitoring hospitals and their capacities in off-screen space due to a natural disaster (compare to Chapter 5). If not indicated otherwise, the hospitals share the same importance level and thus do not directly relate to a point of interest within the viewport. In this case, and because participants of the present study intuitively used this strategy, the orthographic projection makes sense to apply. In summary, the results show that users did intuitively choose the orthographic projection strategy. However, the task at hand can influence the design for the consumer.

Based on the task at hand,
DC3: *apply radial projection for navigational tasks.*
DC4: *apply orthographic projection for monitoring tasks.*

I am aware of the lack of evidence for **DC3** and **DC4**. However, particularly in systems where the direction plays a major role, such as navigation systems or computer games, the radial projection is applied. Despite discussions in related work, such use cases heavily relate to graph representations and along the edge routing. I consider it as natural to apply the radial projection for a point-to-point navigational task. The results yet show a clear preference for the orthographic strategy when no reference point within the viewport is given.

Preservation of Data Characteristics The results show that users favor the orthographic projection strategy for topology-preserving off-screen visualizations. However, compressing off-screen information into the border region is based on the distortion of distances. This raises the question whether the orthographic projection also preserves spatial data characteristics as accurate as possible. I, therefore, conducted a systematic evaluation using Scagnostics (scatterplot diagnostics) by Wilkinson et al. [216] to compare the radial and the orthographic projection. Scagnostics are measures which indicate data-specific characteristics such as outliers (outlying), density (clumpy, skewed, sparse, striated), form (convex, skinny, stringy), and connection of points (monotonic). This approach connects all data points by a minimum spanning tree and applies predefined metrics to it. I conducted a computational study using Scagnostics to get an idea of which projection strategy preserves the relations in the data best. I used similar ground-truth data sets to [216] and moved the data off-screen so that the data was projected to the dedicated border region. For each dataset and Scagnostics measure, I computed the deviation to the undistorted dataset measures. For each projection strategy, I further considered the distortion for the sides left/right, top/bottom, as well as the corner region in a standard $16:9$ viewport. Since all values are normalized using feature scaling, the size of the border does not interfere with the results. Figure 6.11 shows the results. The table reveals that there is no deviation for the orthographic projection. One valid way to interpret this finding is that points are distorted uniformly in x- and y-direction, which is why the Scagnostics measures remain unchanged due to feature scaling. The results of this systematic evaluation back up the results of the user study. In conclusion, the orthographic projection strategy is not only favored by users but also preserves the data characteristics.

Limitations and Future Work

The present study has some limitations, which I aim to cope with in future work. The evaluation aimed at pointing out which projection strategy meets the users' intuition in a topology-preserving off-screen environment, as well as the effect of an adaptive border.

Border Size We conducted the evaluation with a maximum border size similar to EdgeRadar [82]. However, for the adapted border design the intrusion can decrease, raising the question: How big should the initial border size be? I assume that the dimensions of the border should be based on the amount of data being presented as well as the dimensions of the data space. Also, one can consider how much focus the user wishes for. In future work, I plan to investigate these factors regarding the dimensions of the border region. Also, it can be of interest to apply the concept of an adaptive intrusion to other non-topology-preserving techniques such as Halo [12] or Wedge [81], this is, a non-uniform intrusion of visual cues dependent of the navigated space.

Applicability and Scalability The purpose of this study was to investigate which projection strategy meets the users' intuition in a topology-preserving environment and whether the adaptive border impairs the perception. We conducted the study using a data space that is no

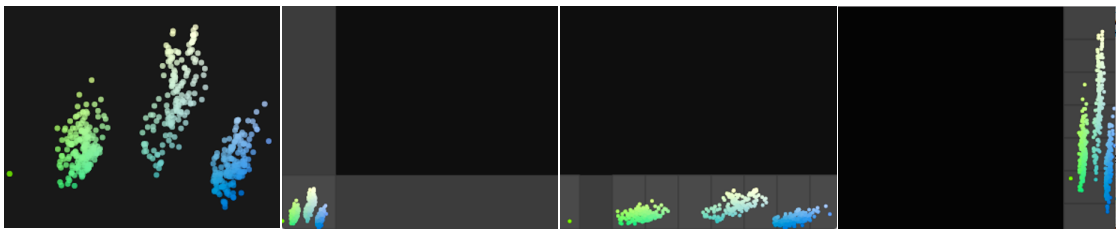


Figure 6.12: *Example of a scatterplot using the adaptive border design and the orthographic projection strategy for approximately 500 data points. (a) Overview of the dataset. (b) In this case, all the data was panned to the off-screen corner area. Even though this area is the smallest compared to the vertical and horizontal extents, we perceive the general structure of the data. (c) The user panned the data into the lower off-screen space. (d) The user panned the data into the off-screen space on the right-hand side.*

more than 1.3x larger than the viewport including the off-screen visualization. The results of the study scale to data spaces larger than 1.3x the viewport: Imagine extending the data space to, for example, 10x larger than the viewport. An object positioned in the center of the off-screen space is still projected back to the center of the border extent, regardless of projection and off-screen space dimensions (illustrated in Figure 6.2). This is traced to the adaptive border that proportionally compresses the adjacent off-screen space and distances between objects. Task 1 tests the converse projection via a pointing hub forcing us to provide a restricted extent (as appeared in [158]).

Another concern arises regarding the limited space in the corner regions when applying the orthographic projection strategy. At this point, I like to emphasize that an investigation of how well off-screen techniques scale to large datasets is not part of the contribution of this study. Despite this, I like to show an example for a scatterplot including around 500 data points. Figure 6.12 illustrates the result of an adaptive border intrusion as well as the orthographic projection applied to the scatterplot. While I agree that the amount of space reserved for the corner regions is limited, the overall structure of the data can still be perceived.

7
Reflection & Conclusion

THE present thesis presented novel methods towards supporting the *identification, interpretation, and navigation of patterns in multivariate data spaces*. In particular, I investigated methods that integrate the user into the automated analysis process using interaction. I showed the usefulness of the methods in various use cases and provided evidence for the effectiveness via different user studies. The four presented core chapters of this thesis are connected as follows: First, Chapter 3 investigated whether users untrained in advanced statistics can interpret the depiction of a multivariate projection. The study revealed that the users adapted effectively to this abstract representation of the data. Furthermore, the study showed that it can be a tedious task to find the correct attribute combinations (subspaces) to identify relevant patterns. Therefore, I proposed *Pattern Trails*, a technique that helps to investigate the structure of patterns among different subspaces. The technique can be applied to any multivariate data and enables the efficient identification of relevant subspaces as well as patterns. Also, Pattern Trails helps to interpret the meaning of certain attribute combinations based on a proposed taxonomy of structural pattern changes.

Patterns not only change structurally among subspaces but in some cases also evolve over time. I presented *Temporal Multidimensional Scaling* (TMDS) in Chapter 4, a technique that processes temporal data window-wise. Each window is projected into one-dimensional space and then aligned one after another on a temporal axis. The result is a two-dimensional plot that makes temporal patterns salient. A fingerprint matrix helps to interpret the patterns, and I enable the user to find similar patterns based on selections. TMDS proofed to be effective in a study together with a domain expert, who analyzed evolving network data on a daily basis.

