Visual Data Analysis in Device Ecologies

Tom Horak

August 2021

Technische Universität Dresden

Fakultät Informatik
Institut für Software- und Multimediatechnik
Interactive Media Lab Dresden

Visual Data Analysis in Device Ecologies

Dissertation
zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

Tom Horak
Master of Science in Medieninformatik
geboren am 05.03.1991 in Gera

Erstgutacher und Betreuer Prof. Dr.-Ing. Raimund Dachselt
Technische Universität Dresden
Interactive Media Lab Dresden

Zweitgutachter Jo Vermeulen, Ph.D.
Autodesk Research
HCI & Vis Group

Kommissionsvorsitz Prof. Dr. Markus Krötzsch
Fachreferent Prof. Dr. rer. nat. habil. Gerhard Weber
Weiteres Mitglied Prof. Dr.-Ing. Wolfgang Lehner

Tag der Verteidigung 07. Juli 2021

Dresden, im August 2021

Tom Horak
Visual Data Analysis in Device Ecologies
Dissertation, Submitted on March 25, 2021
Reviewers: Prof. Dr.-Ing. Raimund Dachselt and Jo Vermeulen, Ph.D.
Supervisor: Prof. Dr.-Ing. Raimund Dachselt

Technische Universität Dresden
Interactive Media Lab Dresden
Institute of Software and Multimedia Technology
Faculty of Computer Science
Nöthnitzer Straße 46
01187 Dresden

Abstract

With the continued development towards a digitalized and data-driven world, the importance of visual data analysis is increasing as well. *Visual data analysis* enables people to interactively explore and reason on certain data through the combined use of multiple visualizations. This is relevant for a wide range of application domains, including personal, professional, and public ones. In parallel, a ubiquity of modern devices with very heterogeneous characteristics has spawned. These devices, such as smartphones, tablets, or digital whiteboards, can enable more flexible workflows during our daily work, for example, while on-the-go, in meetings, or at home. One way to enable flexible workflows is the combination of multiple devices in so-called *device ecologies*. This thesis investigates how such a combined usage of devices can facilitate the visual data analysis of multivariate data sets. For that, new approaches for both visualization and interaction are presented here, allowing to make full use of the dynamic nature of device ecologies. So far, the literature on these aspects is limited and lacks a broader consideration of data analysis in device ecologies.

This doctoral thesis presents investigations into three main parts, each addressing one research question: (i) how visualizations can be adapted for heterogeneous devices, (ii) how device pairings can be used to support data exploration workflows, and (iii) how visual data analysis can be supported in fully dynamic device ecologies. For the first part, an extended analytical investigation of the notion of responsive visualization is contributed. This investigation is then complemented by the introduction of a novel matrix-based visualization approach that incorporates such responsive visualizations as local focus regions. For the two other parts, multiple conceptual frameworks are presented that are innovative combinations of visualization and interaction techniques. In the second part, such work is conducted for two selected display pairings, the extension of smartwatches with display-equipped watchstraps and the contrary combination of smartwatch and large display. For these device ensembles, it is investigated how analysis workflows can be facilitated. Then, in the third part, it is explored how interactive mechanisms can be used for flexibly combining and coordinating devices by utilizing spatial arrangements, as well as how the view distribution process can be supported through automated optimization processes. This thesis's extensive conceptual work is accompanied by the design of prototypical systems, qualitative evaluations, and reviews of existing literature.

Zusammenfassung

Die fortschreitende Entwicklung hin zu einer digitalisierten Welt geht auch mit einer steigenden Bedeutung der *visuellen Datenanalyse* einher. Solch eine Analyse ermöglicht es, bestimmte Daten durch die Verwendung von Visualisierungen interaktiv zu explorieren und Erkenntnisse abzuleiten. Dies hat eine große Relevanz für eine Vielzahl von Anwendungsbereichen, sowohl im persönlichen, beruflichen als auch öffentlichen Umfeld. Zur gleichen Zeit hat auch die Verbreitung von neuartigen Geräten mit heterogenen Eigenschaften stark zugenommen. Geräte wie Smartphones, Tablets oder digitale Whiteboards erlauben flexiblere Abläufe im Arbeitsalltag, beispielsweise wenn unterwegs, in Meetings oder zu Hause. Die Kombination mehrerer Geräte zu sogenannten *'Device Ecologies'* ist eine Möglichkeit, solche flexibleren Abläufe zu unterstützen. Diese Dissertation untersucht, wie eine solche kombinierte Nutzung moderner Geräte die visuelle Datenanalyse von multivariaten Datensätzen unterstützen kann. Dafür werden neue Ansätze für sowohl die Visualisierung als auch die Interaktion präsentiert, welche es erlauben, den dynamischen Charakter von *Device Ecologies* voll zu nutzen. Diese Aspekte der visuellen Datenanalyse in *Device Ecologies* wurden bislang nur unzureichend untersucht.

Diese Dissertationsschrift umfasst drei Hauptteile, welche je eine Forschungsfrage adressieren: (i), wie Visualisierungen für heterogene Geräte angepasst werden können, (ii), wie Gerätepaarungen zur Unterstützung von typischen Explorationsabläufen genutzt werden können, und (iii), wie die visuelle Datenanalyse in komplett dynamischen Set-ups unterstützt werden kann. Für den ersten Teil wird eine analytische Untersuchung des Konzepts von *responsiven Visualisierungen* bereitgestellt. Diese wird dann ergänzt durch die Einführung eines neuartigen matrixbasierten Visualisierungsansatzes, in dem responsiven Visualisierungen als lokale Fokusregionen genutzt werden. Für die beiden anderen Teile der Dissertation werden mehrere konzeptionelle Frameworks vorgestellt, welche innovative Kombinationen von Visualisierungs- und Interaktionstechniken sind. Im zweiten Teil werden solche Frameworks für zwei ausgewählte Display-Paarungen eingeführt: für die Erweiterung von Smartwatches mit Armbändern mit integrierten Displays sowie für die gegensätzliche Kombination von Smartwatch und großem Display. Für diese Gerätekombinationen wird untersucht, wie Analyseabläufe unterstützt werden können. Anschließend wird im dritten Teil betrachtet, wie einerseits die dynamische Koordinierung von Geräten durch die interaktive Einbeziehung der räumlichen Anordnung sowie andererseits die Verteilung von Ansichten auf verschiedene Geräte durch automatisierte Optimierungsprozesse erfolgen kann. Diese umfangreiche konzeptionelle Arbeit wird durch den Entwurf prototypischer Systeme, qualitative Evaluationen und umfassende Analysen existierender Forschung begleitet.

Acknowledgements

The past years have been an incredible experience that now peak in writing this thesis. This journey was coined by the support, guidance, and feedback of several great people as well as many friendly encounters along the way. I am truly grateful for the manifold ways these people shaped my life.

First of all, I want to sincerely thank my supervisor *Raimund Dachselt*. Raimund, you have always been encouraging and supportive, constantly trying to enable me in reaching my goals. From time to time, this also involved forcing me to push beyond my comfort zone in order to hold my work accountable to the highest standards. Thus, it is very safe to say: this thesis would not exist without your support and I would also not be the person I am now. Thank you, Raimund!

I also want to explicitly thank *Jo Vermeulen* for agreeing to serve as second reviewer of my thesis and putting the therefor needed time and effort into it.

A special thanks is due to my current and former colleagues. *Ricardo Langner*, thank you for the countless discussions, fruitful joint research work, much-needed breaks, and all the deep reflections on research and life. Your supervision and mentorship during my Master's played an crucial part in me starting a PhD and I am particularly grateful that this developed further into a friendship. Since 2016, I am sharing an office with *Konstantin Klamka* and I could have not have wished for a better office mate. Thank you, Konstantin, for all the conversations, mutual support, and the over 1.500 shared coffees (rough estimation)! A thank you is also due to *Ulrike Kister* for our enjoyable collaboration on graphs and digital boardrooms. You were always a great company during work and when traveling!

Further, I want to thank *Wolfgang Büschel* for all the entertaining as well as intense discussions, *Ramona Behling* for all your support in fighting the red tape, *Annett Mitschick* for all your helpful and wise advices, and *Tamara Flemisch* for the pleasant company within CPEC. During my time at the Interactive Media Lab I had the chance to work with or alongside many more great colleagues: *Eva Goebel*, *Patrick Reipschläger*, *Marc Satkowski*, *Andreas Peetz*, *Udo Wähner*, *Ulrich von Zadow*, *Weizhou Luo*, *Severin Engert*, *Robert Fuhrmann*, *Anke Lehmann*, and *Fabrice Matulic*. Thank you for making the IML to such an inspiring workplace!

I feel very lucky that I also got the opportunity to jointly work together with many awesome people from around the world. I want to hugely thank *Niklas Elmqvist* for hosting me for six months at the University of Maryland. The many friendly conversations as well as the intense and joyful work on Vistribute were truly inspiring. Together with our earlier joint work on David meets Goliath, two core papers of my thesis came together through our successful collaborations. In the context of these particular publications, I am also thankful to *Karthik Badam* for the smooth teamwork across time zones as well as *Andreas Mathisen* for the productive work on two papers in parallel and making my stay abroad even more pleasant (thanks for all the IPAs!). For the joint work on the GEMS 2.0 project, I want to thank *Christian Tominski* and *Heidrun Schumann* for all the intense discussions as well as *Philip Berger* for the mutual support while working on the RMC project.

Most importantly, I am very grateful to several people outside the research bubble. To my *family*: Who would have thought that I will end up writing a doctoral thesis? All this would not have been possible without your support during my whole life! Then, *Lisa*: Thank you for helping me to stay sane, particularly during the final phase of my dissertation. I'm endlessly thankful that we always managed to find joy and relaxation; despite all the workload, despite a global pandemic. I'm very much looking forward to our joint future endeavors. Lastly, I also want to thank *Luke*, *Celina*, and *Franzi* for proof-reading the different parts of this thesis.

I want to conclude these acknowledgements with a personal statement. Within the last years, we all had to repeatedly listen to news from across the world that were worrisome, at least. From my personal experience, I can only stress that a diverse society and a healthy planet is an inspiring and exciting treasure that we have to defend everyday. I would not be the same person, if I would not have had the chance to enjoy visiting places all over the world and meet so many fantastic people with completely different backgrounds—and I am thankful that I can call many of them friends now. As a tiny contribution to keep our planet healthy, I will place a donation to the City of Dresden for planting a tree. I hope it will grow and become part of a decent future.

Funding

Most of the research presented here was supported by the Deutsche Forschungsgemeinschaft (DFG) by funding the *GEMS 2.0* research project (grant 214484876, see `imld.de/gems2`) as well as the Transregional Collaborative Research Centre (TRR) 248 – CPEC (grant 389792660, see `perspicuous-computing.science`).

Publications

During the work on this thesis, the following publications have been published (all peer-reviewed, in reverse chronological order). Where applicable, material used in this thesis is indicated at the beginning of each chapters.

Full-Paper Conference Publications

- Konstantin Klamka*, **Tom Horak***, and Raimund Dachselt. "Watch+Strap: Extending Smartwatches with Interactive StrapDisplays". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, pages 72:1–72:15. **The first two authors contributed equally.*

  Material from this publication appears in Chapter 5.

- **Tom Horak**, Andreas Mathisen, Clemens N. Klokmose, Raimund Dachselt, and Niklas Elmqvist. "Vistribute: Distributing Interactive Visualizations in Dynamic Multi-Device Setups". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, pages 616:1–616:13.

  Material from this publication appears in Chapter 8.

- **Tom Horak***, Sriram Karthik Badam*, Niklas Elmqvist, and Raimund Dachselt. "When David Meets Goliath: Combining Smartwatches with a Large Vertical Display for Visual Data Exploration". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, pages 19:1–19:13. **The first two authors contributed equally.*
  The publication received an **Honorable Mention Award**.

  Material from this publication appears in Chapter 6.

Journal Articles

- **Tom Horak***, Philip Berger*, Heidrun Schumann, Raimund Dachselt, Christian Tominski. "Responsive Matrix Cells: A Focus+Context Approach for Exploring and Editing Multivariate Graphs". In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pages 1644–1654. **The first two authors contributed equally.*

  Material from this publication appears in Chapter 4.

- Andreas Mathisen, **Tom Horak**, Clemens N. Klokmose, Kaj Grønbæk, and Niklas Elmqvist. "InsideInsights: Integrating Data-Driven Reporting in Collaborative Visual Analytics". In: *Computer Graphics Forum* 38.3 (Jun. 2019), pages 649–661.

- Ricardo Langner, **Tom Horak**, and Raimund Dachselt. "VisTiles: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration". In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018), pages 626–636.

  Material from this publication appears in Chapter 7.

Book Chapters

- **Tom Horak**, Wolfgang Aigner, Matthew Brehmer, Alark Joshi, Christian Tominski. "Responsive Visualization Design for Mobile Devices". In: *Mobile Data Visualization*, AK Peters Visualization Series. CRC Press, 2021, 34 pages. To appear.

  Material from this publication appears in Chapter 3.

- Tanja Blascheck, Frank Bentley, Eun Kyoung Choe, **Tom Horak**, Petra Isenberg. "From Perception to Behavior Change: Characterizing and Evaluating Glanceable Mobile Visualizations". In: *Mobile Data Visualization*, AK Peters Visualization Series. CRC Press, 2021, 25 pages. To appear.

- Ricardo Langner, Ulrich von Zadow, **Tom Horak**, Annett Mitschick, and Raimund Dachselt. "Content Sharing Between Spatially-Aware Mobile Phones and Large Vertical Displays Supporting Collaborative Work". In: *Collaboration Meets Interactive Spaces*. Springer International, 2016, pages 75–96.

Extended Abstracts

Doctoral Consortium

- **Tom Horak**. "Designing for Visual Data Exploration in Multi-Device Environments". In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, pages DC11:1–DC11:6.

Poster Publications

- **Tom Horak**, Ricardo Langner, Raimund Dachselt. "Towards Visualizing and Exploring Multivariate Networks on Mobile Devices". In: *Companion Proceedings of the ACM Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2020, pages 5–8.

  Material from this publication appears in Chapter 4.

- **Tom Horak**, Ulrike Kister, and Raimund Dachselt. "Comparing Rendering Performance of Common Web Technologies for Large Graphs". In: *Poster Program of the IEEE VIS Conference*. 2018, 2 pages.

- Javid Abbasov, **Tom Horak**, and Raimund Dachselt. "Smartwatch-based Pointing Interaction". In: *Mensch und Computer 2018 - Tagungsband*. Bonn, Germany: Gesellschaft für Informatik e.V., 2018, pages 261–265.

- **Tom Horak**, Ulrike Kister, and Raimund Dachselt. "Improving Value Driver Trees to Enhance Business Data Analysis". In: *Poster Program of the IEEE VIS Conference*. 2017, 2 pages.

- **Tom Horak**, Ulrike Kister, and Raimund Dachselt. "Presenting Business Data: Challenges during Board Meetings in Multi-Display Environments". In: *Proceedings of the ACM International Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2016, pages 319–324.

- Ricardo Langner, **Tom Horak**, and Raimund Dachselt. "Towards Combining Mobile Devices for Visual Data Exploration". In: *Poster Program of the IEEE VIS Conference*. 2016, 2 pages.
  The publication received an **Honorable Mention Award**.

Demonstrations

- Ricardo Langner, **Tom Horak**, and Raimund Dachselt. "Demonstrating VisTiles: Visual Data Exploration Using Mobile Devices". In: *Proceedings of the International Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2018, pages 69:1–69:3.

  Material from this publication appears in Chapter 7.

- **Tom Horak**, Sriram Karthik Badam, Niklas Elmqvist, and Raimund Dachselt. "Demonstrating David Meets Goliath: Combining Smartwatches with a Large Vertical Display for Visual Data Exploration". In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, pages D414:1–D414:4.

Workshop Publications

- **Tom Horak** and Raimund Dachselt. "Hierarchical Graphs on Mobile Devices: A Lane-based Approach". In: *Proceedings of the CHI 2018 Workshop on Data Visualization on Mobile Devices*. 2018, 7 pages.
- **Tom Horak**, Ulrich von Zadow, Matthias Kalms, and Raimund Dachselt. "Discussing the State of the Art for 'in the wild' Mobile Device Localization". In: *ISS 2016 Workshop on Interacting with Multi-Device Ecologies 'in the wild'*. 2016, 7 pages.
- **Tom Horak**, Ulrike Kister, Konstantin Klamka, Ricardo Langner, and Raimund Dachselt. "Logging in Visualizations: Challenges of Interaction Techniques Beyond Mouse and Keyboard". In: *Workshop on Logging Interactive Visualizations & Visualizing Interaction Logs (LIVVIL)*. 2016, 3 pages.
- Wolfgang Büschel, Ricardo Langner, Ulrich von Zadow, **Tom Horak**, and Raimund Dachselt. "Towards Cross-Surface Content Sharing Between Mobile Devices and Large Displays in the Wild". In *CHI 2016 Workshop on Interacting with Multi-Device Ecologies 'in the Wild'*. 2016, 5 pages.

Supervised Student Theses

The following student theses that I have co-supervised during the time of my PhD work contributed in part to this dissertation (in reverse chronological order):

- Tobias Knothe. *Responsive Graphs: Adapting Node-Link Diagrams for Mobile Devices*. Master's Thesis, 2020.
- Florian Semt. *Interactive Comparison of Graphs on Mobile Devices*. Bachelor's Thesis, 2020.
- Eva Goebel. *Responsive Visualizations: Systematic Investigation und Characterization*. Master's Thesis, 2019.
- Lars Arne Beck. *Strategies for Adapting Visualizations in Dashboards for Usage on Mobile Devices*. Master's Thesis, 2019.
- Paul Hoffmann. *Embedded Micro Visualizations in Interactive Matrix Visualizations of Multivariate Graph*. Student Project Thesis, 2019.
- Anselm Bunsen. *Novel Approaches for Visualizing and Interacting with Data on Smartwatches*. Bachelor's Thesis, 2019.
- Elisabeth Baudisch. *Visual Data Exploration of Graphs with Cross-Device Workflows between a Large Display and Mobile Devices*. Master's Thesis, 2018.
- Javid Abbasov. *Pointing Interaction with Mobile Devices Using Internal Sensors*. Master's Thesis, 2018.
- Antonio Pietzsch. *Semi-automatic Editing of Elements and Layout in Multivariate Graph Visualization*. Bachelor's Thesis, 2018.
- Robin Thomas. *Performance Analysis and Optimization of Web-based Multivariate Graph Visualizations*. Bachelor's Thesis, 2018.

I want to thank the students for their interest in the topics and their contributions to the overall research.

Contents

List of Figures

List of Tables

Part I

Introduction and Research Background

1 Introduction

We are living in a data-driven world. With the continuing digitalization process affecting all areas of our daily lives, there is also an increase in data being recorded or created on the way—as well as an increased interest in making sense of this data in order to be able to derive insights and actions from it. This holds true at a personal, professional, and public level, for example, when looking at private fitness data, companies' processes data, or countries' public health data. At all stages of working with such data, no matter if personal or big data, visualization and data analysis are required to access and make sense of it as well as to communicate the gathered insights [Spe01; TS20]. At the same time, data analysis is not limited to take place on desktop computers anymore. Modern devices, such as smartwatches, smartphones, tablets, laptops, or digital whiteboards, allow to access information anytime and anywhere [Wei91] and can provide the opportunity to interact with information in more natural ways [Lee+12; Rob+14]. This thesis contributes concepts for how visual data analysis can be facilitated on modern devices by using them in combination.

Visualization & Data Analysis

At the core of visual data analysis are *visualizations* that can encode the data in various ways by mapping values to visual channels, and thereby creating external representations [Mun14]. Having these visual representations is in most cases essential and the only way to properly represent the often complex, high-dimensional, and large data collections. Instead of parsing single data entries, visualization can allow to represent the entirety of a data (sub)set making the characteristics of it apparent, such as the distribution of attribute values, correlations, clusters, or outliers. The possible designs of visualizations are manifold and have been researched extensively [Mun14; Spe01; Tuf06]. While visualizations can be used statically for a pure communication purpose, the *interaction* in and with visualizations is often crucial to fully understand the different aspects of the data [Tom15]. For example, by selecting data points to access details, zooming into a representation to focus on a subpart, or filtering elements that are not of interest, it becomes possible to actually explore the data, investigate its various characteristics, and derive insights [Shn96].

With such interactive visualizations, *visual data analysis* (VDA) is enabled. Consequently, a VDA interface (e.g., shown in Figure 1.1a) incorporates a set of visualizations, data operations, interface components, as well as suitable interaction means [TS20]. The objective of every data analysis is deriving specific insights, although they do not necessarily have to be known beforehand. Specific goals and how they are reached in VDA interfaces can be described with *analysis tasks*, which express in what aspects analysts are interested and which steps they follow to derive the required insight [BM13; HS12; YKS07]. For example, for identifying a specific data point and retrieving its value, it might be necessary to navigate in the visualization first before selecting the data point. In the context of this thesis, the focus is on how such a VDA interface can be controlled and how interactions can be supported specifically.

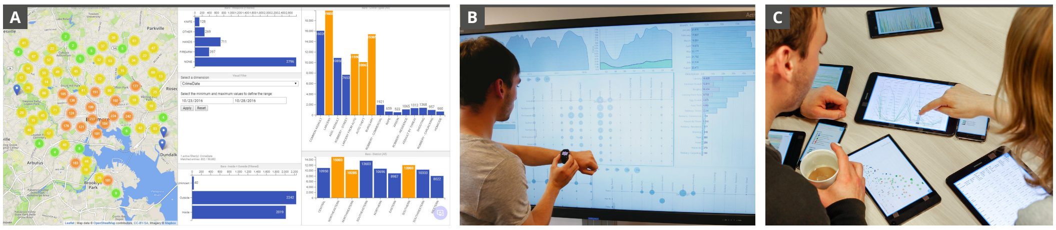


Fig. 1.1.: Visual data analysis interfaces (a) enable analysts to explore a given data set through interaction with multiple visualizations. When bringing such analyses to ecologies of modern devices (b+c), new ways of conducting an analysis emerge and have the potential to support a more natural type of working with the data.

Modern Devices & Device Ecologies

The advances in technology over the last few decades have also led to a ubiquity of new devices that now shape our everyday lives [BH10]. As illustrated in Figure 1.1b+c, these *modern devices* liberate people from the bonds of their desks, allowing them to conduct computer-assisted work in various contexts and with different devices [DP08; Hus+18; SW13]. However, as the devices' characteristics notably differ from traditional desktop systems, it must be carefully considered how existing concepts, such as visualization and analysis interfaces, can be brought to them. Within the visualization research community, this is often referred to as *visualization beyond the desktop* [Lee+12; Rob+14]. Among other aspects, it was investigated how touch input can be incorporated or how the visualization can be provided within the limited screen estate of mobile devices. However, prior research mainly focused on solutions for a specific visualization or specific device. With the space of both available devices and visualization techniques being vast, further research is required.

Within the context of visual data analysis, one promising way to incorporate modern devices is their combined use. Such *device combinations* can allow to overcome limitations of one device and create synergies emerging from the different strengths of the devices [Bru+19]. For example, while a large display allows to show much information in parallel, the reachability of content can be challenging, particularly when standing at an overview distance [BNB07; LKD19]. Adding a mobile device such as smartwatch or smartphone can allow to interact with the content from a distance as well as to provide further information on its display (Figure 1.1b). Thus, within these combinations, devices can take on different roles that express which interface parts or mechanisms can be covered by them. With the high diversity of devices, many possible combinations as well as device roles exist [Bru+19]. Yet, particularly in a visualization context, existing research has been investigating only few selected device combinations so far.

The vast number of possible device pairings also indicates that visual data analysis should not be optimized for a few selected ensembles, but for dynamic and changing ***device ecologies***. I follow the notion of a device ecology, which I consider to be *a changeable set of co-located computing devices that are used in combination and for which a suitable and dynamic interface is provided that actively considers device characteristics and user context*. On the one hand, this means that the interface features a view distribution and interactive mechanisms that lead to the experience that devices symbiotically complement each other and are behaving as one dynamic unit (Figure 1.1c)—similar to organisms in real ecologies. On the other hand, the specific interplay of devices and views in the interface is not fixed and can change anytime, either because of changed user goals or a changed device setup, which is supported through suitable automated processes.

This idea of realizing synergistic multi-device setups is also shared with other research work, for example, titled as "society of devices" by Fitzmaurice et al. [Fit+03], "device ensembles" by Schilit and Sengupta [SS04], "device ecology" by Indrawan et al. [ILL07], or "display ecologies" by Chung et al. [Chu+15]. For such synergistic setups, novel interaction and visualization concepts as well as new software architectures become required. The work presented here contributes such concepts within the context of data analysis, where device ecologies are particularly promising as they have the potential to effectively support the dynamic nature of analysis processes. So far, this space of visual data analysis in device ecologies has remained underexplored in the literature and is extended by this thesis.

1.1 Research Questions and Objectives

The overall goal of this thesis is to investigate how visual data analysis can be facilitated in device ecologies. Consequently, the work builds on existing knowledge on multi-device environments within the human-computer interaction (HCI) research area as well as the knowledge on data analysis coming from information visualization (InfoVis) research (Figure 1.2). From this starting point, it is explored how visualization can be brought to a diverse set of devices, how device roles can facilitate data analysis when using the devices in combination, as well as how the dynamic nature of device ecologies can be supported. These major research questions are detailed in the following.

RQ1: How can visualizations be adapted for heterogeneous devices?

So far, most research focused on adapting a specific visualization to a specific type of device, with the applied strategies often not being applicable for other visualization techniques or devices. In order to enable data analysis in device ecologies, more *generalized adaptation strategies* for bringing a visualization to a wider range of heterogeneous devices must be investigated. Specifically, this addresses the notion of responsive visualization [AS17; HLL20], which, however, has not been systematically considered yet. Thus, the goal is to develop an improved understanding of responsive visualization as well as possible strategies for adapting visualizations for various devices.

RQ2: How can specific device pairings be used to facilitate analysis workflows?

It has been shown that device combinations have the potential to support workflows during data analysis [Kis+17; LKD19; Woź+14]. While the existing work provides valuable insights, investigation of further device ensembles is required to be able to identify promising device roles and specific interplay between visualization views. This includes the design of *interaction and visualization concepts* for different device combinations. In combination with existing work, this can extend the knowledge on how synergies between devices and views can be reached within cross-device VDA interfaces.

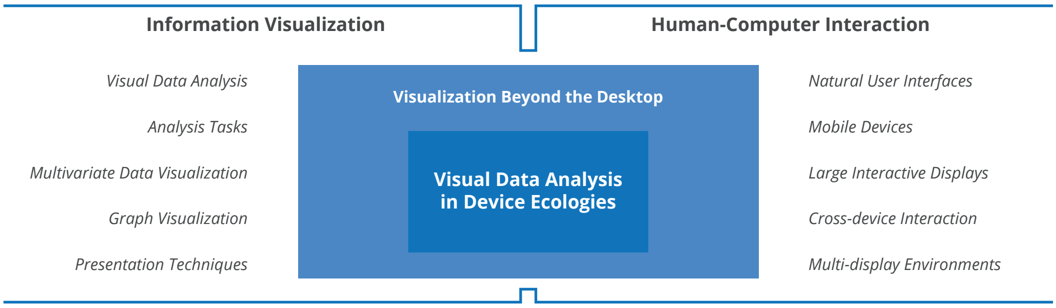


Fig. 1.2.: Within the intersection of *information visualization* (InfoVis) and *human-computer interaction* (HCI), one research area is *visualization beyond the desktop*. The here considered visual data analysis in device ecologies is one particular instance of it and incorporates knowledge from multiple sub-areas of InfoVis and HCI.

RQ3: How can visual data analysis effectively be supported in fully dynamic device ecologies?

> Within dynamic device ecologies, interface distribution and coordination between devices have to be flexible and must be established on the fly. While for pre-defined device pairings the applied coordination between visualization and devices can be directly wired into the interface, for device ecologies novel concepts are required that allow analysts to easily establish similar coordination functionalities in an ad-hoc fashion. Similarly, manually configuring and adapting the analysis interface to a changed device setup, e.g., requiring to re-assign views to devices, is a time consuming effort. Minimizing such configuration overheads is key for dynamic device ecologies and requires the development of suitable automated algorithms and processes. At the same time, the analyst has to remain in control over the overall interface and must be supported by providing suitable interaction means.

1.2 Research Scope

Both HCI and InfoVis research cover a huge spectrum of sub research areas, providing many interesting aspects in their intersection that can be considered (Figure 1.2). The work presented in this thesis is one instance of research that took place in the intersection and is closely related to work conducted under the notion of *visualization beyond the desktop* [Lee+12; Rob+14]. There, the general goal is to investigate ways of how visualizations can be provided and be made interactive within novel work environments, e.g., by involving mobile or large devices as well as by using input modalities other than mouse and keyboard. Device ecologies are one specific way to make use of the available devices in those environments. This thesis investigates their utility for visualization and data analysis within the following scope:

2D Visualization for Multivariate Data This thesis focuses on solutions for multivariate data sets as these are the most prominent data type in information visualization research. In such data sets, each data entry is characterized by multiple attributes. In this thesis, the considered visual representations are limited to 2D visualizations, explicitly excluding 3D techniques. Further, the here proposed concepts build mostly on existing visualizations, with creating new ones not being the main focus.

Interactive Data Analysis For the visual data analysis, the focus is on the interaction part taking place when exploring and investigating the visualized data. These interactions are described as visualization tasks in various taxonomies [BM13; HS12; YKS07] and apply to specific visualization elements (e.g., data points) or the visualization view as a whole (e.g., changing the visualization technique). Particularly relevant are common interface schemes or mechanisms for data analysis, such as linked brushing or focus+context approaches. Explicitly out of scope are data analysis aspects that are of interest before or after the exploration, such as data wrangling or presentation of insights.

Commodity Devices The considered devices include mainly common commodity devices such as smartwatches, mobile devices, laptop and desktop systems, as well as large interactive displays. The presented concepts incorporate the specific—and potentially very different—characteristics and strengths of the devices, especially when used in combination. Out of focus are still emerging devices, such as foldable devices or mixed reality devices, as well as devices not commonly found in work environments, like tabletop displays. Due to this focus it is ensured that the majority of currently available devices is considered and a reasonably large spectrum of different devices is covered, while avoiding over-specialized solutions.

Input Modalities & Interaction Techniques To maintain a manageable scope, the considered input modalities are mainly limited to touch input and spatial input. Touch input is the current de-facto standard on commodity devices and therefore also the primary input channel for visualization on these devices. The proposed interactions with the interface are designed with touch input in mind and often incorporate its different variations and gestures, e.g., long tap, double tap, drag, swipe, or pinch. Spatial input considers the relative positioning between devices as well as between devices and users and is particularly promising for multi-device setups. These movements are also incorporated as interactions, for example, for pointing or spatial device arrangements. Both input types can be used for single-device interactions as well as cross-device interactions.

Enabling Multi-user Usage Multi-user scenarios are increasingly common within data analysis and also well supported by larger devices as well as multi-device environments in general [Bru+19]. Consequently, the proposed concepts in this thesis also enable such multi-user usage for data analysis. However, explicit means of collaboration, such as user identification, cross-user interaction, or remote collaboration, are not in the scope and thus not further considered.

1.3 Methodological Approach

The proposed concepts and derived insights of this thesis build upon a thorough initial analytical investigation of related work. Following this first phase, I conducted multiple iterative design processes that resulted in specific conceptual frameworks, implementations, and design spaces. These processes took place within three research strands that directly map to the aforementioned research questions. These strands are also indicated in Figure 1.3. Commonly, the different design processes also included qualitative user studies to validate the proposed concepts and inform a better understanding of analysis requirements. In the following, the used methods are detailed:

Literature Review and Analytical Considerations As the starting point of this research, an extensive literature review was conducted. This review included existing research work concerned with natural user interfaces and cross-device interaction, bringing visualizations to modern devices, as well as foundations of visual data analysis and its support in multi-device environments. In addition, some design processes included a separate literature review or analytical considerations for a specific topic, for example, for multivariate graphs or interface distribution. Finally, for the considerations of responsive visualization, an own systematic exploration was required as this topic is not covered extensively in the literature yet.

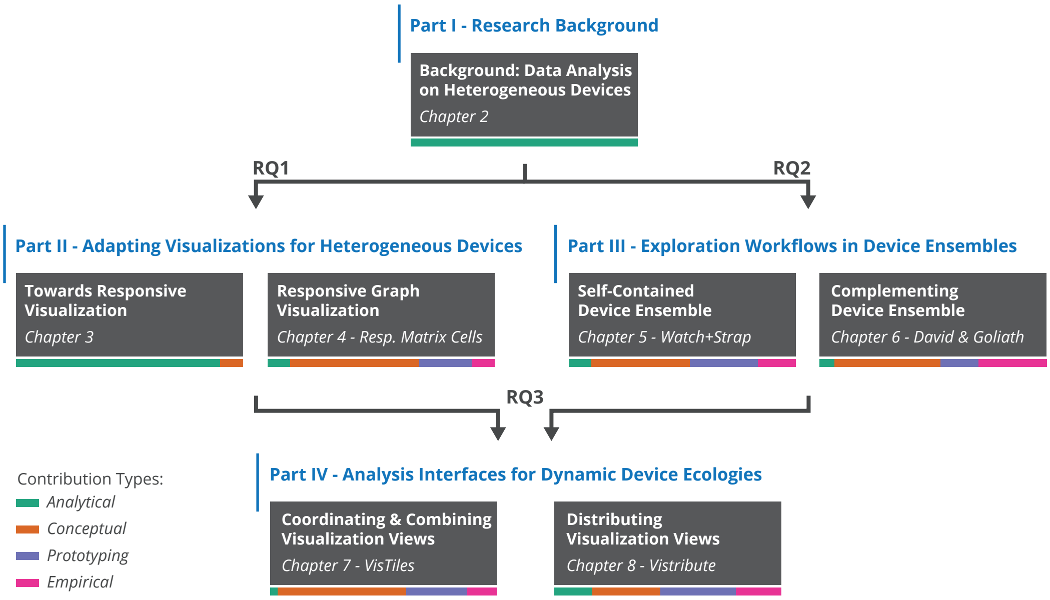


Fig. 1.3.: Overview of the work presented within this thesis. Part II, III, and IV directly relate to the research questions RQ1–RQ3, which have motivated the conducted research. The boxes represent the single chapters, with the colored stripes indicating the relative proportion of contribution type that the underlying work made.

Visualization and Interaction Design Within the works of this thesis, multiple conceptual frameworks are proposed. These frameworks are a combination of specific interaction techniques as well as visualization approaches that together can enable data analysis with modern devices. On the one hand, this involves concepts of how visualizations can be specifically displayed, adapted, arranged, or coordinated with each other. On the other hand, techniques for interacting with the visualizations as well as the devices are designed by considering the modalities of touch and spatial arrangements. Consequently, each chapter of this thesis incorporates both visualization and interaction aspects but can feature different proportions of them. As part of the underlying mechanisms, these concepts can also involve smart or automated functionalities that assist analysts beyond explicit interaction means. All concepts and solutions were developed in a systematic and iterative fashion.

Prototyping and System Design All proposed concepts were at least in parts realized as prototypes. Foremost, these were software prototypes providing a high-fidelity realization of the envisioned system. The required implementation work for these is also a fundamental part of the iterative design processes and important for being able to validate the solutions. Particularly in the context of the dynamic device ecologies, this also included a notable effort that had to be put into the design of these systems, e.g., to enable an efficient communication between devices and to coordinate their behavior. Further, these prototypes are also important for guaranteeing a reproducibility of the conducted research and, therefore, were made open source and publicly available. As a common technological basis, all software prototypes were implemented using web technologies. In addition, low-fidelity prototypes (e.g., paper prototypes) as well as hardware prototypes were realized too for some design explorations.

Design Evaluations: Qualitative User Studies The proposed concepts were evaluated through various qualitative studies, investigating how well users were adopting them. These studies involved both formative and summative instances and were an integral part of the design processes. For the formative ones, feedback sessions as well as brainstorming sessions were conducted, while the summative ones involved observational studies in combination with semi-structured interviews or questionnaires. In general, the dependence on qualitative validation approaches was due to the novelty of data analysis in device ecologies. This novelty meant that adequate baseline conditions for quantitative comparisons were typically not available. Further, qualitative insights can better facilitate understanding the users' perspective onto the concepts.

1.4 Thesis Outline & Contributions

The work presented in this thesis extends the understanding of the specific space of visual data analysis in device ecologies. The main contributions are directly related to the raised research questions: approaches for adapting visualizations for heterogeneous devices (RQ1), concepts for utilizing device pairings for facilitating analysis workflows (RQ2), as well as concepts for effectively supporting data analysis within fully dynamic device ecologies (RQ3). These three-folded contributions are also reflected in the structure of the thesis through three corresponding parts (Figure 1.3). These main parts are accompanied by the required introduction and background as well as the subsequent discussion of the results. In the following, all parts and their main contribution are summarized:

Part I: Introduction and Research Background Following this introductory chapter, the remainder of the first thesis part is devoted to the literature review. Specifically, **Chapter 2** structures and reports on existing research related to the thesis' scope.

More specifically, this involves reporting on the specifics of modern devices and natural user interfaces, existing investigations on cross-device interaction, work on visualization beyond the desktop, as well as relevant aspects of visual data analysis and prior work supporting it within multi-device environments. This survey provides the required background and foundation for the work presented in this thesis.

Part II: Adapting Visualizations for Heterogeneous Devices The second part addresses the research question RQ1 of how visualizations can be adapted for heterogeneous devices. In order to inform an improved understanding, **Chapter 3** contains an analytical exploration of the notion of *Responsive Visualization* [Hor+21a], which is yet underexplored in the literature. Following these generalized considerations, the aspects of responsiveness are then explored specifically for multivariate graph visualization in **Chapter 4**. As the major contribution, the *Responsive Matrix Cells* [Hor+21b] concept is introduced, which is a focus+context approach for matrix visualization that allows embedding responsive detail visualizations in local focus areas. In sum, Part II provides an improved understanding of how visualizations can adapt to different contexts and support data analysis within a compact space.

Part III: Analysis Workflows in Dedicated Device Ensembles For the third part, the focus then shifts to using device ensembles for data analysis (RQ2). In **Chapter 5** [KHD20], the novel display combination of smartwatch and display-equipped watchstraps and its usage for personal visualization is proposed. With this self-contained ensemble called *Watch+Strap*, concepts for using the different displays in combination are developed and tested. Similarly, **Chapter 6** investigates how the contrary device types of smartwatch and large display—like *David and Goliath*—can be used in a complementing synergy for visual data analysis [Hor+18b]. Following the design of a conceptual framework, this device setup was studied more extensively with a focus on which workflows analysts adopt while working in this environment. Both explorations provide novel concepts for mobile visualization and visualization beyond the desktop, as well as a better understanding of how device roles can aid and support common exploration workflows when investigating multivariate data sets in such ensembles.

Part IV: Analysis Interfaces for Dynamic Device Ecologies Taking into consideration the insights from the previous two parts, Part IV investigates how visual data analysis can be supported in fully dynamic device ecologies (RQ3). **Chapter 7** focuses on the interaction design that allows to apply device coordination and view combinations in an ad-hoc fashion. Specifically, the conceptual framework called *VisTiles* [LHD18b; LHD18a] focuses on exploiting changing physical device arrangements to resemble similar device roles as explored in Part III. The flexibility of dynamic device ecologies also means that the present device setup can change often and that the interface has to be adapted accordingly. Here, **Chapter 8** presents the *Vistribute* [Hor+19] framework that proposes multiple heuristics that can support an automatic distribution of visualization views across available devices by considering both view specifications and device properties. The accompanying implementation showed that the quality of such automatic distributions is rated by users similar to manually created ones, while notably reducing the setup effort. In sum, the works presented in this part provide concepts and approaches that can allow to actually enable visual data analysis in dynamic device ecologies and support a more natural way of conducting these analyses.

Part V: Conclusion In the last part, **Chapter 9** concludes this thesis by recapping the gained insights and contributions made as well as providing a thorough discussion of remaining limitations and possible future research directions.

2 Background: Data Analysis on Heterogeneous Devices

This chapter provides the required background for this thesis and presents the existing related work. The background is structured into four parts, starting with general considerations of modern computing devices (2.1) and characterizing their use in device ecologies (2.2). Then, visualization research concerned with visual data analysis (2.3) and utilizing modern devices (2.4) is discussed. Finally, the existing knowledge for conducting visual data analyses in multi-device settings (2.5) is reviewed.

The first two sections have the goal to provide a compact overview of how interaction can be realized with and on modern devices, such as smartphones, tablets, or large interactive displays. The third and fourth sections focus on the general aspects of data analysis and visualization as well as depict how visualizations can be brought to modern devices. The last section will focus on the few existing research approaches focused on bringing data analyses to multi-device environments. Taken together, the required background is coming from both the human-computer interaction (HCI) and information visualization (InfoVis) research community. This serves as the foundation for the contributions of this thesis, which comprise novel approaches for using devices for data analysis not only in a separated way but in synergy.

2.1 Modern Devices for Natural User Interfaces

In the HCI community, more natural ways of interacting with computers have been investigated for decades. Among others, this was done by utilizing devices with different characteristics and capabilities than traditional desktop systems. In the following, an overview is provided on this research with respect to modern devices, involving in general devices with state-of-the-art input and output capabilities as well as explicitly mobile devices. Following the scope of this thesis, only widely available commodity devices are considered, excluding, e.g., mixed reality devices.

2.1.1 Device Landscape and Characteristics

The range of currently available computing devices covers tiny personal ones, such as smartwatches, up to large shared display walls. In between, many device classes can be named, for example, smartphones, tablets, laptops, or desktops. However, nowadays, the boundaries between these classes are highly fluid, transforming it into a continuum. The devices come with very different characteristics, for example, their size, mobility, usage period, or association to users (cf. [PD15, p. 560]) that can be mapped onto respective dimensions as shown in Figure 2.1.

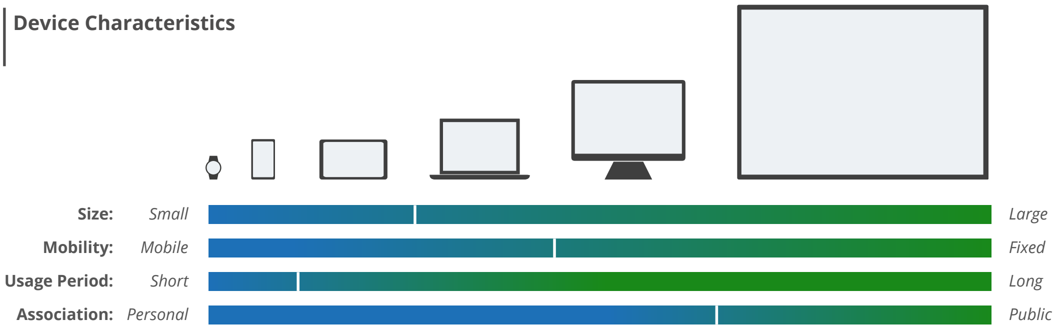


Fig. 2.1.: Modern devices can be characterized along multiple dimensions, here displayed for size, mobility, usage period, and association. As the boundaries between device classes are fluid, there is no exact mapping of device to value but mostly a tendency. The white lines in the figure indicate tipping points, where devices are typically categorized towards the one or other extreme.

Inherited from these characteristics, each device has advantages and disadvantages, rendering them particularly suitable for some usage context and unsuitable for others. This means that in contrast to desktop computers, which served as universal tools for almost all tasks of computer-assisted work, people now have the opportunity to choose the device or devices that fit their current needs most [SW13]. In the following, the typical usage patterns of todays' commodity devices are outlined as well as how they can potentially allow for more natural user interaction. An extended discussion of the device characteristics themselves will be provided later in the context of responsive visualization in Chapter 3.

2.1.2 Usage Patterns of Modern Devices

In a field study published in 2013, Santosa and Wigdor [SW13] found that the interviewed stakeholders used on average 2.9 traditional computers (laptop, desktop, netbook) as well as 2.7 mobile devices (smartphone, tablet, eReader) within their work routines. However, these devices are mostly used in a stand-alone fashion and at different locations, with the content being simply forwarded between the devices. The usage of the different devices was mostly motivated by their availability, for example, a workstation in the office, a personal computer when working from home, or a tablet when doing on-site visits. This situation has not significantly changed since the study was conducted, with a wide range of devices available and more flexible work settings being offered. This introduces the need to support users in achieving their goals on various devices and situations, but also in transferring the content and allowing to continuing the work in a different context.

The goal of supporting the users comes with different challenges for each device type. For example, interactions with smartwatches tend to be short-lived or only involve glanceable usage while the content itself must be provided within a very limited screen estate [Bla+21; Piz+16; Vis+17]. Similarly, smartphones are often used for rather short periods of less than 15 seconds [Fer+14] to check or review specific information [Ban+14]. Further, smartphones serve now as smart companions that are equipped with increasingly more computation power and a multitude of sensors, but still maintain a rather compact form factor—thus, also facing screen space limitations. In contrast, large vertical displays or digital whiteboards allow for displaying vast content and are often used in collaboration with other persons. At the same time, the often standing interaction from varying distances and reachability of distant content must be considered [BNB07; LKD19]. In conclusion, devices have different affordances and different means to provide content, thus requiring different strategies to support a given user task via an interface. As data analysis is no different, this aspect will be a common theme in this thesis, with particularly Part II being focused on adapting visualizations for a wide range of devices.

2.1.3 Natural Interaction Styles

Modern devices are no longer limited to mouse and keyboard-based interaction, but can allow for utilizing more direct and more natural interaction modalities. In research, this class of interfaces has been considered under different terms, such as ***natural user interfaces*** **(NUI)** or *post-WIMP interfaces*. The common goal is to provide interaction means that borrow skills, behaviors, and knowledge from the physical world to provide a more intuitive interaction [WW11]. Such interaction that is based on the skills and awarenesses of people is also known as *reality-based interaction* [Jac+08]. For example, this can involve imitating physical behaviors such as friction with UI elements, allowing to directly manipulate objects via touch input, or allowing to freely arrange devices around us. Ideally, this makes it easier for people to provide the correct input to a system in order to achieve a desired outcome, as well as to interpret the provided output by a system and understanding its state. These gaps in the communication between human and computer are also known as the *gulfs of execution and evaluation* [ND86]. Notably, NUIs are not only characterized by the interaction itself or the used modality (e.g., touch, gestural, spatial, speech, or gaze-based input), but also by the usage context: working on-the-go or at a desk, alone or collaboratively with others. Here, the term *natural user interfaces* is used to describe any system that aims to utilize novel interaction styles in various contexts and to provide a different experience then mouse and keyboard-based desktop environments.

While the altered way of how devices are used was already outlined above, the following discussion aims to detail the specific interaction means provided by modern devices. In general, the space of input modalities is vast, ranging from touch to speech to gaze input. For the purpose of this thesis, the considered modalities are limited to *touch interaction* as well as *spatial interaction*. The first one is considered as the default input type of todays' mobile devices but also of specialized devices such as interactive whiteboards. In addition, spatial interaction is particularly promising when considering the parallel usage of multiple devices, as in these situations spatial arrangements are always present and can potentially be used to facilitate interaction. For now, the focus is on the fundamental aspects of these two interaction types, while Section 2.4 will deepen the discussion of these in the context of visualization.

Touch Interaction

Interacting via touch developed into the de facto standard, particularly on mobile devices but also increasingly on laptops and large displays. Similar to mouse-based interfaces, the direct manipulation paradigm [Shn83; HHN85] is followed, however, as the input and output space is directly overlaid, touch input can better imitate real-world concepts. For example, a virtual object can be 'touched' and moved to a different location by simply dragging it to the specific target. Within the last decade, multiple touch gestures have been established and are now common across applications and operating systems. The simplest one is a tap, with variations in form of a long tap or double tap. Then, there are distinct gestures such as a swipe (or flick) as well as continuous ones such as drag or pinch. The pinch gesture is also an example of multi-touch input, where multiple fingers are used (Figure 2.2a). Lastly, some devices also allow for pressure-sensitive touch input and can therefore distinguish between a normal touch and a "force touch" [AMC17; GMG18].

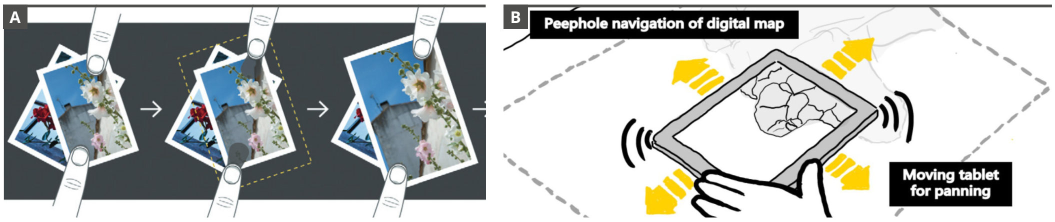


Fig. 2.2.: Modern devices can allow for interaction styles that imitate natural, real-world behaviors, e.g., by (a) supporting direct manipulation via touch input [WW11] or (b) utilizing spatial movements for peephole navigation [Räd+14].

While touch allows for a more direct interaction, the precision is reduced compared to mouse-based input [For+07]. This is due to the fact that the finger occludes a certain area while interacting, but also that there is an offset between a user's expected point of touch and the actual centroid of the finger [WR09; HB10]. This is typically referred to as the *fat-finger problem* and can be addressed by ensuring minimal sizes of touch areas or providing magnifier-like functionalities [SS91; AZ03; WW11; GMG18]. Another limitation is the lack of hover functionality or additional mouse keys. Although some research exists that allows to implement a hover-like technique on touch devices [Hin+16], this is not present in most commodity devices. In consequence, a touch user interface must provide alternative ways for accessing the same functionalities, for example, by using designated touch gestures such as long tap or double tap. Finally, it must also be considered that sometimes touch events can be triggered unintentionally [MC12; Sch+14].

Spatial Interaction

The spatial movements with or around a device can also be utilized as input modality. For mobile devices, this includes specific device movements such as tilting, rotating, or shaking that can be recognized via their internal sensors. These movements can be considered in a continuous way, e.g., tilting or moving for navigating in a certain direction [OO05; Räd+14] (Figure 2.2b), or as a distinct gesture, such as shaking for resetting a view [RLL11] or performing a throw gesture to transfer content [DB09]. Especially continuous movements have proven to be very effective. For example, Spindler et al. [Spi+14] showed that using 3D spatial movements for navigational tasks in map-like application was faster than the established navigation via pinch and drag for zooming and panning.

Besides the movements of the device itself, also the relative position and movement of people around a device can be used. One instance of such movements is *physical navigation*, where the user moves around in front of a (large) device in order to access certain content instead of using a virtual navigation. Here, it has been shown that the involved spatial-temporal memory of people can render physical navigation more efficient than virtual one [BNB07; BN07], although the effects depend on the actual setup, interface, and tasks [JH13; Liu+14; JH15]. A more elaborate way of utilizing people's movements is depicted as *proxemic interaction* [BMG10; Gre+11]. This is based on the anthropological concept of proxemics by Hall [Hal66], characterizing the interpersonal spatial relationships that can be observed between individuals. At its core, this theory describes that the smaller the distance gets the more intimate the relationship is. In the same sense, the spatial relationship can be used to trigger or provide certain functionalities, e.g., showing more detailed or personal content when moving closer to a display [Mar+12; Led+15]. Notably, this relationship can be considered between people and devices, but also between multiple devices.

2.2 Device Ecologies and Cross-Device Interactions

As an extension towards interacting with devices in a more natural way, the devices can be used in synergy by forming device ecologies and supporting specific interactions across devices. For example, this can allow to benefit from the different advantages of available devices while overcoming their shortcomings when used on their own. In 2019, Brudy et al. [Bru+19] published a cross-device taxonomy providing an extensive overview of the research on using multiple displays or devices in combination. In the following, I will provide a compact overview of this research area and its general research themes. Later, in Section 2.5, this discussion will be continued for work focused on data analysis across multiple devices. For a more elaborate reflection, I refer the reader to the work by Brudy et al. [Bru+19].

2.2.1 Notion of Device Ecologies

Already in 1945, Bush [Bus45] sketched the vision of a desk-bound computer system called MEMEX that features multiple displays providing a person with different information. Decades later, this vision developed further into the idea of using different, specialized computing devices in combination, coined as *ubiquitous computing* by Weiser [Wei91]. Nowadays, these visions are partly reality where people have various devices at their hands and research investigating ways of using these devices in parallel or allowing for seamlessly switching between them [Bru+19].

The research in this area took place under various names, such as multi-*display* or multi-*device* environments or as cross-device computing. Partly, this is reflected by the development from additional displays that are attached to a computer towards standalone devices such as smartphones, tablets, or digital whiteboards [Bru+19]. Still, the notion of multi-device environments (MDEs) implies a rather passive role of the devices that are at the disposal of the user and can be used if applicable for the current situation. However, the potential of such environments lies in the interplay of the various *smart* devices: they can detect their relations with each other, adapt to such situations, and can behave as one system. Therefore, to emphasize this vision, I follow the notion of *device ecologies* as described in the introductory chapter.

2.2.2 Classes of Device Pairings

Within this research space, most of the investigated device combinations can be grouped into three main types of pairings: large displays environments, large displays plus mobile devices, and mobile devices only. These groups also illustrate the dimensions of dynamics and scale as defined by Brudy et al. [Bru+19]: environments with multiple larger displays are often part of a fixed setup that cannot easily be changed and is bound to a specific room, while ecologies of mobile devices represent the idea of ad-hoc and in-the-wild setups. With respect to the scale dimension, mobile device ecologies are often used at a personal scale or in smaller groups at a social scale, while having environments with large display is suited for working on a social up to a public scale. This correlation of device characteristics to the cross-device design space dimensions also gets apparent when revisiting Figure 2.1.

Environments with Large Displays

Early on, researchers focused on the question if and how multiple desktop monitors can ease complex computer-assisted tasks [BNB07; And+11; Bra+13]. For example, Andrews et al. [AEN10] found positive effects for sensemaking tasks when users were provided with a 4x2 display setup (Figure 2.3a). However, similar to many wall-sized displays, such setups are behaving as one device with an increased display space. In contrast, multiple large displays have also been used in combination, for example, wall-mounted display plus a tabletop [RL04]. Here, the different devices took on clear roles where the vertical display was used to collect and provide an overview and the horizontal one was used for detailed research purposes. Finally, smart meeting rooms consisting of multiple vertical displays have been of interest as

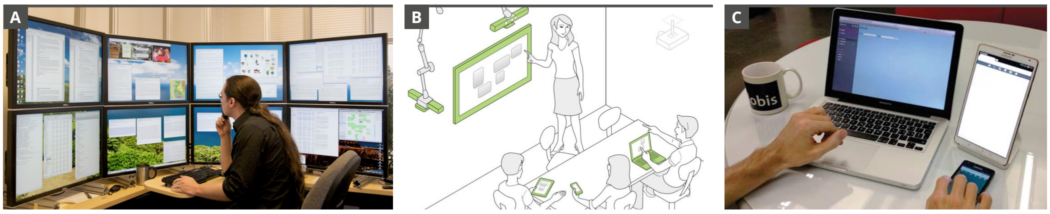


Fig. 2.3.: Different classes of multi-device environments exist: (a) large displays only, where these are often assembled as a multi-display setup [AEN10] forming a wall-like display unit; (b) large displays plus mobile devices, typically allowing remote interaction with the mobiles [Bra+11]; and (c) mobile devices only, allowing for, e.g., distributing interface components in dynamic and ad-hoc device ecologies [Hus+18].

well [Rad+14; FBW17]. For example, Fender et al. [FBW17] presented MEETALIVE, where meeting participants could place content on the different displays. Notably, personal devices where also part of the environment, making it an example for the next MDE group as well.

Large Displays plus Mobile Devices

Accompanying large displays with mobile devices (Figure 2.3b) can allow for improved interaction flows and collaboration or multi-user scenarios. More specifically, the bigger screen estate of large displays allows for overviewing more content at the same time when stepping back from the display [BNB07; Liu+14]. By incorporating mobile devices, users can be enabled to interact from this overview distance with the display, for example, via pointing interaction [Bra+11; Lan+16; Kat+16; AHD18] or general quick-access functionalities enabling remote control [Zad+14; Hor+18b]. Notably, the degree to which the mobile devices provide functionalities on their own can differ: the spatial movements of a smartwatch [Kat+16; AHD18] or a mobile phone [LKD19] can be used to control a cursor on the large display, while the devices themselves do not offer any further functionalities.

Beyond this usage as plain input devices, mobiles can serve as a personal toolbox in various ways: Chapuis et al. [CBF14] proposed to create and control multi-cursors on tablets, while Zadow et al. [Zad+14] placed an minimap-like overview of the whole information space on a smartphone, allowing to quickly change the excerpt shown on the wall display. Within this latter setup, the authors envisioned the mobile as a body-worn device on the arm, which could also host user-specific settings such as colors for used brush. Spindler et al. [SCD13] proposed similar settings via handheld displays around a tabletop display, and Brudy et al. [Bru+16] for the combination of smartwatch and tabletop in the context of content curation. Such personal toolboxes can also serve as clipboard-like content storage [Zad+14; Lan+16], optionally also allowing stand-alone content manipulation on the mobile device [McG+12; Chu+14; BE17]. Finally, the mobile device can also act as lens onto the content hosted on the large display, providing a more detailed or personalized presentation of the targeted area [Spi+13; Kis+17].

In general, the differentiation that mobile devices provide a personal view while the large display provides a shared view is a common theme in all these aforementioned examples. Such a role assignment can also be found in meeting scenarios, where participants take on a more passive role as they mostly follow the statements of one person. Still, having personal devices available can ease following a presentation shown on a large display and can allow for exploring details or making annotations without interrupting the speaker [FBW17; HKD16].

Mobile Device Ecologies

Mobile devices encapsulate one core idea of ubiquitous computing, the capability to perform computer-assisted work anywhere, anytime. In consequence, mobile devices and their combined use were extensively investigated in the context of dynamic and ad-hoc device ecologies. One main motivation is that the combination allows to exploit the strengths of the devices while overcoming their weaknesses, foremost the limited screen estate. For example, they can act as one continuous display space by placing them side-by-side [LK12] or as spatially-aware peepholes into a virtual canvas [Räd+14] (also shown earlier in Figure 2.2). Alternatively, as shown in Figure 2.3c, different views of the interface can be assigned to specific devices. Then, a device is getting a freely moveable view container [HW14; LHD18b; Hus+18].

In general, the roles and tasks a device can take on are more flexible and depend on the specific constellation. Considering the combination of two mobiles with different sizes (e.g., smartphone plus tablet), the roles can be similar as with a large display and a mobile, where the smaller device is used for probing [Woź+14], augmenting the interaction [Che+14], or offloading menus [LHD18b]. In larger ecologies or setups with similar-sized devices, the roles are less pre-defined and flexibly determined by the view assignment. For example, in collaborative settings, a device can be used either as a personal tool or as a shared overview device for the whole group [Bru+18]. While this flexibility can be beneficial, it also leads to the question how many devices are really needed and used by a person [Pla+17]: Is it easier or more convenient to switch between different views on one device compared to distributing them across devices and switching between devices? While Plank et al. [Pla+17] observed that people limit themselves to mostly one or two devices, this heavily depends on how well the interface supports the ecology. This level of support can be assessed based on how much effort a user has to spend on setting up the ecology, distributing content, and exchanging information across devices.

2.2.3 Cross-Device Interaction

In device ecologies, interactions across devices can be categorized by the phases in which they take place as well as the used modalities. For the phases, Brudy et al. [Bru+19] distinguished between configuration, content engagement, and disengagement. Within the content engagement phase, the main interaction takes place and comprises techniques to transfer content between devices or to interact and explore the distributed content. The other two phases are required to setup the ecology, this is, pairing and connecting devices as well as disconnecting them again. As already outlined for natural user interfaces in general, various modalities can be used for the interaction. In the following, these modalities will be detailed in the context of cross-device interaction by discussing existing examples using touch interaction and spatial arrangements.

Touch-based Cross-Device Interactions

Cross-device touch interaction can be grouped into three types: interactions that take place on one device, continuous interaction from one device to another one, and synchronous interactions on both devices. The latter two are often used for pairing the devices, e.g., by stroking from one device to another [Hin03] or by performing a pinch gesture across both devices [OT12]. As with these interactions the touch positions are known, the physical offset between the devices can be inferred and allows for adjusting the content positioning when the devices are used as continuous display space [OT12]. The interaction on one device comprises techniques where a target device is selected via an on-screen interface. This can be realized via traditional menus [BB06; HW14; Räd+15] or proxy representations [BB06; Chu+14; Räd+15], which allow users to define, e.g., to which device content should be transferred. When using proxies, transfer can often also be triggered through dragging or flicking content onto the proxy [Räd+15]. Notably, the positioning of these proxies can correspond to the actual physical arrangement, for example, the proxy of a device lying to the left is shown on the left border. And interesting exception are wrist-mounted devices, for which the relative position to a touch point on a other device is implicitly known [Zad+14; Hor+18b] and concepts similar to synchronous interaction can be applied.

Spatial Arrangements

As it is natural for people to arrange physical items around them as a way to organize and structure their current activity [Kir95], possible techniques exploiting the spatial arrangement of devices in MDEs were investigated various times. Instead of simply indicating the position of others devices, their changing proximity and orientation can be used as a mean of cross-device interaction. Alongside the previously discussed proxemic interactions [Mar+12; Led+15], also the concepts of f-formations and micro-mobility were considered [MHG12]. For example, when tilting a tablet towards the device of another person, the shown content can 'flow' onto this other device. In general, the positioning of devices can be interpreted in ranges [MHG12; LHD18b] (e.g., away, near, side-by-side) or more explicitly as for pointing interactions [Spi+10; Woź+14]. In all cases, precise tracking of devices remains challenging, particularly when an instrumentation of the environment is not possible [Bru+19; Hor+16]. Several investigations considered explicitly using less precise indicators for spatial arrangements that can be recognized by internal device sensors only, for example, by detecting vibrations patterns when bumping devices [Hin03] or slamming a table [Grø+20].

2.3 Visual Data Analysis

Visualization in itself is a huge research area, which continues to grow in importance with the developments towards a data-driven world. The biggest chunk of this research area is focused around how to design visualizations [Mun14; Spe01; War12] as well as how visualization can support people in multiple contexts, ranging from only communicating data-based aspects to fostering the actual analysis of data. For this thesis, the focus is on the latter part. Here, visualization is an interactive tool that allow to represent different aspects of a given data set and, consequently, to explore and analyze its characteristics. This type of data analysis is depicted as *visual data analysis* (VDA) and also is synonymous with *interactive* visual data analysis [TS20]. The interaction is key to such analyses [Tom15], as a fixed set of non-interactive visualizations will not allow to sufficiently derive the often complex and hidden insights from the data. Instead, it is essential to switch between different visualizations, configure them according to the current interests, and interact within them, e.g., by performing selection, zoom and pan, filter, or other operations.

In most instances of such visual data analyses, the specific insights are not known in advance but only discovered because of the analysis. This is also known as exploratory data analysis (EDA) [Tuk77] or exploratory visual analysis (EVA) [BH19]. Such EVAs can be characterized as a high-level goal with a varying precision that is iteratively refined through the subsequent interaction of an analyst [BH19]. This means, the reason for conducting such an analysis can range from exploring specific hypotheses to trying to find interesting insights in general. For that, various subtasks are applied by analysts. Consequently, both EVA and VDA are shaped by the analysts' goals and tasks as well as the interactive mechanisms in an analysis interface that supports these. These two aspects will be detailed in Subsection 2.3.1 and 2.3.2 respectively. The current state of the art for bringing VDA to multi-device environments or even device ecologies will be discussed later (Section 2.5).

At this point, it must also be acknowledged that VDA involves more aspects than the interaction with visualizations in an interface. Explicitly, the preparation of the raw data in order to become able to visualize it (also known as data wrangling [Kan+11]) is an important part as well. With the increasing amount of data it also becomes more and more relevant to apply computer-assisted operations to transform, normalize, or abstract data beforehand, which is then typically depicted as visual *analytics* [Kei+06; TC05]. Finally, at the end of the analysis process, the derived insights need to be communicated to certain stakeholders as reports or presentations. As one example of an analysis interface explicitly allowing such hand overs, I want to point the reader to the InsideInsights tool [Mat+19] that I have co-authored. However, these enclosing aspects of VDA are not in the scope of this thesis and, therefore, will not be discussed further. Instead, the focus is on the interactions that an analyst performs in and with different visualizations in order to reach their analysis goal.

2.3.1 Interaction Tasks during Analysis Sessions

Tominski and Schumann [TS20] name the role and the goal of the user as two main considerations for visual data analysis. These two define, or at least indicate, what requirements are put up for a VDA interface. The required steps to reach a certain goal can be characterized by specific (sub-)tasks that analysts can and will follow. For these tasks, multiple taxonomies and categorizations exist, for example, from Amar et al. [AES05], Yi et al. [YKS07], Heer and Shneiderman [HS12], Schulz et al. [Sch+13], and Brehmer and Munzner [BM13].

Important Interaction Tasks

Example	in [BM13]	in [YKS07]	in [HS12]
Select multiple marks via lasso	Select	Select, Connect	Select
Zoom and pan in a map	Navigate	Explore, Abstract / Elaborate	Navigate
Reorder columns and rows in a matrix	Arrange	Reconfigure	Sort, Organize
Alter the color scale of a visualization	Change	Encode	Visualize
Exclude outliers with extreme values	Filter	Filter	Filter
Aggregate daily values into monthly values	Aggregate	Encode, Abstract / Elaborate	Derive

Tab. 2.1.: Examples for common interaction tasks, following the typology by Brehmer and Munzner [BM13] (limited to their *manipulation* category of the *how* methods). Equivalents from the taxonomies by Yi et al. [YKS07] and Heer and Shneiderman [HS12] are associated accordingly.

Notably, there is not one sharp definition of visualization tasks, instead they can be considered on different levels and grouped in various ways. For example, comparison is considered a high-level task [Gle+11] that can be split up into smaller subtasks. Such lower level tasks can be selecting a mark, changing a sorting, or navigate within the visualization. These low level tasks can be further grouped, e.g., into tasks considered with either data and view specification, or view manipulation, or process and provenance [HS12]. However, the specific terms and definition of the tasks vary and might not always allow for a clear mapping of user interactions.

For this thesis, I will mainly focus on the task typology provided by Brehmer and Munzner [BM13], as it is the broadest one and is also incorporating many of the previously published work on tasks. In this typology, the authors distinguish between three questions that can characterize an abstract task: why, how, and what. The why typically aligns with the motivation or goal of an analyst, which can be instances of *consume*, *search*, or *query* information. Most of the tasks considered by Amar et al. [AES05] can be associated with this class (e.g., retrieve value, find extremum, find anomalies). The how considers the specific methods that are used within the interface, specifically mechanisms to *encode*, *manipulate*, or *introduce* data or views. The tasks listed in the taxonomies by Heer and Shneiderman [HS12] and Yi et al. [YKS07] can be allocated here. Finally, the what specifies the targeted data aspects which are used as the input and/or output of the abstract task.

As stated before, the focus of the work presented in this thesis is mostly on the interaction and interface design that supports analysts in reaching their goals.

Consequently, the *how* question and its associated methods are the most relevant ones. For the *manipulation* methods, Brehmer and Munzner [BM13] name select, navigate, arrange, change, filter, and aggregate as specific instances. These methods are further detailed in Table 2.1. The proposed concepts within this thesis will detail specific interaction and interface designs that allow to apply these methods in a certain way across multiple devices. The *introduce* methods include mechanisms such as annotate, import (of new data), derive (new data via computation), and record (to capture insights), however, are out of scope for the work presented later.

2.3.2 Analysis Interfaces and Interactive Mechanisms

In combination with the visualizations, the analysis interface has to provide sufficient mechanisms for supporting the aforementioned tasks as well as the visualizations itself. Examples of these interfaces are dashboards, which often come with rather interactive mechanisms [Sar+19], commercial tools such as TABLEAU DESKTOP [@Tab03] or MICROSOFT POWER BI [@Mic14] with a wide range of functionalities, or many tools developed within the research community (see, e.g., Tominski and Schumann [TS20] for some examples). These interfaces are then providing functionalities to create or configure the specific layout and visualizations [Che+21], preserve and access provenance or found insights [BH19; Mat+19], as well as to perform data-related operations such as filtering or aggregating. How these functionalities are provided can widely vary, ranging from context menus to specialized settings views, from algorithmic adaptations to novel interactive control mechanisms.

One common theme is the usage of multiple views (MV) in parallel [Che+21; Rob+19]. Having multiple visualizations available allows to maintain different perspectives onto the data at the same time. While these views can be arranged in various ways such as superimposition or nesting [JE12], typically juxtaposition is used. According to an analysis of MCV interfaces in visualization literature by Chen et al. [Che+21], MV interfaces typically consist of less than five views and feature rather simple column-based layouts. This indicates one challenge of MV interfaces: having many visualizations in parallel tends to introduce more visual clutter and may cause information overload. Another interesting aspect reported by Chen et al. [Che+21] is that for certain views common positions exist, e.g., setting panels are often positioned at the outer border of the interface.

While having multiple perspectives onto the data allows to discover different aspects, it can be challenging to relate them across the visualizations. An analyst can be supported in this by connecting and coordinating the different views, resulting in multiple *coordinated* views (MCV) interfaces [Rob07]. Specifically, the coordination simplifies relating views by synchronizing user interactions such as selections across them and is also depicted as an integral part of the interaction in the task taxonomies by Yi et al. [YKS07] and Heer and Shneiderman [HS12]. The coordination can happen in multiple ways [BRR] and is not limited to *linked brushing*, where selections are synchronized [Mun14] (also called *linked highlighting* [Mun14] or *linking and brushing* [Kei02]). For example, the coordination can also involve filter functionalities or simply providing indicators (e.g., for current viewports).

At the same time, providing coordination is not always straightforward. For example, a data point selected in a scatterplot might not be directly represented in a connected bar chart but only occurs as part of an aggregated bar. One solution is allowing to highlight subparts of marks, i.e., the corresponding part of a bar [Koy+18]. In general, it is also possible to provide elaborate configurations for the coordination of selections [Koy+18], however, this can increase the interaction costs for the analyst.

As already stated before, other interactive coordination mechanisms than linked brushing can be commonly found in MCV interfaces as well. To better illustrate this, consider the interactive exploration approaches *overview+detail*, *focus+context*, and *details on demand*. When zoomed in a visualization, it is often beneficial to provide the original non-zoomed view in addition, e.g., as a minimap, resulting in an overview+detail scheme [CKB09]. The coordination can then be indicating the sub-area that is shown in the detail view with a rectangle in the overview—possibly also for multiple detail views. Focus+context follows the same goal, providing a zoomed version while preserving the overall context, but by directly integrating the focus view in the context view by applying local distortions [CKB09; HF01]. This can be interpreted as a MCV arrangement with integrated views [JE12]. Finally, details on demand is a common technique for showing specific values, e.g., in a tooltip. As visualizations are often dense, it can be beneficial to offload this information into a separate view, then representing a primary-secondary view coordination [Rob+19].

2.4 Visualization Beyond the Desktop

Large parts of visualization research have been conducted within the context of desktop environments. Nevertheless, within the the last decade, the potential of modern devices was considered more extensively under the notion of visualization

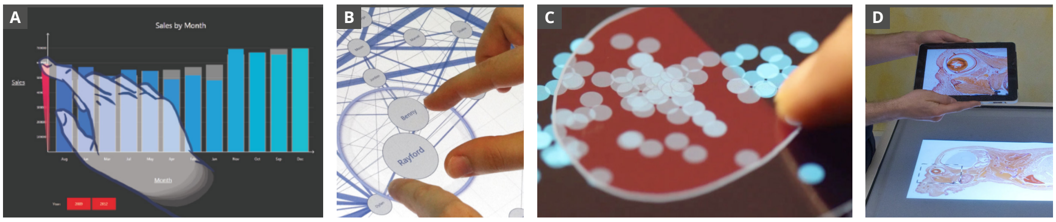


Fig. 2.4.: Modern devices allow and require novel interaction mechanisms, for example, (a) mapping configurations to visualization elements (here, axis sorting) [Dru+13], (b) manipulating lenses via touch [KRD16], (c) allowing alternative selection methods (here, lasso) [SS14], or (d) using spatial movements for interaction (here, zoom and pan) [Spi+13].

beyond the desktop [Rob+14] or beyond mouse and keyboard [Lee+12]. Similar to the general HCI space, the idea is to utilize these devices for a more natural way of interacting with data representations as well as allowing to conduct data analysis in various contexts (also referred to as *ubiquitous analytics* [EI13]). Specifically, the devices allow for novel interaction mechanisms, but also introduce new challenges when being used for displaying data on small or large screens. In the following, the most significant research in this area is presented. Again, the focus here is on visualization for display-equipped devices, excluding mixed reality setups or immersive analytics [Mar+18].

2.4.1 Novel Interaction Mechanisms for Visualization

Most visualizations rely on interactive mechanisms [Tom15], e.g., filtering, zooming and panning, or selecting elements. Consequently, a big body of work focused on how the input modalities of modern devices can be used for novel interaction mechanisms. With respect to touch input, interaction concepts have been investigated for a wide range of visualization techniques, among others, for bar charts [Dru+13; SA19], scatterplots [SS14; RK14], scatterplot matrices [Rie+20], star plots [Lan+15], parallel coordinate plots [RRF20], streamgraphs [BLC12], or node-link graphs [Fri12]. A common theme is that the ideas of direct manipulation [Shn83; HHN85] and fluid interaction [Elm+11] are followed, this is, that interaction is taking place directly on the visualization and with its elements. For example, in TouchViz by Drucker et al. [Dru+13], the sorting of an axis can be changed by swiping on it in the respective direction (Figure 2.4a), or Sadana and Stasko [SS14] as well as Kister et al. [KRD16] incorporate lenses that can be re-positioned or scaled by dragging or pinching them respectively (Figure 2.4b).

A particular relevant challenge for visualization is the fat-finger problem in the context of selecting marks. Marks are the graphical elements that represent the respective data points and are often rather small. As element size typically cannot be increased by default, other selection methods such as lasso selection (Figure 2.4c) or axis-based selections have been explored [SS14; SS16]. Alternatively, zoom and pen mechanisms can be used, either globally or via focus+context techniques such as lenses [SS14; KRD16]. Another way to improve precision is not relying on touch but incorporating pen-based input. Research on using pen is manifold, with Frisch et al. [FHD09], Walny et al. [Wal+12], Zgraggen et al. [ZZD14], Jo et al. [Jo+17], and Romat et al. [Rom+19] being only a few examples for work in this area. However, as for this thesis pen-based input is not in focus, a more thorough overview is not provided. Similarly, work on speech input [SSS20], tangible input [Jan14], or also multi-modal interaction for visualization exists [SS18; Sri+20; SLS20], but will not be detailed here.

For all on-surface input modalities, the performed interactions will cause some parts of the displayed visualization to be covered by the hand. By using spatial interaction, this can be avoided, while also potentially allowing for a more efficient interaction. As already mentioned before, Spindler et al. [Spi+14] found that using the movements of a mobile device for zoom and pan interaction can be more performant than the traditional zooming and panning via pinch and drag interactions. In another study, Spindler et al. [SMD12] found that especially vertical movements can be performed by people with a high precision and is therefore interesting for querying layered information spaces [Spi+10]. The used setup was similar to the one shown in Figure 2.4d. The advantages of spatial movements are also particularly interesting for 3D data visualizations, where it can notably simplify the navigation within these data sets [Büs+17]. For spatial interaction in the sense of proxemics, it has been shown that using the distance to a (large) display for 'zooming' and the position for 'panning' high-resolution information spaces is preferred by users as well as more performant [BN07; End+11; Ise+13; Jak+13]. These works will be presented in more detail within the subsection on large displays.

2.4.2 Going Small – Mobile Visualization

Besides the question how the interaction is changed by modern devices, it must also be considered how the visualization can be fit onto the device's screen—particularly, for mobile devices with rather small displays [Lee+21]. Research on this predates the smartphone era and started with the availability of PDA (personal digital assistant) devices [GF04; BR05; Chi06; HZ07]. These early works focused on how zoomable

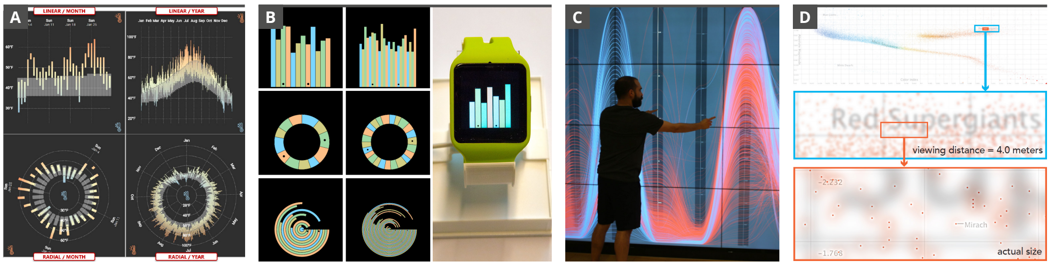


Fig. 2.5.: Visualizations on small or large devices face challenges regarding perception and interaction. Among other aspects, research has investigated (a) different layouts for specific data types (here, time series) [Bre+19a], (b) glanceability of different visualizations on smartwatches [Bla+19], (c) interaction mechanisms tackling reachability issues on large displays (here, axis reordering) [RRF20], or (d) redundant encoding with different granularity to exploit physical navigation.

interfaces can make visualizations explorable on small screen estates, as mentioned before, by using overview+detail and focus+context mechanisms [CKB09]. Later, the research question of how visualizations can specifically be designed for mobile devices became more relevant—and still is. This research involves both adapting the interaction (see above) and the design itself [Lee+18]. For example, Brehmer et al. [Bre+19a] investigated if linear or radial layouts are more suited for visualizing time series on mobile phones (Figure 2.5a), finding that linear layouts proofed to be faster, but no significant difference regarding error rate occurred. Sadana and Stasko [SS16] developed concepts how multiple coordinated views applications, i.e., interfaces with multiple visualizations that are linked, can be realized on tablets.

Similar explorations have been conducted for even smaller devices such as smartwatches. For example, for rectangular watches, Schiewe et al. [Sch+20] visualized stride patterns during running, while Chen [Che17] explored a specialized approach of displaying timelines around the border of the watch. Another display form factor was considered by Wenig et al. [Wen+15; WHS16] with Stripemaps, where they attached strap displays to backpacks in order to show hiking maps. In addition, perception aspects were also of interest, e.g., how quickly certain visualization types (Figure 2.5b) can be interpreted [Bla+19]. As discussed before, devices of this small form factor are often used for very brief interactions (cf. Figure 2.1). Within the visualization context, this is often referred to as glanceable visualization [Bla+21]. Here, Amini et al. [Ami+17] investigated the demands and tasks users face within typical applications such as activity tracking, whereas Gouveia et al. [Gou+16] have explored the design space of the involved glanceable feedback mechanisms concluding that the design of them can have a notable influence on a person's behavior. At the same time, on commodity smartwatches, data is typically represented using icons and text only and rarely with actual visualizations [Isl+20].

One important challenge for mobile visualization remains the highly diverse spectrum of available devices, ranging from smartwatches to mobile phones to tablets, with a huge variety of display sizes, aspect ratios, or even display shapes [Lee+21]. In consequence, a visualization designed for a tablet might not work on a smartphone. Following the notion of responsive web design [Mar11], the idea of designing *responsive visualizations* gained more traction within the last years [And18; HLL20]. Here, the visualizations are capable of adapting to different factors, such as display size, by design. This thesis contributes also to this research area, specifically with the work presented in Part II. This is motivated by the fact that device ecologies can consist of very different devices, while the content has to be display adequately on all of them—thus, strategies to adapt visualizations accordingly are required.

2.4.3 Going Large – Visualization on Large Displays

The use of large displays for visualization and visual analysis has been of research interest for a while, particular with respect to the increased screen estate and the potential for collaborative analysis [And+11]. More screen space means that more data can be displayed in parallel, rendering large displays especially suitable for visualization techniques with a high space demand, such as graphs [Kis+17; PBC17], matrices [Rie+20], maps [PBC16], parallel coordinate plots [RRF20], or simply many views in combination [LKD19]. While displaying more data is straightforward, interacting with visualization within an analysis session requires special consideration. Certain areas of a large display might become hard to reach or are simply further away. One way to address this is using special touch gestures such as a swipe for starting continuous scale mechanisms of views [Rie+20] or bringing axes closer for inspection [RRF20] (Figure 2.5c). Another option is resorting to additional mobile devices for distant interaction, which will be detailed later.

Both the concepts of physical navigation and proxemics have also been investigated for visualization on large displays [And+11; BI12]. One special type of visualization that supports and exploits physical navigation was coined as hybrid-image visualization by Isenberg et al. [Ise+13]. As shown in Figure 2.5d, the idea is to overlay two types of encodings, one which is granular and optimized for the high-resolution when standing close up, and one more coarse that is easily readable when standing apart from the display. The latter one simplifies recognizing aspects of the data on an overview level, while moving closer to display is providing instant access to more detailed information of the focused area of interest. Similar investigations have also been conducted by Endert et al. [End+11]. As an instance of proxemic interaction, both Kister et al. [Kis+15] and Badam et al. [Bad+16] investigated the use of lenses that are attached to the position and movements of a person. While in general providing a personalized view onto the data, the distance to a display was used to adjust parameters (e.g., zoom level), while the position in front of the display defined the position of the lens. In addition, hand gestures and the distance between users allowed triggering further functionalities, such as merging lenses.

In general, large displays can be beneficial for co-located collaborative scenarios [Ise+11] as typically enough space is provided to support multiple persons in parallel. Prouzeau et al. [PBC17] proposed special collaborative selection methods within graph visualization that simplified recognizing the connections between different sub-networks scattered across the display. The aforementioned work by Kister et al. [Kis+15] and Badam et al. [Bad+16] also supported collaboration, for example, via the stated possibility to merge lenses. Importantly, it must be acknowledged that outside of the visualization community a huge body of work on co-located collaboration on large displays exists as well, providing more concepts that could likely be adapted for data analysis too [Bra+13; JH14; Liu+17]. Within this broader research area, further challenges have also been identified, e.g., regarding territoriality [Bra+13; JH14], coordination costs [PBC17], and privacy [Bru+14].

2.5 Data Analysis across Devices

One specific instance of bringing together visualization beyond the desktop and VDA is investigating data analysis within multi-device environments. While the work presented in this thesis is not the first to investigate how multiple devices can be used in synergy for VDA, it makes a significant contribution to the overall limited body of work on this topic. Such research is particularly promising as VDA in general is only slowly establishing outside of the research community [TC05; BE18]. This indicates that there remains a need for further improving the mechanics of it, for example, by supporting a more natural and flexible way for the analysis. This section will detail the existing work as well as put it into relation to the work of this thesis.

2.5.1 Cross-Device Interaction Concepts for Data Analysis

One group of work focuses on proposing novel interaction concepts for cross-device interaction in the context of data analysis, typically targeting multi-user scenarios. Here, a common idea is using mobile devices as personalized views in addition to a shared large display. McGrath et al. [McG+12] considered tablets in combination

Fig. 2.6.: Most work on multi-device VDA has focused on device combinations with differently sized devices to provide personalized views to a reference visualization, e.g., with (a) GRASP [Kis+17], (b) SLEED [Zad+14], or (c) THADDEUS [Woź+14].

with a tabletop to allow a "branch-explore-merge" workflow, where analyst can start off a stand-alone exploration on the tablet and contribute their findings back to the tabletop later on. For the specific use case of text analysis, the VISPORTER system [Chu+14] supported freely moving content and views between different devices (tablets, tabletops, large displays) in order to allow for flexible collaboration patterns. While in VISPORTER [Chu+14] basic physical navigation aspects were considered too, Spindler et al. [Spi+10] greatly extended the usage of spatial navigation. Specifically, the authors used tangible displays which are tracked above a tabletop and mapped their 3D movements to various functionalities, e.g., up and down movements to the zoom level, flipping of displays to changing the encoding, or rotating them to adjust lens parameters (similar setup as shown in Figure 2.4d).

Similar explorations were conducted by Kister et al. [Kis+17], but for the combination of tablets and large display for graph visualization (Figure 2.6a). In their GRASP system, the use of physical navigation was less prominent as pointing interaction was supported as well, while the tablet served again as personalized view onto the dataset (e.g., allowing for applying local filter operations or changed encodings). This was also motivated by the authors aiming to understand the movements of analysts in front of large display, without forcing them into certain patterns. This motivation is shared with Langner et al. [LKD19], who investigated user behavior for a large-scale MCV application on a wall display. Here, participants could either interact by touch or by remote pointing using smartphones (which were solely used for this pointing). In contrast, Zadow et al. [Zad+14] proposed to use arm-mounted displays as personalized views in order to free the users from holding a device all the time as well as allowing to more easily access the personal device (Figure 2.6b). Although not explicitly for visualization, their SLEED system also incorporated concepts for extending the device from a personalized view to a personal toolbox, e.g., by providing clipboard functionalities. This idea was later picked up again and explicitly explored within the context of VDA for the combination of a smartwatch and large display, depicted as *"David meets Goliath"* [Hor+18b]. This work is presented in Chapter 6.

Interaction concepts have also been proposed for the combination of mobile devices only. Rädle et al. [Räd+14] presented HUDDLELAMP, a low-cost tracking approach based on an infra-red camera. Their work also included lightweight visualization examples, such as supporting a peephole navigation for a map and offloading a corresponding menu to a second device. With THADDEUS, Woźniak et al. [Woź+14] proposed the specific combination of a tablet plus a smartphone explicitly for data exploration. This is also one of the few works considering a single-user scenario. Interaction wise, the smartphone's spatial movements in relation to the tablet were used for, e.g., probing a visualization displayed on the tablet (Figure 2.6c). This idea of using the spatial relationship between multiple mobile devices was greatly extended later with VISTILES [LHD18b], which is the basis for Chapter 7 of this thesis. In contrast to THADDEUS, VISTILES is not limited to two specific devices and optimized interactions between them. Instead, the arrangement allows to flexibly structure the distributed analysis interface and control the applied coordination between views. Consequently, the resulting interplay between devices is one example for a true display ecology.

2.5.2 Technological Considerations

Another group of research work focused on the technological considerations required for enabling data analysis across multiple devices. Notably, almost all the proposed frameworks have in common that they are based on web technologies. The one exception is the Java-based MUNIN system [BFE15], which, however, mostly focused on the network architecture that clients have to follow. This indicates one main challenge in multi-device setups, this is, that all devices have to know and support one specific network approach [Bru+19] in order to exchange information. To avoid such a network setup, Badam and Elmqvist [BE17] proposed to encode the data of a visualization in a QR code that can be scanned by a mobile device. While this can be sufficient for transferring a visualization, exchanging further information on the user interaction or device state requires an additional channel.

In general, the advantage of web-based technology is that with the WebSocket standard low-latency communication is already built-in, as well as that almost all devices are capable of running web applications. Consequently, WebSockets were used as backend for more sophisticated frameworks. VISTRATES [Bad+19] provides a component-based system for visualizations that can be freely coordinated by configuring a data pipeline. The system builds upon two existing ones, WEBSTRATES [Klo+15] and CODESTRATES [Räd+17], and inherits their capabilities. WEBSTRATES enables synchronizing the complete DOM (Document Object Model) of the web application across multiple clients, i.e., allowing to synchronize the application's state, while CODESTRATES provides synchronized, notebook-like code editing facilities. The VISTRIBUTE system [Hor+19], which is presented in Chapter 8 and allows to automatically distribute visualization views across devices, is build upon VISTRATES. Recently, Schwab et al. [Sch+21] presented their VISCONNECT tool that also aims to synchronize analysis interfaces across multiple web-clients. In contrast to VISTRATES, the authors rely on synchronizing low-level input events (e.g., '*mousedown*' event) instead of the DOM, with the events then being 'replayed' on the other clients.

2.5.3 Evaluations & Generalized Considerations

Finally, some work also started to provide generalized considerations for conducting data analysis with multiple devices. As mentioned before, Plank et al. [Pla+17] conducted a study on the multi-tablet usage in collaborative visualization settings, finding that participants hesitated to use multiple devices in parallel. While this can partially by explained by a 'legacy bias' (i.e., people are not used to incorporating multiple devices), it has also to be acknowledged that the overall coordination and synergy between devices in this study was limited. A more sophisticated interface design for an actual display ecology could lead to different results. Similarly, Alsaiari et al. [AAJ20] set out to investigate exploration strategies in collaborative VDA, but mainly focused on the collaboration aspects of the analysis itself and less on how devices were used for that.

The broadest consideration of VDA in multi-device settings so far was provided by Chung et al. [Chu+15]. The authors focused on four main aspects crucial for supporting such settings: how displays are composed, i.e., if they form a continuous display or a distributed view, how information can be transferred between devices (e.g., synchronized interaction, only assigning/moving views), how the connection of information is represented, as well as how the display membership can be defined (pre-designed vs. ad-hoc). Many of these considerations are shared with this thesis, for example, the difference between dedicated device combinations (Part III) and fully dynamic device ecologies (Part IV). However, Chung et al. [Chu+15] remained at the level of considerations, which still need to be brought to life with specific concepts. Consequently, it serves as a valuable starting point in which the authors conclude that "further research needs to be conducted about how to support and balance between automatic support and manual adjustments to coordinate, transfer, and connect information" [Chu+15]. This thesis contributes to filling this gap.

2.6 Summary

This chapter provided an overview of the relevant background and related work for this thesis. Located in the intersection of HCI and visualization research, there is a large body of work that can be built upon. Particularly within HCI research concepts exist that can enable a more natural interaction, both on novel devices and when using them in combination. The visualization research provides many insights on how VDA can be characterized, e.g., what tasks need to be considered [BM13; YKS07; HS12], how visualizations can complement each other [Rob07; Che+21; Rob+19], or how visualizations can be designed in general [Mun09; Mun14]. In addition, there exists work that already aimed at merging these perspectives and explored ways to bring visualizations on novel devices or to use multiple devices in parallel for data analysis.

At the same time, it can also be concluded that multiple open challenges remain. First of all, it can be observed that the two main aspects—how to bring visualizations on novel devices and how to use multiple devices in parallel—has been mostly considered separately. However, particular for device ecologies it is an integral part that the visualizations must work on potentially very different devices. Further, the work that considered visualization in, e.g., a mobile context, often proposed only a solution optimized for one specific device type and one specific technique [Dru+13; SS14; Che17; Bre+19a; Rie+20]. In contrast, research aiming at providing more flexible visualization designs can be found under the notion of *responsive visualization*, but is still very limited. Part II of this thesis notably contributes to this research area by first providing an extended background on responsive visualization before examining specific concepts for multivariate graphs.

Second, the understanding of how device pairings can be utilized for VDA can still be extended. While existing work provides valuable insights [Spi+10; Woź+14; Kis+17], investigating further device combinations can help to build a more general understanding of possible multi-device mechanics. This is particularly true for understanding the different roles that devices can take on as well as the specific exploration workflows that can be supported. Further, these roles should be consider in the context of both single-user scenarios and collaborative scenarios. In Part III, such investigations are presented for two different dedicated device ensembles.

Lastly, the large goal remains to support VDA in fully dynamic device settings. Analysts should not be forced into specific device combinations, but have the flexibility to use the devices that are currently available or most suitable. Consequently, it must be investigated how exploration approaches and device roles can be enabled in a dynamic fashion without any pre-defined configurations. The idea that devices are capable of automatically reacting to each other in order to flexibly complement each other and generate synergies is what is at the core of the notion of device ecologies. Notably, this does not only concern the analysis itself, but also aspects such as how the user can be supported in setting up the interface. Consequently, this aims at striking the fine balance between automated support and manual adjustments by the analysts [Chu+15]. Research examining this challenge is presented in Part IV.

Part II

Adapting Visualizations for Heterogeneous Devices

3 Towards Responsive Visualization

One core takeaway from the previous chapter is that today's commodity devices are highly diverse, ranging from small mobile devices to large shared ones. Bringing content to all these devices in a similar quality requires adaptations that fit it to the respective device. However, in the context of data analysis, it must be acknowledged that most visualization techniques used in practice were originally designed with desktop displays in mind. These desktop-oriented techniques are often ill-suited for novel devices due to differences and restrictions in display size, aspect ratio, and interaction capabilities. In addition to these device factors, the usage style or environment can also differ drastically to known desktop environments, such as one-handed interaction with mobile devices, collaboratively working with other persons while standing in front of a large display, or simply working on-the-go in bright sunlight. The combination of these factors requires that data visualization designs must be *responsive* to device constraints and dynamic usage contexts.

Such responsive capabilities are particularly relevant for data analysis in a device ecology. For example, when moving a visualization from one device to another, it must be possible to display it properly on the other device. In addition, the visualization's appearance should be maintained as well, in order to allow a person to quickly re-orientate in the visualization and avoid losing focus. Consequently, providing the possibility to place a certain visualization on many different devices eliminates artificial restrictions and allows to flexibly arrange different perspectives in a device ecology (similar to analysis interfaces on large screens with free view placement). Thus, responsive visualization is an important foundation for multi-device data analyses, as presented later in this thesis.

This chapter provides an extended background on designing *responsive data visualization*. As the research in this area is still in its infancy, the following content is less a typical review of related work but foremost a systematic exploration of this topic based on a mixture of own discussions, existing visualization examples from practitioners, as well as concepts from research work. The resulting overview is a first step towards a more systematic understanding of responsive visualization, which can lead to the development of specific guidelines in the future. In the following, first, the relation of responsive visualization to existing concepts is discussed (Section 3.1). Then, in the main part, the responsiveness for visualization is characterized by first describing the impacting factors (Section 3.2) before an overview on possible adaptation strategies is given (Section 3.3).

Work presented in this chapter is based on the following book chapter (to be published):

Tom Horak, Wolfgang Aigner, Matthew Brehmer, Alark Joshi, Christian Tominski. “Responsive Visualization Design for Mobile Devices”. In: *Mobile Data Visualization*, AK Peters Visualization Series. CRC Press, 2021, 34 pages. To appear. Citation key: [Hor+21a].

Own Contribution: The content of the publication primarily emerged from intense discussions of all authors as well as further participants of a Dagstuhl seminar [Cho+19]; for the writing itself, I took responsibility for the section on impacting factors, and the overall coordination resulting in further contributions to all other parts.

Applied Changes: The content was notably compressed and partly restructured to better address this thesis' topic. The most significant changes were made in the general introduction and discussion, as well as in the strategies section by re-categorizing them and extending the ones concerned with interaction and layout. Further, the previous focus on mobile devices was broadened to also address other device classes.

3.1 Relation to Exisiting Concepts

The term *responsive* is here used to describe the capabilities of visualizations to automatically adapt to various changes during or between analysis sessions. As the idea of both providing adaptive content and optimizing visualizations for specific situations is not new, the relation of responsive visualization to existing concepts will be discussed in the following.

Research on *adaptive interfaces* started at the end of the last century and focused on how the overall interface can be adapted to better support users goals [BM88]. This could either be reached through customizations applied by the user or automated mechanisms provided by the system, which were based on a user model. Thus, adaptation was used to describe how to fit an interface to a user. With the development of mobile devices, this was then also considered as a strategy to optimize interfaces when screen space is limited [Bil+02; FM08], e.g., by placing frequently used items at the top of a menu. The main challenge for adaptive interfaces is accurately inferring the user model, i.e., their goals, expertise, or preferences.

Within the web developer community, the notion of *responsive web design* was coined later and focused on how interfaces can be adapted to various sizes. The goal was to provide a design framework which more easily allows to provide the same content across many contexts instead of developing multiple versions that were optimized for one specific target display. Specifically, Marcotte [Mar11] introduced three pillars of responsiveness in the context web design: (i) fluid grids allowing to specify sizes in percentages rather fixed pixel value as well as to change the grid layout based on interface constraints (e.g., switching from two-column to one-column layout), (ii) flexible images with percentage-based sizing and automatic creation, caching, and delivery of device-appropriate images, and (iii) media queries, which allow web apps to inspect the physical characteristics of the device. In contrast to adaptive interfaces, the goal of responsive web design is to adapt the interface to characteristics of the device on which the content is consumed and not primarily to the user.

In the context of visualization, these challenges of adapting the content has been considered under the term of *scalability* [TC05]. Besides aspects concerning the user (human scalability) or the device (display scalability), also the data (information scalability) and the possible encoding and representation strategies (visual scalability) play an important role. However, while these challenges are known, most visualization interfaces remain optimized for one specific context. For example, in news graphics teams usually multiple versions of a design for desktop-, tablet-, and phone-based consumption are produced instead of a single design [HLL20; Sam18].

In this context, embracing a responsive design mindset would be a way to keep development cost down while increasing the flexibility for users by allowing to consume data visualization on different devices and in various usage contexts.

Particularly with the increasing diversity of modern devices, the need for more flexible visualization design methods is apparent and is increasingly considered under the notion of *responsive visualization*. While inspired by responsive web design, visualization require strategies that go beyond the ones described in responsive web design. Specifically, visualizations are not images but complex and structured objects in themselves that come with a certain composition, data-dependency, and interactivity. For example, consider a simple bar chart: While the bars themselves might be easily scaled down, the same strategy cannot be applied for data or axes labels to the same extent. For these elements, adjustments such as repositioning, abbreviation, or partial omission might be more suitable. Thus, simply scaling down or changing the aspect ratio of a chart is not merely a technical question of size and resolution, but rather that the content and representation needs to be carefully adapted. Further, as interactivity often plays a vital role in visualization [Mun14; Tom15], the specific techniques for, e.g., navigation, selecting, or filtering, must also be adapted in addition to the visual representation.

Within the last few years, multiple research started to investigate aspects of responsive visualization: Early work by Andrews and colleagues [AS17; And18] provided a breakpoint-based approach for how simple charts can be made responsive, while Hoffswell et al. [HLL20] investigated how a visualization editor for news graphics designer can support them in creating responsive designs. In both, strategies for simplifying, rearranging, or removing specific parts of the visualization have been considered during the design phase. Wu et al. [Wu+21] proposed a framework that automatically *fixes* visualization that are insufficiently displayed on mobiles (e.g., overlapping labels, out-of-viewport placement, tiny font sizes), motivated by the fact that over 73 % of the considered web visualizations faced at least one issue when being displayed on mobiles due to missing responsive behavior.

In addition, many visualization techniques have been developed specifically for smartphone, tablet, or large displays (as outlined in Chapter 2 under 2.4). However, these existing techniques tend to focus on single contexts rather than on varied and dynamic usage contexts. Existing literature on responsive visualization design [Jeh14; Hin15; Kör16] is primarily concerned with implementation aspects and how responsiveness can be achieved using web technologies in particular. At the same time, a systematic examination from a conceptual perspective is either rare [Chi11; Fuc11] or predates the smartphone era [EFK95; Chi06; QFZ08].

In conclusion, the goal is to provide interactive visual data representations that can adapt their appearance and behavior to the current usage context—however, both the influencing factors and the possible adaptation strategies are manifold. While responsive visualization can build upon knowledge and concepts from responsive web design, scalability considerations, and adaptive interfaces, more extensive considerations are still required to effectively design visualizations for heterogenous devices and contexts. And, as more and more visualizations are incorporated in non-desktop environments, there is a clear need to conduct more systematic explorations.

3.2 Factors Impacting Visualization Design

Responsive visualizations should be able to adapt to a variety of factors. This includes firstly adapting to the device type and its capabilities, but can also be further extended to other factors: the specific usage (e.g., whether it is held in portrait or landscape mode), the environment (such as indoor or outdoor environments), the user and their visualization and interaction literacy, as well as the data itself (e.g., small vs. large data sets). In general, responsive behavior can be triggered both by explicit changes invoked by a user or by implicit changes sensed in the environment.

3.2.1 Device Factors

As already discussed in Chapter 2 (Subsection 2.1.1), today's devices span a large spectrum of different characteristics. The most prominent difference is the **display resolution and size**. It can range from only dozens of pixels up to ultra-high resolutions such as 8k resolution or even display walls consisting of multiple high-resolution panels. Notably, 'size' can be considered in two ways: as a virtual unit expressed in pixels and as a physical unit in centimeters. A combined measure of both aspects is the pixel density. While in responsive web design often only the virtual size is considered, in visualization design the pixel density is similar important, as it can indicate if marks or other elements are still legible. Further, the **aspect ratio** of a display can change significantly between devices, but also within a usage session when the device is rotated from landscape to portrait mode or vice versa. Finally, the **display shape** can also be non-rectangular, e.g., when using smartwatches. As many visualizations are sensitive with respect to size and aspect ratio, these factors can be limiting.

Within web design, the current approach to handle different display sizes is using breakpoints: hard-coded width values at which content is adapted in some way; between these breakpoints, the content is simply scaled to fit the width [Mar11]. For visualization design, this strategy might not be sufficient to avoid negative effects with respect to readability and graphical perception [Bre+19a; Wei+20]. In addition, designers should also consider that color support, contrast, or refresh rate of displays can also affect the perception. These are particularly important when considering devices with alternative display technologies such as e-ink [Hol+13; KHD20].

The **interaction modalities** supported by modern devices are also relevant for responsive visualization design. As the default, current smartphones support touchscreen input. The challenges and possibilities of touch input in the context of visualization were already discussed in Chapter 2, Subsection 2.4.1). In addition to those on-screen interactions, modern devices offer further input modalities such as hardware buttons (a camera button, a back button, or rotatable controls for watches), spatial interaction (such as tilting recognized through built-in sensors), or speech input. In consequence, a device might also be used with different input modalities, which then can require adaptations to the visualization design.

Finally, the devices' specific **hardware and software** can add limitations to a visualization interface as well. Supported software features defined by the operating system and, particularly for web visualizations, the browser can notably differ between devices. Similarly, connectivity and performance-related hardware components (such as the CPU, GPU, RAM, storage) can influence the speed of interaction as well as how much content can be loaded and rendered: a computer for a large display wall will likely come with more computation power than a smartwatch. At the same time, the large display system has also to handle on a notably larger drawing context, potentially partly mitigating the performance advantage. On mobile devices, these factors may also be in an interplay with battery life: high power consumption will drain the battery until a point at which the operating system will limit the performance to extend battery life.

3.2.2 Usage & Environmental Factors

The way how a device is used can notably differ (see Subsection 2.1.1 in the previous chapter). With desktop computers, the usage is consistent: one is facing the monitor and typically using mouse and keyboard. In contrast, mobile devices can be used in a variety of **postures** and ways: either hand held, placed on a table, or even body-worn as with smartwatches; while sitting, standing, or lying down; and with one or two hands [Bac+15; Ear+18a; Ear+18b]. On the other end of the spectrum, users will most likely stand and walk around in front of a large display. Thus, the viewing angle, orientation, and distance to the display can differ and, then, possibly affect readability of the visualized content.

Further, whether the device is fixed, lying flat on a surface, wrist mounted, or hand held will affect the **interaction style** and, potentially, the precision of a person: on a steady device, small marks can more precisely be selected than on a hand-held device, which can slightly move when interacting. The different ways of holding a device can also imply which parts of the interface or visualization are well reachable. For example, when holding and using the device with just one hand, content in the opposite display corner of the hand is typically harder to access [Ear+18b]. Switching from a one-handed usage to a two-handed usage often comes with rotating the device from landscape to portrait orientation, affecting the aspect ratio available for the visualization content. Reachability is also one major challenge on large displays. In general, the usage type is closely coupled to the device factors such as size, weight, or offered modalities [Ear+17; Zha+19].

Changes in their surrounding environment are beyond the control of a person using a mobile device. These changes can also impact interaction with visualization content directly as well as indirectly via changes in usage as responses to changes in the environment. For instance, consider that one's environment can be in motion when **on-the-go**, such as when traveling inside a bus, train, or car. Jostling around within a busy train car can reduce one's ability to read or interact with the visualization. Further, one might be transitioning to one-handed usage if the other hand is holding on to a safety bar during the ride. Beyond moving environments, crowded spaces also impose limitations such as the availability of voice-based input and auditory output as well as a usually reduced input precision and user attention.

For direct impacts of the surrounding on the visualization content consider the differences between **indoor and outdoor environments**. In outdoor environments, the lighting situation is dynamic and often problematic for visual perception, such as direct sunlight or dark surroundings in the early morning; both situations may require the display brightness to be adjusted. Variable and insufficient lighting affects the readability of visualizations, especially concerning hue and contrast perception [War12]. Other encoding channels might be used to compensate for these deficiencies, or an information display could be simplified for mobile and outdoor environments.