The commonality between multivariate projections is that they span a possibly large two-dimensional space which is challenging to navigate. In Chapter 5 and 6, I, therefore, investigated *off-screen visualization* methods to preserve the overall data topology and to provide a data-driven overview based on aggregation. My studies showed that a dedicated border region improves the overall awareness of the overview compared to state-of-the-art. Also, the studies revealed that the border region can be adapted to provide awareness of the data space without impairing the overview preservation. Finally, I showed that off-screen techniques, based on a dedicated border region, should implement the *orthographic* over the radial projection strategy because it meets the users' intuition.

Each chapter concluded with a comprehensive, systematic discussion of the presented methods, challenges, and future directions. In this chapter, I moreover aim to discuss the overall picture of this work. Therefore, I following discuss this thesis in the context of the

identification, interpretation, and navigation of patterns in multivariate spaces.

Identification of Multivariate Patterns Exploratory data analysis does not assume the user to have any prior knowledge about the data and thus tries to support the exploration phase as good as possible. It is of utmost importance to provide effective means to identify patterns in multivariate data projections. Pattern Trails and TMDS represent first approaches towards the interactive visual analysis of multivariate patterns that either change among different subspaces or over time. There is certainly need to scale to larger amounts of data than considered in this thesis. However, Pattern Trails proofed to be effective to find relevant patterns and subspaces among 120 unprocessed projections. Nevertheless, real-world data can be much bigger. Furthermore, TMDS was applied to closed datasets, although it already includes the necessary design considerations for the application to streaming data. Streaming data introduces new challenges such as the processing speed as well as the efficient use of the limited display real estate when thousands of new data entries come in.

Interpretation of Multivariate Patterns Real-world data requires the users to have knowledge about the data structure or what patterns to expect in the data. Without any prior knowledge, as it is the case in exploratory data analysis, it takes a lot of time to understand the underlying data structure, and even more time to identify and interpret meaningful patterns. I learned this in my studies and during the creation of many different use cases presented in this thesis. The user is key for the interpretation, but so are the techniques that provide adequate support. For example, Pattern Trails presents methods to draw conclusions regarding the influence of attributes to the structure of a pattern and thus helps to identify and interpret relevant patterns. In contrast, TDMS applies a diversity measure to highlight attributes with high fluctuations in the values and whether these patterns repeat over time. However, there is room for more research regarding the interpretation of patterns in multivariate depictions. The related work section in Chapter 3 outlined the lack of user studies regarding interactive data analysis methods. Most methods were evaluated via application examples or use cases, thus lacking proof whether there is an added value for target users who use such methods.

Navigation of Large Spanned Information Spaces I presented first approaches towards preserving the data topology, and thus the overview of off-screen located objects. So far, the development of overview-preserving techniques was never of concern regarding multivariate data. Recent research aimed at preserving locations of points of interests when navigating geo-spatial maps or graphs rather than multivariate data characteristics and distributions. Multivariate data brings in challenges such as the level of detail that can be provided. For example, the presented star glyph insets highlighted the trade-off between topology-preservation and the readability of the data attribute values. There is a clear need to investigate further the overview preservation in large multivariate information spaces. Also, I investigated whether the orthographic or the radial projection strategy meets the users' intuition. However, there is one problem attached to it: the orthographic strategy showed the smaller overall error. This does not imply that there is no error at all. Therefore, I leave it to future work to derive a new projection strategy based on the errors that are introduced by the users.

List of Figures

List of Tables

Bibliography

[1] AGENCY, N. P. I. Practice advice on analysis. Tech. rep., Association of Chief Police Officers by the National Policing Improvement Agency, 2008.

[2] AIGNER, W., MIKSCH, S., SCHUMANN, H., AND TOMINSKI, C. *Visualization of Time-Oriented Data*. Human-Computer Interaction Series. Springer, 2011.

[3] ALSALLAKH, B., AIGNER, W., MIKSCH, S., AND GRÖLLER, M. E. Reinventing the contingency wheel: Scalable visual analytics of large categorical data. *IEEE Trans. Vis. Comput. Graph. 18*, 12 (2012), 2849–2858.

[4] ANAND, A., WILKINSON, L., AND NHON, D. T. Visual pattern discovery using random projections. In *2012 IEEE Conference on Visual Analytics Science and Technology, VAST 2012, Seattle, WA, USA, October 14-19, 2012* (2012), pp. 43–52.

[5] ANDREAS BUJA, DIANNE COOK, D. F. S. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics 5*, 1 (1996), 78–99.

[6] ANDREWS, D. F. Plots of high-dimensional data. *Biometrics 28*, 1 (1972), 125–136.

[7] ANKERST, M., KEIM, D. A., AND PETER KRIEGEL, H. Circle segments: A technique for visually exploring large multidimensional data sets. In *Visualization* (1996).

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

[9] APPERLEY, M. D., TZAVARAS, I., AND SPENCE, R. A bifocal display technique for data presentation. In *Proceedings of Eurographics* (1982), vol. 82, pp. 27–43.

[10] APPERLEY, M. D., TZAVARAS, I., AND SPENCE, R. A bifocal display technique for data presentation. In *Eurographics Conference Proceedings* (1982), D. Greenaway and E. Warman, Eds., The Eurographics Association.

[11] BACH, B., DRAGICEVIC, P., ARCHAMBAULT, D., HURTER, C., AND CARPENDALE, S. A Review of Temporal Data Visualizations Based on Space-Time Cube Operations. In *EuroVis - STARs* (2014), R. Borgo, R. Maciejewski, and I. Viola, Eds., The Eurographics Association.

[12] BAUDISCH, P., AND ROSENHOLTZ, R. Halo: a technique for visualizing off-screen objects. In *Proceedings of the 2003 Conference on Human Factors in Computing Systems, CHI 2003, Ft. Lauderdale, Florida, USA, April 5-10, 2003* (2003), pp. 481–488.

[13] BAUMGARTNER, C., PLANT, C., KAILING, K., KRIEGEL, H., AND KRÖGER, P. Subspace selection for clustering high-dimensional data. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK* (2004), IEEE Computer Society, pp. 11–18.

[14] BEARD, D. V., AND II, J. Q. W. Navigational techniques to improve the display of large two-dimensional spaces. *Behaviour & Information Technology 9*, 6 (1990), 451–466.

[15] BEDERSON, B. B., CLAMAGE, A., CZERWINSKI, M. P., AND ROBERTSON, G. G. DateLens: A fisheye calendar interface for PDAs. *ACM Trans. Comput.-Hum. Interact. 11*, 1 (Mar. 2004), 90–119.

[16] BEDERSON, B. B., AND HOLLAN, J. D. Pad++: a zoomable graphical interface system. In *Human Factors in Computing Systems, CHI '95 Conference Companion: Mosaic of Creativity, Denver, Colorado, USA, May 7-11, 1995.* (1995), J. Miller, I. R. Katz, R. L. Mack, and L. Marks, Eds., ACM, pp. 23–24.

[17] BELLMAN, R. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[18] BENZÉCRI, J. *L'analyse des données: L'analyse des correspondances*. L'analyse des données: leçons sur l'analyse factorielle et la reconnaissance des formes et travaux. Dunod, 1973.

[19] BERNARD, J., STEIGER, M., WIDMER, S., LÜCKE-TIEKE, H., MAY, T., AND KOHLHAMMER, J. Visual-interactive exploration of interesting multivariate relations in mixed research data sets. *Comput. Graph. Forum 33*, 3 (2014), 291–300.

[20] BERNARD, J., WILHELM, N., SCHERER, M., MAY, T., AND SCHRECK, T. Timeseriespaths: Projection-based explorative analysis of multivarate time series data. *Journal of WSCG 20*, 2 (2012), 97–106.