3.2.3 Data & Human Factors

As for visualization design in general, the **structure and size** of the data constrain which visualization techniques are suitable and appropriate [TS20]. In particular, when displaying large amounts of data, one must consider the viewer's ability to read, understand, and interact with the visualization. These challenges are further amplified with mobile devices and their often small screen sizes. For instance, visualization of many data points as individual marks can lead to rendering performance issues and a lagged interface on a mobile device. At the same time, selecting such marks can also become challenging due to the reduced precision with touch input [WR09] or when marks are overlapping.

Similarly, handling large **amounts of data** is also prone to impact performance. On large displays, loading and processing all data points and their attributes may require too much time, while on mobile devices transferring the whole dataset can lead to same result. In these cases, it is possible to subsequently load chunks or only aggregated data, with more detailed information only being loaded on demand (cf. progressive data analysis [Ang+18; Fek+19]). However, as for mobile devices the quality of the data connections can change during a session, loading additional information might not be possible in this case later on.

Connected to the data are also **human factors**, such as the general visualization literacy, subject matter knowledge, or motivation. For example, one may be more motivated to interact with their personal health or finance data via their mobile device than with impersonal data. It can therefore be helpful to think in terms of a person's goals or tasks (see Subsection 2.3.1 of Chapter 2). The visualization design must therefore provide the means to complete these tasks.

An elaborate visualization design may be impractical for some combinations of user and content, though the same design may be appropriate for other combinations or after an initial learning period has elapsed. While a deeper discussion of individual differences in visualization literacy, attention span, motivation, expertise are beyond the scope of this chapter, it is nevertheless helpful to consider these factors during the process of responsive visualization design.

3.3 Responsive Visualization Design Strategies

Many ways exist how a visualization can be adapted and be made responsive. As there is a vast combinatorial space of specific applications, data types, and visual representation techniques, the following list of strategies is not exhaustive, but an overview. Instances of these strategies can mostly be found in the work of visualization practitioners, as there is still little research literature devoted to the this subject. While there is an increased interest in workshops [Lee+18; Cho+19] and tutorials [WS15; @BS18] targeted at visualization researchers, investigations specifically targeting responsive visualization design remain rare. The already mentioned work by Hoffswell et al. [HLL20] as well as by Andrews and colleagues [AS17; And18] are the only notable exceptions by now.

The strategies considered in this section (see Figure 3.1) are grouped by the different components of a visualization that can be adapted or optimized: its *representation*, *supportive elements*, *encoding*, *interaction*, and, for multi-view instances, *layout*. Notably, these strategies are not exclusive approaches, but can be applied in combination. Further, the discussed strategies complement and expand upon those that Hoffswell et al. [HLL20] used to label a corpus of responsive news graphics, which included resizing, re-positioning, adding, modifying, and removing visualization elements such as axes, legends, marks, and labels. Finally, the majority of the considered examples focus on optimizing visualizations designed for desktop systems for mobile devices. However, most strategies are invertible and can also be applied when bringing, e.g., a desktop visualization to a display wall.

3.3.1 Optimizing the Representation

The first set of strategies aims to optimize the representation to the current context, i.e., maintaining the current visualization technique while fine-tuning it. These strategies are focused on the actual visualization and will lead to an adapted static version, i.e., not include any interactive aspects. Supportive elements, interactive mechanisms, or layout arrangements will be considered separately later on.

Fitting to the Viewport The most obvious strategy is scaling the content to fit the display space. In the simplest case, the content is simply *stretched* to match the devices viewport (Figure 3.1a). However, as this distorts the visualization it can lead to perceptual bias, particular for charts with continuous axes [HA06; TGH12]. Alternatively, a *uniform scaling* avoids such distortion at the costs of not making

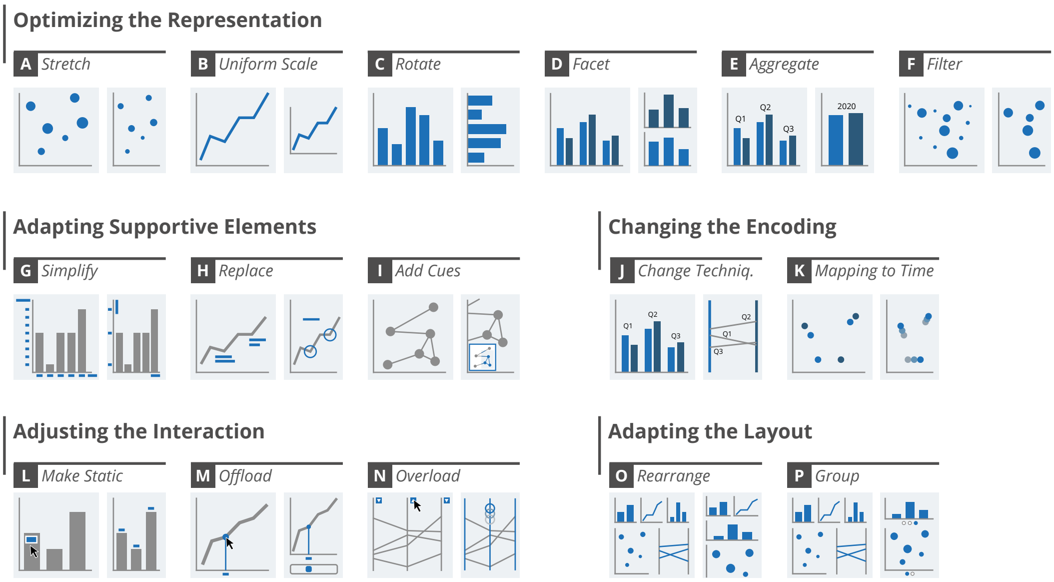


Fig. 3.1.: Multiple strategies exist how visualizations can behave responsively, here grouped by the visualization components that can be adapted: the representation (a–f), supportive elements (g–i), the encoding (j+k), the interaction (l–n), and the layout (o+p).

full use of the available screen estate (Figure 3.1b). Notably, this strategy can still be prone to incurring perceptual biases [Wei+20], thus leveraging an adaptive perception-based approach for resizing visualization might be warranted (see, e.g., VISIZER by Wu et al. [Wu+13]).

In all cases, the specific visualization technique and the involved elements must be considered, but also the display density of the device. Available mobile devices are now often featuring a similar resolution than larger laptop or desktop displays, thus content can be scaled down while still remaining readable. Vice versa, the resolution of large displays can be only slightly larger than of a desktop system but is displayed on physically larger display and viewed from increased distance. In addition, not all elements of a visualization can be treated equally. For example, if the visualization involves text, this text will become illegible when scaled down [WS15]. Here, one strategy is to apply different scaling functions to non-text content and text content [@Bre19]. Alternatively, it is also possible to abbreviate text labels in a systematic and consistent manner [SC17]. Further, it must be considered that the aspect ration can notably differ and scaling can become infeasible, e.g., when stretching content from wide format into a tall one. Consequently, scaling is mostly suited when differences in size and aspect ratio are not too big, e.g., when going from a tablet to a laptop display, or when a visualization features a matching aspect ratio [@Bre19].

Considering Orientation & Aspect Ratio For modern devices the shape and dominantly used orientation can differ from desktop environments, particularly when considering mobile devices. Thus, it is prudent to consider adapting the Visualization to such a changed aspect ratio. One approach is to simply *rotate* content that was otherwise designed for a different orientation. However, it should be carefully considered if text labels should also be rotated, as this can potentially negatively affect it legibility [@BS18]. As illustrated in Figure 3.1c, a simple example of this strategy is converting a vertical bar chart into a horizontal one [And18]. In some cases, rotating might not even be required, as certain visualization techniques are agnostic to the aspect ratio [@Bre19]. For example, for force-directed node-link graphs, the absolute spatial position of the nodes is not meaningful in itself, as only the relative position to other nodes is important. However, when switching between orientations, it can still be useful to consider how to adapt the technique: A node-link diagram could be rotated corresponding to the orientation change, so that the layout remains unchanged, or stretched to the new dimensions, so that the nodes are in the same relative position (e.g., top-right corner).

Further, some visualization types cannot be simply rotated without affecting its expressiveness. In particular map representations rely on a persons' ability to recognize familiar geographic features, which would be hampered by rotation, with the preferred orientation depending on the considered territory [HLL20]. Similarly, rotating a scatterplot would violate conventions of reading directions, as the origin of the coordinate system would no longer be placed in the bottom left and axes growing from right to left and bottom to top. In these situations, the stretch strategy might be more appropriate. For map visualizations, uniform scaling is more suited instead, where the resulting white space could be used for showing additional territory above and below the original excerpt.

Adjusting the Level of Detail Another approach is adjusting the amount of detail that is provided in the visualization. This can either be motivated by a smaller or larger screen estate or by human factors, such as different tasks or literacy being relevant in the current context. While Munzner [Mun14] distinguishes several ways to manipulate the level of detail, the focus here is on a subset of approaches that are particularly relevant to the topic of responsive visualization design. First, a designer could convert one chart into several charts by *faceting* on a dimension of the data, such as faceting a grouped bar chart into a series of bar charts, each displaying one of the group categories (Figure 3.1d). While the grouped bar chart might have required extensive scaling, it can become easier to place the faceted charts. On the downside, marks that were previously sharing axes can then no longer be compared directly, and the newly introduced faceted views may not be simultaneously visible on small screens, leading to the need of scrolling or paging.

Another strategy is to actually change the level of detail via *abstraction*. For example, this can include aggregating a monthly bar chart into a quarterly one (Figure 3.1e), or replacing adjacent nodes in a node-link chart with cluster points. For maps, this aggregation can be employed as a form of cartographic generalization [Mac04; WS15], such as aggregating counties into states and states into countries. Lastly, reclassification is a related concept, in which the number of categories or the number of quantitative bins is reduced and consolidated, such as the reclassification of elevation levels in maps. Other examples are reducing the number of bins in a histogram, or consolidating categories in a color legend.

Controlling the Amount of Data Finally, the number of elements to be displayed can also be reduced by removing marks in a systematic way via *filtering* and sampling. Hoffswell et al. [HLL20] document that this is often done for news graphics when converting a desktop graphic to a mobile one. As one example, they report that filter where linked to the display size by increasing the minimum threshold for shown marks on small displays. This is illustrated in Figure 3.1f. Another approach for reducing the amount of data is sampling based upon a statistical process. Whenever filtering or statistical sampling is employed, it is critical to inform the viewer that this has taken place as a responsive design measure, with some indication or ability to see what has been eliminated from the view.

3.3.2 Adapting Supportive Elements

Visualizations consist of several visual elements [Kim+19; HLL20], including at a minimum some data-bound marks and usually also some visual guides such as legends, axes, and grids. In some cases, there can also be several forms of annotation [Ren+17], such as additional text labels and attention-directing graphical cues (e.g., arrows, color highlights, shapes), as well as peripheral annotation (e.g., titles, captions, or other footnotes). Depending on their importance for the specific representation [HLL20], these guides and annotations are often *simplified* or *replaced* with simpler indicators when switching to smaller screens. Simplifying can involve shortening or abbreviating text, reducing number of marks, rotating elements, or removing them completely (Figure 3.1g), while replacing can involve exchanging text-based annotations with other types of highlights, such as circles or arrows (Figure 3.1h). In exploratory data analysis contexts, a systematic or rule-based approach to manipulating annotation and guides across devices can be applied. For example, with decreasing display size, axis labels of a line chart can be first rotated and then progressively be reduced at equal intervals, until they are removed completely [And18]. Finally, axes and titles are removed as well, leaving only a sparkline with annotated endpoint values.

For larger displays, the same strategies can be applied vice versa, i.e., adding additional labels or annotations. Such additions can also be applied to small-screen devices when combined with other strategies. For example, while it may be feasible in desktop environments to display a large and detailed chart in its entirety, on a mobile display a cropped and zoomed-in version of the visualization with an added minimap view of the entire chart [Chi06] as a form of overview+detail can be more effective (Figure 3.1i). An alternative to the minimap approach is to add graphical annotations that indicate the distance and relative orientation to areas of interest that are currently off-screen [BR03; GJ13].

3.3.3 Changing the Encoding

Until this point, each of the strategies only applied minor changes to the visual encoding. However, the sometimes drastic strategy of changing the fundamental design can be beneficial in certain situations. Practitioners provide multiple instances where such a change was applied, for example, by turning a bar chart into a slope chart (Figure 3.1j), or a population pyramid into a set of overlaid population curves [@Cam19]. However, at the same time this forces users to re-orientate themselves and relate the new presentation to the former one [HLL20]. This can be cumbersome, particularly when switching devices within one session. In consequence, it should be considered if a particular encoding choice should follow a mobile-first approach. However, an over-optimized approach could still lead to the need of switching the encoding: For example, for a multivariate network exploration, Eichmann et al. [Eic+20] proposed to use a sorted tabular representation of nodes instead of the common node-link visualization. While this has advantages for small screens, it is likely that on large screens the traditional node-link representation would be preferred.

In addition to replacing a visualization technique completely, dynamic and smooth adaptation methods for (local) zoom areas can be used. Following the *semantic zooming* concept, depending on the available display space or zoom level, the visual encoding and/or the level of detail can be changed smoothly, either globally for the current viewport or in a focus+context fashion. For a time series representation, the visualization could smoothly morph [RM15] from a line plot to a horizon chart [HKA09] when the height of the graph falls below a certain threshold, or clustered parts could be dissolved [Mor+14]. In fact, such adaptations are then interactive versions of the previously discussed strategies to adapt the level of detail or the amount of data.

Finally, as one special way to encode data, a time-dependent mapping can be applied, e.g., replaying changes for one dimension. This can be of particular interest as a responsive behavior for mobile interfaces (Figure 3.1k). This can allow to save space as, for example, not all elements or the complete chart has to be shown at once, but just the relevant information for the current time step. At the same time, such a timed visualization may require more processing power or some screen space for playback controls. Thus, a series of static snapshots or a looped animated "Data GIF" [@Gro17; Shu+21] may be a suitable compromise in a mobile viewing context. In general, such an approach can also be used for visualizations designed for comparison such as small multiple setups, which then would be transformed into a replay showing one plot after one. Interestingly, it has been shown that static small multiples on mobile devices can provide comparably accuracy to such animated approaches [Bre+20].

3.3.4 Adjusting the Interaction

As already examined in Chapter 2, interaction plays a crucial role for data analysis [Tom15]. While in communicative news graphics designers tend to incorporate no or only scroll-based interaction [@Gol17], this might not be sufficient for more elaborate data exploration tools. One challenge is the reduced target size as well as the lack of hover functionalities. In response, it is possible to incorporate fixed tooltips or tap-to-reveal tooltips [@BS18] instead of hover tooltips (Figure 3.1l). To avoid covering the visualization while interacting, functionalities can also be offloaded around the chart (Figure 3.1m). For example, D'Souza et al. [DSo+17] proposed to provide a slider below a chart that reveals additional guides, annotations, or details-on-demand when skimming through it. In addition, visualization elements can also be overloaded and be used for triggering certain functionalities. For example, as shown in Figure 3.1n, the sorting of an axis could be changed by swiping on it in the respective direction [Dru+13]. This can be particular helpful to avoid extensive menus that can be complicated to navigate on mobiles.

Particularly for explorative data exploration, selecting marks and navigating the overall visualization remain highly important. Due to the smaller screen estate and varied input modality on mobile devices, selection of marks can prove to be difficult, especially for charts with small marks such as scatterplots. One approach to tackle this is increasing the interactive area of a mark by a few pixels beyond its graphical representation. As for dense visualizations this approach might not be sufficient, an invisible Voronoi tessellation can be used as a more sophisticated approach to define the interactive areas [@Bre15]. For some visualizations, it will, however, still be required to further navigate the chart via zoom and pan. In general, scrolling, panning, and zooming work well across platforms, data types, and visual encodings. It might be required to indicate that this interaction is possible. Here, by choosing specific initial zoom and pan positions can provide cues that such navigation is supported [@Bre19].

Finally, a changed input style can also result in a completely changed interaction concept. This is particular true for non-pointer-based interaction modalities such as speech or spatial interaction. For example, when using a mobile device that is capable of tracking its position, its movement in space could be used to navigate a chart instead of the typical pinch-and-drag interaction [Spi+14]. Similarly, multimodal interaction can allow for reducing the number of menus and interface components that are required for conducting complex data analysis tasks [KR18; SLS20].

3.3.5 Adapting View Arrangements

Until this point, the discussion has largely focused on adapting a single visualization. However, analysis interfaces often consist of multiple visualization views that are considered in parallel, including dashboards [Sar+19], documents with combined text and visualizations [Mat+19], and small multiple designs.

In the simplest case, a single row of content arranged horizontally for viewing from a desktop can be stacked vertically [@BS18; @Bre19]. However, consider the more typical case in which content can be seen as occupying a two-dimensional grid, such as in a small multiples design. Here, adaptive grid layout rules have to be defined that anticipate different screen sizes and aspect ratios [@BS18; Hin15]; perhaps a grid of six columns is ideal for a desktop display while a grid of two columns is ideal for a mobile display (Figure 3.1o). The arrangement is typically decided by following a breakpoint-based system, however, the layout could also be decided using sophisticated algorithms such as constrained layout solvers [Jia+19]. A similar approach is also used in the VISTRIBUTE system, which is the topic of Chapter 8. As for all strategies, optimizing the space usage is not the sole goal of responsive measures. In the context of the layout, particularly the order in which views appear—or if they still appear at all—can also be based on other factors, such as, which views are important for the analysts' tasks prominent in the current context.

Independently of the strategy for arranging content, it is likely that content that is displayed simultaneously on a desktop display will cascade off-screen when viewing from a mobile display (Figure 3.1o). Unfortunately, information can no longer be compared at a glance, and viewers' comparisons must rely upon memory. Furthermore, interactive brushing and linking across views is not as useful when the linked views are off-screen, unless there is some visual prompt that directs viewers to that off-screen content [BR03; GJ13]. Despite this drawback, vertical scrolling is commonplace, fluid, and fast [@Cot19]. Scrolling a stacked series of charts interleaved with other content (such as text or images) is often preferable to alternative off-screen layouts, such as swiping or tapping page advance through a series of charts (Figure 3.1p), as these interactions are less common than scrolling and may not be discoverable by viewers [@Gol17].

3.4 Discussion and Summary

For this chapter, it can be concluded that responsive visualization design involves challenges beyond those encountered in web design: First, visualization content is more sensitive to changes in size, aspect ratio, and interaction modalities, thus, this content cannot be simply scaled down or up to fit the screen width. Second, this sensitivity also means that it is not enough to consider the display-related factors in isolation (as it is typical in responsive web design), but also the usage context, environment of the viewer, as well as their tasks. However, the strategies discussed so far provide only a first overview on how such a responsiveness can be achieved. It is thus an exploration, for which it is too early to come to a conclusive taxonomy or even set of guidelines yet.

Role of Responsiveness The investigation presented here of responsive visualization already highlighted multiple important aspects. First of all, it becomes clear that responsiveness should be considered as early as possible and become an integral part of visualization design. While it is possible to fix some of the most prominent issues during run time [Wu+21], the quality will not reach the level as when the factors and the resulting adaptations have been carefully designed beforehand. However, looking at existing visualization design models, such as the nested model [Mun09], the visualization design triangle [MA14], or the five design sheets method [RHR16], it becomes apparent that these do not explicitly consider the aspects of responsiveness yet. Ideally, best practices for responsiveness should also be directly supported within authoring tools, alongside suitable tools for testing them. Hoffswell et al. [HLL20] provide an interesting first example of such a tool.

Further, an often overseen aspect is that responsiveness requires a rethinking of not only basic charts in isolation, but also the combination of multiple representations in more complex visualization interfaces. Visualizations should adapt at similar points in similar ways, providing consistency across the complete interface [QH18]. Also, responsive design is an important building block for data analysis in device ecologies. Consequently, the later chapters of this thesis will repeatedly touch on this topic again. In this context, it should also be emphasized that responsive visualizations should remain agnostic to specific devices as much as possible and not be over-optimized for a few specific devices and contexts, such as smartphones held in portrait mode. Already now device diversity is large and will continue to increase. For example, the device landscape will soon include foldable devices as the Samsung Galaxy Fold [@Sam19] or Microsoft Surface Duo [@Mic20], novel wearable device like watches with interactive strap displays as presented in Chapter 5, or even holographic 'displays' as provided by head-mounted AR glasses such as the Microsoft HoloLens [@Mic16]. In such environments, especially the contextual factors become more prominent as analysis is performed in diverse situations.

Towards Guidelines for Responsive Visualization As already stated before, the investigations conducted here do not provide a general framework for responsive visualization yet, i.e., general guidelines for how to adapt different visualization techniques in cross-device analysis interfaces. Such a generic adoption of responsive design has to be substantiated by investigating, designing, and testing specific examples of responsive visualization. In this context, it could be interesting to either focus on wide-spread visualization techniques such us bar charts, line charts, or scatter plots, in combination with selected explorations into particular challenging visualization techniques. In addition, responsiveness should not be considered in isolation, but as a concept that can be applied within specific representation approaches that incorporate embedded visualizations. This is done in the next chapter of this thesis, by considering interactive visualization solutions for multivariate graphs. With a combination of different investigations types in the future, it can eventually become possible to develop representative guidelines for responsive visualization.

4 Responsive Graph Visualization with Local Focus Regions

With the generalized considerations and strategies from the previous chapter on responsiveness in mind, the focus is now on providing visualization approaches for the specific data type of multivariate graphs. These data sets encode relational aspects alongside multivariate data aspects for both the entities and relations, which leads to a further increased complexity. In the context of this thesis, this is a particularly interesting use case where a visualization solution must provide rich exploration means while still maintaining a relatively compact format.

Specifically, the complexity of multivariate graphs arises from the underlying data structure, which comprises nodes, edges, and multivariate data attributes. An example would be a power grid, where power plants (the nodes) are characterized by quantitative attributes such as maximum capacity or current load. Power lines (the edges) between plants can be characterized by attributes such as throughput or length. In general, the challenge of such graphs is visualizing the two main data aspects, multivariate attributes and graph structure, at the same time. A solution visualizing a multivariate graph, therefore, may not only consist of one visualization technique but multiple, for example, separate techniques representing the structural and multivariate aspects. Being able to provide a solution that behaves responsively and provides access to all data aspects will be beneficial. Further, these insights can also help to inform the design of responsive visualization in general.

What data features must such a solution convey? Typical tasks on multivariate graphs include gaining an overview of the graph structure (what is connected to what?), assessing the overall similarity of nodes (which power plants are alike?), studying the distribution of attribute values (what are the characteristics of plants in a sub-grid?), comparing nodes in detail (which plant produces less carbon dioxide?), and finding relations between attributes and the graph structure (are similar plants interconnected?) [Lee+06; PPS14]. In addition to these analysis-oriented objectives, it is becoming increasingly important to be able to edit or wrangle data [Bau06; Kan+11]. This can be necessary to correct erroneous data values (implausible power line throughput), and also to carry out *what-if* analyses [Spe01] to test how data characteristics change with certain values in the data (would there be sufficient energy when reducing the capacity of some power plants?). Solving the outlined tasks typically requires an interplay of several visual representations [KPW14; Nob+19].

In this chapter, solutions are provided that support these explorations, while being designed with responsiveness in mind. First, an extended background on existing literature on multivariate graph visualization is provided. Then, general responsive measures for node-link visualizations with a focus on a multi-view approach are discussed (Section 4.2). The main part of this chapter is considered with the *Responsive Matrix Cells* (RMCs) technique, which represents multivariate graphs in a matrix visualization in combination with a novel focus+context approach that integrates detail visualizations in the matrix. Specifically, the goal is to exploit the predictable layout of the matrix to support extensive explorations within its fixed dimensions and supporting to go from the overview given by the matrix to details via local focus regions. Here, the design of RMCs is described first (Section 4.3), before discussing the interaction aspects (4.4) and implementation (4.5).

Parts of the discussion of responsive node-link layouts previously appeared in the following publication:

Tom Horak, Ricardo Langner, Raimund Dachselt. "Towards Visualizing and Exploring Multivariate Networks on Mobile Devices". In: *Companion Proceedings of the ACM Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2020, pages 5–8. Citation key: [HLD20].

Own Contribution: I was the major contributor for the complete publication, with the proposed concepts and ideas being discussed with the co-authors in the process.

Applied Changes: The content was significantly changed for this chapter and used in Section 4.2. This includes adding a clearer discussion of attribute-based layouts and better relating the interface concepts to the here stated requirements.

The main parts of this chapter covering the research on Responsive Matrix Cells have been published in:

Tom Horak*, Philip Berger*, Heidrun Schumann, Raimund Dachselt, Christian Tominski. "Responsive Matrix Cells: A Focus+Context Approach for Exploring and Editing Multivariate Graphs". In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pages 1644–1654. **The first two authors contributed equally.* Citation Key: [Hor+21b].

Own Contribution: The publication was a joint effort by the authors with the first two authors generally contributing equally. The parts with me as major contributor were the designs for the embedded visualizations as well the interaction concepts. However, all authors have at least a partial contribution in all parts of the original publication.

Applied Changes: Major parts were re-used for this chapter and partially adapted. Specifically, the background section was extended with work on node-link charts. The requirements were moved up and slightly extended by considering a multiple-view approach as alternative. Further, the provided walkthrough of the original publication was left out here. Finally, parts of the introduction and the discussion were re-used as well, however, the introduction and discussion provided here differ significantly from the original one by focusing on the thesis' topics.

4.1 Background: Multivariate Graph Visualization

Multivariate graphs have been investigated extensively in the visualization community, albeit almost exclusively focused on desktop environments. In the following, this body of work is grouped into three parts: (i) general approaches for visualizing multivariate graphs; (ii) presentation techniques that facilitate interactively exploring these representations, as well as (iii) general interaction and editing techniques for graphs.

4.1.1 Multivariate Graph Visualization

Several approaches exist for visualizing multivariate graphs, with the majority of them being based on node-link diagrams or adjacency matrix visualizations [KPW14; Nob+19]. While node-link diagrams can be considered the default visualization for graphs, their layout can quickly get confusing, particularly when encoding additional data attributes. In contrast, adjacency matrix visualizations feature a clear and predictable layout suitable for providing an overview, even for dense graphs. However, matrices are not as intuitive as node-link charts and can require a higher mental load to map the visible patterns in the matrix to structural characteristics.

In order to make all aspects of a multivariate graph visually accessible, both node-link diagrams and adjacency matrices must be extended. This can be done by incorporating additional views [Ker+17; NSL19], embedding additional visual encodings [MB19; EW14], or laying out the graph based on its attributes [Wat06; WT08]. While incorporating additional views makes it easier to encode more information, such solutions introduce a discontinuity between identifying regions of interest in one view and analyzing the actual details in another view. As a result, relating information across views can impose a higher mental demand to the analyst. Embedding additional visual encodings and varying the layout can avoid this, but it is usually only possible to visualize attributes in an abstract or aggregated form. Therefore, existing solutions often favor one data aspect over another [Nob+19] or are geared towards specific analysis tasks [PPS14].

Node-link representations

Node-link representations are typically extended with on-edge or on-node encodings for showing multivariate aspects [Nob+19]. For both, this can reach from simply using visual attributes such as shape, line width, or color up to embedding or overlaying small charts [Nob+20]. However, node sizes must typically remain rather small in order to be able displaying all of them, while edges can have varying lengths and orientations. Both limits what and how charts can be embedded in the representation. Another approach to encode multivariate aspects is using attribute-based layouts [Nob+19]. For example, a categorical attribute could be used to place nodes in corresponding regions, or node positions can be defined by two attributes (similar to a scatter plot). However, this can quickly cause prevalent node overlaps as well as structural aspects to be harder to recognize than in, e.g., force-directed layouts (there, node positions are calculated based on the interplay of attraction caused by edges and a general repellence by nodes).

Importantly, node-link representations are well suited for supporting a wide range of structure-related tasks, such as, recognizing k-neighbors, path following, or clusters [GFC05; Nob+20; OJK19]. At the same time, node-link diagrams are prone to scalability issues that can quickly lead to clutter representations. Here, interactive mechanisms such as zooming and panning in combination with layout mechanisms such as clustering or edge bundling can help to maintain a (local) readability.

Matrix representations

Adjacency matrices encode the presence of edges (or edge weights) in a tabular layout and are, similar to node-link representations, designed to facilitate visual graph analysis [GFC05; OJK19]. Typical techniques for representing the edge attributes are color-coding and also small glyph-like visualizations placed directly in the matrix cells [Elm+08a; YEL10]. As matrices do not explicitly represent the graph nodes, additional means are required to visualize node attributes. Prior research has assessed that a juxtaposed attribute table is a suitable solution [BST19; Nob+20]. An alternative is to calculate a pairwise attribute-based similarity measure for nodes and visualize it in one half of the matrix (divided by the diagonal), while the other half still encodes the edges [BST19]. This creates an overview of structural and attribute-based characteristics, enabling users to see, for example, whether nodes being similar with respect to their attributes are also connected by edges.

A disadvantage of matrices is their quadratic space complexity, which makes visualizing larger graphs demanding [AKK02]. Moreover, matrices are not very well suited for path-related tasks [Nob+19; Nob+20; Won+13]. A promising approach to mitigate these issues is to combine matrix and node-link representations, as in hierarchical graph maps [AKK02] or NodeTrix [HFM07]. Here, parts of the matrix are replaced with a node-link representation or vice versa for showing regions of interest in an alternative way. Such local replacements and adaptations within the display are also part of general presentation techniques, as discussed next.

4.1.2 Presentation Techniques

Temporary local adaptations of a visual representation can help reveal details for regions of interest while the global context is preserved. Focus+context techniques often apply a local zoom effect while maintaining the overall visualization dimensions. Examples of focus+context techniques are bifocal displays [ATS82], fisheye views [Fur86; RJS01], rubber-sheet navigation [Sar+93], the table lens [RC94], the date lens [Bed+04], or Mélange [Elm+08b]. Focus+context is not limited to geometrical scaling. Semantic zooming can dynamically alter the layout or the very encoding of the focused parts of a visualization [PF93]. Examples would be to change the type of chart embedded into the cells of a table lens [McL+08] or to show meta-nodes for clusters when zoomed out and to automatically expand the clusters to reveal their affiliated nodes when zooming in [AHK06; Shi+09].

Similar to focus+context techniques, magic lenses are lightweight tools that fluidly integrate a transient lens effect into the visualization [Tom+16; KRD16; Kis+17]. In the context of graph visualization, lenses can, e.g., reduce clutter by filtering edges or generate local neighborhood overviews by adapting the layout [TAS09]. Similar to lenses, *in situ visualization* allows users to interactively mark a region in a base visualization for which a different nested visualization is shown [HSS11].

When considering the nesting of views to provide alternative representations locally on demand, the embedded visualizations have to face specific layout restrictions [JE12]. For example, when embedding charts in table cells as in LiveRAC [McL+08] or glyphs in a matrix as in ZAME [Elm+08a] or TimeCells [YEL10], the available space is severely limited. Depending on the application and the user's tasks, different visual encodings for such embedded or micro visualizations are possible [BW17; Fuc+17; Tuf06].

In the context of focus+context and semantic zooming, space constraints are more relaxed because users can freely define and change the zoom level and the dimensions of the focus region. This makes it possible to add details to the visualization (e.g., labels, axes, or guides) or to switch to increasingly detailed visualization metaphors [Mat+02; McL+08]. In other words, such embedded visualizations should behave responsively, too. Besides constraints coming from the layout (i.e., 'screen space'), data density, and interaction-related aspects, the responsiveness here has also to take into account human factors in the form of the tasks of a person (e.g., looking up values, comparing nodes, editing attributes).

4.1.3 Interacting & Editing in Graph Visualization

In general, interaction plays an important role for exploring multivariate graphs [Wyb+14]. Literature suggests that interaction can take place at different levels, including view-level interactions (e.g., brushing and linking), visual-structure interactions (e.g., selections), and data-level interactions (e.g., inserting or deleting edges). Making selections in graphs or filtering nodes and edges are fundamental operations [MJ09; TAS09]. A key interaction for matrix visualizations would be to re-order the rows and columns to reveal different pattern types [PDF14; Beh+16].

Interaction in graph visualization is not limited to mere selections or adjustments of the visual representation. Interaction is also relevant in the interplay of graph exploration [Lee+06; PPS14] and graph editing [Gla+15a]. Following Baudel's direct manipulation[1] principle [Bau06], previous work has proposed to edit node attributes by moving the nodes in a 2D-coordinate system with an overlaid node-link diagram [Eic+16]. For editing a graph's structure, specialized lens tools can be employed [Gla+14]. Specifically for matrix visualizations, interactive editing approaches focus around adding or removing edges by (un)marking the corresponding matrix cells [Gla+15b; Kis+17]. More elaborate and integrated approaches, for example, for editing specific attributes of both nodes and edges, remain under-explored so far.

4.1.4 Open Challenges & Requirements

Overall, it remains challenging to visually explore and also edit multivariate graphs. To balance unwanted attention switches and increased screen space demands, focus+context and semantic zooming have already been applied to node-link presentations [AHK06; Shi+09; Tom+16], data tables [RC94; McL+08] and matrices [AKK02; Elm+08a; YEL10]. However, the existing techniques are typically tailored to showing one specific data aspect of their respective data set. Also, these approaches were designed with desktop systems in mind, and are not explicitly geared towards a responsive behavior. Therefore, one goal is to provide flexible visualization approach that can show multivariate attributes as well as structural

[1] Baudel's direct manipulation regards the direct editing of data values and is not to be mistaken for the classic notion of direct manipulation[Shn83].

aspects of graphs on demand, while being designed with responsiveness in mind. Finally, editing is often not an integral part of existing approaches but considered as as stand-alone task.

With these aspects in mind—as well as the general characteristics of multivariate graphs and the associated tasks [KPW14]—the following application-agnostic **requirements** for a visualization solution can be derived.

R1: Provide overview. The approach must provide an overview of both graph structure and multivariate attributes, enabling analysts to spot general patterns (e.g., cliques or clusters), potential outliers, and possible relations between structure and attributes (e.g., similar nodes are connected).

R2: Allow access to details. For selected regions of interest, it must be possible to access details to refine and complement the findings made with the overview. This includes identifying specific attribute values and comparing nodes or edges for concrete differences.

R3: Enable direct editing. Editing should be possible directly in the visualization to allow users to quickly correct erroneous data or test *what-if* scenarios while observing the resulting changes on the fly.

These requirements are concerned with *what* information and exploration facilities a multivariate graph visualization must convey. On top of that, an additional requirement can be defined that is centered on *how* R1–R3 can be achieved. Notably, this *how* can be addressed in two ways, either as an integrated approach or a multi-view approach:

R4.1: Provide information in separated contexts. The aspects of multivariate graphs can be presented in separated, but coordinated views next to each other. With providing multiple perspectives, the goal is to allow focusing on specific aspects while providing suitable workflows for switching and connecting the respective views.

R4.2: Strive for a fully integrated approach. All aspects inherent in multivariate graphs should be shown in an integrated visualization that supports data exploration and data editing. The integrated approach is to support smooth dynamic workflows and reduce inconvenient attention switches between different tools.

While R4.1 is built on concepts such as multiple coordinated views [Rob07], R4.2 aims to utilize the known advantages of integrating focus within context [CKB09], the visual information seeking mantra [CKB09], and direct editing [Bau06].

4.2 Responsiveness for Node-link Representations

While node-link representations are one of the most common ways to visualize graph structures, they can become quickly chaotic with a large number of nodes or edges. As nodes start to overlap and edges frequently intersect with others, the resulting visual clutter can quickly lead to so called hairballs [Jan+14]. This scalability issue is well known and one of largest challenges for network visualization—even in desktop environments. Consequently, when considering these graphs within smaller display spaces, this challenge is further amplified. At the same time, existing research provides approaches for how the problems can at least partially be addressed, e.g., via different layout algorithms, edge bundling approaches, or clustering mechanisms. In the following, it is discussed how these mechanisms, among others, can work together in order to provide responsive node-link visualizations.

4.2.1 Layout Approaches

As outlined before, the chosen layout algorithm for positioning the nodes defines which aspect are most prominently presented with the node-link representation. For networks, two types of algorithms are dominant: force-directed ones and attribute-based ones [Nob+19]. In addition, for the sub-class of trees, further layout algorithms exist that can build upon the hierarchical structure as, for example, discussed in the context of a business data use case [HD18]. However, for this chapter, the focus remains on general graph visualization with no further knowledge of the specific structure.

Force-directed Layouts The characteristics of force-directed layouts can be beneficial for responsive design as well as small screens. Particularly relevant for the latter one, node overlap tends to be reduced as nodes repel each other during the simulation. Further, the algorithm is agnostic to display size, orientation, and aspect ratio. In order to make the algorithm truly responsive, i.e., explicitly considering device characteristics, it can also be used in a bounded fashion, guaranteeing that the elements remain within a certain area (Figure 4.1a). This then guarantees that by default an overview is provided (R1). Notably, one disadvantage of force-directed algorithms is that they are computational expensive, especially when user-driven changes can restart the algorithm (e.g., when manually re-positioning nodes). One way to minimize this effect is applying the algorithm statically, this is, calculating the layout only once when loading the application.

Attribute-based Layouts In the context of multivariate networks, attribute-based layouts allow to directly encode two data attributes. This can be beneficial when space and density render other encodings mechanisms using visual attributes inefficient. However, this comes at the cost of potentially increased overlap of nodes, particularly when the attributes are categorical or unevenly distributed. In these cases, a jitter for the node position can be applied to indicate that nodes are overlapping [Cha83]. The layout can also be calculated to match the device's viewport by scaling the axes, so that the whole graph is visible (R1).

4.2.2 Encoding Multivariate Aspects

Node-link diagrams represent structural aspects by design, while the inherent multivariate aspect have to be encoded explicitly. Indicating these is also part of providing an overview on the network (R1), i.e., allowing analysts to recognize patterns or outliers within the multivariate aspects. In the following, the general encoding strategies for both nodes and links are recapped before discussing how integrated presentation techniques, such as focus+context or semantic zoom, can allow for a more responsive representation in general as well as for accessing details (R2).

General Encoding Strategies Attributes can be encoded on both nodes and links, for example, by adapting size, color, or shape/style. However, in the context of responsive visualization, these should be used carefully. Network visualization are often very dense presentations that are prone to visual clutter, thus, using additional visual variables can amplify clutter—particular on device with small screen estate such as smartphones. Further, some variables such as size can also lead to overlaps or to elements too small to be touched. The usage of color can be more reasonable for indicating groups or highlights (e.g., link types, connected nodes, value range). For larger networks, it can be more sensible to provide these only when sufficient space and few elements are present, for example when zoomed in, instead of applying encodings to visual variables by default. In general, a responsive behavior can thus be reducing or increasing the use of visual attributes for data encodings depending on the graphs visual density in the current viewing context.

Embedded Visualizations Extending the idea of providing details on demand through zooming mechanisms follows the concept of semantic zooming. As outlined in Subsection 4.1.2, semantic zooming has been intensively used for node-link presentations, e.g., to resolve clusters or provide embedded encodings. These techniques

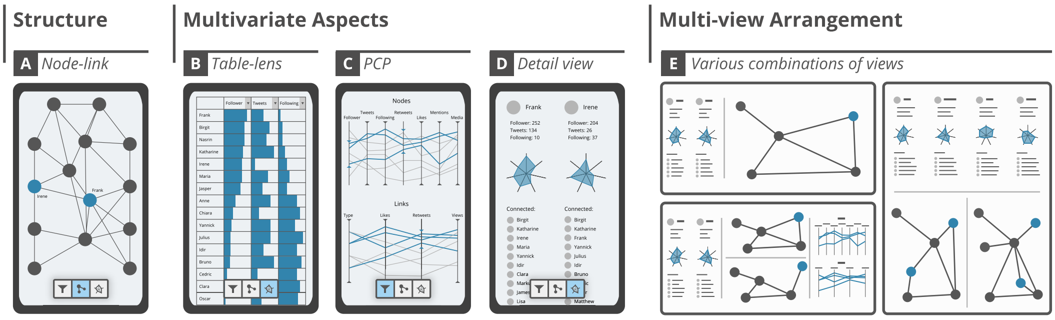


Fig. 4.1.: Multiple specialized views showing the different graph aspects can be provided, e.g., (a) the node-link representation showing the structure and (b–c) visualizations showing the attribute distribution or (d) the attributes for specific objects. Further, (e) these views can also be arranged in various multi-view layouts.

can be incorporated as a responsive mechanism that also addresses R2 (providing access to desired details). For example, an enlarged node can host an integrated bar chart or line chart representation detailing its multivariate characteristics. Notably, multiple ways exist how the objects can be viewed in an enlarged way: either by using a global zoom and pan or by using local focus+context techniques, e.g., via magic lenses [KRD16]. As discussed in a previous publication on business data analysis [HKD17], these integrated visualizations can also allow for direct manipulation techniques to edit the underlying data, e.g., dragging a bar to a different value in order to simulate an alternative business data prediction (R3). Such adaptations could be used to provide a task-dependent responsiveness.

One challenge of embedded visualizations is supporting the comparison of multiple nodes or edges. As embedded visualizations can only be provided within one node and hardly span multiple ones, there is a need for a separate view that allows for a combined comparison of multiple nodes or edges. At the same time, introducing such a separation can also allow to incorporate more targeted adaptations to the single views, and thus, increase the possibilities for responsive mechanisms. The discussion of embedded visualizations, i.e., what techniques can be embedded how, will be continued in the section on Responsive Matrix Cells (Section 4.3).

Handling Separated Views In order to more easily control the amount of displayed data as well as its level of detail, one approach is separately visualizing the different aspects of multivariate graphs in different views and juxta-pose them (requirement R4.1). With the node-link view as starting point (showing the structural aspects, Figure 4.1a), multiple possibilities exist how the multivariate aspects can be shown in an adjacent view. In line with R1, such a view should provide an overview on the multivariate aspects, thus, communicating how attributes are distributed across all nodes or edges, including possible outliers. For this, multiple options exist, e.g., histograms, TableLens-like representations, or parallel coordinates plots (Figure 4.1b+c). Depending on the context, the views can be displayed next to each other (Figure 4.1e) or the interface allows to easily switch between these perspectives. Thus, as a responsive behavior, the general layout can be adapted as well as the specific level of detail of a single view.

Particularly for multivariate graphs, an additional interaction layer is required to provide access to specific details (R2), i.e., values of one or multiple nodes or edges. For example, selecting a region of interest could result in providing the details for the selected objects in a separated view (Figure 4.1d). As the goal is to represent all of the node's or edge's attributes, among others, star plots are one possible visualization that allows to visualize multiple heterogeneous attributes at the same time. In addition, labels, categories, or core attributes can also be shown as plain text elements. In the case where two objects are considered, the goal is to foster a 1:1 comparison, e.g., by providing overlaid or side-by-side placed visualizations (Figure 4.1d+e), which allow for quickly scanning the differences and similarities [Gle+11; JE12]. Overall, instead of visualizing all data aspects at once, introducing an additional interaction layer can help to provide responsiveness. In addition, this also allows to implement a task-dependent responsiveness, providing the details in way that matches the analysis context (e.g., single object, comparison).

In conclusion, for a responsive node-link representation, it makes sense to follow a multi-view approach and to separate the visualization of the different data aspects. This allows to better control the level of detail of the single views, which, in turn, makes it easier to adapt them for various screen estates. However, for such a solution it is crucial to provide suitable interaction means to ensure access to all data aspects and react to the specific needs of the current context.

4.3 Responsive Matrix Cells: Embedded Visualizations in Matrix Representations

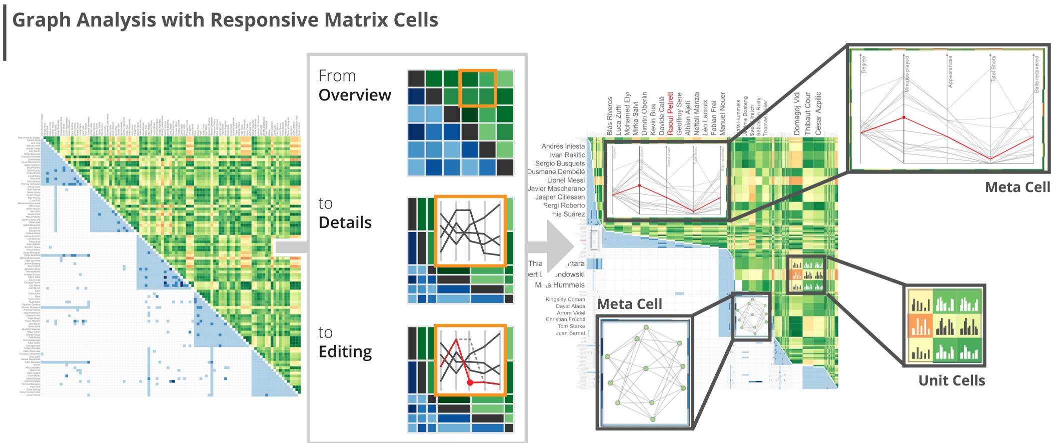


Fig. 4.2.: *Responsive matrix cells* are a focus+context approach that provides details for a multivariate graph via embedded visualizations in a matrix representation and allows analysts to go from the overview in the matrix to details as well as editing within the cells. *Videos and further material are provided at* `imld.de/RMC`.

In contrast to the node-link-based solution, a matrix representation can also easily follow an integrated approach (requirement R4.2). Here, the predictable and compact layout of the matrix as well as the clear position and sorting of nodes and edges renders it suitable for providing details in-place via focus+context mechanisms. In this context, we[2] propose *responsive matrix cells* (RMCs) as a flexible focus+context approach to embed visualizations into a matrix, more specifically, either into individual cells (unit cells) or across cohesive sub-matrices (meta cells). In this section, we elaborate on the visual design of RMCs, while the interaction facilities of RMCs will be described in detail in Section 4.4.

4.3.1 Approach Overview

The core idea of the RMC approach is illustrated in Figure 4.2: A special matrix visualization delivers the overview, while responsive matrix cells embedded into the matrix provide details in various ways and allow users to perform edit operations.

The basis for the overview is a customized matrix visualization [BST19]. As depicted in Figure 4.3, it shows adjacency information and node similarity at the same time.

[2] "We" in this and the following sections relates to the author Tom Horak, as well as Philip Berger, Heidrun Schumann, Raimund Dachselt, and Christian Tominski as co-contributors to this research.

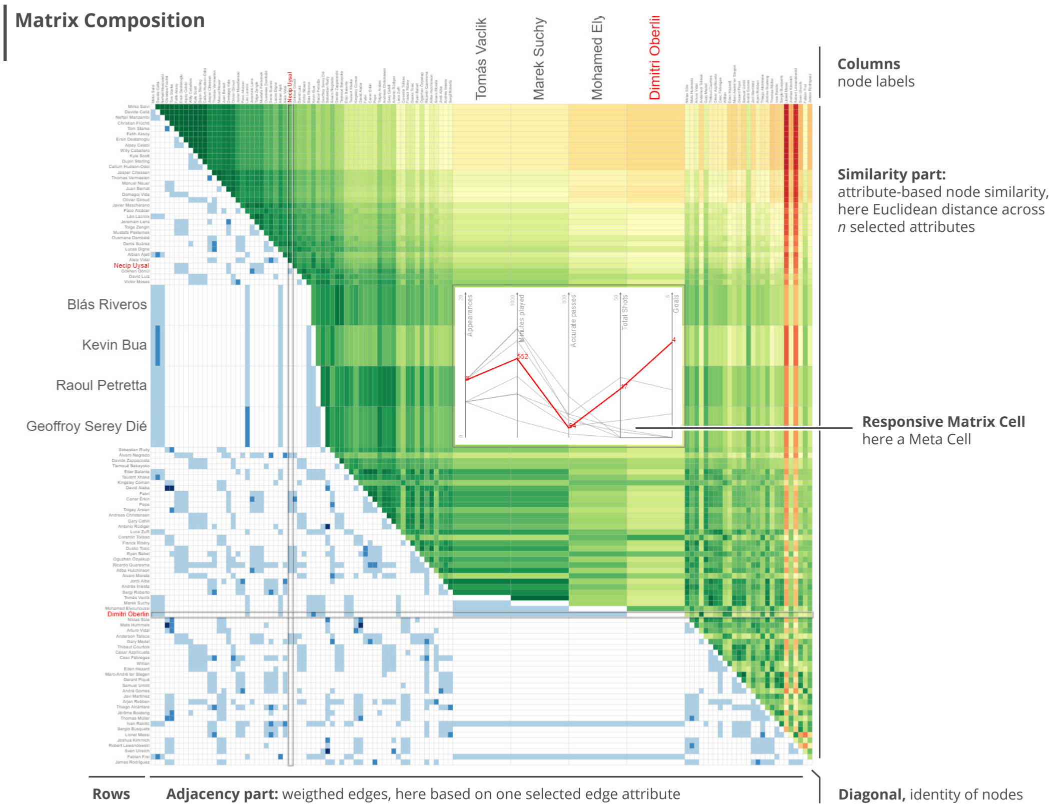


Fig. 4.3.: The matrix is composed of two halves, where the lower triangular half visualizes the weighted edges of a graph and the upper triangular half displays the pairwise similarity of nodes with respect to their multivariate attributes.

As for regular adjacency matrices, rows and columns correspond to the set of nodes. The lower-left triangular half of the matrix visualizes the presence of edges and color-codes a selected edge attribute. Yet, the upper-right triangular part of the matrix shows different information. It color-codes pairwise node similarity as computed based on node attributes. This custom matrix allows users to recognize structural clusters (e.g., hub nodes, cliques, bi-cliques), groups with similar attribute values, and outliers in general (R1). However, as the color-coding visualizes only a single piece of information (i.e., attribute value or node similarity) per cell, multivariate details of edges and nodes are not visible.

To access details and additional functionality, users can initiate responsive matrix cells (RMCs) within the overview matrix. More specifically, users create RMCs either for individual matrix cells (unit cells) or for sub-matrices (meta cells) and scale them up in a focus+context fashion as shown in Figure 4.3. The gained display space is used to embed interactive views that enable users to see and compare details of the data (R2). Additionally, editing facilities are provided when RMCs are shown at a

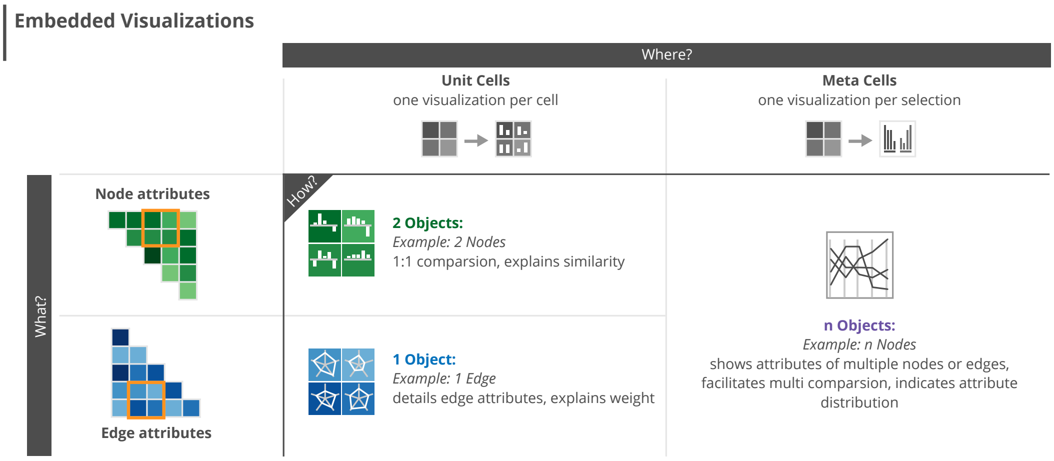


Fig. 4.4.: The embedded visualizations of responsive matrix cells are characterized by *what* they show, *where* they show it, and *how* they show it. The *what* and the *where* define the context for the *how*.

sufficient size (R3). This minimizes interruptions of the analysis workflow as users no longer need to resort to external editing tools (R4.2).

While the overall matrix remains static with a predictable space requirement, the embedded RMC visualization are revealing details and functionality to the analyst in a responsive way. At the core, an embedded RMC visualization adapts to: (a) the origin where the RMC has been created, (b) the space being available for the RMC, and (c) the task (i.e., explore, compare, edit) of the analyst.

There are different design choices for making RMCs responsive. We will primarily be concerned with *what* additional information can be shown *where* in the matrix, and *how* the information can be visualized specifically (Figure 4.4). The what, where, and how will be detailed in the remainder of this section.

4.3.2 What can be Shown?

Multivariate graphs consist of two types of *objects*, nodes and edges, where each object can have several attribute values. By having a matrix with an adjacency part (lower-left) and a similarity part (upper-right), one half of the matrix is primarily focused on the edges, while the other half is focused on the nodes, more specifically on how two given nodes compare. This distinction is crucial to understand *what* information is shown in RMCs. As indicated in Figure 4.4, an RMC being located in the adjacency part (blue) will show information about the edges associated with the underlying matrix cells, while an RMC in the similarity part (green) will show information about the nodes associated with the corresponding rows and columns.

An RMC may span a single matrix cell, in which case it either represents a single edge (adjacency part), a single node (diagonal), or a pair of two nodes (similarity part). Such RMCs allow analysts to study the details of individual nodes and edges or conduct a 1:1 comparison of two nodes. An RMC may also cover an $i \times j$ sub-matrix with $m = i \cdot j$ cells, which means it represents either a group of $n \leq i \cdot j$ edges or a group of $n \leq i + j$ nodes. For such groups of objects, analysts might be interested in studying individual objects as indicated before, but also in investigating the group characteristics as a whole, including the distribution of attribute values or structural aspects of the group's induced sub-graph.

In sum, RMCs support three types of information representation: representations for *1 object* to show its details, for *2 objects* to directly compare them, or for n *objects* to convey group properties.

4.3.3 Where will Information be Shown?

The question of where detail information will be shown depends on a user-specified region of interest (RoI). If the user is interested in an individual edge or an individual pair of nodes, the RoI consist of only a single cell. In that case, a single visualization is embedded into the cell of interest. We call such cells *unit cells*.

When the RoI is defined as an $i \times j$ sub-matrix, it could mean the user wants the details for (a) the individual objects covered or (b) the group comprised of the objects. For case (a), multiple unit cells are created so that there is one embedded visualization for each cell of the sub-matrix. In other words, the cells of the sub-matrix are treated individually as units, similar to *small multiples* [Tuf01]. For case (b), the sub-matrix is treated as a whole and a single visualization is embedded into it. We can also say that the RoI is subsumed into an aggregated *meta cell* being concerned with the data as a group. Figure 4.4 illustrates that unit cells provide visualizations detailing 1 or 2 objects, whereas a meta cell provides the details for n objects in a single visualization.

Unit cells and meta cells differ in their characteristics, which also has consequences for the embedded visualizations. Unit cells generally start in the square aspect ratio of the underlying matrix cells. When unit cells are generated for a sub-matrix, a visualization is placed in each cell. As these visualizations have to share the available display space, they initially cover only a few pixels. Therefore, unit cells typically require zooming before further details are revealed. Figure 4.5 depicts possible unit cell designs. Meta cells span multiple underlying matrix cells and therefore start at a larger size than unit cells. Yet, as illustrated in Figure 4.6, no assumptions can be made about a meta cell's aspect ratio as it depends on the shape of the RoI defined by the analyst. Consequently, the visualizations embedded into meta cells must cope with varying aspect ratios. Next, we discuss the design of embedded responsive visualization in detail.

4.3.4 How is Information Shown?

Based on the discussion in Chapter 3, we outline what it takes to make the embedded visualizations responsive in this context and illustrate this with selected examples. Our discussion focuses on (i) how the visualizations scale and respond, and (ii) what information they can represent.

Making Visualization Responsive

In our case, the embedded responsive visualizations must be able to communicate the characteristics of one or two objects for unit cells, and of n objects for meta cells (Figure 4.4). Depending on the number of objects, the visualizations should facilitate *object visibility* or *attribute visibility* [Spe01]. The focus can be on representing data attributes or supporting comparison tasks (R2). As indicated above, responsive visualization must also be compatible with different aspect ratios.

Most importantly for our focus+context approach, the visualizations must be able to work at different sizes. Ideally, details are conveyed already at sizes of a few pixels. When additional space becomes available, it should be used efficiently by adding more and more details, not only geometrically, but also semantically [PF93; Mat+02; McL+08]. For our RMCs, we consider four major levels of detail (LoD) that represent important breakpoints when increasing the cell size: (1) *Pixel* level with color-coding only, (2) *Miniature* level with a minimal version of the visualization, (3) *Compact* level showing first labels or values, and (4) *Medium* level showing more labels and details.

Note that the medium level is not meant as a maximum, since cells can be increased further and more details can be added. Also, we refrain from defining exact pixel-based values for these sizes because the specific thresholds for showing additional details depend on the visualization (e.g., how space-efficient the visualization is), the used device (e.g., what resolution and pixel density is offered), and preferences of the user (e.g., details as soon as possible vs. abstraction as early as possible).

A general concern though is to help users maintain their mental map as the LoD changes. To this end, we propose to preserve the original matrix cell's color-coding as the background color at the miniature size or as the border color for the larger sizes as illustrated in Figure 4.5. Maintaining the color as a visual residue can make it easier to keep track of specific cells and to recall why they seemed of interest (e.g., dark encoding, light encoding, similar encoding). Yet, when used in the background, the color can potentially compromise the contrast in the embedded visualizations. Therefore, miniature visualizations render their marks using a contrast color (e.g., white or dark gray) that depends on the luminance of the background. This way, we can guarantee a sufficient separation of background and visualization.

Complementing the aforementioned general design aspects, we next discuss specific design considerations for visualizing the multivariate attributes of nodes and edges. Representations of structural aspects and multi-faceted data aspects will be discussed later in this section.

Designs for Multivariate Aspects

This section proposes exemplary designs for multivariate visualizations in RMCs. First, we focus on unit cells, for which the visualization has to encode either one or two objects primarily for *object visibility*. As suitable techniques, we consider bar charts and star plots for a single object as well as adaptations of them for representing and comparing two objects as illustrated in Figure 4.5. Second, we discuss visualization designs for meta cells, for which *attribute visibility* is important. Here, we consider parallel coordinates plots in addition to grouped bar charts, and star plots as indicated in Figure 4.6.

Focusing on Details of a One Object For a single object, the objective is to make its specific attribute values visible (R2). Bar charts are suitable for this purpose. They already work well on the miniature size as bars are easy to distinguish and make good use of the available space (Figure 4.5a). At the compact size, it is possible to start showing labels (e.g., for the maximum), while at the medium size, all values and potentially the attributes can be labeled.

A downside of a bar chart is that all attributes should be in the same or similar value range so that they can share the same axis. Otherwise, certain attributes can be overemphasized if the same normalized axis is being used. Alternatively, each bar can have its own axis, but these are difficult to incorporate on small sizes. Another option is to configure the bars to not show absolute values but relative ones

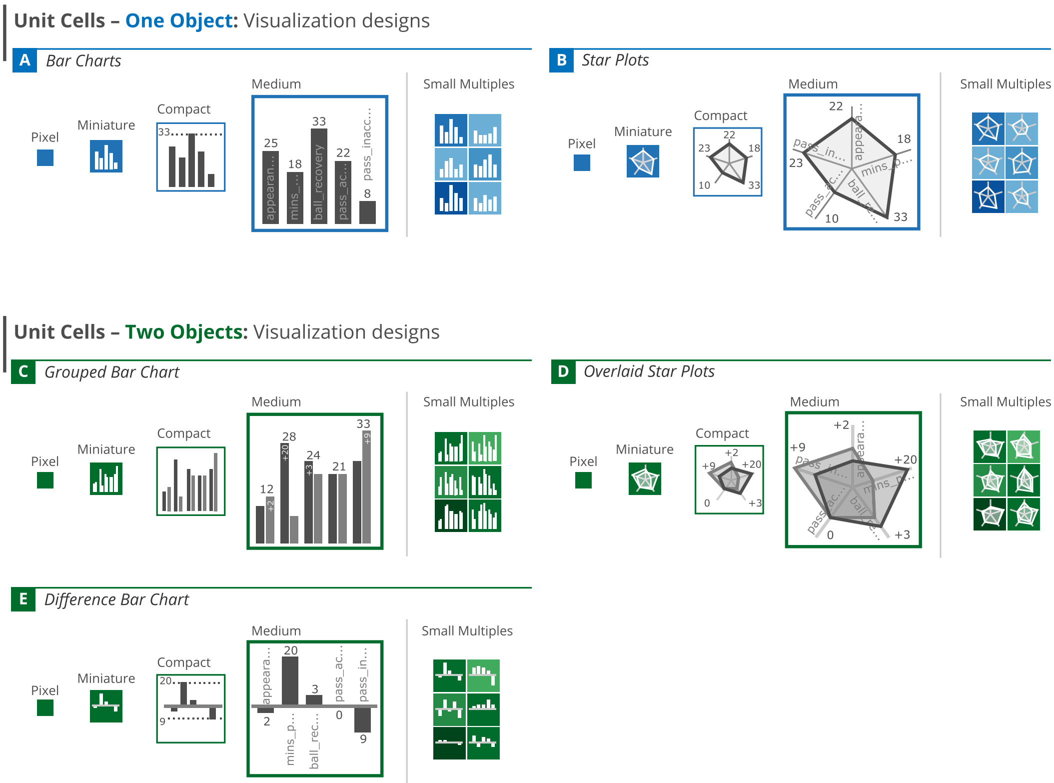


Fig. 4.5.: Visualizations in unit cells have to represent 1 or 2 objects. Variants of bar charts and star plots are suitable for being embedded into unit cells. Depending on the available display space, different levels of detail can be offered. Unit cells also work as small multiples.

corresponding to the global min/max. In both cases, however, interpreting the bars could be difficult as they would contradict typical conventions.

Another technique for representing a single object are star plots. Similar to bar charts, star plots work well on miniature size thanks to their glyph-like appearance [Bor+13] (Figure 4.5b). The glyph-like character is particularly beneficial when multiple unit cells form a small-multiples arrangement. Star plots make sense for three or more data attributes. Each attribute has its own axis, avoiding issues with different attribute ranges. However, the comparison of attribute values at the differently oriented axes can potentially be more demanding. Value labels can be shown starting at the compact size, and axis labels make sense at the medium size. Especially at smaller sizes, overlaps of labels with the plot lines and the axes are hard to avoid.

Comparing Two Objects in Detail In general, visual comparison of two objects can be supported by showing the two objects in parallel (superimposed or juxtaposed) or by computing and visualizing their difference directly [Gle+11]. As before, bar charts and star plots can be used to show two objects at the same time. Particularly useful for comparison are bar charts, where bars are grouped by attribute, and overlaid star plots (Figure 4.5c,d). In both cases, the visual density is increased due to the additional graphical marks, which requires different responsive behavior. For example, labels for the grouped bar charts become visible only at the compact size, as the miniature size already introduces the usage of different shades for the bars as a new detail. For both grouped bar charts and overlaid star plots, it is not inherently clear which marks corresponds to which object (i.e., the node of the row or of the column). This can be mitigated by establishing conventions. For example, the bars corresponding to the row node can always be shown on the left, or its outlined polygon always be rendered on top. Interactive coordinated highlighting further supports users in identifying data objects in RMCs (see Section 4.4).

In addition to showing two objects simultaneously, comparison tasks can also be supported by directly encoding the difference between the objects in a difference bar chart (Figure 4.5e). While this sacrifices the display of the actual values, the comparison is simplified and the chart itself is cleaner with fewer marks being shown. Thanks to the simpler design, difference bar charts work well in a small-multiples arrangement of unit cells. The idea of encoding differences directly can also be expanded to star plots, where the polygonal shapes could encode the differences.

Inspecting Multiple Objects Meta cells provide a visual representation of a group of either nodes or edges. In contrast to the designs discussed before, visualizations embedded into meta cells often divert from the typically square aspect ratio of their unit-cell counterparts. In general, three aspect ratios of meta cells are relevant: a wide shape in horizontal orientation (landscape), a wide shape in vertical orientation (portrait), and an (almost) square shape.

Visualizations whose space demands grow mostly in only one direction work well with landscape and portrait, where different orientations can be supported by 90-degree rotation. A prominent example are parallel coordinates plots (PCPs), which benefit from growing with the number of shown attributes or axes. PCPs offer the necessary degree of flexibility to adapt to different aspect ratios as both the axes and the spacing in between are easy to adjust (Figure 4.6a,b). PCPs can support attribute visibility, which enables users to see how attribute values are distributed, whether attributes are correlated, or if there are any outliers. At miniature size, no labels can be shown, while at compact size it gets possible to indicate minimum and maximum values per axis. At medium size, axis labels can be displayed and the background can show the entire data set in a dimmed fashion to provide additional context.

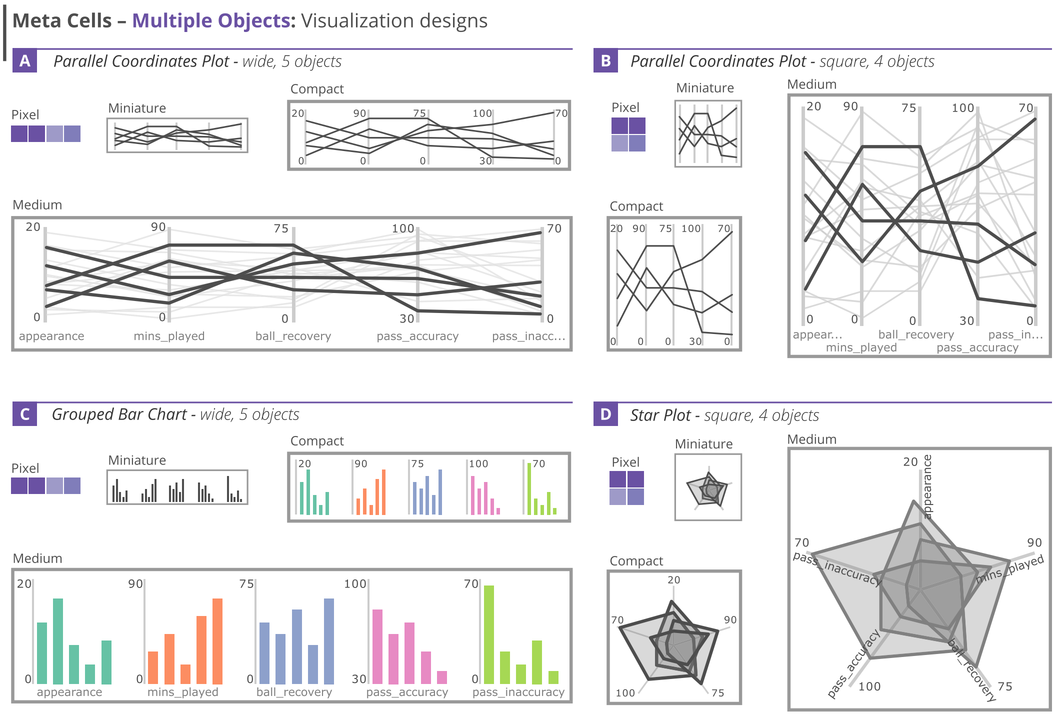


Fig. 4.6.: Visualizations in meta cells of size $i \times j$ must encode n objects, either $n \leq i \cdot j$ edges or $n \leq i + j$ nodes. For example, when studying nodes, a 4×1 meta cell represents 4 column nodes plus one row node (=5 objects), while a 2×2 meta cell represents 2 column nodes plus 2 row nodes (=4 objects). In addition to different levels of detail, visualizations in meta cells have to take varying aspect ratios into account.

Grouped bar charts and star plots also facilitate the inspection of multiple objects. In a grouped bar chart, there are several adjunct groups of bars, one group for each attribute (Figure 4.6c). The grouping makes it possible to show individual axes per group at larger sizes. Within a group, the number of bars corresponds to the number of objects. Hence, the space demand for grouped bar charts grows in only one direction as the number of attributes and objects increases. This makes grouped bar charts suitable for landscape and portrait orientations. Star plots, on the other hand, become distorted for landscape and portrait. They are better suited for square-shaped meta cells (Figure 4.6d). Analog to PCPs, star plots allow to recognize attribute distributions and correlations as well as to compare specific objects.

As for any multivariate visualization, readability in meta cells decreases with a large number of marks due to over-plotting. Yet, our focus+context RMCs are not meant to operate on larger data, but on subsets as defined by regions of interest. Still, it is mandatory to support readability and object identification by means of interactive highlighting as described in Section 4.4.

Designs for Further Data Aspects

So far, we mainly illustrated RMCs for representing multivariate data aspects. Yet, RMCs can also be employed to convey other data aspects, including structural, spatial, or temporal aspects of graphs.

While the adjacency part of the overview matrix already incorporates structural aspects, certain path-related analysis tasks are easier to carry out with node-link diagrams [GFC05; OJK19]. To combine the advantages of both, node-link diagrams can be embedded into meta cells. They show the induced sub-graph corresponding to the set of nodes or the set of edges associated with the RoI. For the layout and encoding, the same strategies as described in Section 4.2 can be used. Embedding a node-link diagram enables users to quickly check how certain patterns in the adjacency matrix look like in an arguably more intuitive representation.

Besides graph structure and multivariate attributes, a graph can have further facets, most prominently spatial and temporal dependencies [HSS15]. Provided that suitable visualizations for such additional facets exist, RMCs can generally be used to also embed them into the matrix. For example, a meta cell could be extended to show a map underneath a node-link diagram and use a geographical layout. Similarly, it would be possible to show nodes or edges along a time line. While these are first ideas for generalizing RMCs, concrete designs are left for future work.

4.4 From Overview to Details to Editing with RMCs

To facilitate the dynamic use of RMCs as a data exploration and editing tool, a suitable interactive interface must be provided to the user. In fact, our approach really lives from interaction. Yet, the combination of focus+context and embedded visualizations makes the interface design a non-trivial endeavor. On the one hand, interaction with the matrix must be possible on a global level (e.g., selecting attributes of interest). On the other hand, users must be able to interact on a local level with the RMCs (e.g., scaling RMCs) and the embedded visual representations (e.g., highlighting and editing data). Careful design is necessary to obtain an easy-to-use and conflict-free interaction repertoire.

The starting point for RMCs is that users spot something interesting in the overview matrix (R1). Therefore, the analyst can initially configure the matrix on a global level by zooming and panning, selecting the attributes to be included in the similarity calculation, sorting rows and columns, and choosing appropriate color scales via a global menu. Once the overview matrix has been set up so that interesting data features stand out, RMCs come into play to inspect and compare the surfaced features in detail (R2). In the following, we discuss how analysts can create RMCs and configure the embedded visual representations. Finally, we turn our attention to data editing by interactively manipulating marks in the visualizations (R3).

4.4.1 Exploring Details with RMCs

The primary steps for exploring details with RMCs are to create and configure RMCs in the first place, to adjust the embedded visual representations appropriately, and to link data points across RMCs and the overview matrix.

Creating RMCs In order to create a new RMC, the analyst simply clicks and drags up a rectangular region of interest (RoI) covering the matrix cells to be studied in detail (Figure 4.7a). A single-cell RMC is created with a single click or tap. As the user-specified RoIs are typically associated with some visual patterns being evident in the overview matrix (e.g., cluster of edges or group of very (dis)similar nodes), the creation process could be eased by offering automatic selection support that fits RMCs to such patterns [Yu+16]. Upon creation, RMCs are initialized based on useful defaults. Whether node or edge attributes will be shown (the what) depends on the triangular matrix part where the RoI is created. By default, meta cells will be created (the where). To generate a small-multiples arrangement of unit cells, a modifier key (e.g., shift) can be held while selecting the RoI. For the embedded visualization (the how), we consider bar charts as a suitable default. All these default settings can be subject to interactive adjustment via a local menu as explained later.

Scaling RMCs A major advantage of RMCs is their flexible level of detail (LoD), which is coupled to their scaling level. On creation, RMCs are automatically scaled up from the pixel to the miniature level revealing initial details in the embedded visualization. The analyst can increase the LoD further by local zooming, for example, using the mouse wheel, dragging the RMC borders, or performing a pinch gesture (Figure 4.7b). The additional space required for enlarging RMCs is obtained by shrinking rows and columns outside of RMCs uniformly like in bifocal views [ATS82]. To deal with the issue of varying aspect ratios, the zooming can happen either uniformly in x and y directions or be restricted to only x or y direction. Upon zooming, responsiveness sets in and RMCs are automatically enhanced with additional information and richer visual encodings. These make it easier for the analysts to read the visual representation and understand details better.

Expanding, Shrinking, and Dismissing RMCs After first conclusions have been drawn from an RMC, the analyst's interest might change. This can result in the need to adapt the region covered by RMCs, that is, to expand or reduce it by adding or removing cells. This can be supported by dragging borders similar to scaling up RMCs, but while activating another modifier (Figure 4.7c). Once an RMC's details have been studied conclusively, the RMC can be dismissed. This is as easy as triggering a shortcut key (e.g., delete) or a designated gesture. A global reset function can be used to dismiss all RMCs altogether and reset the overview matrix.

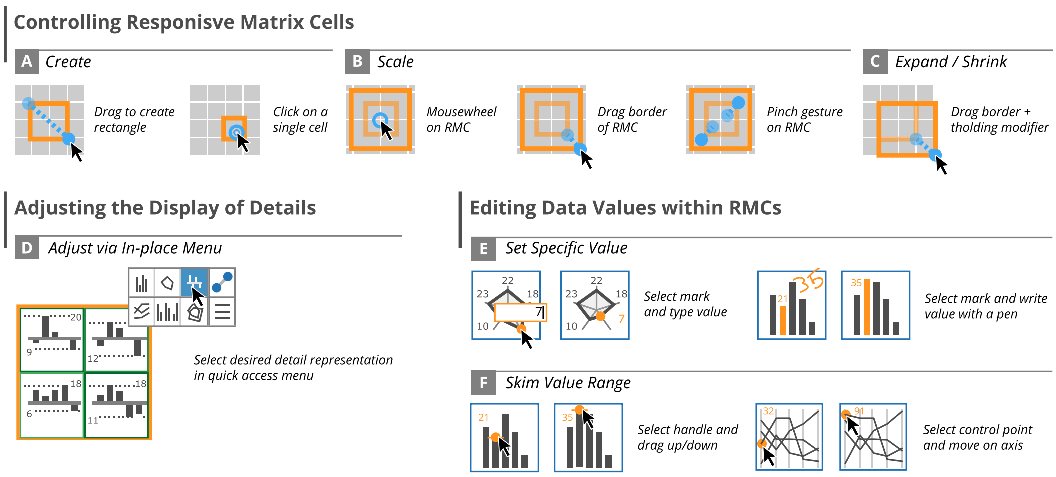


Fig. 4.7.: Interaction techniques for creating and configuring RMCs, and mechanisms for editing attribute values.

Adjusting the Display of Details To facilitate the in-depth exploration of details, RMCs provide an in-place menu interface (Figure 4.7d) for adjusting what (nodes or edges), where (unit or meta cells), and how (embedded visualization) details are made visible. When switching nodes and edges, RMCs are automatically transitioned from one half of the matrix to their corresponding position in the other half. Switching the cell type results in either merging a set of unit cells into a meta cell or splitting up a meta cell into several unit cells. Switching the embedded visualization simply replaces the visual representation in an RMC. To allow analysts to quickly switch back and forth between the different options, the menu stays open until a suitable configuration has been found and the menu is closed explicitly.

As we consider altering the visualization (the how) to be a frequent operation during the data exploration, additional shortcuts are provided. The arrow keys can be used to select different visualizations and layout variants, the space bar toggles between unit and meta cells, and the tab key switches between node and edge attributes. For touch interfaces, horizontal, vertical, and diagonal swipe gestures can be used to adjust visualization type, cell type, and attribute type respectively.

Exploring details typically involves further adjustments of visual representations, for example, reordering axes, selecting attributes, changing scales, and so on. While it is standard to carry out such interactions directly within the visualization, this is impractical for our space-constrained RMCs. Instead, it makes sense to offload further adjustments to external controls or the menu.

Linking Details and Overview A coordinated highlighting is indispensable to support analysts in linking the details provided in one RMC to the overview matrix and the details in other RMCs. In general, hovering graphical marks in RMCs results in highlighting all other marks being associated with the same node or edge. For example, hovering a node in an embedded node-link diagram results in highlighting all corresponding marks in all other RMCs and in emphasizing the corresponding row and column labels in the overview matrix (and vice versa).

4.4.2 Editing Data Values within RMCs

During an in-depth analysis of a multivariate graph, it can be desirable or even necessary to shift from data exploration to data editing. This shift can be motivated by the need of either correcting erroneous data or observing the influence of an attribute on the overall graph. The first case corresponds to Baudel's direct manipulation principle, where data values are edited directly within the visualization [Bau06]. The second case addresses what Spence coined *what-if* analyses, which can help users understand the interplay of different values [Spe01]. In both situations, the edits are supposed to be immediately visible in the visualization.

In general, an edit operation can target the graph structure (add or remove nodes or edges) or the associated attribute values (update) [Gla+15a]. The literature already offers several strategies for editing structural aspects using matrices [Gla+14; Gla+15b; Kis+17]. Therefore, our interest primarily regards the editing of attribute values. Here, depending on the user's goal (correcting error or what-if analyses), editing can mean plainly setting a specific value or involve skimming a range of potential values before a value is eventually set. A specific value is set by entering it via keyboard or, where pen input is available, via handwriting (Figure 4.7e).

For what-if analyses, entering many values in such a discrete fashion is impractical. Instead, it must be possible to quickly check a range of values while observing the resulting changes in the visualization (cf. Horak et al. [HKD17]). This is facilitated by continuous drag gestures where users move the data-encoding marks directly within RMCs. To this end, interaction handles become available as soon as RMCs are sufficiently large to allow for a reasonable range of movement so that edits can be performed more precisely. For most of the previously discussed visualizations, this starts to be doable at the compact size. Figure 4.7f illustrates the editing for bar charts and parallel coordinates. In a bar chart, the upper end of a bar can be dragged up or down to update the underlying attribute value. In parallel coordinates (and star plots), the control points of the polylines can be dragged for editing.

4.5 Applying Responsive Matrix Cells

We implemented the RMC approach in a web-based prototype using the native canvas API for rendering, the D3 library [BOH11] for computing force-directed layouts, and the chroma.js library [@Ais13] for color coding. The prototype is shown in Figure 4.8. The GUI consists of a mix of SemanticUI [@Sem13] and custom controls (Figure 4.8c+d). The prototype supports all key concepts via mouse and keyboard, including creating unit and meta cells, scaling them up, changing the visualizations, and editing attribute values; touch and pen input are currently not supported. Except for difference bar charts and spatiotemporal visualizations, all visualizations discussed in Subsection 4.3.4 are implemented. The prototype is publicly available [@TBH20].

For our implementation, we considered the use case of exploring and editing real-world soccer data. Based on this, we demonstrated the feasibility of our approach by presenting a walk-through for the use case as well as conducting a user feedback session. The walk-through is provided in the original publication [Hor+21b], while the feedback session will be described in the following.

Data & Task

As an example data set, we used a graph of soccer players from the 2017/18 Champions League season. The graph consists of 95 players, the nodes of the

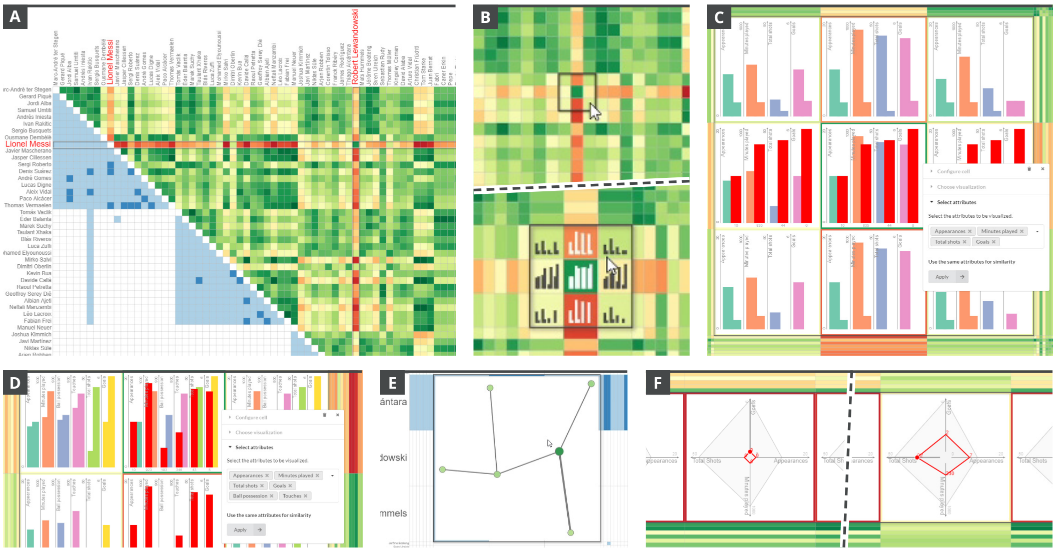


Fig. 4.8.: Screenshots of our realized prototype [@TBH20] showing the main workflow within RMCs: Going from (a) overview, to (b–e) exploring multivariate and structural details, to (f) editing data values.

graph. The players are characterized by up to 39 quantitative data attributes, including general stats (e.g., minutes played), defensive figures (e.g., balls recovered, interceptions), and offensive qualities (e.g., shots on goal, goals scored). Not all players have values for all attributes. While this is partly due to different player types (e.g., goal keeper can have special attributes), some players are actually lacking correct attribute values. An edge represents co-occurrences of two players, that is, if and how often the two players have played for the same club during their career. The edge weight corresponds to the number of shared clubs, however, no further edge attributes are present in the data set. There are 1046 edges in the graph.

For the use case, it is assumed that the goal of an analyst is to identify match-deciding players and compare them with each other. In particular, this includes (i) the exploration of details of an a priori unknown sub-graph and (ii) the correction of found errors within this sub-graph. These goals were set out for both the walk-through as well as for the user feedback session.

Preliminary User Feedback

In order to receive early user feedback on our approach, we invited 4 researchers (2 interaction experts, 2 visualization experts, all PhD-level) from our local institutes for guided hands-on sessions. After an introduction and short demonstration, participants (P 1–4) were asked to interact with the prototype and test the usability of its different functionalities. Sessions were conducted remotely by two investigators via video chat with screen-sharing and lasted around 1 hour. Overall, participants were very positive and attested the implementation a high quality. While all agreed that initial training is required to understand both data set and visualization approach, they adapted to the interface quickly and used all techniques without larger issues. Interestingly, while we did not instruct for, participants started to reason on the data, but with different approaches and focus. For example, participant P 3 started exploring possible matrix sorting, participant P 1 looked into node similarities, and participant P 4 focused on the relations between similarity and adjacency.

Both unit cells and meta cells were considered helpful to understand why nodes are (dis)similar, but we could observe that unit cells required more time to be properly read—likely since attribute labels are only shown on higher zoom levels (Figure 4.8b+c). As participants P 1 and P 4 used global zoom more intensively, they noted that the node labels were quickly becoming invisible, as they are only placed outside of the matrix (cf. Figure 4.8a and b). Showing the labels additionally around an RMC could avoid this. The highlighting mechanisms were considered useful with few suggestions for improvements, e.g., permanent highlights for one or more nodes (P 2 + 4), or highlights of attribute axes when hovering the labels in the sidebar or context menu (P 2). All participants found the editing very useful, particularly for understanding the influence of attributes on the similarity measure (Figure 4.8f). However, while working with larger RMCs at a high LoD, P 2 and P 4 noted that due to the stronger distortion the edit effects are getting harder to observe in the overall matrix. Simplifying editing on lower LoDs could mitigate this issue. With most of the provided mechanisms working smoothly, ideas for further functionalities were proposed, e.g., allowing filtering of nodes within RMCs (P 3).

To summarize, the walk-through as well as the user feedback provide a first indication of the utility of RMCs. In combination, this suggest that RMCs accomplish what they set out to achieve: They support a seamless analysis workflow from an overview to details to editing without resorting to external tools. As a crucial part of this dynamic nature, the responsive mechanisms allow analysts to access the required information that is currently of interest—or, in other words, our approach can provide visualizations that match the current factors of the data exploration.

4.6 Discussion and Summary

This chapter set out the goal to explore possible ways to provide visualizations of multivariate graph data in a compact and flexible manner as well as to incorporate responsiveness as a fluid concept. In the following, further reflections are provided onto the general aspects of responsiveness as well as how both approaches could be developed further.

Integrated Responsiveness As discussed throughout the chapter, a visualization for multivariate graphs can follow a multi-view approach or an integrated approach. In both, particular the way how access to details is provided can allow to incorporate responsive strategies. However, for the overall visualization the extend of possible responsiveness depends on the base technique: For example, while a matrix visualization can provide rich exploration means within a compact space, it still requires a certain minimum space to work properly. Therefore, it might be not as well suited for very small devices such as smartphones. In contrast, a multi-view approach with node-links graph can allow for reducing the visible information density by focusing on one data aspect, while the then hidden aspects are provided via interactive mechanisms.

Based on the discussed adaptations of the graph visualizations, it becomes apparent that responsiveness involves more aspects than adaptations for fulfilling space constraints. First of all, data and human factors are clearly an important part of it, dictating what adaptations are reasonable at all and which data aspects might be negligible based on the users current interest. Within the Responsive Matrix Cells, an analyst can define which region he or she is interest in, while the subsequently provided details and their representation depend on the underlying data structure (here, node attributes vs. edge attributes). Similarly, in a multi-view approach with a node-link chart, only one data aspect (structure or multivariate) could be shown on small screen estates, with a juxta-posed layout used on larger screens. Notably, this also indicates that a responsive adaptation is in most cases a compromise between acknowledging the different factors and providing suitable analysis means. This signifies that such a visualization might not be the optimal or perfect visualization solution for the given context and data, but one that is capable of representing the data adequately in various contexts.

Extending Responsive Matrix Cells The approaches presented here can be developed further in multiple interesting ways. First of all, an important aspect for RMCs, and any graph representation using embedded visualizations, is to have a suitable and diverse set of visualizations that allow to detail specific aspects of the underlying graph. Here, it can be particularly interesting to look at tailored visualizations that work well for specific constellations. For example, further glyph-like visualizations can be effective for small unit cells, scatter plots could show correlations between two attributes in meta cells, miniature maps would be helpful for geo-spatial networks, and horizon graphs could be applied to temporal data attributes. Similarly, the direct editing facilities could be extended to provide more functionalities to an analyst. On the one hand, this involves history and provenance mechanisms for undoing and redoing edits as well as capturing insights respectively [KNS04; NC14; Mat+19]. On the other hand, incorporating additional input modalities of modern environments, e.g., touch [Hor+18b; SS14], pen [FHD09; Rom+19], or speech [SS18], can potentially simplify edit operations and improve precision at lower LoDs. For example, in order to update an attribute value, the new value could simply be spoken, written with a pen, or indicated by 'slicing' a bar at a certain height via touch.

In the future, it would also be interesting to conduct formal user studies to investigate and compare the two main approaches, integrated views versus multi-view, in more detail. In order to understand the direct influence of these approaches, they should be compared for the same representation, e.g., the matrix. In a second step, it can then also be of interest to investigate the differences across different device types to better understand the responsive aspects of them.

Going beyond Single-device Usage In conclusion, this chapter indicated how a suitable data analysis interface can be provided for more complex data types in general, but also how it can be adequately provided on various devices. Having such a flexible visualization approach is also a basis for data explorations in environments with more than just one computing device. At the same time, utilizing multiple devices can allow for overcoming some constraints that cannot be equalized by the responsive behavior alone. For example, in a multi-view approach, the other views would not have to be sacrificed anymore but could simply be placed on another device—or one that is better suited for hosting this view. The idea of utilizing device ensembles for data analysis while specifically considering the devices' suitability for the currently present views or tasks is at the core of the following thesis parts.

Part III

Analysis Workflows
in Dedicated Device Ensembles

5

Self-contained Ensemble: Smartwatches with Interactive Strap Displays

This part of the thesis aims at extending the understanding of how the combination of devices can help to support data analysis. Specifically, the goal is to identify and incorporate different device roles within these combinations that can facilitate exploration workflows and presentation approaches in visual data analysis. In combination with existing work (see Chapter 2, Section 2.5), this can allow deriving general cross-device mechanisms, which can then also be applied within dynamic device ecologies. The research reported in this part consists of two separate investigations. The first one is focused on a very personal, dynamic, and small-scale combination where a smartwatch is extended with interactive strap displays. This is the topic of this chapter. Then, in Chapter 6, the complementing combination of a smartwatch with a large display is investigated, looking at a scenario with extreme differences in device characteristics and multi-user capabilities.

Fig. 5.1.: Watch+Strap ensembles extend a smartwatch with interactive StrapDisplays allowing to, e.g., (a+b) expand information spaces, (c) offer quick access menus, (d) provide glanceable information, or (e) display high-resolution content. *Videos and further material are provided at* `imld.de/watch+strap`.

In general, multi-device ensembles often contain mobile devices, which thus have a special importance for device ecologies. Consequently, the device ensemble that is considered first, is focused around the smallest and most personal mobile device, the smartwatch. Due to the positioning on the wrist, smartwatches are always available for the wearer and feature an unobtrusive design and special familiarity. However, at the same, their input and output capabilities are limited. This can be tackled by transforming the watch itself into a multi-display setup: Here, the “Watch+Strap” setup is introduced, where the watchstraps of the smartwatch are becoming an additional input and output channel (Figure 5.1).

Such a Watch+Strap ensemble is self-contained, i.e., is facing one person as one unit, although consisting of multiple displays. Interestingly, the incorporated displays are characterized by very different properties that have to work together in synthesis. Further, being a wearable device, special usage styles and various display positionings have to be considered. With respect to visualization, wearable devices in general are becoming more popular as smart companions allowing for quickly reviewing visualized data. However, the device size puts up notable limits for displaying and interacting with content. A Watch+Strap device can ease these limitations while maintaining the physical form factor of a smartwatch.

The specifics of the Watch+Strap ensemble are presented in this chapter. First, an extended background on research for extending output and input spaces of smartwatches is provided (Section 5.1). Then, the main contributions are described in Section 5.2: The novel Watch+Strap combination itself (5.2.1), a *conceptual framework* proposing interface principles for a Watch+Strap system (5.2.2) plus a *modular research platform* (5.2.3) featuring multiple prototypes and a flexible web-based software architecture. In addition, specific examples for both general and visualization-related applications (Section 5.3), as well as insights from brainstorming sessions and expert interviews (Section 5.4) are presented.

Parts of the research presented in this chapter have previously appeared in the following publication:

Konstantin Klamka*, **Tom Horak***, and Raimund Dachselt. “Watch+Strap: Extending Smartwatches with Interactive StrapDisplays”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, pages 72:1–72:15. **The first two authors contributed equally.*
Citation key: [KHD20].

Own Contribution: My main contribution to this work is centered in the conceptual framework as well as the software architecture of the research platform. However, the whole research process was coined by joint work and discussions on all aspects, which is also reflected in the shared first authorship.
Applied Changes: The published content was re-used for this chapter to a large extend. However, parts on the physical design space as well as the hardware aspects of the research platform were notably reduced. In turn, the considered application examples have been extended with additional visualization-specific ones (Section 5.3). Also, the reporting on the brainstorming session has been extended (Subsection 5.4.1). Further, changes to the structure as well as an adapted introduction and discussion were incorporated.

5.1 Extending the Input and Output Space of Smartwatches

In the background chapter (Chapter 2), smartwatches and wearable devices were only briefly introduced. Therefore, the following section will provide an overview on literature focused on smartwatches, particularly research that aims at extending the *output space* and/or the *input space* of the smartwatches.

Extending the Output Space of Smartwatches

Besides the display size and form, commercially available wearables can differ notably with respect to the display technology and output qualities. *Hybrid smartwatches* combine integrated smart electronics with mechanic watch parts, however, have no output capabilities (i.e., relying on an associated smartphone) or only limited ones, e.g., small integrated LEDs [XL15] or e-ink screens [@Fos19; Olw18]. Similarly, *fitness trackers* come in a compact format with only small, low-resolution screens integrated (e.g., Fitbit, Microsoft Band). In contrast, *smartwatches* often feature high-resolution displays in either rectangular or circular shape, but remain limited to the size of classical watches. Finally, some startup companies [@Gla14; @Kai14; @Mom14] have been experimenting with curved strap displays or accessories over the last years. However, none of them has yet resulted in a widely available product. Further, a few e-ink-based bracelets and watchstraps have emerged that are commercially available (e.g., Tago Arc [@Lbe18] or Sony FES Watch U [@Son18]), but these lack any input channels or support for interactive content, thus only serve as unique fashion accessories.

On the research side, different approaches have been investigated on how to extend the output space or how to alter the appearance of smartwatches. One direction involved concepts for *transformable smartwatches*. These ranged from reconfigurable tangible dual-face smartwatches [SYV16], over providing holographic mid-air visuals [Wen+17], to origami-inspired design concepts for foldable structures with multiple on-wrist displays [FSS18; ZFÜ18]. However, such transformable concepts remain hard to realize. Extending the *watchstraps* with visual capabilities is more feasible. This has also been explored before [OWS14; SBH17], but without incorporating pixel-based displays. For example, WRISTBAND.IO [SBH17] used tiny LEDs to communicate information, while PRINTSCREEN [OWS14] demonstrated how electroluminescence segments can be used for simple notifications. For truly pixel-based displays, a few concepts for novel *display assemblies* have been proposed instead. Lyons et al. [Lyo+12] jointed multiple displays to a bracelet and mapped generic apps to these, while Olberding et al. [Olb+13] arranged displays along the forearm. Both are valuable design explorations, but remain limited to their particular setups, which have different characteristics than a Watch+Strap ensemble.

As another research direction, the output space was extended along the forearm. For this, one approach is *skin projection*, where UI elements are projected around a smartwatch. Such elements can range from buttons [Lap+14] to notifications [Xia+18] to full-sized content that extends the watch [Gru+15]. Instead of projections, *e-textile displays* can be used for the same purpose [SOA16]. All of these approaches rely on additional instrumentation of the user and are thus not easily deployable. Finally, continuous *sleeve displays* have been discussed [BSV15; SBV15; Zad+14], which could replace smartwatches. The increased display space promises advantages, e.g., for list content [SBV15], and allows for using them similarly to smartphones.

Extending the Input Space of Smartwatches

Almost all commodity smartwatch devices come with touch capabilities and, in some cases, with additional physical input capabilities, such as buttons or rotatable controls (e.g., a rotatable bezel or crown). The latter can, for example, allow for scrolling or alternative keyboard inputs [Yi+17].

Similarly to the output space, the *watch straps* have been used to extend the input space, e.g., for recognizing simple gestures [Per+13], pressure-sensitive touch input [Ahn+15], or specifically text entry [Fun+14]. With such input capabilities, the straps were used for back-of-band interaction that can avoid occlusion issues while interacting [BC09; McI+19; SBH17]. The touch input does not have to be limited to the watch or the straps, but could also be performed on the skin around the device [HBW11; OTI15; Sri+17; Zha+16]. As further gesture-based interactions, specific arm and finger movements were proposed for mode switches or triggering commands, either in the form of mid-air gestures above the watch [Kim+07], or as arm movements [Gon+18a; Sun+17]. Beyond these mostly established interaction concepts, alternative interaction approaches such as stretching the watchstrap [Vog+17], using small everyday items as tangibles [Gon+18b], or gaze-based interactions [Est+15] have been presented.

In summary, while a rich spectrum of research around smartwatches exists, a thorough investigation of the technical aspects of pixel-based straps and applicable interface principles for the combined use have not been proposed yet.

5.2 The Watch+Strap Display Ensemble

With Watch+Strap, we[1] intend to thoroughly explore the design space of *interactive* strap displays *in synergistic combination* with smartwatches. More precisely, we first consider the physical aspects of StrapDisplays (e.g., display type, physical appearance, input modalities) and then investigate the digital aspects of the resulting multi-display interface (i.e., how to distribute, visualize, and interact with content). One important part of this is also a modular research platform, for which we built three functional Watch+Strap prototypes and implemented a flexible web-based software architecture. Notably, the here discussed aspects remain independent from specific applications for now, thus, are not per-se tuned towards data analysis. This remains left for Section 5.3.

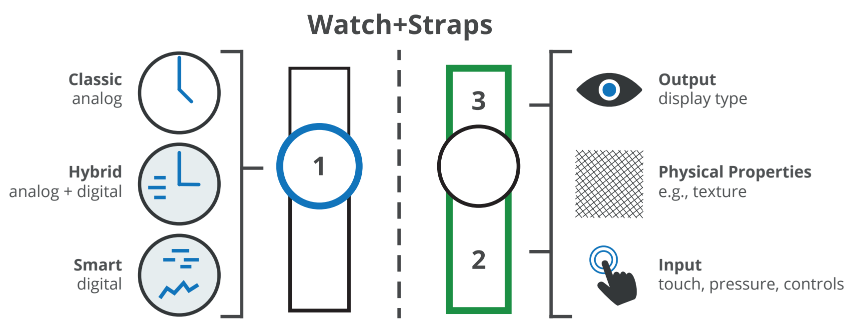


Fig. 5.2.: Wrist-worn watches can be classified in a three-zone scaffold consisting of a watch (1), an inner strap (2), and an outer strap (3). All three can feature different levels of input and output capabilities.

5.2.1 Physical Design Dimensions

The combination of interactive StrapDisplays with a smartwatch forms a distinct system that we call Watch+Strap. For both the StrapDisplays and the watch, a variety of specific designs exist, which we characterized in our *physical design space*. As the basic scaffold (see Figure 5.2), we assume that the watch (1) is always attached to two straps, an inner (2) and an outer one (3). All three components can feature different degrees of input and output capabilities as well as different haptic and aesthetic qualities. In the following, a brief overview of every part and the relevant design options is given. However, for a more detailed discussion of the StrapDisplays capabilities, see the original publication [KHD20].

[1] "We" in this chapter relates to the author Tom Horak, as well as Konstantin Klamka and Raimund Dachselt as co-contributors to this research.

As the central component in a Watch+Strap system, the watch type and its degree of interaction and display capabilities must be considered. On the lower end, *classic watches* do not support any digital functions and therefore take on a passive—but not less interesting—role; here, the StrapDisplays can bring interactivity to the watch while preserving its classical character. *Hybrid watches* already aim to combine these aspects, often providing simple output capabilities that support the indication of, e.g., tracked fitness data. In this context, StrapDisplays are able to extend the interaction capabilities without having to change the watch itself. As the most versatile watch type, high-end *smartwatches* are characterized by touch-sensitive high-resolution displays with a rectangular or circular shape, running diverse applications. In consequence, smartwatches are of special interest to Watch+Strap systems as they enable synergistic and highly dynamic multi-display concepts.

StrapDisplays offer many design possibilities that can significantly alter their usability. A highly important dimension for the design and integration is the underlying display technology, its visual properties as well as technical capabilities. As outlined in the previous section on related work, watchstraps can feature single point displays [Jen+19; OWS14; SBH17], which has advantages regarding their simplicity. However, pixel-based thin-film display technologies (e.g., e-ink or OLEDs) provide the opportunity to visualize dynamic content. Due to their technology, e-ink displays do not emit light, provide high contrast by sunlight, and are able to hold static content without electricity. However, they are typically limited in their color range and refresh rates. In contrast, emerging bendable OLED screens provide a full-color space and fast refresh rates. Both display types are considered for StrapDisplays.

The input capabilities of StrapDisplays can be enhanced in multiple ways. By default, we consider multi-touch and pressure-touch input. However, it is also possible to integrate tactile membrane landmarks or proper physical controls, such as buttons. In addition to touch input on the surface, also the strap edges could be made touch-sensitive [OL14]; the resulting physical guidance can be beneficial when, e.g., moving or interacting eyes-free. In addition, the haptic qualities of the straps can influence the aesthetics and user acceptance significantly. For example, the smooth display surfaces could also be overlaid with sophisticated materials and be used via cutouts or shine-through effects (cf. Klamka and Dachselt [KD17]).

Finally, it must be noted that the visibility and reachability of StrapDisplays play an important role for the interaction, usability, and overall acceptance. However, these aspects are significantly influenced by the wearer's hand posture (cf. Burstyn et al. [BSV15]), which changes during both intentional arm rotations when focusing the watch and natural movements when on the go. While this provides promising opportunities for context-sensitive interface adaptations, it limits which parts of the straps can be directly incorporated. In most situations, the outer strap is pointing away from the body while the inner strap is oriented towards the wearer's body. This constellation provides a semi-public display that is directed to the outside, thus, hardly visible for the user; and a more private, always visible and reachable display directed to the inside. Consequently, it could also be possible to incorporate an asymmetric setup of the straps, for example, using an e-ink display for the outer strap, and an OLED for the inner strap.

5.2.2 Conceptual Framework

In the following, we contribute a conceptual framework for the self-contained display ensemble created by a Watch+Strap system. The framework aims to provide ways of how an interface can be designed by considering common *interface components* as well as discussing possible *content types and their arrangement*. This further includes a suitable *interaction repertoire* for the Watch+Strap setup. Of particular interest is also the influence of the Watch+Strap's special usage style as a body-worn device and how it can, e.g., support *glanceable usage*. Finally, this subsection will also discuss the resulting *display roles* of the ensemble.

Interface Components

For a Watch+Strap interface, multiple relevant interface components exist. *Quick access and function keys* are wide-spread across all types of systems and allow for easily triggering specific functionalities (e.g., media control, camera, home screen). In recent years, also context-aware quick access keys have emerged, both in research systems [BGV10; GTV15] and commercial products (e.g., Apple's Touch Bar). While commodity smartwatches are lacking space to offer such quick access functionalities, here, the additional StrapDisplays can be utilized to act as second screen providing context-aware quick-access controls (Figure 5.3a, Figure 5.1c). These controls can be applied within both an in-app context and a system-wide context. For instance, the StrapDisplay can offer common functions within an app itself (e.g., save, cancel, add) or additional system-wide functions (e.g., switch to last app, screenshot, app drawer) that the user can trigger via touch.

Clipboard and user-defined storage (e.g., favorite list, bookmarks) represent dynamic quick access components. In contrast to fixed function buttons, users actively manage these lists and add or remove elements, e.g., via context menus or special touch

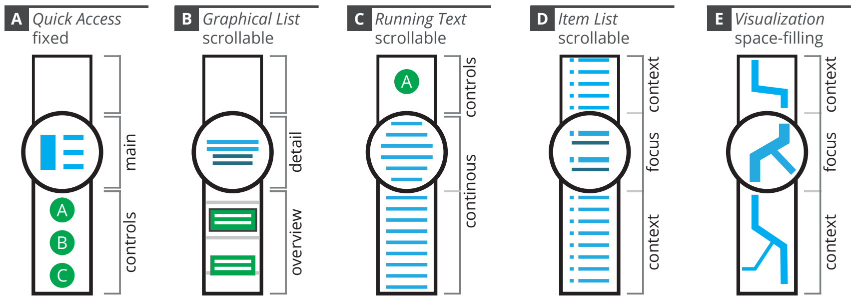


Fig. 5.3.: The variety of possible interfaces allows for different distribution strategies across the three displays; a-e show exemplary interfaces.

gestures such as flicking elements from the watch onto the strap display. Afterwards, the user can scroll through the stored item stack and invoke the respective quick access functionality (e.g., pasting copied item) by touching an element.

Showing relevant *status information* is one of the most important use cases in the context of smartwatches. This can involve showing, e.g., progress indicators, notifications, location-based information, or simply the time. In most cases, this status information is non-interactive or serves only as quick access to the respective app. This information is well suited to be shown on the inner StrapDisplay: firstly, this allows using the watch as a main display for other apps, while, secondly, the orientation of the inner strap towards the user enables quick glances onto the information as well as maintains a higher privacy level with regard to people in close proximity. In contrast, sometimes, specific information has to be shared with the outside; here, we propose to incorporate the outer strap as a semi-public display. We will detail these aspects in the *Glanceable & Ambient Usage* section.

Content Type & Arrangements

Naturally, the content type can vary: In the simplest form, there can be *fixed content*, as it is often the case for quick access menus. In the context of a Watch+Strap system, *scrollable content* is particularly interesting, as the straps' form factor makes them suitable for lists or lengthy content [SBV15]. Typically, the scroll direction would be towards the watch, where the scroll container could be limited to the strap or extends onto the watch (Figure 5.3b-d; Figure 5.1c). The content itself can be running text, lists with text or simple representations, as well as visualizations (e.g., line charts showing heart rate over time). However, especially for text-based content, the usability heavily relies on the actual width and pixel density of the display. Further, also more graphical representations are possible (Figure 5.3b). For example, a calendar grid view can consistently preview the following hours with events moving in and upwards over time, allowing to quickly peek at the remaining time until the next event (Figure 5.3b). In general, the scrollable content can be 'endless' as in such a calendar view.