[21] BOREN, T., AND RAMEY, J. Thinking aloud: Reconciling theory and practice. *IEEE transactions on professional communication 43*, 3 (2000), 261–278.

[22] BORGO, R., KEHRER, J., CHUNG, D. H., MAGUIRE, E., LARAMEE, R. S., HAUSER, H., WARD, M., AND CHEN, M. Glyph-based visualization: Foundations, design guidelines, techniques and applications. *Eurographics State of the Art Reports* (2013), 39–63.

[23] BREHMER, M., AND MUNZNER, T. A multi-level typology of abstract visualization tasks. *IEEE Trans. Visualization and Computer Graphics (TVCG) (Proc. InfoVis) 19*, 12 (2013), 2376–2385.

[24] BREHMER, M., SEDLMAIR, M., INGRAM, S., AND MUNZNER, T. Visualizing dimensionally-reduced data: interviews with analysts and a characterization of task sequences. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization, BELIV 2014, Paris, France, November 10, 2014* (2014), pp. 1–8.

[25] BREMM, S., LANDESBERGER, T. V., BERNARD, J., AND SCHRECK, T. Assisted descriptor selection based on visual comparative data analysis. *Computer Graphics Forum (Proc. EuroVis 2011) 30*, 3 (2011), 891–900.

[26] BROWN, E. T., LIU, J., BRODLEY, C. E., AND CHANG, R. Dis-function: Learning distance functions interactively. In *2012 IEEE Conference on Visual Analytics Science and Technology, VAST 2012, Seattle, WA, USA, October 14-19, 2012* (2012), pp. 83–92.

[27] BURIGAT, S., AND CHITTARO, L. Visualizing references to off-screen content on mobile devices: A comparison of arrows, wedge, and overview + detail. *Interacting with Computers 23*, 2 (2011), 156–166.

[28] BURIGAT, S., CHITTARO, L., AND GABRIELLI, S. Visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches. In *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services, Mobile HCI 2006, Helsinki, Finland, September 12-15, 2006* (2006), M. Nieminen and M. Röykkee, Eds., ACM, pp. 239–246.

[29] BURIGAT, S., CHITTARO, L., AND GABRIELLI, S. Visualizing locations of off-screen objects on mobile devices: a comparative of three approaches. In *Proceedings of the 8th Conf. on Human-Computer Interaction with Mobile Devices and Services, Mobile HCI 2006, Helsinki, Finland* (2006), ACM, pp. 239–246.

[30] BÜRING, T., GERKEN, J., AND REITERER, H. User interaction with scatterplots on small screens - A comparative evaluation of geometric-semantic zoom and fisheye distortion. *IEEE Trans. Vis. Comput. Graph. 12*, 5 (2006), 829–836.

[31] BUTSCHER, S., HORNBÆK, K., AND REITERER, H. SpaceFold and PhysicLenses: simultaneous multifocus navigation on touch surfaces. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces* (2014), ACM, pp. 209–216.

[32] CAO, N., GOTZ, D., SUN, J., AND QU, H. DICON: interactive visual analysis of multidimensional clusters. *IEEE Trans. Vis. Comput. Graph. 17*, 12 (2011), 2581–2590.

[33] CARD, S. K., MACKINLAY, J. D., AND SHNEIDERMAN, B. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[34] CARPENDALE, M. S. T., AND MONTAGNESE, C. A framework for unifying presentation space. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2001), UIST '01, ACM, pp. 61–70.

[35] CARPENDALE, S., LIGH, J., AND PATTISON, E. Achieving higher magnification in context. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology, Santa Fe, NM, USA* (New York, NY, USA, 2004), UIST '04, ACM, pp. 71–80.

[36] CHAKRABARTI, S., ESTER, M., FAYYAD, U., GEHRKE, J., HAN, J., MORISHITA, S., PIATETSKY-SHAPIRO, G., AND WANG, W. Data mining curriculum: A proposal (version 1.0). *Intensive Working Group of ACM SIGKDD Curriculum Committee* (2006), 140.

[37] CHAN, W. W.-Y. A survey on multivariate data visualization. *Department of Computer Science and Engineering. Hong Kong University of Science and Technology 8*, 6 (2006), 1–29.

[38] CHAN, Y.-H., CORREA, C., AND MA, K.-L. The Generalized Sensitivity Scatterplot. *IEEE TVCG 19*, 10 (Oct. 2013), 1768–1781.

[39] CHATFIELD, C., AND COLLINS, A. J. *Introduction to Multivariate Analysis*. Springer, 1980.

[40] CHENG, S., AND MUELLER, K. Improving the fidelity of contextual data layouts using a generalized barycentric coordinates framework. In *2015 IEEE Pacific Visualization Symposium, PacificVis 2015, Hangzhou, China, April 14-17, 2015* (2015), pp. 295–302.

[41] CHENG, S., AND MUELLER, K. The data context map: Fusing data and attributes into a unified display. *IEEE Trans. Vis. Comput. Graph. 22*, 1 (2016), 121–130.

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

[43] CLIBURN, D. C., FEDDEMA, J. J., MILLER, J. R., AND SLOCUM, T. A. Design and evaluation of a decision support system in a water balance application. *Computers & Graphics 26*, 6 (2002), 931–949.

[44] COCKBURN, A., KARLSON, A. K., AND BEDERSON, B. B. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv. 41*, 1 (2008), 2:1–2:31.

[45] COHÉ, A., LIUTKUS, B., BAILLY, G., EAGAN, J., AND LECOLINET, E. Schemelens: A content-aware vector-based fisheye technique for navigating large systems diagrams. *IEEE Trans. Vis. Comput. Graph. 22*, 1 (2016), 330–338.

[46] COOK, K. A., AND THOMAS, J. J. Illuminating the path: The research and development agenda for visual analytics. Tech. rep., Pacific Northwest National Laboratory (PNNL), Richland, WA (US), 2005.

[47] COX, T., AND COX, A. *Multidimensional Scaling, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press, 2000.

[48] CRNOVRSANIN, T., MUELDER, C., CORREA, C. D., AND MA, K. Proximity-based visualization of movement trace data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, IEEE VAST 2009, Atlantic City, New Jersey, USA, 11-16 October 2009, part of VisWeek 2009* (2009), IEEE Computer Society, pp. 11–18.

[49] DASGUPTA, A., AND KOSARA, R. Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Trans. Vis. Comput. Graph. 16*, 6 (2010), 1017–1026.

[50] DINIZ, P. S. R., DA SILVE, E. A. B., AND NETTO, S. L. *Digital Signal Processing: System Analysis and Design*. E-Libro. Cambridge University Press, 2010.

[51] DÖRK, M., CARPENDALE, M. S. T., AND WILLIAMSON, C. Visualizing explicit and implicit relations of complex information spaces. *Information Visualization 11*, 1 (2012), 5–21.

[52] DU TOIT, S., STEYN, A., AND STUMPF, R. *Graphical exploratory data analysis*. Springer-Verlag New York, Inc., 1986.

[53] DUDA, R. O., AND HART, P. E. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM 15*, 1 (1972), 11–15.

[54] DWYER, T., AND GALLAGHER, D. R. Visualising changes in fund manager holdings in two and a half-dimensions. *Information Visualization 3*, 4 (2004), 227–244.

[55] ELLIS, G. P., AND DIX, A. J. An explorative analysis of user evaluation studies in information visualisation. In *Proceedings of the 2006 AVI Workshop on BEyond time and errors: novel evaluation methods for information visualization, BELIV 2006, Venice, Italy, May 23, 2006* (2006), pp. 1–7.