Finally, the content type can also be *space-filling content*. This involves map views (Figure 5.3e), space-filling visualizations (e.g., Table Lens [RC94]), images, or videos (Figure 5.1e). Beyond photos and movies, the latter two can also extend to specific (animated) wallpapers. For all space-filling content, the possible aspect ratio of the content itself, as well as whether it is acceptable to crop parts of the content, heavily influences the placement, i.e., whether the watch or the StrapDisplays are more suitable.

With the different content types also different content arrangements can be realized in a Watch+Strap system. The two extremes are keeping multiple components strictly separated by putting one per display (Figure 5.3a) or stretching one component continuously across all displays (Figure 5.3d+e). In between, multiple possibilities exist of how to combine component types and/or displays. In the following, we adapt two common presentation approaches in visual data analysis, overview+detail and focus+context, as general interface setups. Naturally, these are especially promising for data-driven content.

For *overview+detail* [RC94; HF01], the interface consists of two separated views: a high-resolution detail view and an overview showing the complete content, sometimes in an abstracted fashion (Figure 5.3b). In the simplest case, the overview can be an item list while the detail view shows a selected item with extended details or in a different way. As more advanced examples, the StrapDisplay can show visualizations such as a step histogram or a stock chart. Selecting a bin or a specific point of time would then show detailed information for this selection on the watch. This can also allow for probing a visualization by continuously moving the finger across the chart. In the given examples, the StrapDisplays are hosting the overview, while the watch is used to show the detailed, high-resolution view.

Similarly to overview+detail, *focus+context* [BGS01; RC94; HF01] shows a certain part of the displayed content in a more detailed fashion (i.e., the focus). In contrast, the focus and context area are one continuous component with adaptations in the focus area (Figure 5.3d+e; Figure 5.1a). Specifically, using the watch as the focus area is similar to applying a magic lens [Tom+16] to the content. The applied transformation can include adding textual details, emphasizing specific details (e.g., highlighting POI in a map), or applying zoom effects (e.g., enlarging list items [Bed00]; see also fisheye lens in general [Fur86]). For example, a list would be extended from the watch to the strap displays with items in the watch area shown with further details. Scrolling the list would move other items into the focus area (scrolling would not directly affect the detail area for overview+detail). For a Watch+Strap system, the watch is often suitable as a focus area because of its central position and uniform size.

Interaction Repertoire

For both the watch display as well as the StrapDisplays, established touch interactions can be used (Figure 5.4c+d). These interactions are well known from mobile devices in general and, thus, are already familiar to users. In addition, physical controls, such as a rotatable bezel or buttons, are often offered by smartwatches and can serve as valuable extensions to the touch interactions (Figure 5.4c). More interestingly, the interaction style with a Watch+Strap system differs from handheld mobile devices as the former is used in very different postures (Figure 5.4a+b). Further, we propose to extend the interaction repertoire with additional StrapDisplay-specific interactions (Figure 5.4e-g).

Interaction Style & Posture When purposely interacting with the device, the user's arm is typically bent inwards and rotated, so that the watch is oriented towards the user (Figure 5.4a). This posture affects the visibility of the StrapDisplays: while the upper half from the inner strap is easily visible, only a small portion of the outer strap is visible. Hence, the inner one is suitable for displaying and interacting with content. Here, the content on top is more prominent, as the content below fades away (through the display's curvature) and is eventually not visible anymore. However, the user can quickly rotate the arm further to peek the content on the lower half; also the strap itself is always easily reachable. Notably, when performing short interaction sequences while the watch is not in focus, the inner strap is even easier to reach than the smartwatch itself (Figure 5.4b).

For the outer StrapDisplay, it is hard to increase the visible area as rotating the arm inwards is against natural movements (cf. [TH01, p. 16]). Thus, the strap's suitability for showing the user content is limited. However, it can still be comfortably reached with the other hand and, thus, serve as an additional input channel.

StrapDisplay-Specific Interactions Due to the limited display space, multi-touch gestures, such as pinch, are hardly applicable. However, the Watch+Strap display arrangement allows for performing combined multi-touch gestures on both straps (Figure 5.4e). For instance, touching the outer and inner strap at the same time

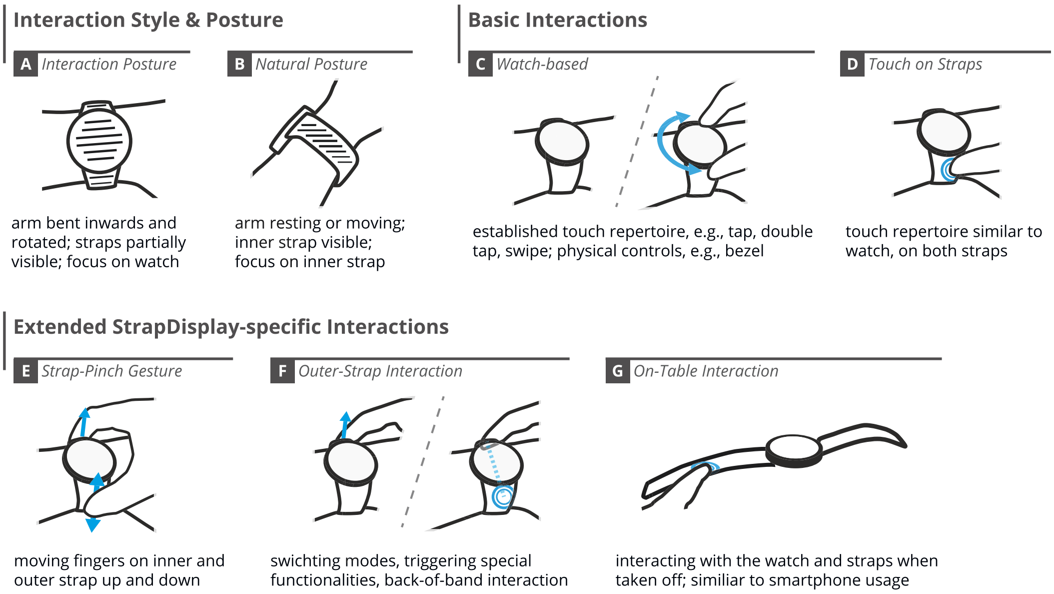


Fig. 5.4.: Our interaction repertoire considers the changed interaction style (a+b), basic interactions (c+d), and further display-specific techniques (e-g).

(similar to a pinch gesture) while moving the fingers up and down can serve as an alternative zoom gesture [Per+13]. This gesture also does not occlude the watch display, which is a notable advantage compared to pinch gestures performed on the watch. Further, one strap can also be used for mode switches. For example, swiping on the inner strap can be mapped to scrolling by default, but switches to zooming when touching the outer strap at the same time.

The outer strap can also serve as a panel for back-of-band interaction [SBH17] (Figure 5.4f). Instead of interacting directly with the content on the inner strap (causing occlusion), the interaction can be performed on the other side, i.e., behind the content. While tapping on the exact position can be difficult, fluid interactions such as scrolling can be easily performed. Further, distinct gestures for controlling application-specific or system-specific functions can be used. For instance, for a music app, simple gestures known from headsets can be used to control the player, e.g., pausing by tapping, skipping a song by double tap or horizontal swipe, or adjusting the volume by swiping vertically. Similarly, gestures for navigation tasks can be used, such as double-tapping-and-scrolling for zooming as known from mobile map applications.

StrapDisplays are more prone to non-intentional interactions than a smartwatch on its own, for instance, when users want to take off the watch, re-position it for comfort reasons, or are just wiggling on the straps while deep in thoughts. As the StrapDisplays allow for interaction during different postures (Figure 5.4a+b), the posture is not applicable for filtering unintentional interactions anymore (as it is done with current commodity smartwatches). Besides incorporating physical buttons, another alternative is using pressure-sensitive touch input, where users have to apply a certain level of force while touching. However, finding reasonable threshold levels is not straightforward and is beyond the scope of this work. Finally, it is also possible to use a Watch+Strap device similar to a smartphone on a table when taken off (Figure 5.4g). With all displays fully visible, we envision that the device can serve as a fully-fledged mobile system on its own, for example, enabling interactive exploration tasks (Figure 5.1a), advanced text entry, media consumption (Figure 5.1e), or effective multi-tasking when handling multiple apps.

Glanceable & Ambient Usage

An important characteristic of smartwatches is the possibility to quickly glance over it without requiring interaction. However, as mentioned before, the arm has to be bent and rotated in order to comfortably see the watch display (Figure 5.4a). In contrast, the inner strap is almost always fully visible during natural postures or movements (Figure 5.4b), e.g., when walking, sitting at a table, holding on to a steering wheel, or carrying a glass. We propose to explicitly exploit this for unobtrusively displaying information that the user can quickly glance at (Figure 5.1d). Notably, the reading direction can change in these situations, i.e., while the inner strap is used in portrait orientation during a focused interaction, here it can also be more useful to use it in landscape orientation or to partially rotate the content [BSV15]. Further, the displayed content is hard to recognize for others, promoting a certain privacy level.

In such a usage context, the Watch+Strap system is suitable to either support a primary task or provide additional information that might get interesting at some point. This is similar to ubiquitous computing research, where glanceable and ambient usage define a short (couple of seconds), mostly passive interaction with content that conveys information en passant [Bla+21; Gou+16]. For instance, while running, we propose to provide relevant measures such as time, speed, or distance on the inner strap, which the user can glance at without altering the natural arm orientations. Independently from other activities, more general information can be shown, e.g., notifications, progress of achievements, or contextual information. This information is not limited to be shown as plain numbers, but can be embedded in other (animated) representations [Ami+17; Gou+16; Isl+20]. These could be similar to existing examples such as pulsating activity indicators [Kay+05] or whereabout clocks [Bro+07; Sel+06].

Further, such an ambient usage is not bound to showing personal or private information, as the outer strap can be used as a semi-public display and can more easily be shown to other people compared to taking out a smartphone. For example, it can allow representing user-specific states such as group affiliations, participants' spoken languages during network events, or mood indicators in daily situations. This information could be encoded through, e.g., patterns, colors, or textual descriptions. Similarly, during events or in secured buildings, specific access credentials could also be provided as a QR-code. Finally, the StrapDisplays can also serve as an aesthetic, stylish accessory by showing passive designs while idling. This can range from imitating structures or materials, to matching cloth colors, to showing abstract patterns, images, or photos (Figure 5.1e). However, these do not convey a specific information but serve a pure design or entertainment purpose.

Display Roles within the Watch+Strap System

In the previous sections, the described interface concepts made use of the distinct multi-displays setup of a Watch+Strap system, i.e., incorporated the displays in different roles within the interface. In the following, we discuss and abstract these roles further into a more generalized characterization of the Watch+Strap concept and its resulting strengths.

In general, the three displays can be used either in a separated way or as one continuous surface; in between, a continuum exists of how strongly coupled or decoupled the displays are. In the extreme case of a functional separation, the content shown on one display does not relate to the others in any way, e.g., applying one app per display or keeping specific displays intentionally blank. In the other extreme, handling the displays as one unit, the content spreads across the displays in order to provide a visual continuity, e.g., to maximize the visible parts of lengthy content or to just provide a special aesthetic appearance. The bigger chunk of our interface concepts is located in between these two extremes (cf. Figure 5.3). Tending towards the extreme of functional separation, we presented concepts where separated interface components are placed across the displays, but provide additional functionalities within the same application scope (e.g., quick access menus, overview+detail). Towards the other side of the continuum, we proposed schemes where the displays act as one visual unit but still address different user needs; focus+context arrangements are one example for these schemes. In particular the overview+detail and focus+context strategies can help to address the need for presenting a larger information space within an overall limited screen estate, as it has already been discussed in the previous two chapters.

Further, as a result of the typical posture during interactions (Figure 5.4a), the watch remains the main display across most interface constellations while the StrapDisplays extend or accompany it. In addition, the StrapDisplays' form, curvature, as well as orientation to the user emphasize the upper part of the display, while the lower part is gradually curving away until it is not visible anymore. Thus, the available display space cannot fully be used at once. However, it also allows to naturally resemble the importance of content parts, i.e., by putting important content closer to the watch and non-important content towards the strap end. Similarly, the StrapDisplays' orientation makes them suitable for different specific roles. For instance, the inner strap is best suited for showing private data during interaction or glanceable content when on the go. In contrast, the outer strap is mostly suited as a pure input channel enriching the available interaction repertoire or as a communication channel to people in close proximity. This consideration of visibility and reachability for content is also one form of responsiveness that becomes relevant for non-planar displays.

5.2.3 Open Research Platform

To thoroughly investigate and test our concepts, we built a modular research platform that allows us to remix strap and watch assemblies with different properties and technologies. Specifically, we built three working prototypes (Figure 5.5). The platform is publicly available, including additional details on the used hardware, the source code, all 3D models, as well as building instructions [@KHD]. In the following, a condensed overview on the platform is provided. Further explanations are provided in the original publication [KHD20] and on the project webpage [@KHD].

Hardware Architecture

For our system, we used primarily a Samsung Gear S3 (⌀ 46 mm; [@Sam16]), and a Samsung Gear S2 (⌀ 40 mm; [@Sam15]) for early testing. Both have rotatable bezels, two additional hardware buttons and a high-resolution (302/278 ppi) multi-touch display. For the StrapDisplays, we focus on pixel-based screens and built three fully-functional prototypes using two bendable e-ink displays (Figure 5.5a), a bendable grayscale OLED display (Figure 5.5b) and a tablet-based prototype that represents two full-color, high-resolution StrapDisplays (Figure 5.5c).

The *e-ink StrapDisplays* featured ultra-thin, bendable 2.13" displays that could show black and white content with a resolution of 212 × 104 px. These e-ink displays also supported partial refreshes, allowing to update the screen without flickering in

Fig. 5.5.: Our research platform currently involves two StrapDisplays (a+b) and a tablet prototype (c). The (a) e-ink displays are paired with touch sensors and put into a housing and connected to the required controllers. The (b) OLED watchstrap is assembled in a similar way. The (c) tablet prototype consists of a stencil cover placed on the full-color displays of a tablet.

many situations. Our *OLED watchstraps* featured bendable 4-bit grayscale displays, measuring 1.81" with a panel consisting of 160 × 32 px. For both types, we designed curved housings in which the displays were put in (Figure 5.5a). While we also 3D-printed bendable straps, we decided to use more stiff straps for our studies since the available electronics and displays have a maximum bending radius. The driver board and all other necessary parts are housed in an external case (Figure 5.5a).

Controller-wise, we decided to build on the popular Arduino ecosystem to make our open research platform user-friendly, extensible, and suitable for further research by minimizing entry barriers. Specifically, we use an Arduino-compatible ESP8266 [@Esp16] System-on-Chip that is capable of connecting to a WiFi network, managing WebSocket messages as well as supporting our different extensions such as touch sensors. To circumvent unintentional touch inputs, we decided to integrate pressure-based touch sensors requiring a small force for triggering actions. Specifically, we used multiple membrane potentiometers (Figure 5.5a) that were placed behind the flexible displays.

We also wanted to evaluate the potential of full-color, high-resolution strap displays. However, the still emerging bendable OLED technology is not yet available for prototyping. To circumvent this, we realized an interactive *tablet prototype* based on a stencil cover with 3D-printed straps and the S3 smartwatch that we placed on a tablet computer (Figure 5.5c). In this case, a web application ran on the tablet and the touch input was directly consumed from its display.

Software Architecture

In order to realize early prototypes as well as application examples, we implemented a web-based prototype system with a central Node.js server controlling both the watch and the StrapDisplays. The system follows a thin-client principle, i.e., both the watch and the StrapDisplays consume the pre-rendered content from a server and forward all input information to this server; all communication is handled via WebSockets. We opted for this setup to avoid having distributed applications with separated deployment processes for each device. Hence, the watch and the microcontroller only receive image data that they can directly flash onto their respective displays (cf. Holman et al. [Hol+13]). In order to further simplify the development process, all the main application logic is provided via a web application hosted on the server. Although this means that the application must be opened on an additional device, it allows for making full use of established web frameworks that are not always available as pure Node.js applications.

More specifically, we use the D3.js library [BOH11] in combination with the SSVG library [Sch19]. While D3 allows for creating interfaces and visualizations in particular, SSVG transforms the SVG element used by D3 to a canvas with offscreen rendering during run-time while still mapping input information to the correct D3 element. From the three offscreen canvases (one for each display), we can then extract the current image data and send it to the corresponding displays. The web application also outputs the canvases directly, providing a live preview of the running system as well as a quick testing environment. Further, the application also allows for defining which StrapDisplay type is currently used, i.e., setting the required resolution and color mode. The examples are implemented as app-like modules and can be loaded via an app drawer.

5.3 Applications & Visualization on Watch+Strap

In the following, the interface concepts are illustrated in the context of specific application examples. First, general but promising examples are discussed, featuring a music player app as an elaborate example and multiple smaller examples for on-the-go usage. Then, the focus is put on personal information visualization applications. Here, an activity tracker is presented as a main example, alongside smaller examples for displaying nutrition charts and governmental voting casts. With the exception of some touch gestures, the main examples are implemented fully functionally for the e-ink and tablet prototype, while the others are provided as mock-ups with limited functionality.

5.3.1 General Application Examples

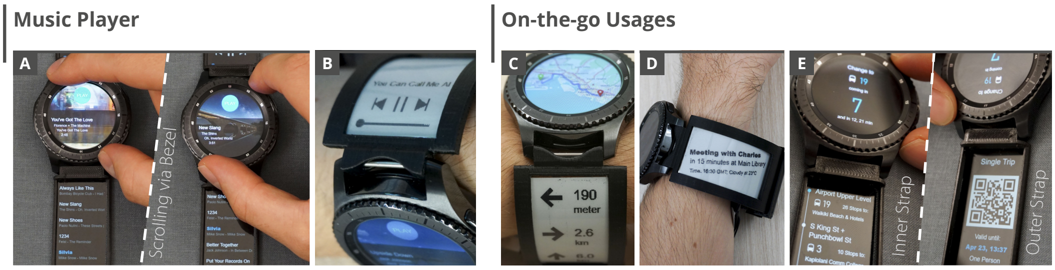


Fig. 5.6.: General application examples illustrating our interface concepts: (a+b) music player and (c–e) on-the-go usages.

Music Player As a very common application for mobile devices in general, the first example represents a music player. One major activity within music apps is browsing through different lists, e.g., genres, albums, playlists. Here, we propose to show these lists on the combined space of the watch display and the inner strap with a focus+context arrangement, where items in focus on the watch are shown with additional details (Figure 5.6a). For example, in a playlist, songs on the strap are shown with title and artist name; on the watch, also the duration, album name, and album cover are shown. Scrolling through the list can be controlled by touch on the strap or the watch, as well as by rotating the watch bezel (Figure 5.6a). Items can be selected by touch on both displays. Further, performing a hold on an item opens a context menu with additional functionalities (e.g., adding song to playlist, show artist page, share list). On top of the outer strap, controls and a progress bar for the currently played song are shown (Figure 5.6b). In addition to controlling the player via these buttons, the lower part of the outer strap can be used for gestures, e.g., swiping horizontally for loading the previous/next song, vertically for volume adjustments, or tapping for pausing.

On-the-go Usages As a Watch+Strap device is a body-worn device, it is highly suited for on-the-go usage. To illustrate such usage, we created a collection of application snippets, which are implemented as mock-ups. As already indicated, due to its orientation, the inner strap is highly beneficial for displaying information on the go, such as directions, notifications, or transportation instructions (Figure 5.6c–e; Figure 5.1d). This information can be either shown as text or as simple, glanceable visualization [Bla+21]. With regard to public transportation, it could also be an interesting possibility to use the outer strap as a semi-public display by showing ticket information (possibly encoded as QR-code) for inspection (Figure 5.6e).

5.3.2 Visualization-specific Examples

Activity Tracker Fitness and activity trackers belong to the most commonly used applications on smartwatches [Isl+20] and represent an interesting example for personal data visualization. Here, we implemented an example for tracking and reviewing sport activities, specific for runs. The app initially shows some brief statistics about the recent runs on the watch, a list of all activities on the inner strap, and a quick access button for starting a new activity on the outer strap. The latter one switches the app to a streamlined interface, where only live data during the run is shown (e.g., pace, distance, time; Figure 5.7a). These numbers are displayed in a larger font and are rotated into a landscape orientation in order to improve readability while running (glanceable usage). In our current prototype no sensor data is used, instead, the data of an existing run is replayed.

Selecting a run from the start screen of the app shows a detailed view. On the watch, the route, total time, and duration are shown, while the inner strap displays sparklines [Tuf06] for pace, heart rate, and elevation alongside their average values (Figure 5.1b). These sparklines are one instance of responsive behavior, as these charts can also be shown in a higher resolution. Specifically, by tapping on the inner strap the sparklines are transformed into a full line chart visualization (rotated by 90 degrees), where the three attributes are overlaid on the strap (Figure 5.7b). In addition, the watch shows an enlarged sub-part with specific values. This combination resembles an overview+detail arrangement. The shown part on the watch can be adjusted by either tapping on the strap or by rotating the bezel.

Nutrition Chart & Voting Visualization Besides assigning the display roles (overview, detail) based on the different display qualities, also their shape and size can be considered. For example, the circular display of the watch is particularly suited for

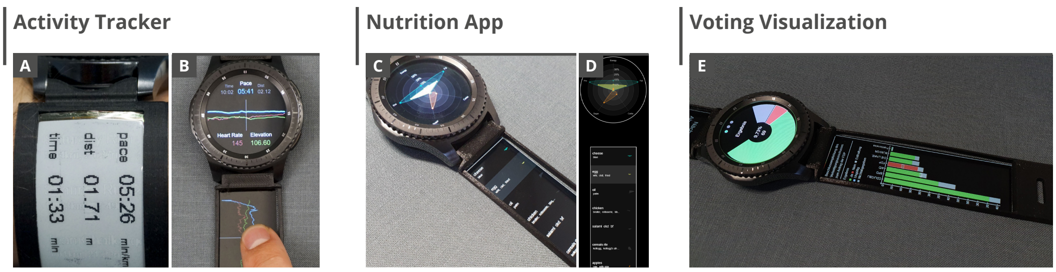


Fig. 5.7.: Visualization application examples: (a+b) activity tracker, (c+d) nutrition app, and (e) voting visualization.

radial visualizations, while the straps are appropriate for hosting oblong visualizations or views. Specifically, within a nutrition application, the watch display is used for showing a star plot that visualizes selected food items, while the strap hosts the full list of all available food items. This list does not only consist of text, but also includes a small preview of the corresponding star plot for every item. This allows for quickly scrolling through the content and identifying items of interest. By tap, these are then selected and overlaid in the larger star plot on the watch. A second example for utilizing the display form factors is a voting app. Here, a pie chart indicates the overall voting behavior on the watch, while the inner strap shows a more detailed bar chart where votes are split up by, e.g., party (Figure 5.7e). This is another example where the strap display hosts a more detailed visualization, while the watch provides a simplified overview.

5.4 Brainstorming Sessions and Expert Interviews

The development of our Watch+Strap concepts and platform was a highly iterative process. Within this process, we created various low-fidelity and high-fidelity prototypes that also allowed us to test the conceptual aspects and eventually resulted in the open research platform. More specifically, we conducted brainstorming workshops with students and HCI researchers at an early stage, while later in the process we ran expert interviews. In the following, we report on both of these formats.

5.4.1 Early Brainstorming Sessions

After deriving first concepts, we conducted two types of brainstorming sessions in parallel to validate our early explorations and to better identify further user interface patterns: an unsupervised sketching diary and workshop-like brainstorming sessions in groups. Both types involved paper prototyping techniques that we also used among the authors to iterate and discuss our own ideas. Specifically, we created a postcard-sized template with the contour of two Watch+Strap devices, allowing us to easily sketch and discuss ideas in a fixed format (Figure 5.8a–d).

For the unsupervised sketching, we asked three lab members (age M=30.33 yrs, SD=7.37 yrs; all male; 2 smartwatch owners) to create multiple sketches of how a StrapDisplay could be used. We provided them with brief instructions on the envisioned device and the procedure of the experiment. Afterwards, they were given two weeks for producing sketches. For the on-site brainstorming sessions, we invited

Fig. 5.8.: In the brainstorming sessions, participants created sketches (a–d); afterwards, the clustered groups (e) helped to inform our concepts.

12 students from our university (age M=20.5 yrs, SD=2.24 yrs; 3 female, 9 male; 5 owned a smartwatch) and conducted 90-minute supervised workshop-like sessions. The participants were assigned to one of four groups beforehand (3 students each); the session with these groups were conducted within a few days. Procedure-wise, we provided each group with four sample applications (calendar, music, activity tracking, navigation) alongside matching screenshots from existing smartwatch applications. Then, we asked them to discuss and sketch ideas for bringing these on a Watch+Strap device. These discussions were moderated by one investigator, while a second investigator recorded participants' statements in writing. For all sessions, we provided the printed template cards and collected them afterwards.

We clustered and sorted the comments and sketches (Figure 5.8e). First of all, all participants found the envisioned setup interesting, with most of them directly commenting on the importance of the visual appearance. For the specific utilization, the collected sketches illustrated that participants thought of a wide range of applications, also going beyond the ones we proposed, e.g., general text messages and notifications, smart home controls, app drawer (Figure 5.8b), or navigation instructions (Figure 5.8c). Within their interface sketches, it also became apparent that mostly the inner strap and the watch were considered as suitable parts for providing content. However, this does not mean that the outer strap was not considered. Here, participants proposed to use it for, e.g., touch gestures, flashlight use, or as public displays encoding a group affiliation via color or showing status information (Figure 5.8a). In general, these insights helped us to ground our own considerations, which were mostly inline with participants take. Consequently, the brainstorming sessions were one part that informed the development of our concepts.

5.4.2 Follow-up Expert Interviews

After we had developed concepts and prototypes, we conducted combined semi-structured interviews and hands-on sessions. Participants were provided with the e-ink prototype as well as the tablet prototype, which both ran the implemented music player and activity tracker examples as well as the on-the-go usage examples. In total, we invited six participants (P 1–6; age M=30.7 yrs, SD=5.39 yrs; 2 female, 4 male; 3 wear smartwatches). Three of them (P 3–5) participated in the earlier brainstorming sessions (two are members of our lab), the other participants were external HCI researchers: participant P 1 was an HCI professor at a University of Applied Sciences, participant P 2 was a post-doc working on visualization design, and participant P 6 was a 3rd-year PhD student also focusing on interactive visualization. The sessions lasted one hour and consisted of an introduction, walkthroughs, and discussion of our implemented apps and mock-ups, as well as a concluding questionnaire. For the walkthroughs, first the e-ink, then the tablet prototype was used. Also, we gave participants small tasks within the different apps (e.g., selecting a specific song) to foster the engagement with the examples and prototypes.

Results All participants successfully interacted with our prototypes without facing major issues. In particular, those who participated in the brainstorming sessions (P 3–5) stated the high-fidelity as useful. For the provided examples, a few missing functionalities were mentioned (P 1–3, P 5), e.g., list sorting, search functionalities, or specific touch gestures.

During participants' interaction with our apps, we could observe multiple interesting aspects. When browsing the music player's playlists, some participants (P 2, 3, 5) focused on the strap display (it showed multiple songs at once), while others (P 1, 4, 6) mostly focused on the watch because of its richer detail level. The latter three also commented on the required gaze switches between watch and inner strap. Due to the notable physical gap between strap and watch, they found this to hinder the perception of the displays as a continuous content container. However, they also believed that this effect can be eased by reducing the gap. In the context of the activity tracker, all participants liked the idea of showing glanceable information on the inner strap while on the go; participants P 2 and P 4 explicitly mentioned the privacy aspect of it. Extending on this, the idea of further rotating the arm to access more content on the inner strap was mentioned multiple times (P 1, 3, 4, 6) as well as the advantages of posture-sensitive content placement (P 3–6), e.g., positioning and rotating content to maintain readability (cf. Burstyn et al. [BSV15]). For the outer strap, all participants acknowledged the limited suitability for content placement; however, half of them (P 2–4) explicitly mentioned the possibility for 'blind' or back-of-band interactions, and for semi-public display usages (P 1, 2, 6).

Participants also agreed that the main applications will remain checking notifications, tracking health and fitness aspects, and providing status information (P 1 explicitly emphasized showing the time). Some participants also mentioned further application examples, e.g., using the outer strap as flashlight (P 2 + 6) or running parallel apps (P 5). As the feasibility of some apps depends on the display characteristics (e.g., color, resolution, brightness), we asked participants for their preferences. Most of them (P 1, P 3–6) believed that full-color OLED displays are the best option because of their display quality and the low latency (P 3, 4, 6). However, participant P 2 preferred e-inks because of their advantages regarding readability in sun light and battery life. Relevant to this are also the overall ergonomics and aesthetics, with all participants agreeing that it is a major criterion. For example, it was stated that the strap width should be constrained to available watchband widths (P 1, P 3–5). In context of the StrapDisplays' interchangeability, participant P 1 could imagine to switch them between different watches (e.g., dress-watch, everyday watch). Consequently, possible lifestyle usages, such as showing aesthetic design patterns, were also mentioned (P 1, P 4–6).

5.5 Discussion and Summary

By combining commodity smartwatches with StrapDisplays, they can be enhanced into a synergistic Watch+Strap system. As a result, a watch can also take on more elaborate tasks independently. By outlining the physical design dimensions and proposing multi-display interface concepts, a systematic exploration of this device class is provided. Further, the brainstorming and expert sessions showed that both manifold usage scenarios and interest in such devices exist.

Enhancing Watch+Strap For the Watch+Strap concept, there are many ways of how the setup can be investigated further. For example, the relation to different wrist sizes was not considered, i.e., how a changed circumference affects the displays' bending radius and, thus, their visibility. One aspect mentioned in the expert sessions was that some people place the watch on the inner arm side, which would also affect the interface. Further, one expert also discussed the potentials of the buckle, e.g., it could be placed asymmetrically up to the point where only one strap display is needed, or it could host an additional physical button. In general, a Watch+Strap assembly strongly builds on the familiar appearance of a classical watch and aims to maintain their appeal and aesthetics. However, future work could explore variations of this ensemble, e.g., incorporating a rectangular smartwatch that allows for minimizing the gaps between displays, or pairing the StrapDisplays with analog watches.

One major aspect for further developments are technical improvements. While the research platform demonstrated the basic feasibility, the further miniaturization of all components (e.g., batteries, sensors, processors) remains challenging, especially with respect to device thickness, heat generation, and power consumption. Nevertheless, it is more than likely that this will become possible in the next years; as a case in point, the instrumentation of watchbands has also been described in recent patent applications [Car+19]. While not including pixel-based displays yet, this highlights that the Watch+Strap approach is highly timely and should be investigated further. Specifically, the StrapDisplay concept could be evolved into a modular platform providing straps with different capabilities and aesthetics. With such flexibility, Watch+Strap devices could emerge as more versatile smartwatches taking on information needs that are currently reserved for phones, while still preserving the familiar and well-liked form factor of wristwatches.

Coordination of Displays This exploration also highlights how different displays can work together and how these can potentially support data analysis tasks—here, in particular for personal data explorations. First of all, the research shows that the different device characteristics can notably affect what content is best placed where, i.e., here either on the watch display or on the straps. Having different characteristics then also simplifies distributing and combining visualization views that provide very different representations, e.g., a tall line chart onto the strap and a square bar chart onto the watch. Particularly in an arrangement like Watch+Strap, the adjacency of displays also allows for applying strong coordination between the visualizations, such as focus+context. As already mentioned in the discussion on display roles in Subsection 5.2.2, the strength of a coupling between displays or devices can be seen as a continuum, defining if two devices are complementing each other or acting as one unit. This thought is especially of interest in the context of dynamic device ecologies and will be revisited in Chapter 7.

At the same time, it has already been indicated that smartwatches are also a promising device for accompanying other devices. For example, Chen et al. [Che+14] paired a smartwatch with a smartphone or Brudy et al. [Bru+16] with a tabletop. Sharing the motivation of these investigations, the next chapter will show how a smartwatch and a large interactive display can be combined into a complementing display ensemble and address specific analysis tasks. Although not considered in the following chapter yet, a Watch+Strap device could further extend the interaction and visualization possibilities of the used watch.

6 Complementing Ensemble: Smartwatches plus Large Display

As anticipated at the beginning of the thesis, the combination of multiple devices promises to offset their weaknesses and create new synergies by merging their strengths. For large displays, the strengths are the extensive screen estate and possibilities for multi-user scenarios, while reachability [Liu+17] and providing personalized content is challenging [Zad18]. Vice versa, smartwatches are limited by their size but are highly personal. Thus, the devices represent two extremes—like David and Goliath—of interactive surfaces in many ways (e.g., small vs. large, private vs. public, mobile vs. stationary). This yields several fundamental design challenges for their combination. The goal of the present work is to investigate this complementary combination specifically for data analysis by looking at how specific analysis workflows can be efficiently supported (Figure 6.1).

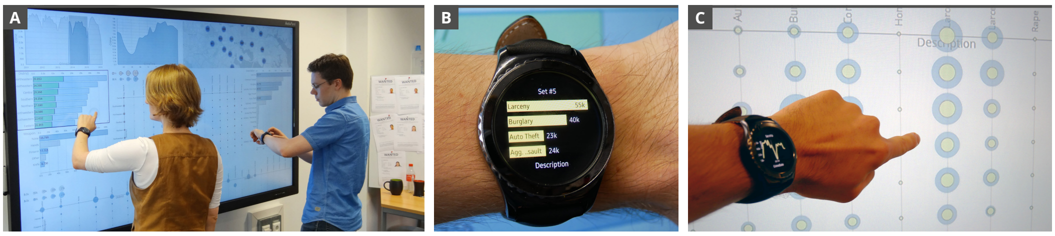


Fig. 6.1.: Visual data analysis using large displays and smartwatches together (a). Cross-device interaction workflows discussed in our conceptual framework allow for an interplay between these two types of devices. The watch enhances the large display by acting as a user-specific storage (b), a mediator (c), and a remote control. It further aids multiple users to either work in concert or by themselves. *Videos and further material are provided at* `imld.de/david-meets-goliath`.

While the combination of hand-held mobile devices and large displays has been studied before (see Chapter 2, Section 2.5), the focus on smartwatches allows incorporating the specific advantages that they have over traditional handheld devices. As illustrated in the previous chapter, a wearable device is not only more lightweight and non-intrusive but also provides anytime access without the need for persistent handheld usage. Moreover, it also leverages proprioception for eyes-free, on-body interaction [Ash+08; Rek01]. This characteristic also applies the other way around, when considering a combination with other devices: established and familiarized workflows on the large display are in no way affected; instead the smartwatch offers the possibility to enhance these workflows in an unobtrusive way. Given these advantages, the combination with large displays is compelling, yet this idea has so far not been explored in the literature. Specifically, the smartwatch is not only used to provide a personalized view but serves as a personal analysis toolbox. In this function, the watch supports the multivariate data exploration on a large display interface containing multiple coordinated views.

The chapter is structured along the contributions of the work. First, an example data analysis scenario is discussed (Section 6.1), illustrating the requirements and possibilities for such a setup. Then, the main part is presented (Section 6.2): First, generalized design considerations are discussed (6.2.1) that serve as the foundation for the actual concepts. These concepts are embedded into a conceptual framework (6.2.2), defining the specific interplay between the smartwatch and the large display for a single-user. Within this framework, users can interact with the large display alone and also benefit from the watch as a container to store and preview content of interest from the visualizations as well as manipulate view configurations (Figure 6.1). While collaboration is not explicitly considered yet, the concepts allow for simultaneous (parallel) work of multiple users during the visual data analysis. Finally, the framework was realized in a prototype system (Section 6.3) and the concepts were evaluated through two experiments (Section 6.4), a formative evaluation and a summative user study, with an explicit focus on occurring interaction patterns.

The research presented in this chapter has previously been published in:

Tom Horak*, Sriram Karthik Badam*, Niklas Elmqvist, and Raimund Dachselt. "When David Meets Goliath: Combining Smartwatches with a Large Vertical Display for Visual Data Exploration". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, pages 19:1–19:13.
**The first two authors contributed equally.*
The publication received an **Honorable Mention Award**. Citation key: [Hor+18b].

Own Contribution: Overall, the major contributions by the first two authors were equally and apply to all parts and aspects of the work.
Applied Changes: The changes to the original content incorporated here are minor and involve mostly adaptations for connecting the presented ideas to other parts of the thesis and adjustments of the headings. Further, the introduction and discussion were updated.

6.1 Scenario: Analyzing Crime Data on Large Displays

To better understand the requirements of visual data analysis, as well as to illustrate and validate our interaction concepts, we[1] consider an application scenario of a law enforcement department planning patrol routes within a city. For that, we build on a real dataset of crimes in Baltimore [@Bal11]. Here, we will describe the scenario and its involved users, their goals, the setup, and the challenges.

Consider two police analysts trying to build a tentative plan for patrol routes based on historical crime data within the city. Their goal is to design routes that cover as much as possible of the high-crime areas while still maintaining a police presence throughout the entire city. The analysts meet in an office space that has a large digital whiteboard featuring a high-resolution display and multi-touch support, as seen in Figure 6.1a. As outlined at the beginning of this thesis in Chapter 2, such large displays have become increasingly popular for visual sensemaking scenarios since they enable analysts to work in concert or on their own, view the data from a distance or up close, o even leave the room and continue their exploration later [BE17; Bra+13; Kis+15; LKD19]. In our law enforcement scenario, the analysts use standard visual analysis techniques [Rob07] to construct an interactive dashboard on the large display. This interface is capturing the attributes in crime data by incorporating different visualizations, such as line charts, histograms, or scatterplots. A similar interface has later also been used by Langner et al. [LKD19].

To actually create the patrol plan, the analysts need to observe the crime distributions in these different visualizations. Now, to identify in-depth characteristics of the city's crimes, analysts need to investigate multiple hypotheses over different crime patterns of interest. For instance, to evaluate effects of crime prevention measures in certain districts they have to visually verify if downward tendencies are present. These tendencies could exist in an overall trend, but also only for a few districts, crime types, or certain time periods. This sensemaking task by itself involves multiple visual exploration tasks [BM13; Shn96; YKS07]: selecting data items (i.e., crimes) of interest, filtering them, accessing more details about these crimes (elaborate), encoding them on visualizations for other attributes, connecting them across visualizations, and comparing multiple collections of crimes. This exemplifies how, similar to other visual analysis scenarios, crime analysis is also centered around working with data items—collections of crimes—of analysts' interest. During sensemaking, multiple such collections have to be considered in parallel threads of visual analysis and by groups of analysts in collaborative scenarios.

[1]The use of "we" in this chapter refers to the author Tom Horak, as well as Sriram Karthik Badam, Niklas Elmqvist, and Raimund Dachselt as co-contributors to this research.

The large display can provide multiple views on the shared large screen real estate to support multiple visual perspectives and help users utilize the space. However, this is not enough; analysts need to deal with two types of challenges. (1) **Display space management:** when interactively exploring the crime records on the large display, analysts need to develop spatial memories of visualized information when seeing or comparing multiple parts of the large display. Also, adding further views for comparison is not possible when the amount of space is fixed and already taken by other views. (2) **Interaction management:** at the same time, they also need to keep track of the visualizations for multiple crime collections over time to fully develop their insights. Beyond this, the users should be able to manage their personal focus (views of interest) as well as data points of interest within the focus, and to access interactions to explore these points without affecting other users. Further, these interactions should not be bound to the display, instead they should be accessible from both close proximity and distance, e.g., to examine visualizations in detail or from an overview distance.

6.2 When David meets Goliath: Smartwatches and Large Display for Data Exploration

To support such scenarios, we propose to combine a smartwatch with a large display, where the smartwatch acts as a personal toolbox that can support analysts with the display space management as well as interaction management. In the following, we present first design considerations that serve as the basis for the conceptual framework proposed afterwards.

6.2.1 Design Considerations

For the illustrated visual data analysis, we need a platform to view data records, store them as separate groups, and compare groups to each other. Further, the platform should support modifying visualization properties to make comparisons more effective. To answer these challenges, we use secondary devices to augment visualization components, enhance user interactions, and ease the visual exploration. Specifically, we incorporate personal smartwatches into the environment in order to take advantage of the unique characteristics of such wearables [Ash+08; Rek01; Zad+14]. Here we explore the design space of combining smartwatches and large displays to allow for cross-device interaction in visual data analysis.

Roles of the Devices

Each device in our cross-device setup—smartwatch and large display—has a specific role during visual analysis. By virtue of its size and affordance, the large display serves as the primary display that provides multiple coordinated visualizations for a multivariate dataset. As described in Chapter 2, touch interaction can be supported and applied on different levels: the data level (selecting data elements), the view level (sorting the data by swiping on an axis), or the layout level (moving a view via drag). Thanks to its size, the large display can also be used by multiple analysts in parallel, thus serving as a *public and shared display*.

In contrast, the smartwatch is a personal—and significantly smaller—device, only used by its owner. Consequently, the watch is suitable as a secondary device, but can take on different roles. Given the challenges when using the large display in the crime analysis scenario, the secondary device should keep track of the user's interaction activities and corresponding data items. The device can therefore act as a *user-specific storage*—a container for points of interests or parameter settings—that can be easily accessed at any time. This role can further be extended by allowing the user to manage the stored content on the watch itself (e.g., combining, manipulating, or deleting content items). In the interest of managing the available display space while supporting multiple users, the secondary device enhances the interaction capabilities to support a wide range of exploration tasks. The smartwatch can serve as a *mediator* (cf. CurationSpace by Brudy et al. [Bru+16]), i.e., defining or altering system reactions when interacting with the large display. This mediation can happen in both an active and a passive way: either the watch is used to switch modes, or it offers additional functionalities based on the interaction context and the user. Finally, to flexibly use the space in front of the large display, the smartwatch can also be used as a *remote control* and allow the users to interact from a distance.

Elementary Interaction Principles on the Smartwatch

Generally, the smartwatch supports four types of input: simple touch, touch gestures, physical controls, and spatial movements. As the analysts mainly focus on the large display during exploration, the input on the watch should be limited to simple, clearly distinguishable interactions, which can also be performed eyes-free to reduce attention switches (cf. Pasquero et al. [PSS11], Zadow et al. [Zad+14]). Therefore, we propose to primarily use three interactions on the watch (see Figure 6.2a-c): *swiping horizontally* (i.e., left or right), *swiping vertically* (i.e., upwards or downwards), and, if available, *rotating a physical bezel* of the smartwatch. For

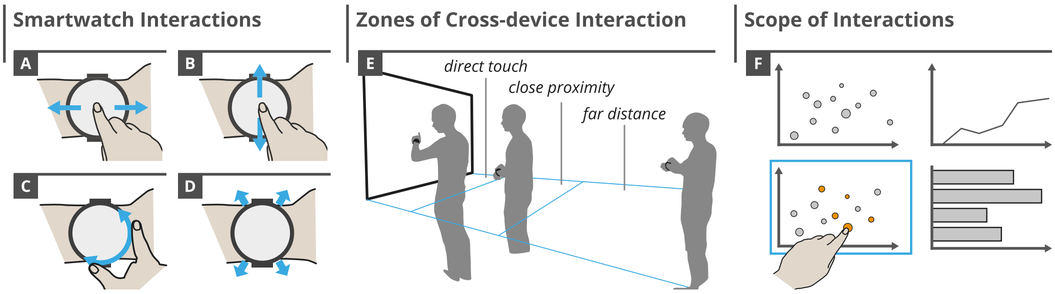


Fig. 6.2.: Primary smartwatch interactions: (a) swiping horizontally, i.e., along the arm axis for transferring content; (b) swiping vertically or (c) rotating a physical control for scrolling through stored content; and (d) moving the arm for pointing interaction. Further, (e) the cross-device interaction can happen with direct touch, in close proximity, or from intermediate or far distance. For all user interactions, (f) the scope is limited to the views in focus.

advanced functionalities, long taps as well as simple menus can be used. Finally, using internal watch sensors, the users' *arm movements* or poses (Figure 6.2d) can be used to detect different states or to support pointing [Kat+16; Rek01; Wil+16]. Such a smartwatch-based pointing has also been explicitly investigated in an own follow-up work [AHD18].

When the smartwatch takes the role of user-specific storage, we assume that users have a mental model of two directions for transferring content; towards the smartwatch or towards the large display. Based on this, a specific axis of the smartwatch can be derived: The *proximodistal axis* (i.e., along the arm) is suitable for transferring content; swiping towards the shoulder (i.e., left or right, depending on the arm on which the user wears the watch; Figure 6.2a) can pull content from the large display onto the smartwatch. Vice versa, swiping from the wrist towards the hand, i.e., towards the large display, can allow to push content back to the visualizations. Additionally, the *axial axis* (i.e. orthogonal to the arm) can be defined as a second axis [Zad+14]. We suggest scrolling through the stored content by either swiping vertically (Figure 6.2b) or rotating the bezel or crown of the watch (Figure 6.2c).

Zones of Cross-Device Interaction

In general, the cross-device interaction can happen in three zones: either at the large display using *direct touch*, in *close proximity* to the display but without touching it, or from intermediate and even *far distance* (Figure 6.2e). We expect analysts to work directly at the large display most of the time, thus the touch-based connection is primarily used. As the users' intended interaction goal is expressed in the touch position, i.e., defining on which visualization (part) the analyst is focusing, the smartwatch should act as a mediator and incorporate this knowledge to offer or apply functionalities. In contrast, the remote interaction enables the analysts to work without touching the display, possibly even from an overview distance or while sitting. As the contextual information of the touch is missing, the user has to perform an additional step to select the view of interest, e.g., via pointing.

Existing work on physical navigation [Bad+16; BNB07; JH14; LKD19] illustrates that working from an overview distance, close proximity, or directly at the large display is not an either-or decision. There is always an interplay between the three: analysts interact in front of the large display to focus on details, step back to orient themselves, and again move closer to continue exploration. Consequently, the cross-device interaction should bridge these zones. For instance, an analyst may first work near the large display and perform interactions incorporating the watch (e.g., store data selections). Then, stepping back allows to continue the exploration from a more convenient position as well as to analyze other views on the large display based on the stored data.

Scope of Interactions in Multi-User Setups

In common multiple coordinated view applications, changes in one visualization (e.g., selection, filter, encoding) have a global impact, i.e., they are applied to all displayed views. As discussed in our motivating scenario, this behavior may lead to interference between analysts working in parallel. To avoid this issue, the effects of an interaction should by default only be applied to the visualization(s) currently in focus of the analyst (Figure 6.2f). Further, we also propose to constrain the scope of an interaction mediated by the smartwatch to a short time period. More specifically, when touching a visualization to apply a selected adaptation from the smartwatch, the resulting change is only visible for a few seconds or as long as the touch interaction lasts. Yet, there also exist situations where changes should be applied permanently, i.e., merged back into the shared visualization [McG+12]. Therefore, it must be possible to push these adaptations to the large display and keep the altered data visualization.

6.2.2 Conceptual Framework

By incorporating the different roles of the smartwatch and the large display, our conceptual framework supports a multitude of tasks during visual exploration [BM13; YKS07]. In the role of a user-specific storage, the smartwatch provides access to

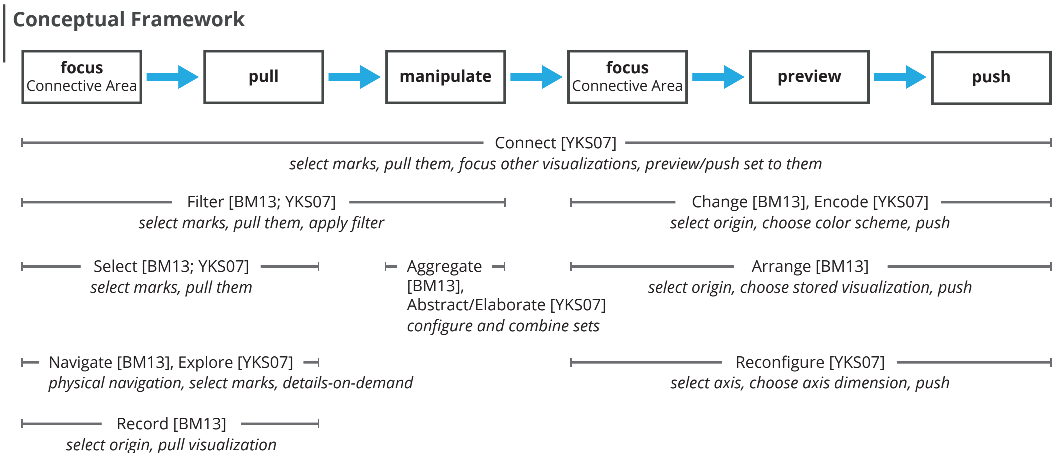


Fig. 6.3.: Our framework addresses a wide range of tasks, here illustrated by mapping two established task classifications [BM13; YKS07] onto interaction sequences that are enabled by our framework (examples in italics). For some tasks, certain aspects are also still supported by the large display itself, such as zooming and panning. For the typology by Brehmer and Munzner [BM13], their *how* methods are focused (similar as in Chapter 2, Subsection 2.3.1).

the data, i.e., points of interest. Both the shared large display and the smartwatch (as remote control) determine or define the context of an interaction. Regarding the task topology from Brehmer and Munzner [BM13], the combination of these two aspects—data and context—represents the *what* of an interaction and enables the smartwatch to act as mediator defining the *how*. This mediation enables the analyst to solve a given task deriving from questions raised in the scenario (*why*). Our framework provides components that blend together into specific interaction sequences and address the various task categories, as indicated in Figure 6.3. In the following, we will introduce these components and describe their interplay. We will also reference the matching tasks from Figure 6.3 in small caps (Example).

Item Sets & Connective Areas

The primary role of the smartwatch is to act as a personalized storage of *sets*. We define *sets* as a generalized term for a collection of multiple entities of a certain type. In our framework, we currently consider two different set types: data items and configuration properties (e.g., axis dimension, chart type). These sets can also be predefined; for instance, for each existing axis dimension, a corresponding set is generated. On the smartwatch, the stored sets are provided as a list. As shown in Figure 6.4, each set is represented with a description, a miniature view or icon, and further details (e.g., value range). Consistent with the set notion, sets of the same

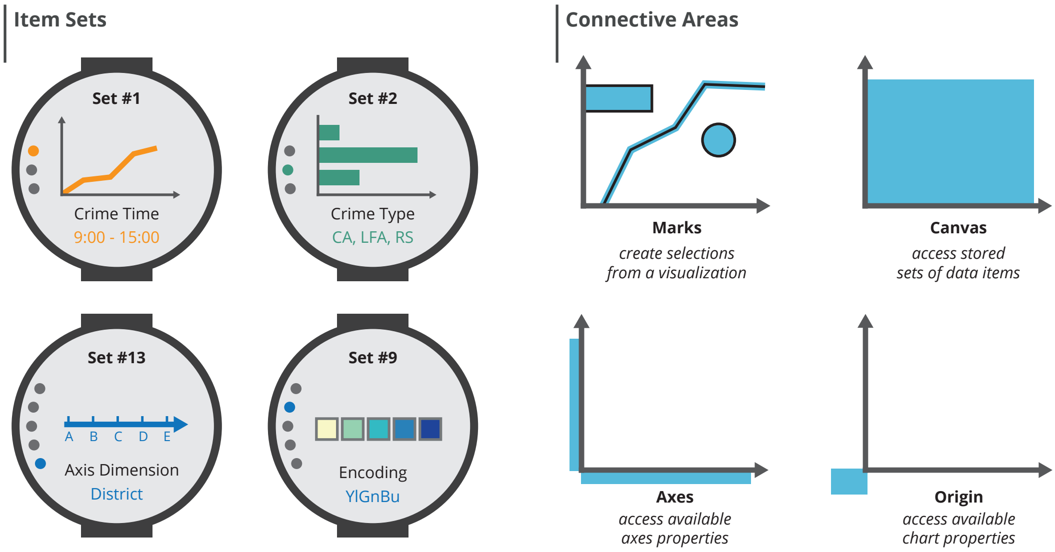


Fig. 6.4.: *Sets* are represented by labels and a miniature: for sets with data items, the miniature is based on the view where it was created (top); for sets containing configuration items, an iconic figure is shown (bottom). *Connective Areas* (CA) represent semantic components of a visualization that have a specific interaction context with respect to a secondary device (a smartwatch in our case).

type can be combined by using set operations (i.e., union, intersection, complement). Finally, to allow managing sets over time, they are grouped per session. Former sessions can be accessed via the watch.

During the data exploration, the region that a user interacts with can provide a valuable indication of the user's intent. We therefore define four zones for each visualization—called *connective areas* (CA)—that will provide the context (*what*) of an interaction: the marks, canvas, axes, as well as a special element close to the origin. Connective areas define the set type (Figure 6.4) and control the functionalities accessed on the two devices. To focus on a connective area, the interaction comprises of tapping or circling marks (i.e., data points) for selection. For other connective areas, users can set the focus in two ways: by performing a touch-and-hold (long tap), the focus is set onto the respective area underneath the touch point but stays only active for the duration of the tap; by performing a double tap, the focus is kept as long as not actively changed. Setting the focus activates suitable functionalities for the specific connective area on the watch. On focus, the stored set content can also be previewed on the large display.

While we consider working in close proximity to the large display as the primary mode of interaction, there are certain situations where this is not appropriate or preferred. For instance, a common behavior when working with large displays is to step back to gain a better overview of the provided content. To remotely switch the focus onto a different view or connective area, the user can perform a double tap on the smartwatch to enable *distant interaction* and enter a coarse pointing mode. As shown in previous work [AHD18], the pointing can be realized by detecting the movements of the watch using its built-in accelerometer. Alternatively, it is also possible to scroll through the visualizations instead of moving the arm. In both cases, the current focus is represented as a colored frame around the corresponding view on the large display.

After confirming the focus, the analyst can select the desired connective area within the focused visualization and then can access and preview stored sets. This remote interaction provides the same functionality as the direct touch interaction. Users can explicitly switch between interaction based on direct touch or on remote access from both close proximity and far distance. This transition could also be extended by incorporating proxemic interactions [BMG10; MG12].

Creating & Managing Sets for Visual Exploration

To develop insights through visual exploration, the interactions in our framework are focused on selecting, manipulating, and previewing data points of interest, as well as applying the previews permanently to a visualization. These interactions are mediated by the smartwatch based on the context of the user. The concepts enabling these four functionalities also define the *how* of the analyst's task.

To **pull** (i.e., create) a set, the analyst first selects marks in the visualization on the large display by tapping or lasso selection, and then swipes towards him or herself on the watch (SELECT). The resulting set is stored on the smartwatch. Now, by again switching the focus to another view on the large display (i.e., by long tapping, double tapping, or pointing), the set currently in focus on the watch is instantly **previewed** on the target visualization. The preview is only shown for a few seconds, or, in the case of long tapping, for the duration of the tap. Depending on the visualization type and the encoding strategy (aggregated vs. individual points), the items are inserted as separate elements or highlighted (Figure 6.5a+b).

While the focus is set on a connective area, the smartwatch can still be used for further exploration. For example, by swiping vertically on the watch or rotating its bezel, the user can switch through the list of stored sets and preview others for comparison. Again, the preview is shown only for a few seconds. To permanently **push** the changes to the view on the large display, a horizontal swipe towards the large display, i.e., the visualization, can be performed on the watch (CONNECT). As

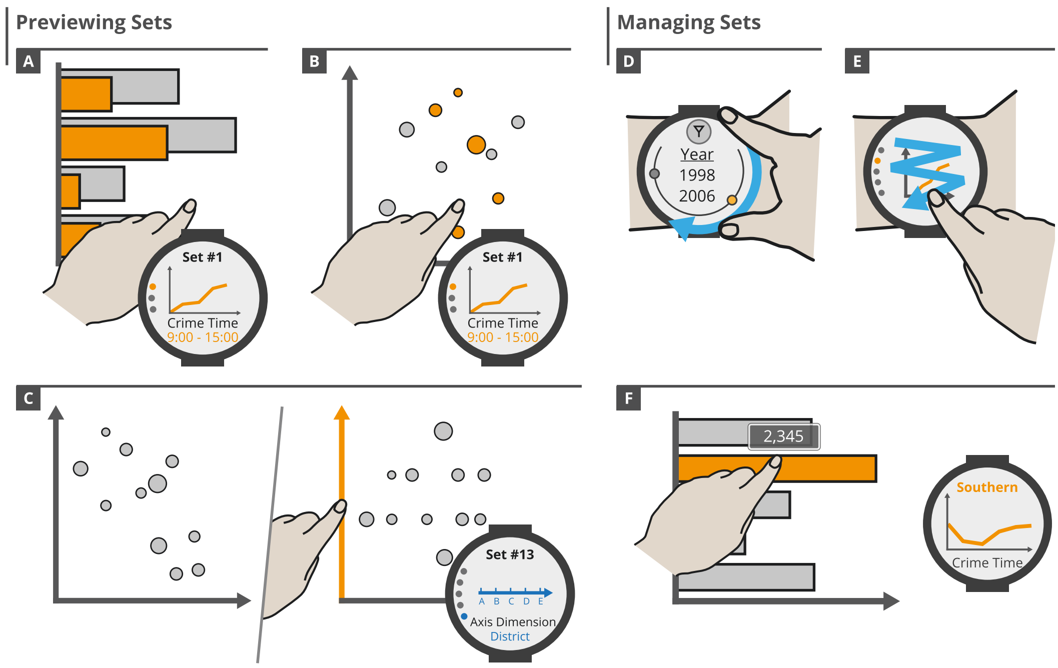


Fig. 6.5.: Previewing stored sets results in (a+b) inserting or highlighting the containing data points in the visualization, or (c) adapting the visualization to the respective configuration item (here: axis dimension). Further, the smartwatch allows to manage sets by: (d) applying filters to data item sets; (e) deleting sets by wiping; and (f) displaying additional details-on-demand.

push is considered a concluding interaction, the system then switches back to a neutral state by defocusing the view.

Besides data items, visualization properties can also be accessed and adapted. Based on the connective areas, we distinguish between axis properties (e.g., dimension, scale) and chart properties (e.g., chart type, encoding). These configuration sets are mostly predefined, as only a limited number of possible values/configurations exist. For instance, when tapping on an axis, all dimensions as well as scales are offered as individual configuration sets on the watch. As with data items, scrolling through this list of sets results in instantly previewing the sets, e.g., the marks would automatically rearrange according to the changed dimension or scale (Figure 6.5c).

By performing a push gesture, this adaptation is permanently applied to the visualization on the large display (Change, Encode, Reconfigure). Naturally, more possibilities for visualization configuration may exist; however, covering all of them is beyond the scope of this work. In addition to single configuration properties, the origin can also provide access to the visualization in its entirety, i.e., a set containing all active properties at once. This allows for storing a visualization for later use, or moving it to another spot in the interface (Arrange, Record).

As an extension to storing sets, the smartwatch also offers the possibility to **manipulate** and combine sets on the watch. By performing a long tap on a set, these operations are shown in a menu. For all set types, this involves the possibility to combine sets based on a chosen set operation (e.g., union or intersection), which results in a new set (AGGREGATE). Sets that contain data items can also be bundled. Previewing or pushing such a bundle shows all the contained sets as overlays at once, thus, merging them on the view itself. Furthermore, it is possible to create new filters and change the set representation on the watch. The filter option allows the analyst to first select a property and then define the filter condition (e.g., crime date in July 2015). For numeric filter options, sliders are provided (Figure 6.5d). To delete a set on the watch, a wipe gesture can be performed (Figure 6.5e).

All in all, the *set* metaphor is ideal for visually comparing multiple regions of interest on the large display because data items can be extracted from the views, manipulated or combined on the watch, and then previewed on multiple target visualizations (CONNECT). The ephemeral nature of our proposed preview techniques enables analysts to explore aspects without worrying about reverting to the original state of a visualization. In addition, the set storage further acts as a history of user interactions, to undo, replay, or progressively refine the interactions [Shn96] (RECORD). During the exploration, the watch can also be used for tasks not involving sets. For example, existing details-on-demand mechanisms on the large display (e.g., displaying a specific value for a mark) can be extended by displaying further details on the watch, such as further attribute values, an alternative representation or related data items (Figure 6.5f; NAVIGATE).

Feedback Mechanisms

For cross-device setups, it is important to consider feedback mechanisms in the context of the interplay between devices, especially to avoid forced attention switches. In our setup, we are able to use three different feedback channels: visual feedback on the large display and on the smartwatch, as well as haptic feedback via the watch. On the large display, the feedback equals the system reaction on user interactions, e.g., previewing content. To further ease the exploration of different sets, a small overlay on the large display indicates the set currently in focus when scrolling through the list on the watch, thus reducing gaze switches between the two devices. The colored frame around a view indicates if a connective area is focused. The watch can hence act as a mediator.

We use haptic feedback, i.e., vibrations of the smartwatch, for confirmation. When successfully performing an interaction, e.g., pulling a set onto the watch or pushing it to a visualization, the watch confirms this by vibrating. Alongside with the small overlays described above, this behavior also supports eyes-free interaction with the smartwatch. Further, the watch also vibrates to indicate that additional information or tools are available on it: While moving the finger over a visualization, the watch briefly vibrates when a new element is hit to indicate that details-on-demand or more functionality are available. To some degree, this also enables to ‘feel’ the visualization, e.g., through multiple vibrations when moving across a cluster of data points in a scatterplot.

6.3 Applying and Realizing the Framework

We applied our conceptual framework by the means of an interaction walkthrough as well as the implementation of a prototype [Hor+18a].

6.3.1 Scenario: Enhanced Crime Analysis

In the following, we revisit the motivating crime data scenario and provide an interaction walkthrough to illustrate our concepts.

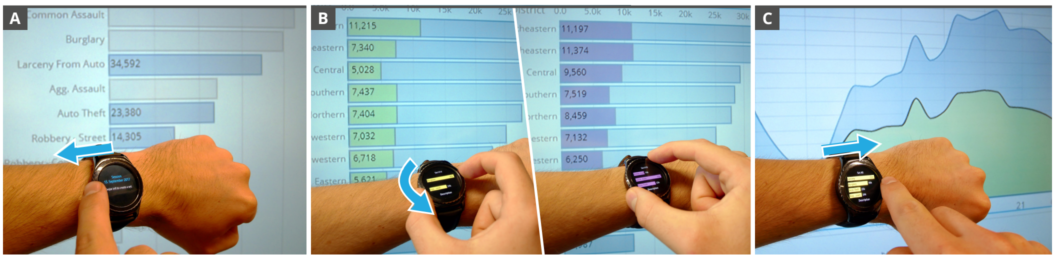


Fig. 6.6.: (a) Pulling, (b) previewing, and (c) pushing of sets.

The first question that one of the police analysts has is whether there are specific high-crime regions within the city over time. She starts by selecting multiple bars representing different types of assaults in a bar chart and saves them into her user-specific storage on the watch by performing a swipe on the watch towards herself (Figure 6.6a). The watch immediately creates a set and represents it with a miniature of the original bar chart and the selected bar. Further, she also selects the corresponding bar of burglaries, and creates another set. As she can carry the sets, she investigates how the assaults occur in various districts. Triggered by double tapping on other visualizations, the smartwatch mediates the interaction and induces the large display to show a preview of the analyst's set in these views. By rotating the bezel of the watch back and forth, she switches between the previews of the two stored sets and compares their distribution on the large display (Figure 6.6b).

She notices that assaults have happened in neighborhoods surrounding Downtown, while burglaries happened more often in specific suburbs. In order to investigate patterns of assaults during daytime, she taps on a line chart to focus on this view and swipes towards the large display. As a result, the current set on the watch is pushed to the focused chart (Figure 6.6c). She continues this process for other crime types (e.g., robberies) by identifying data items and previewing them on other views, while tracking the multiple sets on her smartwatch.

A second analyst wants to evaluate the effects of measures taken in a neighborhood. First, he restores a set of crimes for this neighborhood from a former session via the watch menu. By selecting the crimes for the neighborhood on the large display and pulling them, he creates a set similar to the restored one with current data. To compare them, he pushes both sets onto a weapons histogram and recognizes a drop of crimes with firearms but not for crimes with knifes. By double tapping the axis of the histogram, the smartwatch displays the list of available dimensions, and the analyst switches from weapons to crime types (cf. Figure 6.7a). This allows him to quickly validate his assumption that the drop in firearms is caused by a reduced number of assaults, while the number of robberies is almost unchanged. He can now conclude that the introduced measures only affected assaults.

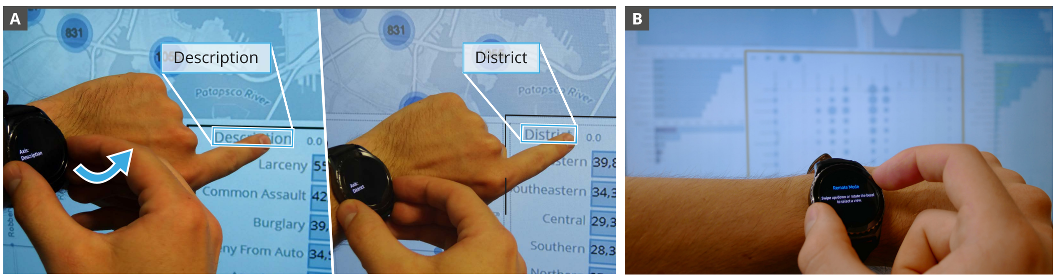


Fig. 6.7.: (a) Changing the axis dimensions, and (b) remote control from a distance to set the focus onto a specific visualization view.

Afterwards, both analysts start discussing their insights and step back to get a better overview of all considered visualizations. The first analyst pushes her stored set remotely to the histogram used by the second analyst. She performs a double tap on the smartwatch, moves the pointer onto the visualization by moving her arm, confirms the focus by tap, selects the canvas (connective area) on the watch, and applies her set (push). They recognize that the patterns are opposed, i.e., assaults dropped in the one neighborhood but raised in the other. With this insight, one analyst leaves to report their observations while the other continues the exploration.

6.3.2 Prototype Implementation

We developed a web-based prototype to instantiate our conceptual framework for demonstrating and evaluating our ideas. For deployment, we used two different large display setups in our respective universities (TUD, UMD), a 84" Promethean ActivePanel (TUD) and a 55" Microsoft Perceptive Pixel (UMD). Both setups used the Samsung Gear S2 smartwatch [@Sam15]. The watch features a rotatable bezel as additional input. All devices connect to a Python server that serves the web-based front-end, handles communication, and performs required data operations. The server also stores the created sets and manages the sessions. Visualizations are developed with D3.js [BOH11]. The dataset contains roughly 250,000 crimes in Baltimore, MD, USA between 2011 and 2016. Each crime within this dataset is characterized by location, date, time, type, weapon, and geographical district.

With this prototype version, we focused on the interaction with data points and sets to test the core principles of our framework. The large display shows bar charts, line charts, scatterplots, and a map to visualize different dataset attributes. In each view, users can select marks by touch. On the smartwatch, it is possible to pull a set from the large display onto the watch as well as preview and push it onto other views on the large display (Figure 6.6). Currently, it is only possible to push one set to a view. Pushing a second set replaces the first one. Both pull and push are confirmed by vibration feedback on the watch. Furthermore, the watch allows to combine sets and to remotely select views by scrolling through the displayed ones on the large display; remote pointing is not yet supported. Also, the current version is not able to distinguish multiple users and to change visualization configurations.

6.4 Evaluation of Concepts and Interaction Patterns

Our evaluation consisted of two studies. First, we ran a formative study in order to collect feedback on our implementation as well as to inform a design iteration of our concepts. Second, we conducted a more extensive user study where we aimed to investigate the interaction patterns that participants follow when using our system compared to a standalone large display interface.

6.4.1 Formative Evaluation: Design Feedback

The formative evaluation took place during our design process, using earlier versions of our concepts and prototype. We detail the study setup in the following.

Participants Five unpaid researchers (4 Ph.D. students, 1 post-doc; age 30–48 yrs; 1 female; 4 male) from our HCI lab (thus, experts in interaction design) at the Technische Universität Dresden (TUD) participated. Three participants focus on visual data analysis in their research, all are familiar with large interactive displays, and one uses a smartwatch on daily basis.

Apparatus and Dataset We used the setup and dataset as described above. The prototype was an earlier version, thus some of the interaction concepts differed from the framework presented here. In this earlier version, the cross-device interactions required the users to persistently touch the large display. For example, to preview a set and to perform a pull/push interaction it was required to long tap the visualization at the same time.

Procedure In each session, we first introduced the participants to our application scenario, i.e., setup, users, and their tasks and goals. Then, we presented our framework and sequentially explained the different techniques in the prototype. We asked participants to try the techniques on their own while thinking aloud. Afterwards, we illustrated further concepts of our framework using figures and discussed possible implications. Ratings were assessed via a questionnaire (Appendix A.1).

6.4.2 Feedback and Iteration of Concepts

Overall, all participants (P 1–5) liked the idea of our proposed setup for visual analysis: for example, they commented that the watch is a multi-purpose device personalized for a single user and—in many cases—available ubiquitously (P 1). Consequently, it can provide access to content in different setups, e.g., first at a desktop for preparation, and then later at the large display (P 4). It could even be integrated further, for example, to authenticate a person when accessing confidential data (P 3). Two participants (P 1 + 4) also noted the advantage of having their hands free for, e.g., performing pen and touch interaction or taking notes.

The feedback also helped us to iterate our concepts. The main concern of the participants was the interface complexity, especially regarding the handling of sets. For example, they suggested to provide functionalities for grouping and sorting of sets on the watch (P 4), which we address now through grouping sets by sessions. We also followed the recommendation to provide an additional description instead of only showing the miniature view for sets on the watch (P 3). Regarding the reconfiguration of visualizations, one participant stressed that the offered possibilities should be limited to a list of presets (P 2). Participants P 3 and P 4 suggested to keep menus for complex adaptations on the large display itself. In general, participants cited our proposed mechanisms for adapting views as a good way to manage user-preferred settings (P 1 + 3) and to support a dynamic view layout (P 4).

Regarding the cross-device interactions, four participants (P 1, 2, 4, 5) positively commented that our techniques already kept forced attention switches between the devices at a minimum. Two of them also stressed the importance of interacting from close proximity and their preference to avoid enforced long taps for the pull/preview/push interactions. They felt that without the need of holding on to the display, a more casual interaction would be enabled (P 1) and fatigue prevented (P 2). We considered these comments in our iteration by introducing the double tap for setting the focus and streamlining the transition between remote and touch-based interaction. For the remote interaction, opinions diverged whether pointing is adequate (P 5) or scrolling through the views with virtual controls is sufficient (P 4), therefore we kept both options as possible concepts. Participant P 3 added that this presumably depends on pointing precision and display size.

6.4.3 User Study: Interaction Patterns

As illustrated in the interaction walkthrough, our conceptual framework has the potential to ease visual exploration. However, the way the techniques are utilized during sensemaking and how they affect the developed observations from the visualized data is not clear. With our study, we aim to further investigate this.

Study Design

We conducted a user study with the dual usage of large display and smartwatch (**Dual**) as proposed by us, against an equivalent single usage of the large display (**Single**) for visual analysis tasks. This allows us to investigate the interaction patterns during visual exploration, and especially how the context-aware smartwatch and in its different roles alters these patterns.

Experiment Conditions The study comprised two conditions: the DUAL condition, where large display and smartwatch were used, and the SINGLE condition facilitating the large display only. In the DUAL condition, the interface allowed participants to: (1) pull data from the large display to create sets (each set gets a unique color), (2) show a preview of sets on target visualizations, (3) push sets to the large display, (4) use the smartwatch as remote control to focus views on the large display, and (5) combine sets on the smartwatch. Except for the last two, equivalent capabilities were created on the SINGLE condition using an freely movable overlay menu with a scrollable set list that appears on long tap. Following a within-subject design, all participants worked with both conditions, with the order being counterbalanced.

Participants We recruited 10 participants (age 22–40 yrs; 5 female, 5 male) at the Technische Universität Dresden (TUD, P 1–4) and the University of Maryland (UMD, P 5–10). Participants were visualization literate with experience in using visualization with tools such as Excel and Tableau; 4 of them used visualization for data analysis (for their course or research work). Two of the participants had already taken part in the formative evaluation (TUD).

Apparatus and Dataset The study was conducted in two setups as described in Subsection 6.3.2. They only differed in the size of the large display (TUD: 84", UMD: 55"). The smartwatch (Samsung Gear S2 [@Sam15]), the prototype version, as well as the dataset (Baltimore crime) were the same.

Tasks We used the crime dataset to develop user tasks that can be controlled for the study purposes. Tasks contained three question types: (**QT1**) finding specific values, (**QT2**) identifying extrema, and (**QT3**) comparison of visualization states [Bad+16]. In general, the complexity of a task results from the number of sets and the target visualizations that need to be considered to answer it. After pilot testing with two participants, we settled on a list of questions with different complexities: for question types QT1 and QT2 the number of targets was increased to create complex tasks, while for question type QT3 both the number of sets and the target visualizations were increased. In the following, sample questions are provided:

1. How many auto thefts happened in Southern district? (**QT1**)
2. What are two most frequent crime types in Central? (**QT2**)
3. What are the differences between crimes in the Northern and the Southern districts in terms of weapons used? (**QT3**)
4. For the two crime types that use firearms the most, what are the differences in crime time, district, and months? (**QT3**)

Two comparable lists with 9 questions each were developed to enable a within-subject study design. The lists and all questions are provided in appendix A.3.

Procedure The experimenter first trained participants in the assigned interface by demonstrating the visualizations and interactions. The participants were then allowed to train on their own on a set of training tasks. Following this, they worked on the nine tasks, answering each question verbally. They then moved on to the other condition and repeated the procedure. Afterwards, they completed a survey on the perceived usability of the two interface conditions, as well as on general interaction design aspects (A.2). On average, sessions lasted one hour.

Data Collected Participants were asked to think out aloud to collect qualitative feedback. Their accuracy for the tasks was noted along with the participant's interactions, movement patterns, as well as hand postures by the experimenter in both conditions. All sessions were video recorded and used to review the verbal feedback as well as noted observations.

Results

After analyzing the collected data, we found three main results:

- The combined usage of large display and smartwatch in the DUAL condition allows flexible visual data analysis patterns.
- Set management tends to be easier in the DUAL condition due to fewer attention switches; thus, simplifying comparison tasks.
- Participants rated the interactions provided by our prototype in the DUAL condition as seamless, intuitive, and more suited for the tasks.

Here, we explain these results in detail within their context.

Interaction Patterns and Observed Workflows As we expected, the interaction abilities of both devices in the DUAL condition and the ability to work from any distance lead to flexible workflows for visual analysis. Therefore, we focused on observing when and how these workflows manifest in our tasks. Table 6.1 lists the observed workflows. In simple tasks within question types QT1 and QT2, participants used the basic touch interaction (long tap, double tap) to preview a set in the target view (workflow **WF1**). Eight participants used physical navigation to move from one part of the display to the other to perform such tasks, while others did this remotely with their watch. For most of them (7 of 10), the long tap action was seen to be sufficient

Code	Description	Pattern	Question Types	Participants
WF1	Set Preview	back-and-forth	QT1, QT2	8 of 10
WF2	Set Comparison	back-and-forth	QT3	8 of 10
WF3	Set Comparison	remote	QT3	3 of 10
WF4	Set Manipulation	remote	QT3	5 of 10

Tab. 6.1.: Overview of observed workflows during the study.

to quickly answer these tasks when only a value or extrema must be determined. For comparisons between two sets (QT3) on a target, eight participants preferred to disconnect from the large display by double tapping it and taking two or three steps back to gain a full view of the target visualization (**WF2**), while only two remained close and used long tap. On the SINGLE condition, it was not possible to step back since participants had to stay close to the display to switch between sets.

In more complex tasks where two or more targets were considered, participants again stepped back to get a better view of the large display in the DUAL condition. While eight participants mostly performed these tasks by moving back-and-forth in front of the display to collect sets and pick target visualizations to make comparisons (**WF2**), three participants (P 7 did both) used remote controls to access target views to avoid this movement to an extent (**WF3**). To track the sets on their smartwatch, four participants held their hand up to view both displays at the same time, while the majority (seven) differentiated sets based on their assigned color. This set awareness was weaker in the SINGLE condition. There, the participants had to shift repetitively their focus between the sets menu and the visualizations to achieve the same. Finally, five participants used the combine option when related sets were already created for previous tasks, avoiding large display interaction (**WF4**).

Overall, we observed that participants followed the pattern of *interact, step back, and examine* (WF1, WF2), as well as *interact remotely* from a distance (WF3, WF4). Further, they often interacted eyes-free with the watch, although the prototype could be further improved in that regard (e.g., by displaying set labels on the large display as more sets are being previewed). The rotatable bezel of the watch was exclusively used for switching sets, thus played an important role acting as a tangible control.

Differences in Developed Insights Workflows WF1–WF4 were observed for different tasks on the DUAL condition. Given these observations, we were interested in the differences in task answers from these workflows compared to their counterpart in the SINGLE condition. In QT1 and QT2 tasks, participants answered accurately on both conditions. However, the SINGLE condition was less preferred, e.g., participant P 1 stated, "*the interaction [in the* SINGLE *condition] was a little complicated and felt slower than with the watch.*" More nuanced patterns existed in participant answers to visual comparison of two or more sets in target visualizations: they made observations about specific values, trend differences in the target, and relative differences in specific data items. To begin with, all participants mentioned specific value-based differences between the sets in the target visualization. To observe trend and relative differences more effectively in the DUAL condition, participants (following workflows WF2 and WF3) made use of the possibility to step back from the large display and to switch back-and-forth between sets with the help of the rotatable bezel on the watch. In the SINGLE condition, participants tried to switch back-and-forth by alternately tapping on the sets in the menu, however, this was more error-prone due to the missing physical guidance. As a result, this forced attention switches between set navigation and visual comparison and required some participants to repeat the interaction multiple times to develop their answers. For instance, one participant (P 10; worked first in the DUAL condition) answered a comparison task (QT3, three sets on two targets) by rotating the bezel between the sets twice for each target, while he switched between the sets five times for each target to make a similar comparison in the SINGLE condition.

Finally, in the two large display setups (84" vs. 55"), the workflows differed slightly regarding the extent of physical navigation (stepping back) and distant interaction (WF2, WF3), while the answers given by the participants were similar.

Qualitative Feedback After each session, participants rated the two conditions on a Likert scale from 1 to 5 for two groups of metrics: (1) the overall efficiency, ease of use, and utility, as well as (2) suitability of the devices for set-based tasks and the intuitiveness of the specific interaction designs. Participants rated both conditions to be similar in efficiency, ease of use, and utility for visual exploration.

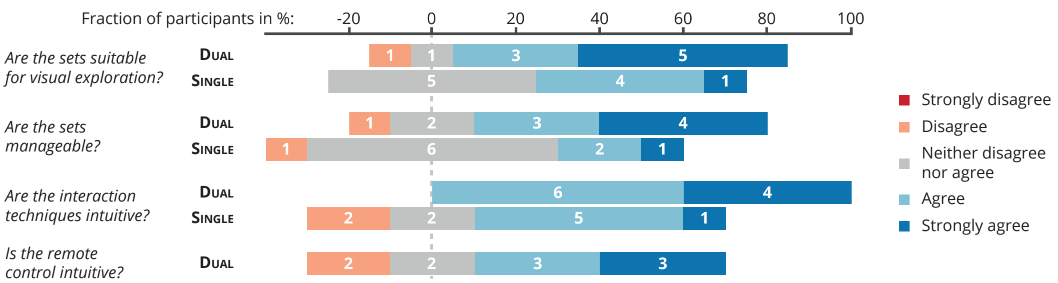


Fig. 6.8.: In the DUAL condition, sets were more suitable for exploration, and more manageable. The interactions were also rated more intuitive for the DUAL condition.

This was expected as the SINGLE condition supported equivalent operations to the DUAL condition. The one negative rating for the DUAL condition was due to the perceived increase in interaction cost with an additional device. For remaining questions, participants found the DUAL condition to be more suited for set creation and management, and the interactions in the DUAL condition to be more intuitive. In Figure 6.8, this pattern is visible with more participants strongly agreeing to these questions in case of the DUAL condition. As participant P 6 says, "*The interactions correspond to the [cognitive] actions: pull reads data in, and preview/push by activating a focus visualization gives data back.*"

6.5 Discussion and Summary

The conceptual framework presented here supported visual analysis tasks within the complementing device ensemble of smartwatches plus a large interactive display. The devices fulfilled different roles based on their strengths: the large display provides a multi-view interface, whereas the smartwatch augments and mediates the functionalities by serving as a personalized toolbox. In interplay with connective areas on the large display, the smartwatch supports exploration based on sets of both data items and visualization properties, which can be stored, manipulated, previewed, as well as applied permanently. This framework, among the corresponding design considerations as well as the evaluation, extends the understanding of such device ensembles and serves as valuable foundation for further discussions of data analysis in multi-device scenarios.

Device Roles The specific combination of smartwatches and large displays for data analysis is very interesting in respect to two aspects. First, the two devices are on the opposite side of most device characteristic dimensions, providing a maximal contrast. Second, while hand-held devices were already incorporated as secondary devices [BFE15; Chu+14; Kis+17], this is rarely the case for wearable devices in general [Zad+14] and especially not for smartwatches. The role of a wearable is to remain invisible [Wei91] and seamlessly improve the user's primary task. In contrast, hand-held devices generally have more screen space and can show alternate visual perspectives to augment the large display. For example, Kister et al. [Kis+17] have studied the large display and mobile tablet combination, and found workflows where users either stayed at a certain distance or crisscrossed in front of the display wall. Their study participants exhibited two distinct exploration styles: distributed between the combined devices, or focused on the mobile. This is in contrast with the here presented user study, where most participants focused on the large display while interacting eyes-free on the watch.

It goes without saying that neither handheld devices nor wearable devices are better suited for being paired with other devices, but rather that they have their specific roles and affordances during visual exploration. What is common is that in the context of large displays, analysts make extensive use of the possibility to work from a distance. From that, it can be concluded that mobile device should be able to serve as a remote control to a stationary device. Similarly, particularly wearable devices are suited to behave as a personal toolbox, for example, by offering access to stored data selections or visualization configurations, or by allowing to mediate the interaction on other devices. Further, for every combination between devices, the type and strength of the applied coordination should be able to vary. In some cases device are supposed to work on their own, in other cases they can influence the connected device in a ephemeral way (e.g., previewing data), and, finally, there is also the option of a permanent and closely coupled coordination. These aspects are particular important for informing the design of device ecologies, which is the topic of the next part of this thesis.

Future Work For the specific work presented in this chapter, multiple ways to develop it further exist. Regarding multi-user scenarios, current interactive displays are generally not able to distinguish which user is interacting, thus associating a touch point with one smartwatch is not directly possible. Here, it would be interesting to incorporate existing experimental solutions that allow to track and recognize users [HB13; Mur+12; Zad+16]. Then, the conceptual framework should be extended with mechanisms explicitly promoting collaboration during visual data analysis (e.g., supporting concurrent tasks, group awareness, or overall communication). With respect to analysis workflows, an in-depth study of open-ended visual exploration (cf. Reda et al. [Red+15]) would broaden this to a larger group of tasks as covered in the here presented framework and potentially allow for further deriving insights of analysts' behavior. Similarly, an explicit comparison of using handhelds to wearables for enhancing visual analysis tasks can allow for further improving the understanding of these setups and how they influence the typical exploration workflows.

Part IV

Analysis Interfaces for Dynamic Device Ecologies

7 Coordinating and Combining Visualization Views in Device Ecologies

Based on the insights from the previous part, it can be concluded that supporting visual data analysis within cross-device settings is a promising endeavor. However, as Part II on the responsive aspects has illustrated, there is a wide range of devices that are considered for visualization use. Consequently, it cannot be taken for granted that an analysis session is always conducted within the same device environment. Instead, the goal should be to enable such analyses in any device ensemble, independently from what specific devices and how many of them are present. This also means that the coordination between devices is not pre-defined beforehand, but can change depending on the constellation. In other words, the devices are supposed to automatically adjust to each other and form a device ecology.

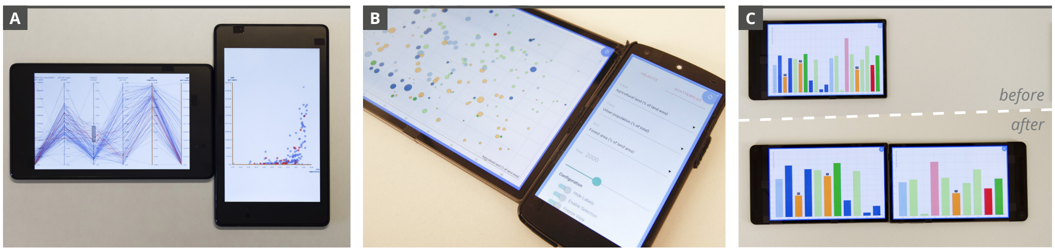


Fig. 7.1.: VisTiles supports visual data analysis in device ecologies by, e.g., (a) aligning visualizations, (b) UI offloading, and (c) display extension. *Videos and further material are provided at* `imld.de/vistiles`.

It is the topic of this part of the thesis how this can be done specifically. First, in this chapter, the focus is on the conceptual VisTiles framework, where techniques for flexibly combining and coordinating visualizations on multiple mobile devices are proposed. In Chapter 8, it is then investigated how analysts can be supported in distributing visualizations in dynamic device setups. There, the Vistribute system provides heuristics that enable an automatic distribution process, significantly reducing the setup efforts for users. Together, these two works are providing important insights into how dynamic device ecologies can be enabled for data analysis.

The remainder of this chapter details the VisTiles framework, which is motivated by two observations. The first one is shared with the overall thesis, the general availability of modern devices and that they allow for novel interaction mechanisms. The second observation is that people are used to spatially organize working artifacts around them, such as printouts, notebooks, or post-its. Kirsh [Kir95] described it as *intelligent use of space*, expressing that the way "how we manage the spatial arrangement of items around us, is not an afterthought; it is an integral part of the way we think, plan and behave." Mobile devices can allow for imitating that as they can easily be picked up, moved around, and physically organized, thus be used similar as people handle paper for sense-making activities [HNC14; ITC08]. Within a multiple coordinated views application, where one device hosts one view, the device arrangement can be used for configuring the views and their coordination.

This idea is followed with VisTiles. In the following, first, the foundations are provided (Section 7.1), including concepts for general view handling and basic coordination between devices such as linked brushing. Then, the concepts for dynamic arrangements are detailed (Section 7.2), which enable advanced adaptations that provide stronger couplings of the visualizations (Figure 7.1). These concepts can work either in a spatially-agnostic way or in a spatially-aware way, then actively considering the relative positioning of devices. Finally, the applied design and implementation process are detailed (Section 7.3).

Parts of the research presented in this chapter has previously appeared in:

Ricardo Langner, **Tom Horak**, and Raimund Dachselt. "VisTiles: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration". In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018), pages 626–636. Citation key: [LHD18b].

Own Contribution: The whole work was done in an extreme close collaboration among the first two authors over the course of two years, as well as shaped by extensive discussions among all authors. Consequently, I hold a shared contribution to all parts.
Applied Changes: For this chapter, the content was significantly shortened by removing the design space and reducing the reporting on the studies and prototypes. The conceptual aspects have been restructured and streamlined for the use in this thesis. Finally, the introduction and discussion were re-written.

Ricardo Langner, **Tom Horak**, and Raimund Dachselt. "Demonstrating VisTiles: Visual Data Exploration Using Mobile Devices". In: *Proceedings of the International Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2018, pages 69:1–69:3. Citation key: [LHD18a].

Own Contribution: The publication is a follow-up of the VisTiles publication with the presented concepts originating from the joint work on the overall project.
Applied Changes: From the publication, concepts for the spatial-agnostic device pairing as well as parts of figures have been incorporated here.

7.1 The VisTiles Framework: Foundations

In this section, we[1] describe the fundamental concepts as well as the design of VisTiles. In general, the VisTiles framework builds on the idea of enabling multiple coordinated views (MCV) [Rob07] on multiple mobile devices, i.e., transferring a traditional visualization interface from a desktop-based environment to mobile devices. However, in contrast to the strategies for responsively adapting such a layout for one smaller device (cf. Chapter 3), the idea is to distribute the different visualization views across multiple mobile devices (Figure 7.1). This forms an analysis interface that permits a varying number of users to visually explore and analyze data with their mobile devices. Analysts benefit from the use of a *physical workspace*, which allows to grasp, move, and spatially organize visualization views. For example, a devices can be put aside for coming back to it later or views related to one hypothesis can be physically grouped.

As a basic mechanism, for such physical groupings a coordination as in MCVs can be provided. Going beyond that, the spatial arrangements can also be used in an active way for stronger adaptations. For example, by placing two devices side by side, a person can trigger a comparison mode or a screen extension, and easily resolve the state later by detaching the devices again. This illustrates that VisTiles builds on both the distribution of visualization views across multiple devices as well as cross-device interaction techniques. In addition, and in contrast to the device combinations described in the two previous chapters, the number and type of participating mobile devices is not constrained. In consequence, a dynamic device ecology is formed that can change or be adapted at anytime. In this section, we will detail the basics of VisTiles, including common coordination mechanisms. Then, in the next section, the more elaborate adaptations and combinations enabled by VisTiles are presented.

View Types and Assignment

We propose to assign one view to one device in order to reduce the interface complexity and avoid hidden content. Further, we distinguish between two view types, visualization views and control views. The latter one can contain UI elements such as menus or widgets that provide further configuration options to visualization views. To support the specific mapping of views to devices, the system could provide a recommendation system proposing which views are most suited for a

[1] The "we" in this chapter relates to the author Tom Horak, as well as Ricardo Langner and Raimund Dachselt as co-contributors to this research.

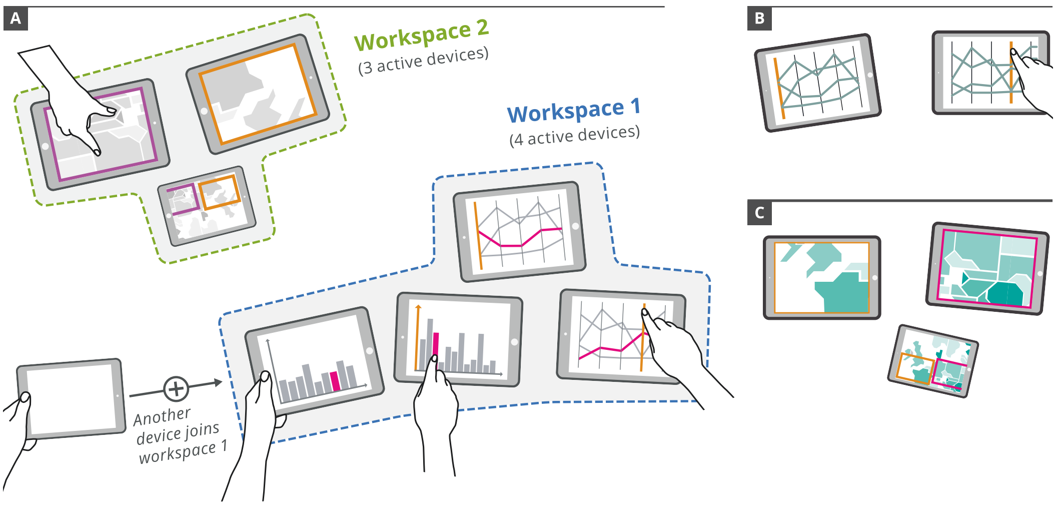


Fig. 7.2.: Coordination of visualizations in workspaces: (b) highlighting shared axis by touch (extended linked brushing); and (c) indicating position and size of other viewports for overview & detail.

given device. In general, larger devices such as tablets are particularly qualified for visualizations, whereas small devices such as smartphones are well-suited for displaying menus or other UI elements. As outlined in detail in Chapter 3, many more device characteristics are relevant for visualizations and, thus, could be considered for such recommendations. However, this discussion is left for the next chapter.

Workspaces and View Coordination

After the setup of the devices, they can be coordinated by joining them in *workspaces*, as indicated in Figure 7.2a. The number of both workspaces and devices in a workspace is flexible. To minimize the setup effort, we suggest that new devices/views automatically join a main workspace. In general, workspaces allow to easily control the coordination as well as ensure flexibility for typical multi-user situations, where data analysts join or leave ongoing working sessions. For the coordination itself, we incorporate the common mechanisms as detailed next.

As the most fundamental technique, linked brushing is supported, providing synchronized selection and highlight functionalities within a workspace. In addition, we propose extending linked brushing to other visualization elements. For example, when an axis of a scatterplot is touched, other devices also highlight this axis (Figure 7.2b). This is particularly useful as visualizations can be distant. Instead, of highlighting data items, some visualizations constellations can also allow for indicating overview+detail relations with other devices. Specifically, when two or more devices show a subset of the same data space, they automatically indicate the position and size of other viewports by displaying corresponding bounding boxes (Figure 7.2c). This concept can also allow for a remote manipulation of such views by interacting with the bounding boxes.

7.2 Dynamic Arrangements for Combining Visualization

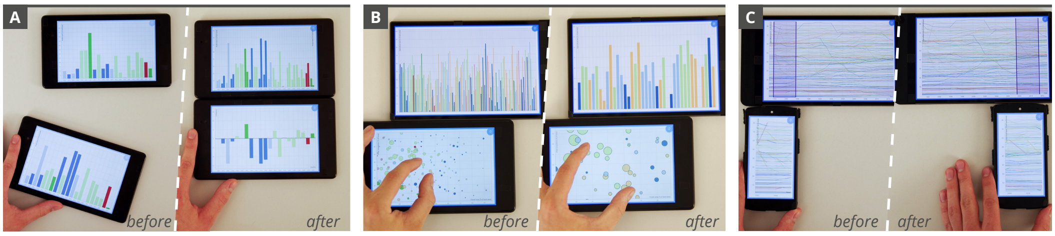


Fig. 7.3.: Advanced device coordination techniques supported by VisTiles: (a) view merging, (b) coordinated filtering, and (c) indicating overview+detail arrangements.

As noted at the beginning of this chapter, people make inherently use of a physical space and arrange artifacts in a meaningful way [AEN10; Kir95]. This also includes aligning and arranging artifacts to each other, e.g., when comparing two paper printouts. In this section, we propose how such close-proximity arrangements of physical devices can be utilized for the data analysis. We will first detail concepts that directly use *side-by-side arrangements* (Figure 7.3a+b), and then consider *continuous device movements* (Figure 7.3c) for interactive combinations of visualization views. Lastly, we will describe how these arrangements can be detected and how analysts can be provided with full control over the adaptations and combinations.

7.2.1 Use of Side-by-Side Arrangements

The side-by-side arrangement of devices can be used to trigger enhanced visualization adaptations that support common analysis tasks, such as, visually comparing objects, retrieving details, or identifying outliers. Specifically, we propose seven adaptation mechanisms detailed in the following.

Alignment As a simple but powerful concept, we propose to align visualization that are displayed on apposed devices. By automatically translating, rotating, or scaling the displayed views, they adapt to the properties of their counterpart (e.g.,

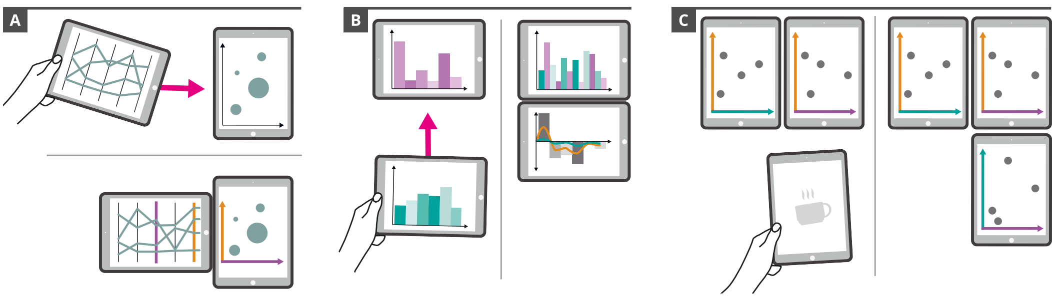


Fig. 7.4.: Alignment and rearrangement: (a) aligning visualizations and rearranging axes; (b) merging two bar charts and displaying computed data on the attached device; and (c) creating a scatterplot matrix by sequentially attaching devices.

resolution, pixel density, position, orientation). This ensures that visualizations and their elements are visually aligned and use the same physical scale, for instance, by adapting the length of axes (Figure 7.4a). Although visualizations can become smaller and display space can remain unused, the alignment increases readability across devices and counteracts misinterpretations caused by differences in size or shifted visualizations.

Rearrangement In addition to aligning two views, it can also be beneficial to apply a rearrangement of visualization elements, such as data items or axes. For example, when two bar charts are combined vertically, the sorting of data items (bars) could be equalized, thus reducing the visual gap between corresponding marks and simplifying comparison. Moreover, we suggest an extended type of rearrangement for providing additional computed information (Figure 7.4b). The idea is to merge two similar views (e.g., two bar charts) onto just one device. Then, the other device can be used to display calculated data, such as differences, as an explicit encoding [Gle+11].

Both alignment and rearrangement can also be applied on a view level. For example, a set of side-by-side scatterplots can be automatically turned into a scatterplot matrix (Figure 7.4c), i.e., resulting in aligning the plots as well as rearranging their axes accordingly. This can be generalized for axis-based visualizations. Consider the combination of a scatterplot and a parallel coordinates plot (PCP) (Figure 7.4a). Here, (i) all axes are scaled to the same height, (ii) the axis in the PCP corresponding to the y-axis in the scatterplot is moved to the shared border, and (iii) all other shared axes are highlighted.

Adaptation of Encodings Besides manually configuring visualization properties via a configuration view, the encoding of views can also be shared automatically. For

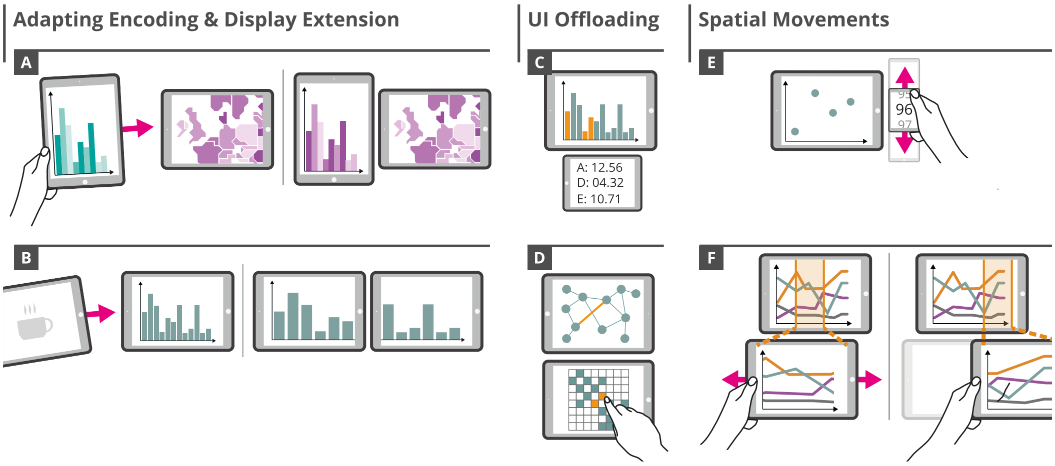


Fig. 7.5.: Further visualization adaptations: (a) adapting the color scheme; (b) extending displays and thus visualizations over two devices; (c) offloading details of selected data items; (d) editing relation of a node-link diagram through the corresponding adjacency matrix; and using continuous spatial movements for manipulating (e) visualization parameters and (f) overview+detail views.

example, placing a bar chart side by side with a map visualization, the bars color encoding is re-mapped according to the color scheme of the map visualization (Figure 7.5a). Additionally, this can also apply to other configurations of the encoding, such as the scale ranges.

Display Extension Another concept in the context of visualizations on mobile devices is to increase display size via side-by-side device combinations [LK12; LHJ11; OT12], i.e., to extend display space by 'stitching' multiple devices together (Figure 7.5b). Specifically, a visualization could be expanded across the adjacent device screens and make use of the higher resolution. Similar to the concepts for alignment and rearrangement, the technical device properties such as pixel density should be considered to avoid differences in the physical scale.

Dynamic UI Offloading In addition to using a device as a workspace-wide control view, we also consider a dynamic UI offloading. Here, a side-by-side arranged control view can automatically display the corresponding controls for the other visualizations, such as data mapping, encodings, or color scheme (Figure 7.1b). Besides view properties, additional or detailed information about selected data items can be provided as well (Figure 7.5c). Alternatively, a control view can also access and manipulate the properties of other views from the workspace remotely. However, as the workspace can consist of many views, analysts then have to select the corresponding view from a menu.

Extended View Synchronization The side-by-side arrangement can also be used to initiate stronger data-based coordination mechanisms. One concept for this is using the viewports of other devices as a filter condition. The idea of this 'filter-by-viewport' mechanism is that the filtered data items are defined by the ones visible in the current (zoomed) viewport of a visualization. For example, when zooming and panning a scatterplot, coordinated devices immediately filter data items that are not visible anymore in the scatterplot (Figure 7.3b). A second concept considers network visualization. Here, a node-link diagram arranged side by side with an adjacency matrix can be used to simplify the editing of links (Figure 7.5d; cf. Gladisch et al. [Gla+15b] and Kister et al. [Kis+17]). While the space of possible view synchronizations is vast, they heavily depend on the specific combination of visualizations.

7.2.2 Use of Continuous Spatial Movement

The adaptations described above build on explicit side-by-side arrangements of devices, where the specific arrangement is used as a trigger. As an extension to this, continuous movements of a device within a given arrangement can be used to further control adaptations. For example, when two devices are already arranged side by side, one device can be moved along the other one. Similar to Woźniak et al. [Woź+14], we suggest mapping spatial movements to a selected visualization parameter or data dimension and thereby allow users a continuous manipulation.

One example is to use such movements for manipulating a slider widget. Consider a smartphone that is arranged side by side to a tablet showing a scatterplot for a time-dependent data set. Here, the smartphone can act as a physical slider, i.e., moving the smartphone along the tablet manipulates the year for which data items are shown in the scatterplot (Figure 7.5e). A similar mechanism can be provided for an overview+detail configuration. As illustrated in Figure 7.5f, the smartphone can provide a zoomed, more detailed excerpt of another visualization displayed on the adjacent tablet. Again, analysts can pan the detail view by physically moving the smartphone alongside the other device.

The concept of using continuous spatial movement is designed as an alternative to surface interactions for adjusting parameters. On the one hand, this allows to avoid occlusion issues during interaction. On the other hand, the dexterity of persons, i.e., their ability to grasp and manipulate real-world objects, supports a more natural interaction style. In particular, this can also allow to enable eyes-free interaction as well as potentially benefit from the spatial memory when recalling specific information [Kir95].

7.2.3 Managing Adaptations and Combinations

By proposing a number of adaptation concepts, it becomes clear that several possible system reactions exist for a specific arrangement. Thus, it must be consider which adaptations or functions are activated when as well as how analysts can control these. In general, two strategies can be applied: (i) The system suggests or recommends a set of useful options, which can be activated by the analysts; or (ii) the system activates the most 'appropriate' option and the user can correct it afterwards. For VisTiles, we propose to follow the first strategy—'the application suggests, users confirm'. When moving devices towards each other, possible suggestions include: extending a visualization, aligning a view, filtering data items, or enabling the synchronization of zoom and pan between two devices. By offering such options, we aim to avoid that users feel irritated or even lose control by sudden changes of the visualization.

Fig. 7.6.: (a) On spatial combination, the adaptations can be explicitly activated via slide-in menu. If the arrangement cannot be sensed automatically by device, (b) a manual combination by a synchronous pinch is possible.

In addition to the suggestion and activation of options, the direction of an adaptation is important. For example, when combining two bar charts as illustrated in Figure 7.4b, it is ambiguous which view could show the merged data items and which should display additional information. Therefore, we propose to present options on each view that is involved (Figure 7.6a). Then, users apply the action to a specific view by activating the corresponding option on that device. However, for some cases it is sufficient to consider the type of a view: since the control view is attached to a visualization, it is this control view which should show details of selected data items or offloaded user interface elements of the corresponding visualization, not vice versa. Finally, we also suggest employing these options to deactivate previously applied adaptations, instead of moving a device. For example, this then allows to pick up a device for better readability while the adaptation and coordination is maintained.