[56] ELMQVIST, N., AND FEKETE, J.-D. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *Visualization and Computer Graphics, IEEE Transactions on 16*, 3 (2010), 439–454.

[57] ELMQVIST, N., HENRY, N., RICHE, Y., AND FEKETE, J.-D. Melange: space folding for multi-focus interaction. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI* (New York, NY, USA, 2008), CHI '08, ACM, pp. 1333–1342.

[58] ENDERT, A., FIAUX, P., AND NORTH, C. Semantic interaction for sensemaking: Inferring analytical reasoning for model steering. *IEEE Trans. Vis. Comput. Graph. 18*, 12 (2012), 2879–2888.

[59] ESTER, M., KRIEGEL, H., SANDER, J., AND XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA* (1996), E. Simoudis, J. Han, and U. M. Fayyad, Eds., AAAI Press, pp. 226–231.

[60] FANEA, E., CARPENDALE, M. S. T., AND ISENBERG, T. An interactive 3d integration of parallel coordinates and star glyphs. In *IEEE Symposium on Information Visualization (InfoVis 2005), 23-25 October 2005, Minneapolis, MN, USA* (2005), pp. 149–156.

[61] FAYYAD, U. M., PIATETSKY-SHAPIRO, G., AND SMYTH, P. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*. 1996, pp. 1–34.

[62] FERNSTAD, S. J., SHAW, J., AND JOHANSSON, J. Quality-based guidance for exploratory dimensionality reduction. *Information Visualization 12*, 1 (2013), 44–64.

[63] FISCHER, F., MANSMANN, F., KEIM, D. A., PIETZKO, S., AND WALDVOGEL, M. Large-Scale Network Monitoring for Visual Analysis of Attacks. In *Visualization for Computer Security*, J. R. Goodall, G. Conti, and K.-L. Ma, Eds., no. 5210 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 111–118.

[64] FRISCH, M., AND DACHSELT, R. Off-screen visualization techniques for class diagrams. In *Proceedings of the ACM 2010 Symposium on Software Visualization, Salt Lake City, UT, USA, October 25-26, 2010* (2010), pp. 163–172.

[65] FRISCH, M., AND DACHSELT, R. Visualizing offscreen elements of node-link diagrams. *Information Visualization 12*, 2 (2013), 133–162.

[66] FUCHS, J., FISCHER, F., MANSMANN, F., BERTINI, E., AND ISENBERG, P. Evaluation of alternative glyph designs for time series data in a small multiple setting. In *2013 ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '13, Paris, France, April 27 - May 2, 2013* (2013), W. E. Mackay, S. A. Brewster, and S. Bødker, Eds., ACM, pp. 3237–3246.

[67] FUCHS, J., ISENBERG, P., BEZERIANOS, A., FISCHER, F., AND BERTINI, E. The influence of contour on similarity perception of star glyphs. *IEEE Trans. Vis. Comput. Graph. 20*, 12 (2014), 2251–2260.

[68] FURNAS, G. W. Generalized fisheye views. In *Proceedings of the SIGCHI Conf. on Human Factors in Comp. Systems* (New York, NY, USA, 1986), CHI '86, ACM, pp. 16–23.

[69] FURNAS, G. W., AND BEDERSON, B. B. Space-scale diagrams: Understanding multiscale interfaces. In *Human Factors in Computing Systems, CHI '95 Conference Proceedings, Denver, Colorado, USA, May 7-11, 1995.* (1995), I. R. Katz, R. L. Mack, L. Marks, M. B. Rosson, and J. Nielsen, Eds., ACM/Addison-Wesley, pp. 234–241.

[70] GAMES, P. S., AND JOSHI, A. Visualization of off-screen data on tablets using context-providing bar graphs and scatter plots. In *IS&T/SPIE Electronic Imaging* (2013), International Society for Optics and Photonics, pp. 90170D–90170D.

[71] GEYMAYER, T., STEINBERGER, M., LEX, A., STREIT, M., AND SCHMALSTIEG, D. Show me the invisible: visualizing hidden content. In *CHI Conference on Human Factors in Computing Systems, CHI'14, Toronto, ON, Canada - April 26 - May 01, 2014* (2014), pp. 3705–3714.

[72] GHANI, S., RICHE, N. H., AND ELMQVIST, N. Dynamic insets for context-aware graph navigation. *Comput. Graph. Forum 30*, 3 (2011), 861–870.

[73] GONÇALVES, T., AFONSO, A. P., CARMO, M. B., AND DE MATOS, P. P. Evaluation of halodot: Visualization of relevance of off-screen objects with over cluttering prevention on

mobile devices. In *Human-Computer Interaction - INTERACT 2011 - 13th IFIP TC 13 Internat. Conf., Lisbon, Portugal, September 5-9, 2011, Proc., Part IV*. 2011, pp. 300–308.

[74] GONÇALVES, T., AFONSO, A. P., CARMO, M. B., AND PAULO, P. Halodot: Visualization of the relevance of off-screen objects. In *SIACG 2011: V Ibero-American Symposium in Comp. Graph.* (2011), pp. 117–120.

[75] GOODALL, J., LUTTERS, W., RHEINGANS, P., AND KOMLODI, A. Preserving the big picture: visual network traffic analysis with TNV. In *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05)* (oct 2005), pp. 47–54.

[76] GOWER, J. C. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika* (1966), 325–338.

[77] GOWER, J. C. A general coefficient of similarity and some of its properties. *Biometrics 27*, 4 (1971), 857–871.

[78] GRAHAM, R. L., KNUTH, D. E., AND PATASHNIK, O. *Concrete Mathematics: A Foundation for Computer Science*, 2nd ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994.

[79] GRATZL, S., LEX, A., GEHLENBORG, N., PFISTER, H., AND STREIT, M. Lineup: Visual analysis of multi-attribute rankings. *IEEE Trans. Vis. Comput. Graph. 19*, 12 (2013), 2277–2286.

[80] GROENEN, P. J., AND BORG, I. Past, present, and future of multidimensional scaling. *Visualization and Verbalization of Data* (2014), 95–117.

[81] GUSTAFSON, S., BAUDISCH, P., GUTWIN, C., AND IRANI, P. Wedge: clutter-free visualization of off-screen locations. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008, 2008, Florence, Italy, April 5-10, 2008* (2008), pp. 787–796.

[82] GUSTAFSON, S., AND IRANI, P. Comparing visualizations for tracking off-screen moving targets. In *Extended Abstracts Proceedings of the 2007 Conference on Human Factors in Computing Systems, CHI 2007, San Jose, California, USA, April 28 - May 3, 2007* (2007), pp. 2399–2404.

[83] GUTWIN, C. Improving focus targeting in interactive fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2002), CHI '02, ACM, pp. 267–274.

[84] GUYON, I., AND ELISSEEFF, A. An introduction to variable and feature selection. *Journal of Machine Learning Research 3* (2003), 1157–1182.

[85] GUYON, I., AND ELISSEEFF, A. An introduction to feature extraction. In *Feature extraction*. Springer, 2006, pp. 1–25.

[86] HARROWER, M., AND BREWER, C. A. Colorbrewer. org: an online tool for selecting colour schemes for maps. *The Cartographic Journal 40*, 1 (2003), 27–37.

[87] HINNEBURG, A., AGGARWAL, C. C., AND KEIM, D. A. What is the nearest neighbor in high dimensional spaces? In *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt* (2000), pp. 506–515.