Finally, the technical questions how arrangements are detected remains. For our concepts, we assumed that a side-by-side arrangement is simply recognized by the participating devices. In fact, the increasingly sensitive internal sensors of modern devices can allow for detecting these [Bru+19; Grø+20; Hor+16]. The alternative is using external tracking systems, as we did for our prototype implementation. However, this is impractical for real-world scenarios. Instead, another alternative is not relying on sensors at all, but have arrangements explicitly activated by a synchronous pinch gesture [Hin03; LHD18a; OT12], as illustrated in Figure 7.6b. This spatially-agnostic option is implemented in our prototype as well.

7.3 Design Process and Prototype Implementation

The development of the concepts followed an iterative design process: first we focused on a proof-of-concept prototype, before we then conducted a preliminary user study. Both informed the revised and final concepts as presented above. These iterated concepts were then also implemented in a running prototype. An overview of this process as well as the prototype implementation is provided in the following. A more elaborate explanation is provided in the original publication [LHD18b].

Design Process

We first elaborated initial ideas and principles as well as showcased those using paper prototypes and a conceptual software prototype. Then, we discussed with colleagues about different variants of both interaction and visualization concepts to identify promising approaches as well as challenges [LHD15; LHD16]. The conceptual software prototype used simple UI mockups (static images) and scripted transitions to simulate system reactions in a spatially-aware setup. This prototype was later developed further into a first web-based prototype, better implementing the proposed concepts. Devices were tracked with an external tracking system [@Nat20], with all coordination functionalities being triggered based on the device proximity (close-range and side-by-side).

We then ran an early user feedback session to validate the fundamental functionality and feasibility of our concepts. We invited seven participants (age M=24.43 yrs, SD=2.37 yrs; 3 female, 4 male; 5 post-graduated) and demonstrated out concepts with a set of five mobile devices (one smartphone, four tablets). Using the first web-based prototype, participants received an introduction and were then asked to try out the concepts by themselves. After this hands-on part, we finished with an overall discussion. Sessions lasted 45 minutes and were videotaped.

Among other findings, user feedback confirmed that device movements need to be used carefully for activating changes. As such movements might happen during other actions with mobile devices (e.g., taking a closer look at a display), unintentional system reactions could be the consequence. Similarly, for every device arrangement, multiple valid coordination functionalities exist. Analyst should be able to select and confirm the ones that are most suited for the current analysis situation. Based on device arrangements, we initially envisioned several means of automatic adaptations. Overall, the feedback also confirmed that using mobile devices for organizing an analysis interface is indeed an interesting and promising option. For example, comments showed that offloading UI widgets is a very simple, but powerful technique. A more detailed discussion of the findings is provided in the original publication [LHD18b].

Based on the feedback and experiences from this process, we then refined our concepts to the version as described before as well as created a more elaborate prototype. The implementation of this prototype is described in the following.

Prototype Implementation

The final prototype [@LHD] build upon the first web-based prototype, thus used web technologies as well to support the majority of mobile devices. To provide, manage, and drive communication between involved devices, we realized a client-server-architecture. On the server side, a Node.js [@Ope09] server handles the client communication via WebSockets. It consumes the tracking data provided by the tracking system to detect proximity-based device combinations and controls corresponding system reactions. We used D3.js [BOH11] to create visualizations and the Materialize CSS framework [@Mat14] for a consistent user interface.

As the entrance point into the application, an option screen allows analysts to choose between several visualization techniques as well as UI widgets, thus to assign the role of a Visualization view or control view to a device (Figure 7.7a). The control view features a tab-based menu offering control elements such as sliders or drop downs for connected visualization views (Figure 7.7b). The interface can also provide notifications, e.g., to confirm applied adaptation between two devices. When

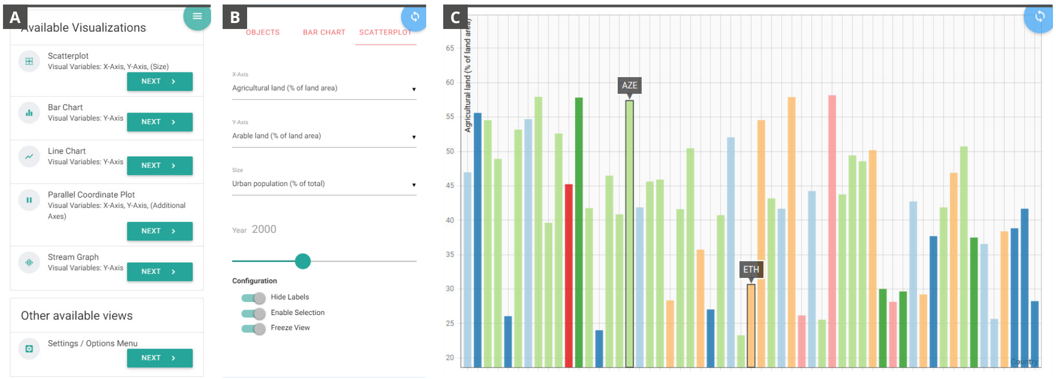


Fig. 7.7.: Realization of our web-based prototype: interface design of (a) the main menu, (b) a control view, and (c) a visualization view, here a bar chart with multiple selections.

multiple workspaces are used, the association of devices is indicated by colored borders (one color per workspace).

On the visualization side, we implemented bar charts, scatterplots, line charts, parallel coordinate plots, stream graphs, and tables (spreadsheets). In all of them, a 10-class qualitative color scheme from ColorBrewer [@BHT13] was used to indicate data groups (Figure 7.7c). Data items can be directly selected by touch, with a small tool tip then showing details (Figure 7.7c). As example data, a subset of the World Development Indicators (WDI) time series data collection [@The20] was used, containing 52 dimensions for 215 countries (data items) over 24 years (1991 to 2015).

Our VisTiles prototype provides specific visualization adaptations as options (Subsection 7.2.3). These are listed in an icon bar (drawer), which appears at the border of arranged devices (Figure 7.6a). As soon as one device is moved away, the option menu slides back in and offers to disable the former selected adaptations or configurations. However, the user can also decide to keep synchronizations active and to manually deactivate them later. We implemented the following side-by-side combinations: alignment and rearrangement, display extension, dynamic UI offloading, view synchronization for overview+detail, and 'filter-by-viewport'.

Interaction Walkthrough for Data Exploration

In the following interaction walkthrough it is illustrated how data analysts can use a VisTiles-powered device ecology for analyzing multivariate data. Consider a group of three people who want to investigate a multivariate data collection collaboratively. Since two of the three users are interested in particular data dimensions, they decide to directly cooperate. Using their own devices, they load a bar chart and a line chart respectively, visualizing urban population growth. They add both devices into the shared main workspace. For the year 2007, the bar chart reveals that some data items have significantly higher values, specifically the State of Qatar and the United Arab Emirates are outliers. The one analysts selects the two corresponding marks. Looking at the line chart on the other device, both countries are again highlighted, showing that for both these high values are not a linear trend over the last 15 years, but a peak of a fluctuation.

The third analysts investigates the data independently within a separate workspace. Reviewing the data with respect to distributions and correlations by using a scatterplot, she finds one correlation for the income per person and the child mortality rate. To share her insights, she adds her device to the other workspace by moving the scatterplot towards the line chart and arranging them side-by-side. In response, the menu for managing adaptations appears, allowing to add the device to the workspace. The previously selected outliers are now also highlighted in the scatterplot, showing that contrary to the overall correlation, the State of Qatar and the United Arab Emirates have almost equal child mortality rates but quite a different income per person. Further, the 'filter-by-viewport' option can be activated via the adaptation menu. By zooming and panning the scatterplot, the line chart is then filtered, allowing to discover, for example, that especially for African states the child mortality rate decreases, while the urban population growth increases.

7.4 Discussion and Summary

The VisTiles framework illustrates how mobile devices can be used in synergy for visual data analysis. An important key characteristic is the flexibility in which analysts can arrange, coordinate, and combine the different devices and, thus, visualizations. The arrangement itself can be used in a passive and active way. In the passive way devices are placed at positions that are meaningful for the analyst, but no interaction is directly triggered. In contrast, the side-by-side arrangement is used actively where devices propose suitable adaptations to the user.

Dynamic Device Combinations Further, the proposed concepts involve coordination and adaptation mechanisms that can work either across a distance or require adjacent displays. For example, the UI offloading for the control views shares similarities with the toolbox concepts for the smartwatch in the "David meets Goliath" device ensemble from Chapter 6, which is designed as a distant control. At the same time, the proximity to a specific view (or device) allows for quickly setting the target context. In contrast, concepts such as the display extension or adapting and rearranging visualizations rely on the side-by-side placement. In fact, these considerations to use the devices as a continuous display are similar to the discussion within the WATCH+STRAP setup. The here presented concepts could also be further extended with ideas from WATCH+STRAP, such as using displays with a higher resolution as a focus display and lower-resolution displays as context areas.

In difference to previous work investigating specific device combinations, for dynamic device ecologies such as VISTILES it is not known what specific combinations with how many and which devices will occur. For example, devices might have very different characteristics but can also be of the exact same device type. This results in two conclusions. Firstly, specific adaptations should not become over optimized for one specific device combination, but should be beneficial for a wider range of combinations. Secondly, the number of theoretically possible adaptations is extremely vast. Besides adaptations considering certain device characteristics, also visualization-specific combinations can be proposed, however, resulting in a theoretically endless combinatorial space. In consequence, it might be more beneficial to reduce the number of supported adaptations as well as ensuring that these can be easily discovered and controlled by the data analysts.

Future Work The VISTILES framework itself can be further investigated in several ways. On a technical side, this involves efforts for precisely detecting device proximity with internal sensors only. On a conceptual side, it can also be interesting to transform the concepts for physically arranging views to virtual views, e.g., for wall displays or mixed reality applications. This would allow for investigating view-specific adaptations in a more generalized way. Another interesting continuation can be followed by relaxing the one-to-one mapping of visualizations to devices. Effectively, this would result in displaying MCV arrangements on one device [SS16], which can be beneficially if more views than devices are present. Here, it would then be required to investigate how the adaptions can be applied to such MCV devices. In general, the mapping of views to devices in a certain arrangement represents a notable setup effort for analysts, both for MCV layouts on one device and across devices as in VISTILES. Therefore, in the following chapter, it is investigated how this effort can be mitigated through automated distribution approaches for dynamic device ecologies, allowing analysts to purely focus on the data analysis and not device configuration.

8 Automatically Distributing Visualization Views in Device Ecologies

A major part of the existing research in the intersection of HCI and visualization focuses on how specific tasks can be performed during an analysis—similarly as VisTiles proposed for dynamic mobile device ensembles. However, particularly for multi-device usage the setup and configuration of the analysis interface and its views pose a notable extra effort, which is often not addressed by existing research. Further, such a configuration is not a one-time effort, but an integral part of the analysis in dynamic device ecologies. For example, consider an oncologist in a hospital using patient tumor data to inform her practice (analyzing, e.g., tumor growth rate, blood levels). In the morning, the doctor may spend some time in her office to plan treatment (desktop and tablet), continuing during a coffee break (laptop and phone) with a colleague spontaneously joining after a while (adding a tablet), then at a tumor board with other doctors (large displays, laptops, and mobiles), and finally in a treatment room consulting the patient (tablet and large TV)—and every time, the interface must be adapted accordingly.

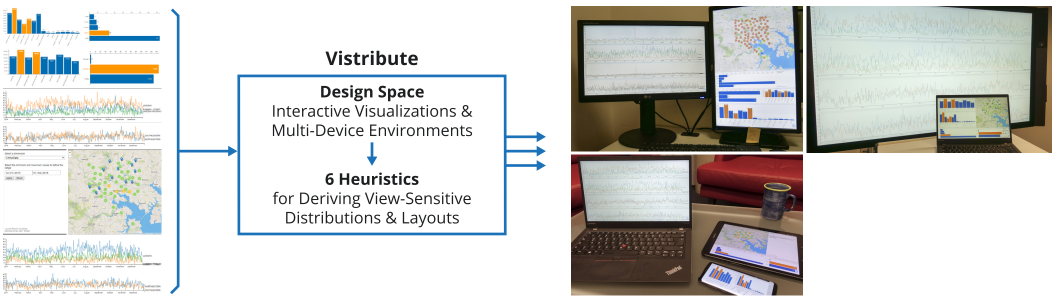


Fig. 8.1.: The Vistribute system: Based on a design space we derived six heuristics that can guide an automatic distribution of visualizations in changing device setups, e.g., dual desktop, laptop and large display, or mobile device ensemble. *Videos and further material are provided at* `imld.de/vistribute`.

This chapter introduces the Vistribute framework, which aims to support data analysis in all these situations while eliminating the effort of manually setting up the interface, i.e., distributing and positioning views. At its core, an automatic approach for distributing visualization interfaces across dynamic device ecologies is proposed (Figure 8.1). Unlike existing automatic distribution mechanisms for general user interfaces, VISTRIBUTE uses in-depth information about visualization views, the data they visualize, and the tasks users want to perform on them to optimize the layout. The resulting algorithm can then detect changes in the current setup, consider all the relevant information, and determine how the interface could be distributed in useful ways across the devices.

The VISTRIBUTE framework consists of a design space, a set of heuristics, and an example implementation. After providing further background on distributed interfaces in general (Section 8.1), these parts of VISTRIBUTE are presented in this chapter. The design space for cross-device visualization (Section 8.2) draws on the literature as well as an analysis of existing visualization interfaces, and explicitly considers dynamic factors such as view properties and relationships, device properties and the current device ensemble, as well as user preferences. Building upon this design space, the VISTRIBUTE system is introduced (Section 8.3). With it, several heuristics are proposed as high-level constraints for distributing visualization views (8.3.1). Further, the web-based implementation of VISTRIBUTE (8.3.2) can automatically collect information about the devices, the dataset, and the visualizations to derive a suitable distribution. In addition to the distribution itself, the users are also enabled to adapt the interface distribution according to their needs and preferences. The here presented implementation of VISTRIBUTE was also used for a user study. The reporting of the gained insights is provided in Section 8.4.

The research presented in this chapter has been published before in:

Tom Horak, Andreas Mathisen, Clemens N. Klokmose, Raimund Dachselt, and Niklas Elmqvist. "Vistribute: Distributing Interactive Visualizations in Dynamic Multi-Device Setups". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, pages 616:1–616:13.
Citation key: [Hor+19].

Own Contribution: I am the major contributor to this work, with the other authors having provided valuable feedback during discussions as well as partly contributing to the writing.
Applied Changes: The original content was minorly adapted for this chapter. These changes involved adjustments to the structure as well as updated introduction and discussion parts.

8.1 Background on Distributed Interfaces

Utilizing heterogeneous devices in parallel introduces multiple technical challenges how to coordinate and synchronize the devices. In Chapter 2 (2.5), the few existing visualization-specific frameworks concerned with synchronizing analysis interfaces across devices have already presented, including VisConnect [Sch+21], Vistrates [Bad+19], or Munin [BFE15]. To recap, the synchronization can happen here on multiple levels: For example, Webstrates [Klo+15] (and, thus, Vistrates [Bad+19]) operates on the level of the Document Object Model (DOM), effectively maintaining exact copies on different devices, while VisConnect [Sch+21] transmits and replays interaction events.

In general, keeping distributed interfaces in sync is not a challenge limited to visualization interfaces, and, thus, has been considered repeatedly in the HCI community. For example, some also focused on simplifying creating device spanning graphics from a developer's perspective, i.e., when one canvas is stretched out across multiple devices [Räd+14; Sch+15]. Also, functionality supporting cross-device interaction techniques can be provided [HM15]. However, these frameworks are designed to ease the development of new applications, while requiring that programmers or users manually arrange interface components.

This gap is partly addressed by research proposing specific distribution algorithms or frameworks, most of which automatically derive a candidate distribution based on interface semantics provided by the developer, and then let the user adjust the result. Panelrama [YW14] introduced a lightweight specification that allows programmers to provide additional semantics for HTML elements, which are then consumed by an interface optimizer. Park et al. [Par+18] proposed an optimizer called AdaM, which is based on a constraint solver; however, AdaM requires users or developers to provide additional semantics for each interface component, too. The XDBrowser [ND16; Neb17] segments web pages and distributes the parts across devices. In all examples, the layout is not guaranteed to be optimal, and serves rather as a starting point.

More specialized applications may allow for automatically determining dependencies between interface components and how to organize them. As a case in point, Husmann et al. [Hus+18] presented a similar system in the context of an integrated development environment, but applied automatic assignments only for a few selected view constellations. Such an approach has not been proposed for visualization and data analysis yet.

8.2 Design Space: Visualizations in Multi-Device Environments

The distribution and layout of views in a visualization interface are not arbitrary, but often follow certain patterns [Che+21; Rob+19]. Based on related work, considerations of existing interfaces, and our own experience in cross-device research, we[1] aim to provide a conceptual framework that is able to reproduce these patterns when distributing and arranging views across multiple devices. The framework consists of a design space, distribution heuristics, and a prototype implementation.

In creating our framework, we were guided by multiple considerations. First of all, individual visualizations encode richer semantics compared to other user interface components [QH18], such as the data being visualized, the visual representation chosen, and the typical tasks supported. By considering these aspects, it is possible to automatically derive properties required for a distribution that otherwise would have to be provided by analysts, designers, or developers.

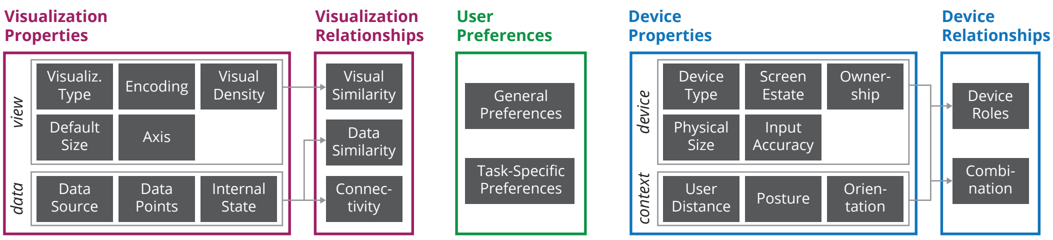


Fig. 8.2.: The design space comprises aspects coming from the visualizations, users, and devices. When considered pairwise, relationships emerge for both devices and visualizations.

Second, these semantics also reveal relationships between multiple visualizations [QH18], which allows for further refining the distribution. In existing interfaces or dashboards (see, e.g., Tableau dashboards [@Tab17a; @Tab17b] or examples analyzed by Sarikaya et al. [Sar+19]) it is possible to observe such relationships, e.g., two bar charts are aligned for comparison. Similar aspects can be observed in research focusing explicitly on large displays or multi-device ensembles [LHD18b; LKD19], as well as for the involved devices, where their properties and relationships imply their strengths or possible roles in a distributed interface.

The design space aims to give an overview of interactive visualizations in multi-device setups, considering all relevant properties and relationships occurring (Figure 8.2),

[1]The use of "we" in this chapter refers to the author Tom Horak, as well as Andreas Mathisen, Clemens N. Klokmose, Raimund Dachselt, and Niklas Elmqvist as co-contributors to this research.

which we group and discuss as five dimensions in the following. At the end, this design space will eventually provide a fundamental understanding of the incorporated dimensions. By molding this knowledge into easy-to-apply heuristics, we aim to provide a guidance for new distribution approaches (i.e., specific implementations) for interactive visualizations.

8.2.1 Visualization Related Aspects

Visualization Properties In comparison to traditional UI components, visualization views feature a rich body of properties that depends on their configuration, visual representation, or encoded data. These properties can be used to construct visualizations (as in, e.g., D3 [BOH11], Idyll [CH18], Vega [SWH14; Sat+17]) as well as to analyze them (essentially the inverse of construction), as in our case.

First, visualizations can be characterized through properties related to their visual appearance: the actual *visualization type* (i.e., used visual marks), the applied *encoding* and mapping (i.e., visualized data dimensions), the *axis* configuration (e.g., orientation, scale, sorting), as well as the *default size* (and also implicitly the aspect ratio). Although these properties are often defined in the context of the considered data, they do not fully depend on the actual data: two views can have the same visual configuration but show disjoint data subsets. We also consider a *visual density* property, resulting from the mark size, potentially occurring overlaps, and existing additional elements (e.g., guides). This density can affect the comprehension and supported interaction, as the selection of small marks is more difficult and requires a certain minimum precision (cf. Park et al. [Par+18]).

For data-related properties, we consider the used data source, the data points themselves, as well as the internal state. The *data source* can describe only the source or the complete data flow prior to the view, i.e., from the dataset through filters or aggregation components. Depending on the visualization system, certain functionalities such as aggregation can be part of the view itself (e.g., VEGA-LITE [Sat+17]) or a separate component (e.g., VISTRATES [Bad+19]). Nevertheless, we consider them as pre-processing and not part of the visualization itself. The *data points* allow comparing the data of two views or analyzing the view regarding the number of visualized marks, e.g., to estimate how dense the visualization is. Finally, visualizations often maintain an *internal state* which can be accessed by other views, for example the currently selected marks or ranges.

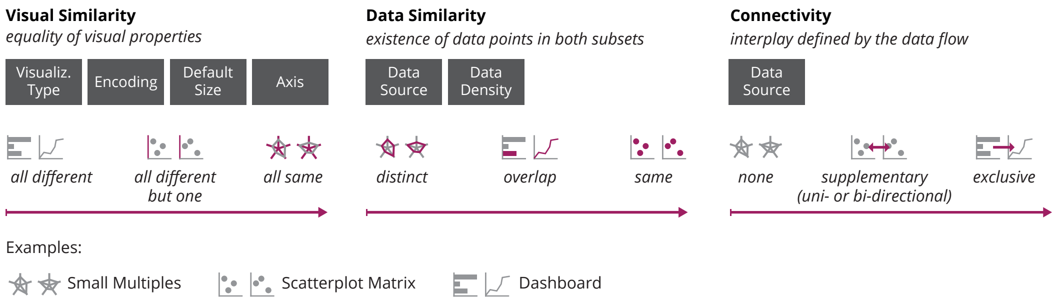


Fig. 8.3.: Three dimensions of visualization relationships based on view properties; many combinations can be useful.

Visualization Relationships Typical visualization interfaces consist of multiple (coordinated) visualizations [Che+21; Rob+19; Sar+19] where the views complement each other by showing different aspects of the data and, in combination, help the user gain insights. Thus, the views stand in an interplay and maintain relationships on different levels. Here, we characterize these within three types: visual similarity, data similarity, and connectivity (Figure 8.3). These relationships yield patterns for grouping and aligning views common in existing interfaces [Sar+19].

Visual similarity considers how similar the two views appear, regardless of the actual encoded data points. We use the visual-appearance properties described above (e.g., type, encoding, axis configuration) to rate the consistency of two views [QH18]; by comparing the properties, the similarity can range from *all different* to *all same*. Similar views can support visual comparison when placed in juxtaposition [Gle+11; JE12; QH18]. For instance, two views with the very same visual configuration is an example of *small multiples*, where the single instances differ only in the shown data. Slightly weaker relationships can be found in scatterplot matrices, where two plots differ in one dimension. In contrast, dashboards may feature multiple views that are not or partly consistent and, thus, have only a weak visual similarity [Sar+19].

The second relationship type is *data similarity*, and expresses how big the overlap between the visualized data points of two charts is. When the data is exactly the *same*, this indicates that the two views show different representations for the same data subset. A weaker similarity is a data *overlap*, and no similarity means the data is *distinct*. These constellations can indicate certain schemes, e.g., overview+detail (overlap). However, in many situations, data similarity must be considered with respect to visual similarity. For instance, some combinations of the two measures are not practical, e.g., a perfect visual similarity and a perfect data similarity describes the same visualization. In conclusion, data similarity provides an indication which views are related data-wise and, thus, can potentially provide complementing insights to this data.

Finally, views can also have a relationship with respect to the data flow, which we define as *connectivity*. This involves mechanisms such as linked brushing in multiple coordinated views [Rob07], or incorporating a selection in one chart as a filter condition in another [LHD18b]. We distinguish between different connectivity levels. The strongest is an *exclusive* connectivity, where a view receives its data purely from another (e.g., a filter component). Linked brushing, instead, is an example of an additional, *supplementary* connectivity. Here, both views would still be able to display data without this connectivity. Such rankings of the connectivity can also be found in other work. For example, in VisTiles, the connectivity level was split between connections triggered by side-by-side combinations (i.e., stronger ones) and general connections (e.g., selections). Notably, the connectivity extends also to non-visualization views, such as UI components for defining filters or aggregations.

8.2.2 User Related Aspects

User Preferences Visualization interfaces are typically flexible and can be adapted to user preferences. We distinguish here between two types of preferences: general and task-specific preferences. *General preferences* are independent of a specific situation and derive mostly from how a user prefers to arrange things or what overall strategy for device organization he follows [HW14]. For instance, a user may want to keep a filter component on the right device border, or prefers to have one specific visualization on a specific device. *Task-specific preferences* emerge during the data exploration [AES05; BM13; YKS07], and also affect the distribution. This can involve, e.g., aligning views for visual comparison, temporarily enlarging a visualization, or moving a view to another device to simplify interaction.

While multiple distributions of the same quality exist, they may fit analyst's preferences differently. Thus, considering these user preferences helps to improve the system's usability. However, retrieving such information automatically is challenging. Instead, interfaces should provide adequate functionalities that allow users to express their preferences.

8.2.3 Device Related Aspects

Device Properties As elaborated repeatedly in this thesis, the current device generations feature a very wide spectrum of distinct characteristics. Likely the most important property is the available *screen estate*, determining how many visualizations can be displayed at what size. As done within VisTiles, the screen resolution should not be a sole measure, as the resulting physical size is often relevant for relating information across devices. Further, devices differ in the available input modalities, i.e., no input, touch, pen, mouse, or keyboard, and the resulting *input accuracy* [Par+18] of these. The *device type* can also indicate useful information with regards to mobility or computation power. In combination with the *ownership*, this allows to distinguish between personal smartphones (mostly used by one person) or public large display (shared with multiple users) [Hor+18b; Kis+17; McG+12].

Besides these basic properties, further characteristics can be considered. Contextual information about the device's *posture*, *orientation*, and *user distance* (i.e., user-to-device proximity) provide insights on how the device is used by analysts. For example, hand-held devices are more likely to be used for input. Similarly, a distant device may require scaling up views for readability reasons. Further, advanced display specifications could be considered (e.g., viewing angles, color accuracy, brightness). However, such properties are hard to access and typically require external sensors or knowledge.

Device Relationships Depending on the actual device ensemble, devices can step into different relationships during the interaction—as it was extensively illustrated in the previous chapters. While the theoretically possible combinations are manifold, we focus here on realistic device combinations. The simplest *combination* is a two-display desktop setup, where the displays are aligned and form one big surface. The WATCH+STRAP setup illustrated such a constellation as well. In contrast, in a scenario where a laptop is connected to a projector, these two screens act as separate units with different properties. The second case can also be applied to mobile devices (i.e., smartphones and tablets). For example, they can be used in combination with a larger display or a desktop, as shown with the *"David meets Goliath"* setup in Chapter 6, or also by Kister et al. [Kis+17] and McGrath et al. [McG+12]. Or, they can be used with multiple other mobiles, as shown with VISTILES or by Plank et al. [Pla+17], Rädle et al. [Räd+14], and Woźniak et al. [Woź+14].

In these situations, devices differ regarding their type, size, input modality, posture, and distance, which makes it possible to assign certain *device roles* to them. As one example, smaller devices in addition to a larger device are most often suitable to host additional details and UI elements [LHD18b; Kis+17; McG+12], or devices closer to the user can act as remote controls for a more distant device [Hor+18b; LKD19; Led+15; Woź+14; Zad+14].

8.3 The Vistribute System

Our design space and its dimensions can be used to both describe and generate layout strategies for cross-device visualization. In our work, we use these dimensions to derive six heuristics for distributing components of a visualization interface across multiple devices. With these heuristics, we aim to provide comprehensible and replicable high-level constraints. We found that a formal specification, such as in the AdaM framework [Par+18], is often costly with little practical gain, and—most importantly—results in definitions that are hard to relate to. In contrast, our heuristics are prescriptive also to human designers and can be used to guide the design of manual distribution, algorithms, or even optimizers.

8.3.1 Heuristics for View-Sensitive Distributions and Layouts

Each heuristic contributes to different aspects of a distribution, such as view grouping or device assignment, while they also allow for promoting common analysis tasks (e.g., visual similarity supports comparison tasks). Specifically, we consider the heuristics to be applied in a step-wise process (Figure 8.4), where a later heuristic can contradict earlier assignments. In this process, the heuristics can be detailed, weighted, and transformed into a specific algorithm implementation. Our VISTRIBUTE implementation serves only as one example.

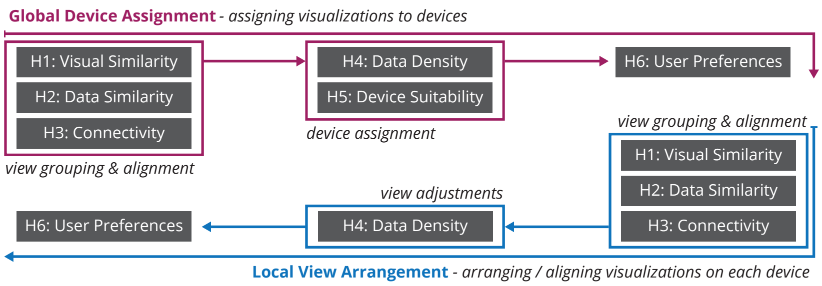


Fig. 8.4.: The heuristics are incorporated for both the global device assignment and the local view arrangement on devices. This results in a step-wise process, where heuristics can also contradict each other.

Grouping & Alignment Based on View Relationships

The relationships between visualizations can serve as indicators for how views should be grouped or aligned [QH18]. Therefore, we introduce three corresponding heuristics.

As pointed out above, views with a high visual similarity promote visual comparison. Based on common practice, such as in small multiple displays and scatterplot matrices, it is beneficial to place these views next to each other. Reducing the screen distance facilitates the user alternating their focus between the two views and, thus, to actually compare them. Aligning the views along a shared axis will further support comparison. Here, we utilize the visual similarity as an indicator if and how well two views are comparable. We consider a high visual similarity as the strongest type of relationship between views that motivates juxtaposing them. At the same time, a lower visual similarity is often not of interest. We define the heuristic as follows:

Heuristic 1 (Visual Similarity). *If two views are visually very similar, they should be both juxtaposed and aligned.*

The second driver for grouping is data similarity. Placing the views with a high data similarity close to each other, i.e., forming view groups, can support the search-related tasks of users [BM13] as well as focusing on related aspects (cf. the semantic substrate concept by Chung et al. [Chu+15]). For example, if multiple views encode the exact same data subset and are placed next to each other, they will provide different visual representations of the same subset. Similarly, this applies to other constellations, such as overview+detail schemes (i.e., one view shows a subset of the other view). However, this relationship is not as strong as the visual-similarity-based one, and typically does not require an alignment of the views. Further, it may also depend on the type of visual similarity, e.g., a subset relationship eventually represents a useful overview+detail setup if the views are of the same type. Thus, this heuristic focuses on data similarity, but also incorporates visual similarity:

Heuristic 2 (Data Similarity). *If two views have a high degree of data similarity and a corresponding visual similarity, they should be placed close to each other.*

As described before, views can consume data from another view and either rely on it exclusively (e.g., filter), or use it as a supplementary input (e.g., linked brushing). In the first case, the component providing the input must be accessible so that the other view can be used. Therefore, it is beneficial to place it close to the affected view, in order to emphasize their dependency. Also, and similar to visual similarity, proximity helps to reduce the cost of attention switches between the input component and the affected components. This is also true when the connection provides supplementary input. In all cases, a close proximity of the views is desirable:

Heuristic 3 (Input Connectivity). *If an interface component serves as data input for others, it should be placed close to the affected components.*

As a result of these heuristics, we expect two types of view groups: (i) strong groups that result in guaranteed alignment, and (ii) weak groups that lead to view proximity, but also can be split up in case of insufficient space.

View Adjustments and Device Assignments

The next step towards the distribution is considering the single views with respect to the current device ensemble.

First, it should be identified how much space a view requires: although exceptions may exist [JH13], generally, the more data points a visualization encodes, the more it benefits from being scaled up [Liu+14]. For instance, a bar chart showing three bars requires less space than one with 50 bars. Similarly, a scatterplot encoding hundreds of data points should be allocated more space than one with 10 marks. While the optimal size in relation to the number of data points always depends on the visualization, it is still a good estimation of relative space requirements. Finally, many visualizations are sensitive to changes in their aspect ratio [HA06; Wei+20]. Thus, scaling should be mostly uniform to avoid tampering the original perception.

Heuristic 4 (Data Density)**.** *A view should be allocated space proportional to the number of data points it encodes.*

Second, we consider the device suitability, which expresses how well a certain device can fulfill the requirements derived from a view or a group of views. These requirements mainly comprise the space requirement, input accuracy, and relations arising from the connectivity. For instance, views with a high space requirement are likely to be placed on a larger display. However, the suitability has not always an impact, i.e., when all devices are very similar, and, thus, interchangeable. For example, when only tablets are available, it does not matter which part of the interface is distributed to which device. In contrast, with high diversity in the device ensemble, device suitability can be used for assigning different device roles (see device relationships described in design space). This can lead to exceptions of the grouping, e.g., components serving as an input can be moved to a mobile device and act as a remote control for the larger displays. In summary, device suitability is a main constraint in diverse ensembles:

Heuristic 5 (Device Suitability)**.** *If devices are heterogeneous, views should be assigned to devices which characteristics can match the views' requirements.*

User Preferences

No matter how advanced a view distribution system is, users should be able to change the layout based on their preferences or current situation. These preferences can involve, e.g., a fixed placement of some views, an altered alignment, or even the exclusion of certain devices or components. These constraints should always be reflected in the distribution and overwrite the definitions coming from the other heuristics. Furthermore, these preferences should be stored and reapplied automatically, but must be editable by the user.

Heuristic 6 (User Preferences)**.** *If user preferences are applicable, they outweigh all other heuristics.*

In the context of analysis tasks [AES05; BM13; YKS07], i.e., temporary user interests, it could be theoretically possible to infer these automatically based on user interactions. For example, if a user makes alternating selections in two views, this can express the need to bring the views closer together. As we explicitly left room for weighting the heuristics, this allows for optimizing the distribution for the current task, e.g., emphasizing data similarity (H2) and connectivity (H3) to support investigating related items (*connect* [YKS07]). However, too many (unexpected) interface changes must be avoided.

8.3.2 Realizing a Web-based System

We implemented a web-based system [@Hor+] that is able to (i) extract required properties from both visualization and UI components as well as connected devices, (ii) derive and apply a distribution, and (iii) allow user adaptions via a control panel. However, the implementation is only one of many possible instances of our heuristics. For each feature, we will reference the related heuristic. Stated quantifications and values were determined empirically.

Underlying Systems and Dependencies

Our implementation builds upon three existing system layers: Webstrates, Codestrates, and Vistrates. Webstrates [Klo+15] provides the underlying synchronization across devices, while Codestrates [Räd+17] provides a package management system based on Webstrates (besides an in-browser computing environment). Vistrates [Bad+19] is a visualization layer for Codestrates offering specific visualization components and a data-flow-based execution model. This combination provides common visualizations and the possibility to connect them to a data source or with each other, hence, providing all tools to create an adaptable and full-fledged visualization interface.

Our distribution layer is implemented as a VISTRATES meta-package and makes use of the offered functionality of the before-mentioned layers, e.g., when accessing view properties (including states and data flow configurations). The distribution algorithms are run on one client; the resulting distribution is provided to all clients as a JSON object via the underlying DOM synchronization. Then, the clients move their assigned views to the given position on an interface layer. The instantiation of the visualizations and their connections is, however, left to the user.

Deriving Properties

The first step for the distribution is to derive all required information, i.e., visualization and device properties.

View Properties and Relationships To extract these properties, we directly access the standardized state of the VISTRATES components [Bad+19], e.g., template, size, data source(s), and accessed data properties. Based on the rendered view, we can distinguish between visualization and UI components. We also identify the incoming data as a basis for following steps.

The visual similarity is calculated by comparing selected properties and assigning points for matches. Specifically, we consider the component template (3 pts; comprises type and encoding), dimensions (i.e., consumed data properties; 2 pts), number of data points (1 pt), and size (1 pt). By traversing the components' data source, we extract the connectivity (H3, *exclusive* or *supplementary*) and the data similarity (H2, *none* or *same*). For performance reasons, data points were not compared directly; instead, we determine the closest common source and check if the data structure changes on the way (by, e.g., aggregation). While this does not allow detecting data overlap, it provides an indication if the data structure is the same.

Device Properties Current browsers provide access to a set of device specific properties, allowing us to characterize as well as (re)identify them. Besides common properties such as resolution, language, platform, and user agent, in many cases also hardware-specific properties (e.g., parallel threads, memory size, CPU, GPU) are available. However, some device information is missing, e.g., advanced display properties, physical size, or attached input devices (e.g., keyboard, mouse). As a result, we cannot distinguish larger displays (e.g., digital whiteboard, projector) from desktop displays, as their resolution is identical. Similarly, contextual information (e.g., user proximity, ownership) would require external sensors.

Notably, one physical device can host multiple clients (e.g., laptop with projector), where each client should be considered independently. At the same time, in some setups multiple clients must be perceived as one unit (e.g., display wall consisting of multiple displays), even if they are not hosted on the same device. Therefore, we introduce an abstracted representation of a device called *surface*. Each surface represents one or more clients and maps its resolution to them. For the distribution, only these surfaces are considered. Except for resolution, the surface inherits all the properties of the hosting device.

Distribution: Grouping, Assignment, and Adjustment

As described before, we consider the distribution to be a multi-step process. The first step is to identify the view groups and their types (strong and weak). To qualify as a strong group, views must have an exact visual similarity (=7 pts; H1), while weak groups are formed based on data similarity (H2) and connectivity (H3). An example distribution is given in Figure 8.5. In addition to these groups, we also calculate a relative space requirement V_{SR} for each view based on the number of visualized data points (damped via log_2) and normalized so that $\sum V_{SR} = 1$ (H4). Similarly, based on the available area, we calculate a relative screen estate S_{SE} for each surface, again with $\sum S_{SE} = 1$.

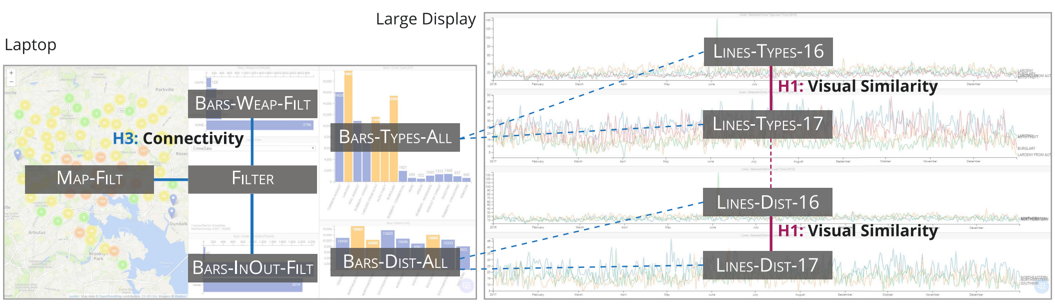


Fig. 8.5.: Example distribution of 10 views illustrating H1 and H3: On the laptop, views form a block based on their connectivity to the filter (H3, *exclusive*); the line charts form two pairs based on their visual similarity (H1, *all-same*, 7 of 7 pts).

Next, we identify special view-device pairs, e.g., offloading input components to smaller mobile devices, and assign the views directly to the surface (H5). Then, we proceed with the default assignment of views to surfaces based on the space requirement (H4,5). We consider strong view groups first, then weak view groups, and finally all other views. If no surface is big enough to exclusively host a group, we either accept to scale down the views (strong groups), or to split them up across multiple devices (weak groups).

The last step is arranging the views on each surface. Here, we applied an approach similar to bin packing [GJ81]: basically, we create columns and fill them up until the available surface height is no longer sufficient. The initial size of views is based on their space requirement in relation to the surface's screen estate (i.e., $V_{Area} = V_{SR} \times S_{Area}$; H4). Because of different aspect ratios and sizes, some rows may not fill up the whole column width. In these situations, we try to fill up the spots with smaller views. While adding views to columns and rows, we allow for a flexibility in view size and aspect ratio (up to 25%). As constellations can exist, where views cannot be fit into the available screen space (e.g., because of contrary aspect ratios), we scale the whole layout down to fit into the surface. Finally, we again adjust view height and width up to 50% to eliminate any free space. Although our implementation does not explicitly align views yet, this approach typically maintains the alignment/grouping implicitly as the views are processed in order of their group membership.

Control Panel for User Adaptations

Our implementation provides a control panel allowing users to fine-tune the distribution (H6). The panel shows the surfaces and distributed views in both a preview and lists. The lists provide indicators for group membership and space requirement as well as allows ignoring surfaces and views, making them ineligible for automatic layout. Views can also be manually assigned to surfaces by drag and drop. The system reacts differently to these changes: while ignoring views or surfaces triggers a recalculation of the complete distribution, the manual assignment only re-runs the local layout. Here, we expect users to have the mental model of reassigning one specific view, regardless of its relations to other views. Therefore, we skip the view assignment to avoid side effects. The user can also switch to a completely manual process, where they can place and scale views freely.

Currently, distribution updates are only triggered on major changes, such as a changed device configuration or when new views are added to the interface. In these situations, we fade in a miniature overview map of the surface configuration highlighting moved views and/or new surfaces. However, smaller view-specific changes, such as changed filter conditions or a changed view density due to zooming, are ignored in order to avoid interrupting the user.

8.4 Study: User-created Distributions

In order to back up our heuristics, we compared the distribution and layout generated by our system to multiple user-created ones as well as report on user ratings of the provided distributions.

8.4.1 Study Design

Participants We recruited six paid participants (age M=36.8, SD=12.59 yrs; 1 female, 5 male) at the University of Maryland. We required that all of them have both a theoretical and a practical background in data analysis and/or visualization theory, i.e., are actively conducting research in the area or work with such interfaces regularly. All participants have been active in the field for over 3 years (M=9.8 yrs, SD=10.26 yrs). This data was inquired through a questionnaire (Appendix B.2).

Apparatus and Dataset We used the VISTRIBUTE system as described before on a crime dataset from the City of Baltimore [@Bal11]. The example interface consisted of 10 views (Figure 8.5). Two bar charts showed the overall crime distribution for districts and crime types (BARS-DIST-ALL, BARS-TYPES-ALL). Selections in these were used as a filter for two connected line charts each, showing the distribution over time for 2016 and 2017 (LINES-DIST-16/17, LINES-TYPES-16/17). A filter component allowed for filtering the data to explore subsets (FILTER). The filtered output was consumed by two bar charts (weapons, BARS-WEAP-FILT; inside/outside location, BARS-INOUT-FILT) and a map (MAP-FILT). We extended the prototype with a manual layout mode, allowing a free view assignment and arrangement via the control panel. In addition, views could also be moved and resized directly in the interface.

Physical Setup We used three different device setups similar to the ones shown in Figure 8.1. These setups represented realistic device combinations that are already in use or are likely to be commonly used in the near future. Specifically, the three device ensembles were defined as follows:

S1 A *traditional dual-display desktop* setup, with both displays being of the same type (24 ”, full-HD) but mounted in different orientations (one in landscape, one in portrait orientation);

S2 A *novel desktop setup* with a laptop (13 ”, 1600 × 900 px) on a standing desk and a large display (55 ”, full-HD) within arm's reach; and

S3 A *mobile device ensemble* consisting of a smartphone (Samsung Galaxy S8, 5.8 ”, 2690 × 1440 px, landscape), a tablet (HTC Nexus 9, 9 ”, 2048 × 1536 px, landscape), and the laptop from before.

Procedure Participants first received a short introduction on view distribution as well as the experimental dataset. We provided them with an initial understanding for the requirements of a distribution by explaining typical scenarios and tasks in the context of the crime dataset. We also explained the abilities and connections of the existing views as well as provided a printout showing these connections (see Appendix B.3).

In **Phase I**, participants were asked to distribute all views across the available surfaces for all three setups (within-subject design, counter-balanced order). None of VISTRIBUTE's automatic layout functionalities were active during this phase. We asked participants to think-aloud while distributing views and logged the created distributions. As the interface offered no support for alignment, we carefully adjusted them afterwards to remove smaller and unintended overlaps or offsets. These adjusted distributions were used for Phase II; they are listed in Appendix B.1.

In **Phase II**, participants were shown three existing distributions for each setup. For all distributions they were asked to rate its quality on a 5-point Likert-scale and provide free-form comments. Since we included three physical setups, each participant rated nine distributions. The setup order was the same as in Phase I. From the three distributions, two were created by prior participants (randomly selected), while one was generated by *Vistribute*. Their order was also randomized per participant. We did not indicate to participants how these distribution were created. For the first two participants, we used distributions created during earlier pilot runs by different persons. In total, sessions lasted approximately one hour.

8.4.2 User Feedback and Findings

We found three main results: when considering a distribution, (1) participants make decisions based on very similar aspects as embodied in our heuristics, but

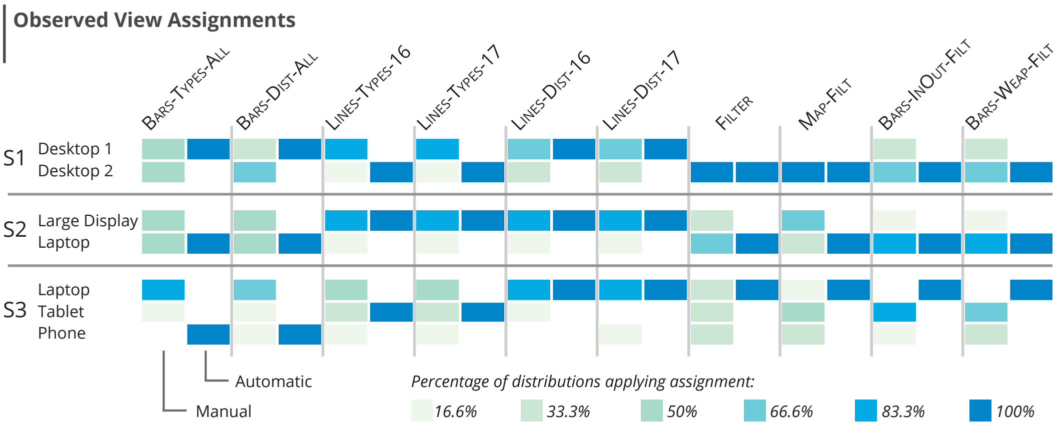


Fig. 8.6.: The observed view-to-surface assignments for each setup are shown in this heatmap. The columns are the views, the rows the surfaces per setup. For each column, the aggregated manual distribution is shown on the left and the automatic one on the right. For example, in S2, 50 % of participants placed the Bars-Types-All view on the laptop and the other on the large display, while almost all placed the line chart views on the large display (similar to the automatic assignment). As our implemented distribution algorithm is deterministic, the automatic assignment is always 100 % for one surface.

(2) personal preferences have a strong influence leading to diverse distributions across participants (Figure 8.6), and (3) the manual distributions were rated slightly better than the automatic ones (Figure 8.7).

When stating their thoughts during the distribution, participants touched on similar principles as covered in our heuristics. For instance, they explicitly stated that views with more data points should be placed bigger (P 1–6), that connectivity must be valued (P 1–4, P 6), or that similar views should be aligned for comparison (P 3, 4, 6). Figure 8.6 also shows some of these patterns: for example, the four line charts (Lines-Types-16/17 and Lines-Dist-16/17) form clear pairs as they are often assigned to the same device, especially for S2 (also shown in Figure 8.5). We also observed participants considering the influence of device size (P 1, 3, 4, 6) or input capabilities (P 2, 3, 4, 6).

However, multiple aspects were considered differently across participants. While most participants valued smaller devices as appropriate for input purposes, P2 used the mobile devices explicitly for visualizations, as these *"can be easily passed around."* For connectivity, we observed that some participants strongly favored placing connected views adjacent to each other (P1, P6), while others found it useful to split them between devices. We also found that some aspect are not covered in our framework yet: multiple participants had a higher-level definition of data similarity by considering their semantics. As an example, the views encoding districts (Lines-

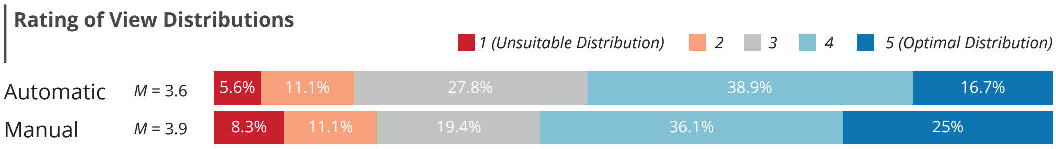


Fig. 8.7.: Percental fraction of the participant's rating for the presented distributions. The manual ones were rated slightly better than the automatic version.

DIST-16/17, BARS-DIST-ALL), the map, and the Inside-Outside bar chart were classified as geographical data, and therefore combined by three participants (P 2, 4, 6). Half of the participants (P 1, 2, 6) also mentioned the importance of surface adjacency and its influence on the perceived proximity between views. For example, views placed on the display borders in S2 can also be considered as adjacent by users, while the current implementation does not incorporate such constellations.

As a result, we could observe a high diversity across the created distributions. In Figure 8.6, this can especially be observed for the bar charts in S1 and S2, as well as for most of the views in S3. Further, no two distributions were similar. Three distributions for S1 and two for S2 used the same view-to-surfaces assignment; however, they had different local layouts. This diversity in user preferences can also be observed in the ratings in the form of high standard derivations. On average, participants rated the manual distribution (M=3.9, SD=0.99) slightly better than the automatic ones (M=3.6, SD=1.21; see Figure 8.7). However, the ratings must be considered carefully: our study included only a small number of participants and they all worked only for a limited time on the distributions without performing specific analysis tasks.

Interestingly, multiple participants found the manual distribution *"exhausting"*, with one participant explicitly stating that *"the computer should suggest where to put things; there should be some optimization for this"* (P 5), and stressing that a manual placement is considered a burden (P 1 + 5). On average, participants spent 8 minutes on the second and third distribution (M=19.6 minutes for the first one). Although a certain part of this time is caused by the think-aloud design and lacking interface support for aligning, even in a real-world system users would eventually have to spend a couple of minutes for the distribution. Any shortcut offered by an automatic distribution would therefore be an improvement. Finally, participant P 1 also noted that *"semantically beautiful is much more important than aesthetically beautiful."* Hence, even if an automatic approach is not able to reach the visual quality of a manual one, it may still be able to provide a valuable layout. All created distributions are listed in the appendix B.1.

8.5 Discussion and Summary

The work on VISTRIBUTE shows that the flexibility coming with dynamic device ecologies should be accompanied with suitable automated mechanisms that eliminate the setup complexity. Providing such a smart configuration of a visual data analysis interface, where properties of views but also of all participating devices are considered, is representing the core idea of device *ecologies*: strength and weaknesses are weight up to form synergies that maximize the capabilities of the devices. As illustrated with VISTILES, this is not only limited to view distribution, but can also be incorporated alongside novel interaction mechanisms.

Challenges of View Distribution The VISTRIBUTE system itself is considered primarily as a foundation for future research on distributed visualization systems. First of all, feedback by the participants indicated that the proposed heuristics could be refined. For example, a semantic data similarity (e.g., all location-related views) or contextual device aspects (e.g., physical device arrangements) are currently not represented, as they are hard to capture. This latter aspect was already subject within the previous chapter, concluding that improving internal sensors will allow to better facilitate these aspects. Secondly, the current algorithm can lead to a radical rearrangement of the views when, e.g., one device is added. Therefore, additional measures to support the analysts in reorienting themselves and understand the changes should be incorporated. For example, such visualizations of changes [APP11; HR07] could involve animated transitions or transient color highlights [Bau+06]. Also, distribution layout changes may require explicit user confirmation or should be listed in a history view.

Towards Smart Device Ecologies While the VISTRIBUTE framework does not stipulate a specific distribution algorithm, the current example implementation is a rather simple algorithm realizing the heuristics, rather than a formal user interface specification such as AdaM [Par+18]. This strategy helped to inform a better understanding of the mechanics coming into play, while the evaluation showed that the quality was similar to layouts hand-crafted by experts. Nevertheless, extending the current algorithm towards an optimizer can help to improve the distribution quality. This could also be further extended by applying machine learning approaches for deriving weights for the heuristics, although the reliance on large training data sets renders this more suited as a future step when distributed visualization interfaces were used more broadly. Finally, even when following this vision towards a distribution purely based on formalism, it remains central that users are able to control and modify the result. Notably, it should be possible to apply these adaptions in a natural way, e.g., by drag-and-drop, and not through abstract parameters, as it is often the case for current optimizers [Par+18].