[88] HOFFMANN, R., BAUDISCH, P., AND WELD, D. S. Evaluating visual cues for window switching on large screens. pp. 929–938.

[89] HORNBÆK, K., AND HERTZUM, M. The notion of overview in information visualization. *Int. J. Hum.-Comput. Stud. 69*, 7-8 (2011), 509–525.

[90] HORNBÆK, K., BEDERSON, B. B., AND PLAISANT, C. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Trans. Comput.-Hum. Interact. 9*, 4 (Dec. 2002), 362–389.

[91] HOSSAIN, Z., HASAN, K., LIANG, H., AND IRANI, P. Edgesplit: facilitating the selection of off-screen objects. In *Mobile HCI '12, Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services, San Francsico, CA, USA, September 21-24, 2012* (2012), pp. 79–82.

[92] HOWARD, D., AND MACEACHREN, A. M. Interface design for geographic visualization: Tools for representing reliability. *Cartography and Geographic Information Systems 23*, 2 (1996), 59–77.

[93] HU, Y., WU, S., XIA, S., FU, J., AND CHEN, W. Motion track: Visualizing variations of human motion data. In *IEEE Pacific Visualization Symposium PacificVis 2010, Taipei, Taiwan, March 2-5, 2010* (2010), IEEE Computer Society, pp. 153–160.

[94] HUBER, P. J. Projection pursuit. *The Annals of Statistics 13*, 2 (1985), 435–475.

[95] HUND, M., BÖHM, D., STURM, W., SEDLMAIR, M., SCHRECK, T., ULLRICH, T., KEIM, D. A., MAJNARIC, L., AND HOLZINGER, A. Visual analytics for concept exploration in subspaces of patient groups. *Brain Informatics 3*, 4 (2016), 233–247.

[96] INGRAM, S., MUNZNER, T., IRVINE, V., TORY, M., BERGNER, S., AND MÖLLER, T. Dimstiller: Workflows for dimensional analysis and reduction. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2010, Salt Lake City, Utah, USA, 24-29 October 2010, part of VisWeek 2010* (2010), pp. 3–10.

[97] INGRAM, S., MUNZNER, T., AND OLANO, M. Glimmer: Multilevel MDS on the GPU. *IEEE Trans. Vis. Comput. Graph. 15*, 2 (2009), 249–261.

[98] INSELBERG, A. The plane with parallel coordinates. *The Visual Computer 1*, 2 (1985), 69–91.

[99] INSELBERG, A., AND DIMSDALE, B. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings IEEE Visualization '90, San Francisco, California, USA, October 23-26, 1990.* (1990), A. E. Kaufman, Ed., IEEE Computer Society Press, pp. 361–378.

[100] IRANI, P., GUTWIN, C., AND YANG, X. Improving selection of off-screen targets with hopping. In *Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006* (2006), pp. 299–308.

[101] JACCARD, P. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, 1901.

[102] JÄCKLE, D., FISCHER, F., SCHRECK, T., AND KEIM, D. A. Temporal MDS plots for analysis of multivariate data. *IEEE Trans. Vis. Comput. Graph. 22*, 1 (2016), 141–150.

[103] JÄCKLE, D., FUCHS, J., AND KEIM, D. A. Star glyph insets for overview preservation of multivariate data. In *Visualization and Data Analysis 2016, San Francisco, California, USA, February 14-18, 2016* (2016), D. Kao, T. Wischgoll, and S. Zhang, Eds., Ingenta, pp. 1–9.

[104] JÄCKLE, D., FUCHS, J., AND REITERER, H. Topology-preserving off-screen visualization: Effects of projection strategy and intrusion adaption. *Technical Report* (2017).

[105] JÄCKLE, D., HUND, M., BEHRISCH, M., KEIM, D. A., AND SCHRECK, T. Pattern Trails: Visual Analysis of Pattern Transitions in Subspaces. In *IEEE Conference on Visual Analytics Science and Technology (VAST)* (2017).

[106] JÄCKLE, D., KWON, B. C., AND KEIM, D. A. Off-Screen Visualization Perspectives: Tasks and Challenges. In *Symposium on Visualization in Data Science (VDS) at IEEE VIS 2015* (2015).

[107] JÄCKLE, D., SENARATNE, H., BUCHMÜLLER, J., AND KEIM, D. A. Integrated Spatial Uncertainty Visualization using Off-screen Aggregation. In *EuroVis Workshop on Visual Analytics (EuroVA)* (2015), The Eurographics Association.

[108] JÄCKLE, D., STOFFEL, F., KWON, B. C., SACHA, D., STOFFEL, A., AND KEIM, D. A. Ambient Grids: Maintain Context-Awareness via Aggregated Off-Screen Visualization. In *Eurographics Conference on Visualization (EuroVis) - Short Papers* (2015), The Eurographics Association.

[109] JÄCKLE, D., STOFFEL, F., MITTELSTÄDT, S., KEIM, D. A., AND REITERER, H. Interpretation of dimensionally-reduced crime data: A study with untrained domain experts. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: IVAPP, (VISIGRAPP 2017)* (2017), pp. 164–175.

[110] JAVED, W., GHANI, S., AND ELMQVIST, N. Polyzoom: multiscale and multifocus exploration in 2d visual spaces. In *CHI Conference on Human Factors in Computing Systems, CHI '12, Austin, TX, USA - May 05 - 10, 2012* (2012), J. A. Konstan, E. H. Chi, and K. Höök, Eds., ACM, pp. 287–296.

[111] JEONG, D. H., ZIEMKIEWICZ, C., FISHER, B. D., RIBARSKY, W., AND CHANG, R. ipca: An interactive system for pca-based visual analytics. *Comput. Graph. Forum 28*, 3 (2009), 767–774.

[112] JERDING, D. F., AND STASKO, J. T. The information mural: A technique for displaying and navigating large information spaces. *IEEE Trans. Vis. Comput. Graph. 4*, 3 (1998), 257–271.

[113] JOHANSSON, J., FORSELL, C., AND COOPER, M. D. On the usability of three-dimensional display in parallel coordinates: Evaluating the efficiency of identifying two-dimensional relationships. *Information Visualization 13*, 1 (2014), 29–41.

[114] JOHANSSON, S. Visual exploration of categorical and mixed data sets. In *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration, Paris, France, June 28, 2009* (2009), pp. 21–29.

[115] JOHANSSON, S., AND JOHANSSON, J. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE transactions on visualization and computer graphics 15*, 6 (2009), 993–1000.

[116] JOHANSSON, S., AND JOHANSSON, J. Visual analysis of mixed data sets using interactive quantification. *SIGKDD Explorations 11*, 2 (2009), 29–38.

[117] JOLLIFFE, I. *Principal component analysis*. Springer series in statistics. Springer-Verlang, 1986.

[118] JUL, S., AND FURNAS, G. W. Critical zones in desert fog: Aids to multiscale navigation. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology, UIST 1998, San Francisco, CA, USA, November 1-4, 1998* (1998), pp. 97–106.

[119] KADMON, N., AND SHLOMI, E. A polyfocal projection for statistical surfaces. *The Cartographic Journal 15*, 1 (1978), 36–41.

[120] KANDOGAN, E. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of the IEEE Information Visualization Symposium* (2000), vol. 650, Citeseer, p. 22.

[121] KEHRER, J., AND HAUSER, H. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Trans. Vis. Comput. Graph. 19*, 3 (2013), 495–513.

[122] KEIM, D. A., ANKERST, M., AND KRIEGEL, H. Recursive pattern: A technique for visualizing very large amounts of data. In *IEEE Visualization '95, Proceedings, Atlanta, Georgia, USA, October 29 - November 3, 1995*. (1995), IEEE Computer Society Press, pp. 279–286.

[123] KEIM, D. A., HAO, M. C., DAYAL, U., AND HSU, M. Pixel bar charts: a visualization technique for very large multi-attribute data sets? *Information Visualization 1*, 1 (2002), 20–34.

[124] KEIM, D. A., HAO, M. C., DAYAL, U., JANETZKO, H., AND BAK, P. Generalized scatter plots. *Information Visualization 9*, 4 (2010), 301–311.

[125] KEIM, D. A., KOHLHAMMER, J., ELLIS, G. P., AND MANSMANN, F. *Mastering the Information Age - Solving Problems with Visual Analytics*. Eurographics Association, 2010.

[126] KEIM, D. A., MANSMANN, F., SCHNEIDEWIND, J., THOMAS, J., AND ZIEGLER, H. Visual analytics: Scope and challenges. In *Visual Data Mining - Theory, Techniques and Tools for Visual Analytics*, S. J. Simoff, M. H. Böhlen, and A. Mazeika, Eds., vol. 4404 of *Lecture Notes in Computer Science*. Springer, 2008, pp. 76–90.

[127] KINKELDEY, C., MACEACHREN, A. M., AND SCHIEWE, J. How to assess visual communication of uncertainty? a systematic review of geospatial uncertainty visualisation user studies. *The Cartographic Journal 51*, 4 (2014), 372–386.

[128] KLIPPEL, A., HARDISTY, F., LI, R., AND WEAVER, C. Colour-enhanced star plot glyphs: Can salient shape characteristics be overcome? *Cartographica 44*, 3 (2009), 217–231.

[129] KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological cybernetics 43*, 1 (1982), 59–69.

[130] KOSARA, R., BENDIX, F., AND HAUSER, H. Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Trans. Vis. Comput. Graph. 12*, 4 (2006), 558–568.

[131] KRAUSE, J., DASGUPTA, A., FEKETE, J.-D., AND BERTINI, E. SeekAView: an intelligent dimensionality reduction strategy for navigating high-dimensional data spaces. *Large Data Analysis and Visualization (LDAV), IEEE Symposium on* (Oct 2016).

[132] KRAUSE, J., PERER, A., AND BERTINI, E. INFUSE: interactive feature selection for predictive modeling of high dimensional data. *IEEE Trans. Vis. Comput. Graph. 20*, 12 (2014), 1614–1623.

[133] KRIEGEL, H., KRÖGER, P., AND ZIMEK, A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *TKDD 3*, 1 (2009).

[134] KRUSKAL, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (1964), 1–27.

[135] KRUSKAL, J. B. Nonmetric multidimensional scaling: a numerical method. *Psychometrika 29*, 2 (1964), 115–129.

[136] LAWRENCE E. COHEN, M. F. Social change and crime rate trends: A routine activity approach, 1979.

[137] LAZAR, J., FENG, J. H., AND HOCHHEISER, H. *Research methods in human-computer interaction*. John Wiley & Sons, 2010.

[138] LEHMANN, D. J., AND THEISEL, H. General projective maps for multidimensional data projection. *Comput. Graph. Forum 35*, 2 (2016), 443–453.

[139] LEHMANN, D. J., AND THEISEL, H. Optimal sets of projections of high-dimensional data. *IEEE Trans. Vis. Comput. Graph. 22*, 1 (2016), 609–618.

[140] LICHMAN, M. UCI machine learning repository, 2013.

[141] LIU, H., AND MOTODA, H. *Computational methods of feature selection*. CRC Press, 2007.

[142] LIU, S., MALJOVEC, D., WANG, B., BREMER, P., AND PASCUCCI, V. Visualizing high-dimensional data: Advances in the past decade. *IEEE Trans. Vis. Comput. Graph. 23*, 3 (2017), 1249–1268.

[143] LIU, S., WANG, B., BREMER, P., AND PASCUCCI, V. Distortion-guided structure-driven interactive exploration of high-dimensional data. *Comput. Graph. Forum 33*, 3 (2014), 101–110.

[144] LIU, S., WANG, B., THIAGARAJAN, J. J., BREMER, P., AND PASCUCCI, V. Visual exploration of high-dimensional data through subspace analysis and dynamic projections. *Comput. Graph. Forum 34*, 3 (2015), 271–280.

[145] MAATEN, L. V. D., AND HINTON, G. Visualizing data using t-sne. *Journal of Machine Learning Research 9*, Nov (2008), 2579–2605.

[146] MACEACHREN, A. M. Visualizing uncertain information. *Cartographic Perspectives*, 13 (1992), 10–19.

[147] MACEACHREN, A. M., BREWER, C. A., AND PICKLE, L. W. Visualizing georeferenced data: representing reliability of health statistics. *Environm. and Planning A 30*, 9 (1998), 1547–1561.

[148] MACKINLAY, J. D. Automating the design of graphical presentations of relational information. *ACM Trans. Graph. 5*, 2 (1986), 110–141.

[149] MACKINLAY, J. D., ROBERTSON, G. G., AND CARD, S. K. The perspective wall: Detail and context smoothly integrated. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1991), pp. 173–176.

[150] MANLY, B. *Multivariate Statistical Methods: A Primer, Third Edition*. Taylor & Francis, 2004.

[151] MAO, Y., DILLON, J. V., AND LEBANON, G. Sequential document visualization. *IEEE Trans. Vis. Comput. Graph. 13*, 6 (2007), 1208–1215.

[152] MARTIN, J. P., SWAN, J. E., MOORHEAD, R. J., LIU, Z., AND CAI, S. Results of a User Study on 2D Hurricane Visualization. *Computer Graphics Forum 27*, 3 (2008), 991–998.

[153] MATEJKA, J., AND FITZMAURICE, G. Same stats, different graphs: Generating datasets with varied appearance and identical statistics through simulated annealing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 6-11, 2017* (2017), ACM.

[154] MAY, T., STEIGER, M., DAVEY, J., AND KOHLHAMMER, J. Using signposts for navigation in large graphs. *Comput. Graph. Forum 31*, 3 (2012), 985–994.

[155] MCPHERSON, J., MA, K.-L., KRYSTOSK, P., BARTOLETTI, T., AND CHRISTENSEN, M. PortVis: A Tool for Port-based Detection of Security Events. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security* (New York, NY, USA, 2004), VizSEC/DMSEC '04, ACM, pp. 73–81.

[156] MITTELSTÄDT, S., BERNARD, J., SCHRECK, T., STEIGER, M., KOHLHAMMER, J., AND KEIM, D. A. Revisiting Perceptually Optimized Color Mapping for High-Dimensional Data Analysis. In *In Proceedings of the Eurographics Conference on Visualization* (2014), The Eurographics Association, pp. 91–95.

[157] MOSCOVICH, T., CHEVALIER, F., HENRY, N., PIETRIGA, E., AND FEKETE, J. Topology-aware navigation in large networks. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009* (2009), pp. 2319–2328.

[158] MÜLLER, H., LÖCKEN, A., HEUTEN, W., AND BOLL, S. Sparkle: an ambient light display for dynamic off-screen points of interest. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational, Helsinki, Finland, October 26-30, 2014* (2014), pp. 51–60.

[159] MUNZNER, T. *Visualization analysis and design*. CRC Press, 2014.

[160] NAM, J. E., AND MUELLER, K. Tripadvisor$^{\text{n-d}}$: A tourism-inspired high-dimensional space exploration framework with overview and detail. *IEEE Trans. Vis. Comput. Graph. 19*, 2 (2013), 291–305.

[161] NELSON, E., AND GILMARTIN, P. An evaluation of multivariate, quantitative point symbols for maps. *Cartographic design: Theoretical and practical perspectives* (1996), 191–203.

[162] NESSIE. *Classification of whiskies*. IAS programme Complex Networks, 2015. Available at `https://www.mathstat.strath.ac.uk/outreach/nessie/`.

[163] OELKE, D., HAO, M. C., ROHRDANTZ, C., KEIM, D. A., DAYAL, U., HAUG, L., AND JANETZKO, H. Visual opinion analysis of customer feedback data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, IEEE VAST 2009, Atlantic City, New Jersey, USA, 11-16 October 2009, part of VisWeek 2009* (2009), IEEE Computer Society, pp. 187–194.

[164] PANG, A. Visualizing uncertainty in geo-spatial data. In *Proceedings of the Workshop on the Intersections between Geospatial Information and Information Technology* (2001), pp. 1–14.

[165] PARSONS, L., HAQUE, E., AND LIU, H. Subspace clustering for high dimensional data: a review. *SIGKDD Explorations 6*, 1 (2004), 90–105.

[166] PARTRIDGE, G. A., NEZHADASL, M., IRANI, P., AND GUTWIN, C. A comparison of navigation techniques across different types of off-screen navigation tasks. In *Human-Computer Interaction - INTERACT 2007, 11th IFIP TC 13 International Conference, Rio de Janeiro, Brazil, September 10-14, 2007, Proceedings, Part II* (2007), M. C. C. Baranauskas, P. A. Palanque, J. Abascal, and S. D. J. Barbosa, Eds., vol. 4663 of *Lecture Notes in Computer Science*, Springer, pp. 716–721.

[167] PERLIN, K., AND FOX, D. Pad: an alternative approach to the computer interface. SIGGRAPH '93, ACM, pp. 57–64.

[168] PERTENEDER, F., GROSSAUER, E. B., LEONG, J., STUERZLINGER, W., AND HALLER, M. Glowworms and fireflies: Ambient light on large interactive surfaces. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016* (2016), pp. 5849–5861.

[169] PIETRIGA, E., AND APPERT, C. Sigma lenses: focus-context transitions combining space, time and translucence. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), CHI '08, ACM, pp. 1343–1352.

[170] PIETRIGA, E., APPERT, C., AND BEAUDOUIN-LAFON, M. Pointing and beyond: an operationalization and preliminary evaluation of multi-scale searching. In *Proceedings of the 2007 Conference on Human Factors in Computing Systems, CHI 2007, San Jose, California, USA, April 28 - May 3, 2007* (2007), M. B. Rosson and D. J. Gilmore, Eds., ACM, pp. 1215–1224.

[171] PINDAT, C., PIETRIGA, E., CHAPUIS, O., AND PUECH, C. Jellylens: content-aware adaptive lenses. In *The 25th Annual ACM Symposium on User Interface Software and Technology, UIST '12, Cambridge, MA, USA, October 7-10, 2012* (2012), R. Miller, H. Benko, and C. Latulipe, Eds., ACM, pp. 261–270.

[172] PLAISANT, C., CARR, D., AND SHNEIDERMAN, B. Image-browser taxonomy and guidelines for designers. *IEEE Software 12*, 2 (1995), 21–32.

[173] RAO, R., AND CARD, S. K. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Conference Companion on Human Factors in Computing Systems* (New York, NY, USA, 1994), CHI '94, ACM, pp. 222–228.

[174] ROBERTSON, G. G., AND MACKINLAY, J. D. The document lens. In *Proceedings of the Sixth ACM Symposium on User Interface Software and Technology, UIST 1993, Atlanta, GA, USA, November 3-5, 1993* (1993), pp. 101–108.

[175] ROSARIO, G. E., RUNDENSTEINER, E. A., BROWN, D. C., WARD, M. O., AND HUANG, S. Mapping nominal values to numbers for effective visualization. *Information Visualization 3*, 2 (2004), 80–95.

[176] RUBIN, E. *Synsoplevede figurer, studier i psykologisk Analyse. Ite Del*. Gyldendal, 1915.

[177] SACHA, D., STOFFEL, A., STOFFEL, F., KWON, B. C., ELLIS, G. P., AND KEIM, D. A. Knowledge generation model for visual analytics. *IEEE Trans. Vis. Comput. Graph. 20*, 12 (2014), 1604–1613.

[178] SACHA, D., ZHANG, L., SEDLMAIR, M., LEE, J. A., PELTONEN, J., WEISKOPF, D., NORTH, S. C., AND KEIM, D. A. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. Vis. Comput. Graph. 23*, 1 (2017), 241–250.

[179] SARKAR, M., AND BROWN, M. H. Graphical fisheye views of graphs. pp. 83–91.

[180] SCHADE, O. H. Optical and photoelectric analog of the eye. *J. Opt. Soc. Am. 46*, 9 (Sep 1956), 721–738.

[181] SEDLMAIR, M., MUNZNER, T., AND TORY, M. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Trans. Vis. Comput. Graph. 19*, 12 (2013), 2634–2643.

[182] SEO, J., AND SHNEIDERMAN, B. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization 4*, 2 (2005), 96–113.

[183] SHANNON, C. A mathematical theory of communication. *Bell System Technical Journal, The 27*, 3 (July 1948), 379–423.

[184] SHNEIDERMAN, B. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages, Boulder, Colorado, USA, September 3-6, 1996* (1996), IEEE Computer Society, pp. 336–343.

[185] SIEGEL, J. H., FARRELL, E. J., GOLDWYN, R. M., AND FRIEDMAN, H. P. The surgical implications of physiologic patterns in myocardial infarction shock. *Surgery 72*, 1 (1972), 126–141.

[186] SIMPSON, E. H. Measurement of diversity. *Nature* (1949).

[187] SINGHAL, A. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull. 24*, 4 (2001), 35–43.

[188] SPENCE, R. Information visualization-design for interaction. *UK: Pearson Educated Limited* (2007).

[189] STAHNKE, J., DÖRK, M., MÜLLER, B., AND THOM, A. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Trans. Vis. Comput. Graph. 22*, 1 (2016), 629–638.

[190] STASKO, J. T. Value-driven evaluation of visualizations. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization, BELIV 2014, Paris, France, November 10, 2014* (2014), H. Lam, P. Isenberg, T. Isenberg, and M. Sedlmair, Eds., ACM, pp. 46–53.

[191] STEIGER, M., BERNARD, J., MITTELSTÄDT, S., HUTTER, M., KEIM, D., THUM, S., AND KOHLHAMMER, J. Explorative analysis of 2d color maps. In *Proceedings of WSCG* (2015), V. Skala, Ed., vol. 23, Eurographics Assciation, Vaclav Skala - Union Agency, pp. 151–160.

[192] STEVENSON, A. *Oxford Dictionary of English*. Oxford University Press, 2015.

[193] STOFFEL, F., JANETZKO, H., AND MANSMANN, F. Proportions in categorical and geographic data: visualizing the results of political elections. In *International Working Conference on Advanced Visual Interfaces, AVI '12, Capri Island, Naples, Italy, May 22-25, 2012, Proceedings* (2012), pp. 457–464.

[194] SZYMKIEWICZ, D. *Une contribution statistique a la géographie floristique*. Polskie Towarzystwo Botaniczne, 1934.

[195] TAKASHIMA, K., SUBRAMANIAN, S., TSUKITANI, T., KITAMURA, Y., AND KISHINO, F. Acquisition of off-screen object by predictive jumping. In *Computer-Human Interaction, 8th Asia-Pacific Conference, APCHI 2008, Seoul, Korea, July 6-9, 2008, Proceedings* (2008), pp. 301–310.

[196] TATU, A., ALBUQUERQUE, G., EISEMANN, M., BAK, P., THEISEL, H., MAGNOR, M. A., AND KEIM, D. A. Automated analytical methods to support visual exploration of high-dimensional data. *IEEE Trans. Vis. Comput. Graph. 17*, 5 (2011), 584–597.

[197] TATU, A., ALBUQUERQUE, G., EISEMANN, M., SCHNEIDEWIND, J., THEISEL, H., MAGNOR, M. A., AND KEIM, D. A. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, IEEE VAST 2009, Atlantic City, New Jersey, USA, 11-16 October 2009, part of VisWeek 2009* (2009), pp. 59–66.

[198] TATU, A., MAASS, F., FÄRBER, I., BERTINI, E., SCHRECK, T., SEIDL, T., AND KEIM, D. A. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *2012 IEEE Conference on Visual Analytics Science and Technology, VAST 2012, Seattle, WA, USA, October 14-19, 2012* (2012), IEEE Computer Society, pp. 63–72.

[199] TENENBAUM, J. B., DE SILVA, V., AND LANGFORD, J. C. A global geometric framework for nonlinear dimensionality reduction. *science 290*, 5500 (2000), 2319–2323.

[200] TOMINSKI, C., GLADISCH, S., KISTER, U., DACHSELT, R., AND SCHUMANN, H. A Survey on Interactive Lenses in Visualization. In *EuroVis State-of-the-Art Reports* (2014), Eurographics Association, pp. 43–62.

[201] TORGERSON, W. S. Multidimensional scaling: I. theory and method. *Psychometrika 17*, 4 (1952), 401–419.

[202] TORGERSON, W. S. Theory and methods of scaling.

[203] TUFTE, E. R. Envisioning information. *Optometry & Vision Science 68*, 4 (1991), 322–324.

[204] TUKEY, J. W. *Exploratory data analysis*. Reading, Mass., 1977.

[205] TURKAY, C., FILZMOSER, P., AND HAUSER, H. Brushing dimensions - A dual visual analysis model for high-dimensional data. *IEEE Trans. Vis. Comput. Graph. 17*, 12 (2011), 2591–2599.

[206] TURKAY, C., KAYA, E., BALCISOY, S., AND HAUSER, H. Designing progressive and interactive analytics processes for high-dimensional data analysis. *IEEE transactions on visualization and computer graphics* (2016).

[207] TURKAY, C., LUNDERVOLD, A., LUNDERVOLD, A. J., AND HAUSER, H. Representative factor generation for the interactive visual analysis of high-dimensional data. *IEEE Trans. Vis. Comput. Graph. 18*, 12 (2012), 2621–2630.

[208] VAN WIJK, J. J. The value of visualization. In *16th IEEE Visualization Conference, VIS 2005, Minneapolis, MN, USA, October 23-28, 2005* (2005), IEEE Computer Society, pp. 79–86.

[209] VAN WIJK, J. J., AND NUIJ, W. A. A. Smooth and efficient zooming and panning. In *9th IEEE Symposium on Information Visualization (InfoVis 2003), 20-21 October 2003, Seattle, WA, USA* (2003), IEEE Computer Society, pp. 15–23.

[210] VIÉGAS, F. B., WATTENBERG, M., AND FEINBERG, J. Participatory visualization with wordle. *IEEE Trans. Vis. Comput. Graph. 15*, 6 (2009), 1137–1144.

[211] WANG, Y.-S., AND CHI, M.-T. Focus+ context metro maps. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2528–2535.

[212] WARD, M. O. Multivariate data glyphs: Principles and practice. In *Handbook of data visualization*. Springer, 2008, pp. 179–198.

[213] WARD, M. O., AND GUO, Z. Visual exploration of time-series data with shape space projections. *Comput. Graph. Forum 30*, 3 (2011), 701–710.

[214] WARD, M. O., AND MARTIN, A. R. High dimensional brushing for interactive exploration of multivariate data. In *IEEE Visualization* (1995), p. 271.

[215] WARE, C., AND LEWIS, M. The dragmag image magnifier. In *Human Factors in Computing Systems, CHI '95 Conference Companion: Mosaic of Creativity, Denver, Colorado, USA, May 7-11, 1995.* (1995), J. Miller, I. R. Katz, R. L. Mack, and L. Marks, Eds., ACM, pp. 407–408.

[216] WILKINSON, L., ANAND, A., AND GROSSMAN, R. L. Graph-theoretic scagnostics. In *IEEE Symposium on Information Visualization (InfoVis 2005), 23-25 October 2005, Minneapolis, MN, USA* (2005), p. 21.

[217] WOOD, J., DYKES, J., SLINGSBY, A., AND CLARKE, K. Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup. *IEEE Trans. Vis. Comput. Graph. 13*, 6 (2007), 1176–1183.

[218] YANG, T., LIU, J., MCMILLAN, L., AND WANG, W. A fast approximation to multidimensional scaling. In *IEEE workshop on Computation Intensive Methods for Computer Vision* (2006).

[219] YI, J. S., MELTON, R., STASKO, J. T., AND JACKO, J. A. Dust & magnet: multivariate information visualization using a magnet metaphor. *Information Visualization 4*, 3 (2005), 239–256.

[220] YU, B., LIU, R., AND YUAN, X. Mlmd: Multi-layered visualization for multi-dimensional data. *The Eurographics Association 5* (2013), 103–107.

[221] YUAN, X., GUO, P., XIAO, H., ZHOU, H., AND QU, H. Scattering points in parallel coordinates. *IEEE Trans. Vis. Comput. Graph. 15*, 6 (2009), 1001–1008.

[222] YUAN, X., REN, D., WANG, Z., AND GUO, C. Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data. *IEEE Trans. Vis. Comput. Graph. 19*, 12 (2013), 2625–2633.

[223] ZANABRIA, G. G., NONATO, L. G., AND NIETO, E. G. istar (i*): An interactive star coordinates approach for high-dimensional data exploration. *Computers & Graphics 60* (2016), 107–118.

[224] ZANELLA, A., CARPENDALE, M. S. T., AND ROUNDING, M. On the effects of viewing cues in comprehending distortions. In *Proceedings of the Second Nordic Conference on Human-computer Interaction* (New York, NY, USA, 2002), NordiCHI '02, ACM, pp. 119–128.

[225] ZELLWEGER, P., MACKINLAY, J. D., GOOD, L., STEFIK, M., AND BAUDISCH, P. City lights: contextual views in minimal space. In *Extended abstracts of the 2003 Conference on Human Factors in Computing Systems, CHI 2003, Ft. Lauderdale, Florida, USA, April 5-10, 2003* (2003), pp. 838–839.

[226] ZHOU, F., LI, J., HUANG, W., ZHAO, Y., YUAN, X., LIANG, X., AND SHI, Y. Dimension reconstruction for visual exploration of subspace clusters in high-dimensional data. In *2016 IEEE Pacific Visualization Symposium, PacificVis 2016, Taipei, Taiwan, April 19-22, 2016* (2016), pp. 128–135.