Within the context of visual data analysis, creating the actual visualizations as well as hooking them up in the data flow remains an important and time-consuming task. Consequently, being able to automatically generate and modify the views (instead of working with existing views) would represent one further step to better support data analysis in device ecologies in general but also to support ideas as presented in VisTiles. For example, instead of just aligning two views in order to promote visual comparison, an even more sophisticated approach would be to rebuild the views to use the same chart type and normalize both of their scales to further increase consistency [QH18]. Notably, this also involves better incorporating strategies from responsive visualization design, such as changing the view configuration to optimize the level of detail or preserve a correct perception [HA06] when scaling views. This step, to either manipulate or generate views to complement existing ones, or even to generate a complete dashboard from scratch [Mor+19], is not far. Unlike the human designer, who can only enumerate so many variant visualizations for a finite set of possible device ensembles, a fully automated visualization generation engine would be able to construct precisely the visual representations that are best suited to the available hardware, physical context, and overarching analysis tasks.

Part V

Conclusion

9 Discussion and Summary

This thesis started out with the goal to explore ways of how visual data analysis can be supported in dynamic device ecologies. I argued that with the availability of modern computing devices and an increased desire to conduct data analysis in a more flexible way, the combined use of such devices can not only fulfill this desire but also potentially lead to a more natural and intuitive way of conducting such analyses. I underpinned this overall goal with three research questions that need to be addressed: how visualization can be supported on all the different devices available (RQ1), how specific device combinations can allow facilitating analysis workflows (RQ2), and how the overall data analysis can be enabled in fully dynamic device ecologies (RQ3). Each research question is addressed by one part of this thesis, providing valuable contributions to the specific aspects. Overall, I am confident that my research confirms the initial assumption that device ecologies can facilitate visual data analysis. Moreover, my research highlights this potential and provides concrete solutions, which can serve as a strong basis for future explorations.

In this chapter, I will conclude my thesis by recapping the gained insights and made contributions (Section 9.1) as well as reflecting on the remaining challenges and resulting opportunities for future work (Section 9.2), before ending it with closing remarks (Section 9.3).

9.1 Insights and Contributions

The research presented in this thesis contributes to the understanding of prevalent mechanisms and requirements of data analysis in device ecologies. It investigates this special visualization environment from multiple perspectives, resulting in multiple major and minor contributions. These are summarized in the following:

Strategies for Supporting Visualization on Heterogeneous Devices

The concept of responsive visualization describes a flexible visualization design approach, which allows to create visualizations that can adapt to certain factors such as device properties. Chapter 3 contributes an **extensive overview on factors and strategies of responsive visualization**, i.e., the various aspects that can influence the work with visualizations as well as the different adaptations that can be applied as a responsive behavior. As this overall topic is underexplored within the visualization research area, this work is one of the first providing a broader discussion of it. While further work investigating the various strategies in more detail is required, the chapter presents a valuable foundation that could be developed into more specific guidelines in the future. In the context of device ecologies, responsive visualization takes on an important role, as views have potentially to be displayed on a wide range of different devices.

In Chapter 4, the aspects of responsiveness are considered for the specific use case of multivariate graphs. Here, the main contribution is the *Responsive Matrix Cells* concept, which comprises a **focus+context approach for a matrix visualization**. By allowing to embed different visualizations into local focus regions, the exploration of details of the multivariate graph is enabled, while maintaining the overall context as well as the dimensions of the original visualization. These focus regions are a promising approach to provide a rather compressed visualization as an overview while still **supporting extensive analysis and editing workflows within a compact space**. Importantly, aspects of responsiveness are incorporated for the embedded visualizations, illustrating how factors such as available size but also underlying data or user intents can be used.

In sum, Part II of this thesis underlines the importance of flexible and responsive visualization within data analysis interfaces. This is especially relevant for device ecologies, where a wide range of devices can be considered and combined. Further, the works also highlights that responsiveness is a fluid concept relating to interaction, e.g., also allowing for addressing task-specific needs of analysts.

Concepts for Facilitating Data Exploration using Device Roles

When considering device ecologies, it is crucial to understand how the interplay between devices can be used for facilitating data analysis. By proposing *Watch+Strap*, Chapter 5 contributes the **novel device type of a smartwatch with display-equipped wristbands**, thus representing a very specific display ensemble. Besides the technical and physical aspects, one major part of the research on *Watch+Strap* is the conceptual framework proposing **strategies for using the displays in synergy**. It can be concluded that a continuum exists of how strongly coupled the displays are, i.e., ranging from acting as one joint surface to behaving as separate ones. Consequently, multiple roles exist that the displays can take on, and thus, multiple ways how they can support data analysis. Particularly promising for the *Watch+Strap* setup are combinations that resemble overview+detail or focus+context schemes.

Another device ensemble is considered in Chapter 6 with the combination of a small and personal device with a large and shared device. Specifically, the work contributes a conceptual work supporting **visual data analysis within the novel combination of smartwatch and large display**, two very contrary but complementing devices that were depicted as *When David meets Goliath*. The watch serves as a personal toolbox allowing to control and augment the interaction with the main analysis interface provided on the large display. As the watch is a body-worn device, it provides anytime access to stored sets and configurations, can mediate the interaction taking place on the large display, as well as serve as a tool for interacting from a distance. Thus, specific interaction and visualization **concepts that apply device roles for supporting analysis workflows** were proposed.

Taken together, Part III contributes to the understanding of how devices can complement each other by taking on different roles within visual data analysis workflows. The roles and mechanics applied in these specific and novel device ensembles then also inform how ad-hoc combinations in dynamic device ecologies can be used to generate synergies.

Concepts for Enabling Data Analysis in Device Ecologies

The insights gained so far are important foundations for providing beneficial analysis interfaces for device ecologies. With Chapter 7, the conceptual framework *VisTiles* is proposed, contributing **interactive mechanisms for controlling device coordination and combination** in dynamic setups through spatial device arrangements. In these ecologies, device roles are not predefined, but can be spontaneously established. Then, these result in specific coordination of views, e.g., linked brushing, synchronized configurations, or even merged representations. For controlling these combinations, considering the spatial arrangement of devices is promising as it is also an integral way of how people structure work artifacts, providing a novel and more **natural interaction style for data analysis environments**. In addition with recommendations proposing suitable coordination functionalities, analysts become able to easily control their device ecology.

Besides managing and controlling the interface during the analysis process, provisioning the analyst with a ready-to-use interface for a device ecology is a crucial element to truly support visual data analysis. By contributing the *Vistribute* framework in Chapter 8, an **approach for automatically distributing an visualization interface across devices**. This approach is based on an **in-depth consideration of both device and visualization properties as well as relationships**, which are then taken into account for deriving a distribution. By following intelligible heuristics, this distribution process can be fine-tuned for various constellations and remains explainable to the user. Overall, *Vistribute* shows that it is possible to eliminate the exhausting effort of manually positioning visualization views across devices while still providing distributions with a similar quality to manual created ones.

The two chapters of Part IV provide important approaches for actually realizing analysis interfaces for *device ecologies*. This includes smart mechanisms for controlling the coordination and applied roles for devices as well as for providing an initial interface for starting or continuing the analysis process.

Overarching Contributions

Lastly, the thesis as a holistic work provides further overarching contributions. First of all, this thesis proposes multiple **strategies for how mobile devices can facilitate visualization and data analysis**, highlighting their potential and central part of future developments. Specifically, mobile devices can allow for working with data while on the go (*responsiveness*), can support and enhance the analysis taking place on other larger devices (*David meets Goliath*), or even form powerful device ecologies on their own (*VisTiles / Vistribute*). Second, it has been indicated that following **multi-view approaches for data analysis interface in device ecologies** could provide powerful analysis environments. Besides the possibility to map visualizations to devices or physically group related views, it is also possible to apply interactive interface schemes such as overview+detail or focus+context more easily as well as to make full use of the devices capabilities and characteristics by providing suitable and adapted views considering, e.g., size, ownership, input modalities.

On the technical side, this thesis also underlines the **high suitability of web technologies for cross-device data analysis**, which were used in all implemented software prototypes. While existing libraries such as D3.js [BOH11] or Vega [Sat+17] already led to a widespread usage of high-quality web visualization across various domains, such a common strategy has not been established for cross-device interfaces yet. With the prototypes of this thesis being open-source, the followed architectures and communication approaches can ease this development and inspire future implementations.

Overall, this thesis contributes analytical, conceptual, and empirical insights in combination with technical aspects that together notably extend the knowledge on visual data analysis in device ecologies.

9.2 Remaining Challenges & Future Opportunities

While my work provides new strategies and considerations for conducting data analysis in device ecologies, it can only serve as a foundation within this vast space and naturally faces certain limitations due to the chosen scope and the extent possible within a dissertation. In the following, I will recap the limitations before reflecting more extensively on the remaining challenges but also opportunities that can be picked up by future research.

9.2.1 Limitations

Extent of Considered Visualizations & Data Types For visualizations, the number of existing techniques and different data types that could be explicitly considered is huge. Therefore, the scope was limited to 2D representations of multivariate data. While this already covers a large spectrum, other data types as well as visualization techniques might come with different requirements and would behave differently in device ecologies. For example, analyzing time-dependent data, dynamic data, or even scientific data provides new possibilities for interaction concepts, such as slicing time by using spatial device movements, which are not covered here.

Extent of Considered Device Types & Combinations Similarly to the considered visualization aspects, the variety of devices and the many different device combinations that are possible cannot be covered within one dissertation. While the considered combinations were carefully chosen to be representative, other device ensembles requiring different mechanisms can exists. This holds particularly true for device types that were out of scope here, such as tabletops, shape-changing devices, or mixed reality devices.

Input Modalities Within the presented interaction concepts, the input was primarily limited to touch interaction as well as instances of spatial interaction. While these are the most relevant input modalities for modern devices and their combined use in device ecologies, more modalities exist that might proof beneficial. Among others, this includes pen input for more precise interactions or speech input, which can allow to even better support a more natural interaction style. Particularly the combination of multiple modalities into multimodal interaction mechanisms are promising [Fri12; Sri+20; SLS20], but have not been considered here. Within this thesis, the only example of multimodality can be found within the *David meets Goliath* chapter (Chapter 6) where remote interaction involved spatial movements for pointing and touch for confirmation.

Qualitative & Quantitative Evaluations The proposed concepts and systems were evaluated in multiple small-scale studies focusing on qualitative aspects. This allowed us to collect valuable feedback and iterate the involved techniques as well as to better understand how people are conducting data analysis within the considered device environments. At the same time, this means that it cannot be judged yet if such modern environments are more effective or even efficient than traditional desktop environments. For these aspects, quantitative studies would be required, however, finding truly comparable setups is challenging. For example, working on a large display promotes a totally different interaction style compared to desktops, as the user is standing and others can easily join. In addition, it can also be discussed if the overall goal should be to reach a similar efficiency, or rather to enable people to conduct data analysis in a wide range of contexts. Still, for the work presented here it remains a limitation that the quality of the supported data analyses in device ecologies has not been investigated in detail and is left for future work.

Collaboration In particular the concepts presented in Chapter 6 (*David meets Goliath*) and in Chapter 7 (*VisTiles*) indicate the possibilities for multi-user scenarios that device ensembles can offer. Due to a mix of personal and shared devices, analysts are enabled to conduct data analyses on their own, in parallel, or together without the need to switch the environment or interface. The gained feedback from participants indicated that they could well envision collaborative analysis sessions. However, to fully support and investigate such sessions both conceptual as well as technical extensions are still required. For example, additional concepts for avoiding conflicting interactions, supporting sharing information among users, as well as joint interaction means would be required [Ise+11]. Further, technical challenges remain, for example, immediate and reliable user identification on shared large screens [Zad+16].

9.2.2 Reflection of Visualization Aspects

As the discussion of the limitations already indicates, there are multiple conceptual aspects that still have to be investigated in more detail. While the single chapters already discussed directions for future work in their respective context, I want to further reflect on more broader aspects related to device ecologies that seem particularly interesting and promising for future research.

Generalizability & Design Challenges

Even within the limited scope of this thesis, the large number of different devices as well as visualization techniques that can be considered and possibly combined is huge. Consequently, neither developing one big technical framework that covers all instances of data analysis in device ecologies nor optimized solutions for every single combination are feasible. Based on my own investigations, I argue that indeed both should not be the goal. Instead, it will be more beneficial to first further advance a general understanding of how analysts want to make use of devices and how they try to set up an overall interface. In other words, future work should continue exploring different device ecologies for different types of visual data analyses. Based on such explorations, the next step can then be to derive guidelines or best practices for designing distributed analysis interfaces for device ecologies.

Those guidelines could describe, for example, which coordination functionalities between devices should be provided, how they can be mapped to specific interactions, or which degree of automatization should be incorporated. Naturally, outlining design guidelines will not allow to cover all instances and special cases, but would likely be an important piece to promote the wider use of device ecologies for data analysis. The work presented and discussed here is contributing many insights for that, e.g., by discussing device roles in different setups, proposing technical approaches as with VISTRIBUTE, or exploring novel interaction mechanisms as in VISTILES. Still, they are specific instances, and deriving representative guidelines will involve considering further instances.

When thinking about providing guidance in the form of a design method, it is also important to develop tools that enable prototyping and testing. This includes previewing setups that might not exist yet as well as to test specific visualization and interaction concepts. On the one hand, this can be addressed by open prototype platforms, which provide both easy-to-use software as well as hardware components. Instances are, for example, WATCHCONNECT by Houben and Marquardt [HM15] or this thesis's WATCH+STRAP [@KHD]. The technical implications for such a platform's software architecture will be discussed in more detail later in Subsection 9.2.3.

On the other hand, such platforms might not always be feasible. Particularly in early stages of a design process, low-fidelity prototyping methods can be better suited to illustrate early concepts and ideas. Specifically this can be supported through paper prototypes and sketches (see the brainstorming section of *Watch+Strap*) or freely placing static image content in the considered environment [Bre+19b]. Finally, device ecologies can involve devices that are not always available, such as display walls. To still be able to test these, virtual reality environments are a promising way for simulating a desired device ecology [Jet+20].

Responsive Visualization

As the work presented in Part II has shown, responsive visualization is an important aspect for any design process of analysis interfaces for device ecologies. As already discussed, the current knowledge on responsiveness for visualization is not sufficiently covered through systematic explorations or guidelines, particularly with respect to factors other than screen size. Any future work on responsive visualization that contributes a systematic abstraction or an embedment into existing visualization design processes can have a big impact. I want to point out that responsiveness should not only be considered in the context of modern devices, but as a fluid concept that considers its surrounding interface context. Specifically, a visualization can be responsive itself, but also serve as the context for further responsive visualizations that are embedded in it via techniques such as semantic zooming or focus+context. Consequently, this means that some factors can be overwritten or become irrelevant, while other factors can be inherited from the hosting visualization. For example, the display size might not be the relevant limitation for an embedded visualization, but rather the dimensions of a container element. At the same time, factors such as interaction modalities, environment, or visualization literacy would be the same.

In addition, the question remains how visualizations should respond specifically to the present factors, i.e., how they can be adapted in a useful way. For future work, one goal would be to investigate what 'useful' adaptations are and what users would expect as responsive behavior. As it might not be possible to know what the most useful adaptation would be, e.g., as it is not clear what the specific tasks or interests of analysts are, it can be an alternative to provide the user with multiple options. For example, within the *Responsive Matrix Cells* approach, this is done by allowing the user to switch between different visualizations and their orientation. In addition, it would also be possible to provide different adaptations of the same visualization, e.g., one version with outliers filtered out, one with a different clustering approach, or one with an altered color encoding.

When considering switching visualization techniques completely, it might be interesting to also automate such changes based on recurring adaptation patterns in order to reduce overall interaction costs [Lam08]. This would result in a time-dependent responsiveness that follows mostly human factors, i.e., analysts tasks and goals. As such analysis patterns are typically highly application- and user-dependent, machine learning methods could be required to infer automatable adjustments from previous user interactions [Bro+14; End+17; OGW19]. Particularly in the context of device ecologies and multi-view interfaces, this can also help to improve automatically setting the layout and arrangement of the views, similar as it was here proposed with VISTRIBUTE for the overall view distribution. In general, better considering the possible responsive adaptations in the context of the overall distributed interface, thus in relation to other views, is also an interesting direction for future work.

Automation & Human-in-the-Loop

As demonstrated with VISTRIBUTE and discussed for responsiveness, smart and automated optimizations play an important role for device ecologies. The high dynamic leads to the need of repeated adaptations of the interface and its components, e.g., because new devices are added, visualizations have to be moved and aligned with other views, or the analysts simply wants to focus on a different data subset. As iterated before, updating a view distribution alone is an exhausting effort for users and therefore should be simplified as much as possible. At the same time, these automations can hardly provide an optimal solution for all situations, particularly when considering an unlimited space of devices and visualization techniques. Further, there is the chance that automated optimizations come as surprise to users and are unintended. For example, the early iterations of VISTILES applied specific view combinations based on pre-defined device arrangements, such as a horizontal side-by-side arrangement or a vertical one. However, users stated that these required arrangements were hard to remember and also sometimes triggered by accident when simply moving devices closer to each other. Consequently, the later version of VISTILES followed an 'application suggests, users confirm' approach, thus applied a basic recommender system that kept the user in control.

In general, a distributed interface can be optimized with respect to multiple aspects: the view distribution, the applied visualization adaptations, as well as the supported interactive combinations. For all of them, it must be considered how automated optimizations are incorporated specifically. Weighing user control against reducing user effort must be done carefully and can be realized in different flavors. On the one end, the system could (a) only propose possible changes that an analyst has to accept. One step further, the system could (b) apply a candidate optimization as a preview, but still requesting the analyst to confirm this change. Then, the same could be done (c) without asking for confirmation but offering an undo. Or, finally, (d) changes can be directly applied without allowing any user participation. Exploring these options in more detail is another direction for future work, and especially relevant for visual data analysis in device ecologies. This is also closely related to other research investigating how to learn from user interaction and reapplying them in other situations [Bro+14; End+17; OGW19]. Finally, it must also be considered how smart optimizations can be provided in a way, in which they remain explainable and understandable, so that analysts can understand why an optimization has been applied in a certain way [Abd+18].

Transforming Devices and Detached Views

Lastly, an important discussion has to be held on how the long term perspective of device ecologies will be shaped. It is likely that the diversity of devices will further increase, progressively blurring the boundaries between device classes. In fact, multiple novel devices featuring multiple displays [@LG20; @Mic20], foldable displays [@Len20; @Sam19], or expandable displays [@LG19] have been presented or announced in recent years. In consequence, it can neither be taken for granted that a device has only one specific output space (i.e., display) nor that this output space has fixed properties. Thus, the possibilities of displaying content on one device are getting more flexible and with that also the possibilities for device ecologies. In return, existing concepts and strategies from device ecologies can also inform the design of interfaces for, e.g., foldable displays. Similar to the discussion made within the *Watch+Strap* chapter, the display could be interpreted as a continuous surface but also as separated display parts that are divided by the fold, which then can take on different roles in the ensemble.

Besides advances in display technology, also mixed reality (MR) technology is progressing. While currently rather heavy headsets such as the HoloLens [@Mic16] have to be worn for providing MR environments, it can be assumed that in future notably smaller glasses or even contact lenses can enable MR immersion. This can lead to a shift from device ecologies to *view ecologies*, as in those environments visualization views are no longer bound to a specific surface and are truly becoming available anytime, anywhere [Pav+21]. Still, concepts for device ecologies can serve as a starting point for investigating how analysis interfaces, or any multi-view interface, can be realized in such view ecologies as well as how views can step in an interplay with each other. In addition, it is also likely that mixed reality will be used in combination with physical displays or devices [Lan+21; RFD21], as these can provide physical affordances that are hard to replicate in mixed reality.

9.2.3 Reflection of Technical Aspects

Distributed computing environments pose special challenges for implementing the interfaces as well as setting up the connection between devices. Based on the experiences gained by implementing this thesis' web-based prototypes, I want to further reflect on how device ecologies can be realized from a technical perspective.

Communicating and Syncing across Devices The most important piece is the communication between the available devices. This involves *how* they communicate with each other and *what* information is exchanged. In all prototypes, the WebSocket protocol was used for the communication (by using Socket.io [@Aut14] as a wrapper). This protocol is a de-facto standard for low latency and low cost messaging in web environments and has proven to be very suitable for the use within device ecologies. The information that is exchanged, however, can follow different approaches: Either discrete low-level events (e.g., touch) are sent or cumulated states (e.g., active selections). This is directly related to the question how the overall system is designed and where the main interface logic is hosted.

The interface logic can be either distributed, thus, each client knows how to react to certain events, or be hosted on one distinct client or server. The latter approach can help to reduce the amount of communication that is required as well as to simplify the overall complexity of the system. In these setups, often low-level events and simple cumulated states are sent. Most of the developed prototypes follow such an architecture. In a distributed approach, there is an additional overhead required to ensure that all information is provided to all clients as well as to avoid contrary reactions from them. Such a setup is used by the VISTRIBUTE system. There, to avoid conflicting reactions, one client is dynamically selected as master client that runs the parts of the interface logic concerned with the cross-device functionalities.

Prototyping and Testing For prototyping and testing, running a centralized architecture proved to be very effective. As the logic and code is hosted at only one point, changes can be more easily incorporated and do require no or only minor updates to the other clients. This is particularly interesting for clients that rely on a deployment process for code updates, such as the Arduino-based strap displays, which were used within the WATCH+STRAP system. In fact, this setup is an example for an even more extreme architecture using an *interface streaming approach*. To circumvent programming efforts in limited or very different environments than web-based ones (e.g., Arduinos), the respective clients only display a pre-generated image stream received from the main client and feed back observed low-level events. This allows to make full use of the rendering capabilities of web technologies that would otherwise not have been available. A similar approach was also used for an early prototype of VISTILES and other research work [Hol+13; Kis+17; Lan+16].

The downside of such a streaming approach is its scalability as well as a slightly reduced output quality and increased latency. The required bandwidth for sending image content in a sufficient framerate can easily be provided for a few clients, but with an increased number of endpoints, wireless networks can hit capacity limits. Further, the required computation power is increased on the main client, while the applied image compression can lead to a reduced output quality. The developed prototypes of DAVID MEETS GOLIATH and VISTILES follow a hybrid model of distributed and centralized architectures instead. Here, the clients process simple interactions themselves, such as rendering visualizations or processing touch events. The resulting changes in the visualization state, such as made selections, are then communicated to the main client, which determines if further adaptations on the devices are required and sends respective instructions to them. While the complexity is then slightly increased, such setups still proved to be powerful for prototyping purposes.

Towards Productive Systems For the development of productive environments, one major challenge is allowing true ad-hoc setups. In the prototypes of this thesis, clients always connected to one specific server that is previously known. This server provides then the application page as well as the WebSocket server, forwarding messages to the target devices. Depending on the prototype, the server could also host parts of the interface logic. However, the goal of device ecologies should not be requiring users to enter a certain URL on all devices, but to support some auto-discovery approaches (see, e.g., TRACKO by Jin et al. [JHH15]). Further, such setups ideally would also run on a fully distributed architecture without requiring a central server at all. While approaches exist for realizing such setups (see, e.g., SurfaceFleet by Brudy et al. [Bru+20]), they remain complex to deploy. This becomes also apparent when looking at existing distributed visualization solutions such as VisConnect [Sch+21] or Vistrates [Bad+19]. They all require that another client connects to a specific URL, which is a suitable solution for remote collaboration but less for co-located ad-hoc setups. In conclusion, it can be an interesting direction for future work to explore how such setups could be realized for co-located and dynamic device ecologies.

9.3 Closing Remarks

With the increasing diversity as well as sheer number of available devices, the potential of using them in combination in device ecologies is big. Particularly for visual data analysis, device ecologies can enable more flexible workflows by incorporating the devices that are most suited for the current situation and handle the overall interface in a more natural way. Further, device ecologies can be available and set up anywhere, anytime. The work presented in this thesis contributes to different aspects relevant for supporting visual data analysis in device ecologies: Responsive visualization design can provide more flexible data representations that can be used on a wide range of devices and contexts. Incorporating device roles for providing interaction and visualization concepts allows to better support specific analysis workflows within a used device ensemble. And, novel interaction concepts in combination with automated optimizations allow to handle and simplify the dynamic aspects of device ecologies for analysts. Taken together, these contributions have the potential to both inform future research in this area and guide the realization of productive analysis tools for device ecologies.

A

Appendix for Chapter 6 (Study on DAVID MEETS GOLIATH Setup)

This appendix provides supplementary material for the conducted user study for Vistribute as described in Chapter 8. First, B.1 shows the distributions created by participants as well as by Vistribute for each setup. B.2 provides the used questionnaire. B.3 shows the printout that was handed out to participants during the study.

A.1 Questionnaire for Formative Evaluation: Demographics and Ratings

The following questionnaire was used to query the participants' demographic information as well as to assess their ratings of our concepts after the session.

Demographics

1. **Gender:** □ Male □ Female
2. **Age:** ___________
3. **Experience with smartwatches:**
   No experience □—□—□—□—□ Expert
4. **Do you own and use a smartwatch?** □ Yes □ No
5. **Experience with data analysis using visualization:**
   No experience □—□—□—□—□ Expert

6. **Please briefly describe your experience with data analysis. Also, mention which visualization tools you used:**

____

____

Post-Session Ratings

7. **Generally, the combination of smartwatches and a large display is useful:**
   Strongly Disagree □—□—□—□—□ Strongly Agree

8. **Utilizing the smartwatch to access personalized content (instead of directly on the large display) is useful:**
   Strongly Disagree □—□—□—□—□ Strongly Agree

9. **I prefer to use a smartwatch for accessing personalized content over a hand-held smartphone or tablet:**
   Strongly Disagree □—□—□—□—□ Strongly Agree

10. **Defining the context of an interaction by touching the visualization is suitable:**
    Strongly Disagree □—□—□—□—□ Strongly Agree

11. **Previewing and pushing is useful for comparing sets:**
    Strongly Disagree □—□—□—□—□ Strongly Agree

12. **The proposed interaction vocabulary for pulling and pushing content (swiping left/right) is suitable:**
    Strongly Disagree □—□—□—□—□ Strongly Agree

13. **Further suggestions:**

____

____

____

____

A.2 Questionnaire for User Study: Demographics and Ratings

The following questionnaire was used to query the participants' demographic information as well as to assess their ratings of the conditions after the session.

Demographics

This part is identical to the questionnaire part of the formative evaluation.

1. **Gender:** □ Male □ Female □ Other: ________________

2. **Age:** __________

3. **Experience with smartwatches:**
   No experience □—□—□—□—□ Expert

4. **Do you own and use a smartwatch?** □ Yes □ No

5. **Experience with data analysis using visualization:**
   No experience □—□—□—□—□ Expert

6. **Please briefly describe your experience with data analysis. Also, mention which visualization tools you used:**

   ____

   ____

Post-Session Rating: Dual Condition

7. **Is the combination of smartwatches and a large display *useful* for data exploration?**
   Not very useful □—□—□—□—□ Very useful

8. **Is the combination of smartwatches and a large display *effective* for data exploration?**
   Not very effective □—□—□—□—□ Very effective

9. **Is the combination of smartwatches and a large display *easy to use* for data exploration?**
   Not very easy □—□—□—□—□ Very easy

10. **Is creating data item collections (sets) with the smartwatch suitable for data exploration?**
    Not very suitable □—□—□—□—□ Very sutiable

11. **Explain why:**

    ____

    ____

12. **Are the proposed interactions for pulling/previewing/pushing sets suitable for the comparison tasks?**
    Not very suitable □—□—□—□—□ Very sutiable

13. **Explain why:**

    ____

    ____

14. **Are the proposed interactions for configuring sets and remote interaction suitable for the comparison tasks?**
    Not very suitable □—□—□—□—□ Very sutiable

15. **Explain why:**

    ____

    ____

16. **Further comments:**

    ____

    ____

Post-Session Rating: Single Condition

17. **Is the large display condition *useful* for data exploration?**
    Not very useful □—□—□—□—□ Very useful

18. **Is the large display condition *effective* for data exploration?**
    Not very effective □—□—□—□—□ Very effective

19. **Is the large display condition *easy to use* for data exploration?**
    Not very easy □—□—□—□—□ Very easy

20. **Is creating data item collections (sets) on the large display suitable for data exploration?**
    Not very suitable □—□—□—□—□ Very sutiable

21. **Explain why:**

    ____

    ____

22. **Is the large display suitable for managing sets?**
    Not very suitable □—□—□—□—□ Very sutiable

23. **Explain why:**

    ____

    ____

24. **Are the proposed interactions for creating and previewing sets suitable for the comparison tasks?**
    Not very suitable □—□—□—□—□ Very sutiable

25. **Explain why:**

    ____

    ____

26. **Further comments:**

    ____

    ____

Post-Session Rating: Dual versus Single Condition

27. **Between the two platforms, which one do you prefer for visual data exploration?**

☐ Large display + smartwatch (Dual)

☐ Large display (Single)

28. **Explain why:**

____

____

29. **For multi-user interaction, which condition is more useful?**

☐ Large display + smartwatch (Dual)

☐ Large display (Single)

30. **Explain why:**

____

____

31. **Do you prefer to use a smartwatch for accessing content over a handheld smartphone or a tablet?**

☐ Yes

☐ No

32. **Explain why:**

____

____

A.3 Study Tasks: Question List

The following table shows the questions that had to be answered by the participants during the user study. For each question, the question type is stated. The questions were always provided in the same order. Which condition was tested first was, however, counterbalanced.

Training

ID	Type	Question
0.1	QT2	What are the two most frequent crimes in 2015?
0.2	QT3	What are the differences between crimes with knifes and with firearms regarding time periods (CrimeTime)?
0.3	QT1	How many Burglaries happened in July?

Condition 1

ID	Type	Question
1.1	QT1	How many auto thefts happened in the Southern district?
1.2	QT1	What is the difference for crimes committed in all Northern districts between April and August? (value)
1.3	QT3	What are the differences between crimes in the Northern districts (Northern, Northwestern, Northeastern) and the Southern districts (Southern, Southwestern, Southeastern) regarding weapons used?
1.4	QT2	What are the two most frequent crime types in the Central district?
1.5	QT3	What are the differences between car related crimes (Larceny from Auto, Auto Theft, Robbery - Carjacking) and residence related crimes (Burglary and Robbery - Residence) regarding crime time as well as district?
1.6	QT3	What are the differences between all eastern districts (i.e., Northeastern, Southeastern, Eastern), all western districts, and all central districts (Northern, Southern, Central) regarding crime type as well as weapon?
1.7	QT2	In Q1 (Jan-Mar), at which time of the day did most crimes happen and what are the most common crimes?
1.8	QT2	What are the two time periods (Crime Date) with the most larcenies for all southern districts as well as all northern districts?
1.9	QT3	For the two crime types that are most often conducted with firearms, what are the differences regarding crime time, districts, and months?

Condition 2

ID	Type	Question
2.1	QT1	What is the difference for crimes committed with hands between the Northeastern and the Southwestern district?
2.2	QT1	How many street robberies are committed with firearms?
2.3	QT3	What are the differences between crimes in the Northwestern and the Southern district regarding crime types (Description)?
2.4	QT2	Which two months of the year have most Auto Thefts?
2.5	QT3	What are the differences between crimes in Q1 (Jan-Mar) and Q4 (Oct-Dec) regarding crime type and weapon?
2.6	QT3	What are the differences between crimes happening in the morning (0:00-6:00), the afternoon (12:00-18:00), and the evening (18:00-0:00) regarding weapon as well as district?
2.7	QT2	What are the two months with the most burglaries and larcenies in all northern districts?
2.8	QT2	For robberies, which weapon is most common and in which months do they most occur?
2.9	QT3	For the month with most crimes during night (0:00-06:00) and the month with most crimes during day (9:00-15:00), what are the differences regarding crime type, districts, and weapon?

B

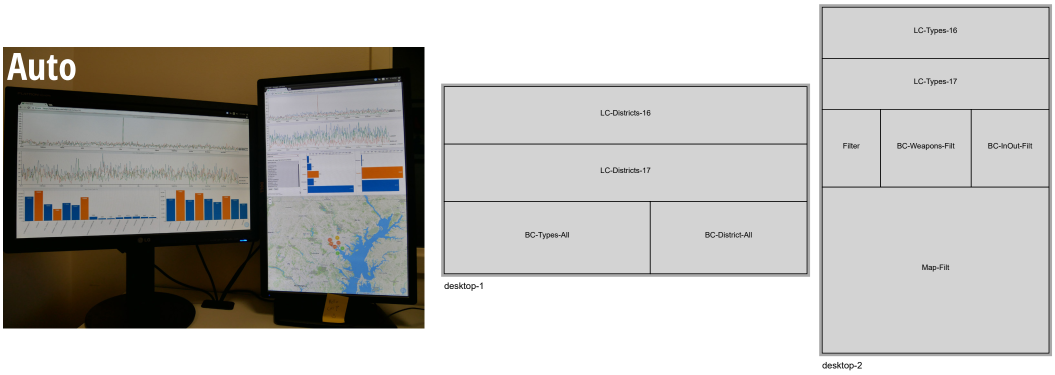
Appendix for Chapter 8 (Study on VISTRIBUTE)

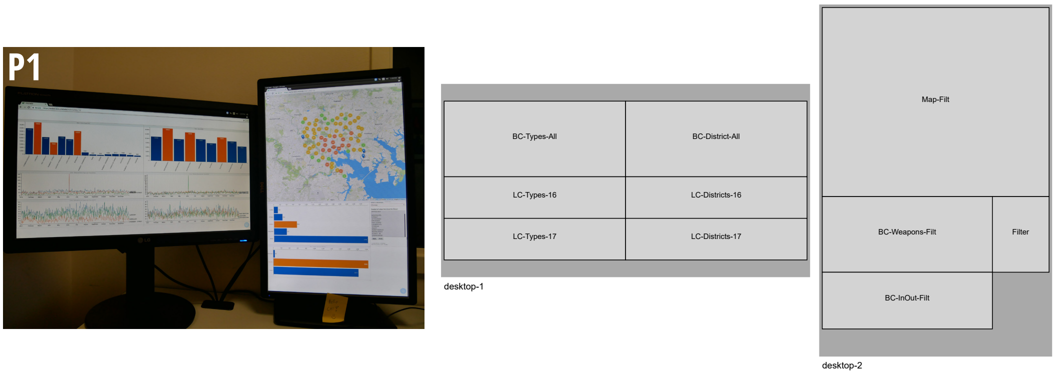
This appendix provides supplementary material for the conducted user study for Vistribute as described in Chapter 8. First, B.1 shows the distributions created by participants as well as by Vistribute for each setup. B.2 provides the used questionnaire. B.3 shows the printout that was handed out to participants during the study.

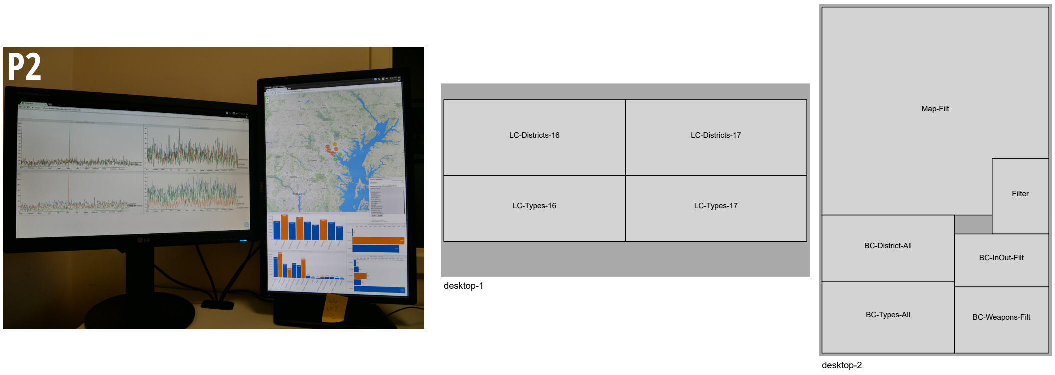
B.1 Created View Distributions

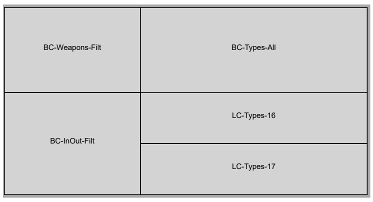
In the following, the view distributed created by the participants (P 1–6) as well as the distribution generated by Vistribute (Auto) are shown for each setup. For each distribution, a photo is shown on the left and the abstracted view arrangement on the right. In the abstracted representation, some view names are further abbreviated: BC-* equals BARS-* and LC-* equals LINES-*.

Setup 1: Dual-Display Desktop

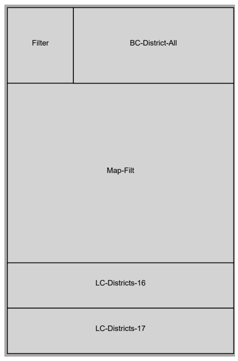




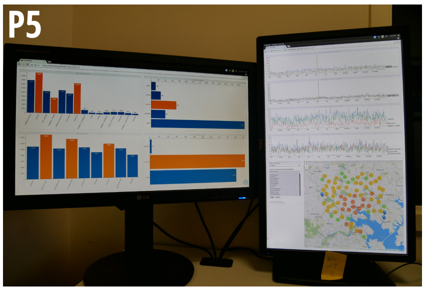


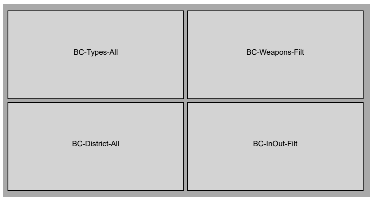


desktop-1

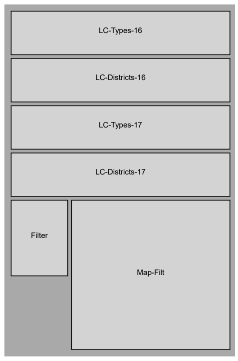


desktop-2

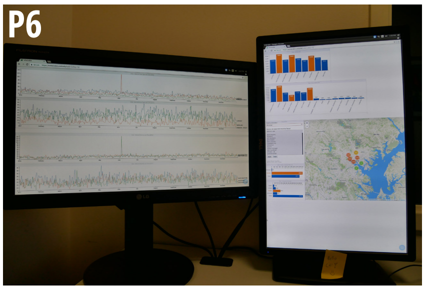


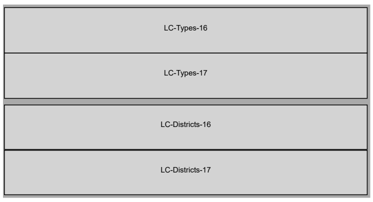


desktop-1

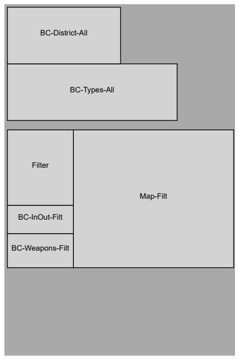


desktop-2



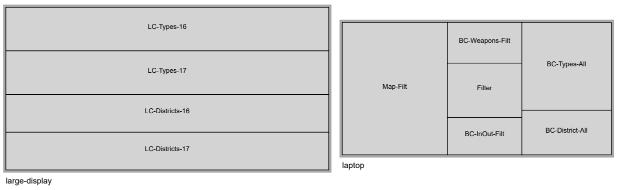


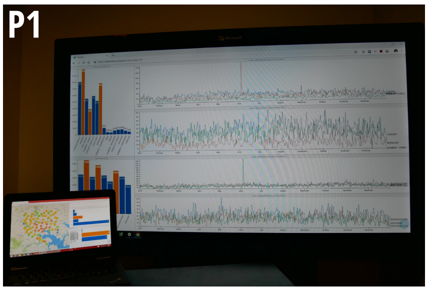
desktop-1

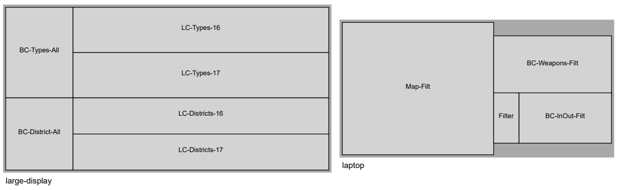


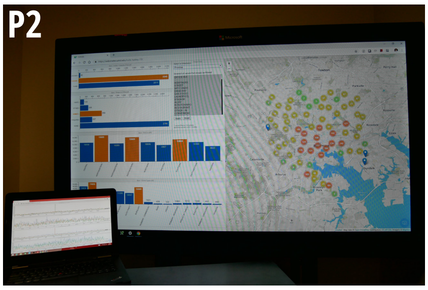
desktop-2

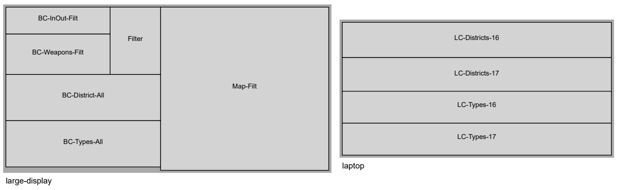
Setup 2: Large Display and Laptop

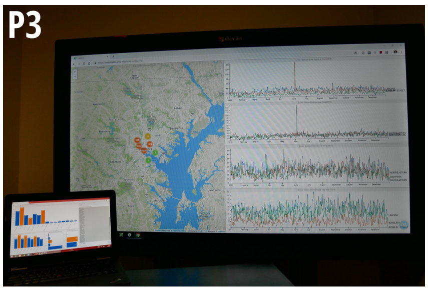


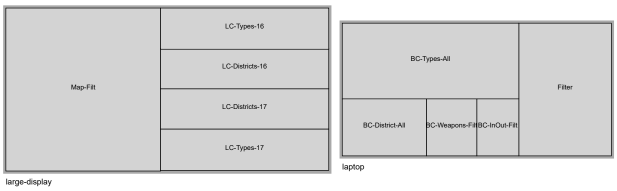


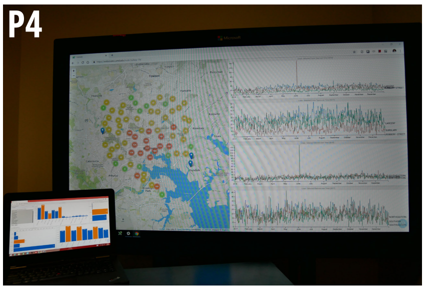


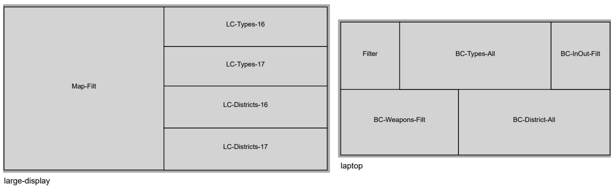


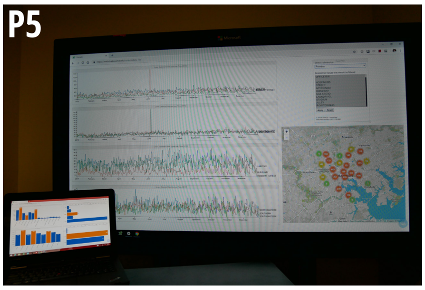


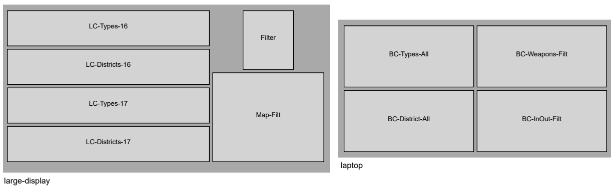


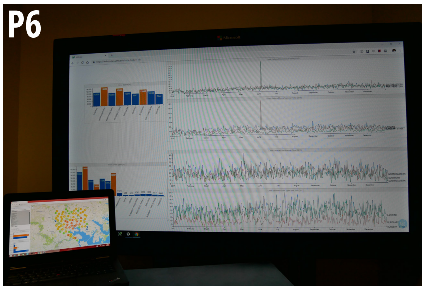


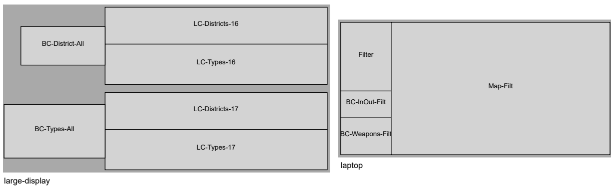

P4


Map-Filt
LC-Types-16
LC-Types-17
LC-Districts-16
LC-Districts-17
large-display
Filter
BC-Types-All
BC-InOut-Filt
BC-Weapons-Filt
BC-District-All
laptop

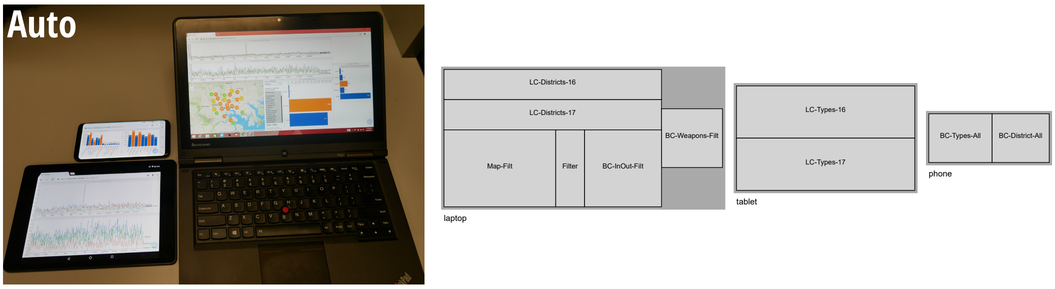

P5

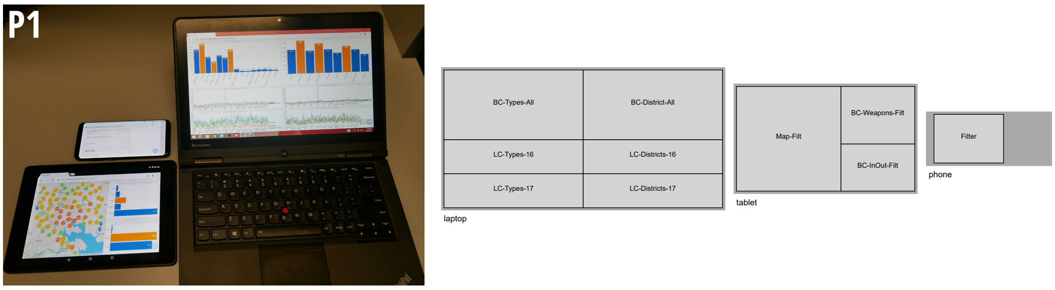

LC-Types-16
LC-Districts-16
LC-Types-17
LC-Districts-17
Filter
Map-Filt
large-display
BC-Types-All
BC-Weapons-Filt
BC-District-All
BC-InOut-Filt
laptop

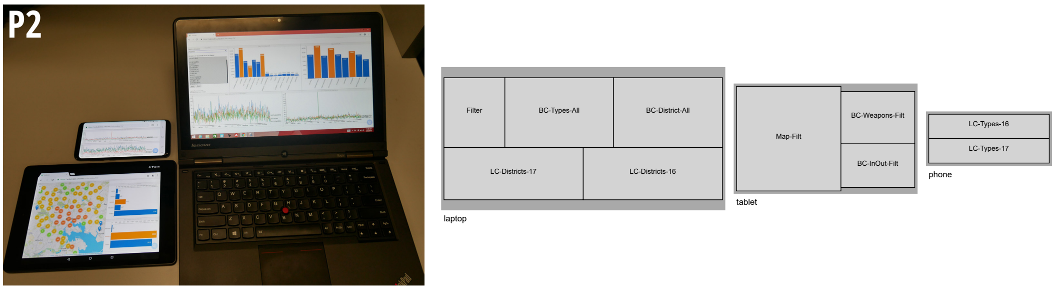

P6

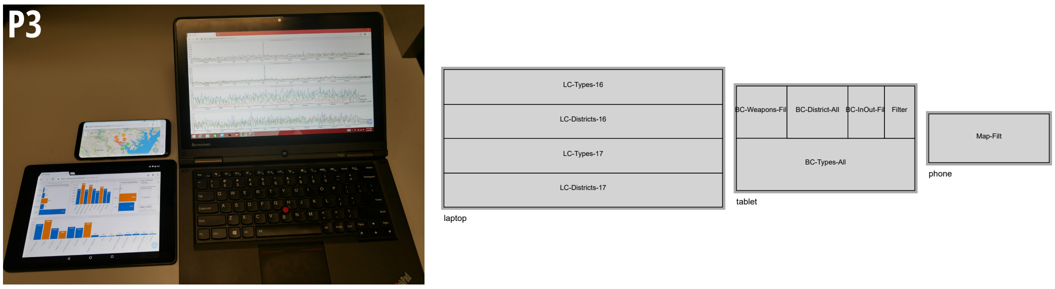

BC-District-All
LC-Districts-16
LC-Types-16
BC-Types-All
LC-Districts-17
LC-Types-17
large-display
Filter
BC-InOut-Filt
BC-Weapons-Filt
Map-Filt
laptop

Setup 3: Laptop, Tablet, and Smartphone

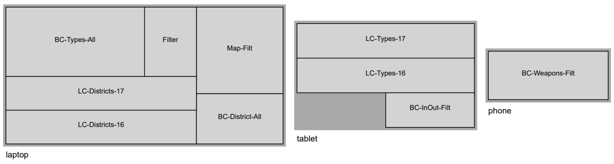




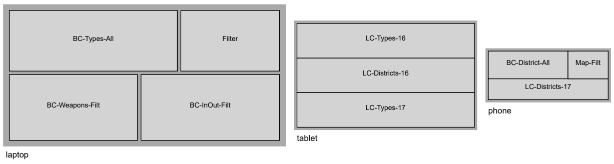




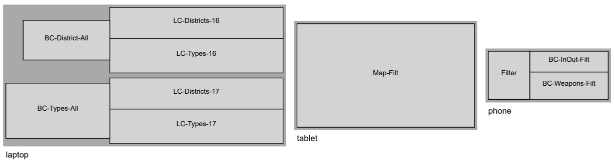

P4


BC-Types-All
Filter
Map-Filt
LC-Districts-17
LC-Districts-16
BC-District-All
laptop
LC-Types-17
LC-Types-16
BC-InOut-Filt
tablet
BC-Weapons-Filt
phone


P5


BC-Types-All
Filter
BC-Weapons-Filt
BC-InOut-Filt
laptop
LC-Types-16
LC-Districts-16
LC-Types-17
tablet
BC-District-All
Map-Filt
LC-Districts-17
phone


P6


BC-District-All
LC-Districts-16
LC-Types-16
BC-Types-All
LC-Districts-17
LC-Types-17
laptop
Map-Filt
tablet
Filter
BC-InOut-Filt
BC-Weapons-Filt
phone

B.2 Questionnaire: Demographics and Experiences

The following questionnaire was used to query the participants' demographic information as well as to assess their prior experience.

33. Gender:

☐ Male

☐ Female

☐ Prefer not to say

☐ Other: ____________

34. Age: ________

35. Theoretical knowledge of visualization:
No knowledge ☐—☐—☐—☐—☐ Expert

36. Practical experience with visual data analysis:
Never used ☐—☐—☐—☐—☐ Daily usage over multiple years

37. Years in the field: ________

38. Description of own activities linked to visualization / data analysis:

____

____

B.3 Connectivity Printout

The following printout was provided to participants showing all present views as well as the configured connections between them.

Bibliography

[AHD18] Javid Abbasov, Tom Horak, and Raimund Dachselt. "Smartwatch-based Pointing Interaction". In: *Mensch und Computer 2018 - Tagungsband*. Gesellschaft für Informatik e.V., 2018. DOI: 10.18420/MUC2018-MCI-0373 (cit. on pp. 22, 122, 126).

[Abd+18] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y. Lim, and Mohan Kankanhalli. "Trends and Trajectories for Explainable, Accountable and Intelligible Systems: An HCI Research Agenda". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, 582:1–582:18. DOI: 10.1145/3173574.3174156 (cit. on p. 190).

[AHK06] James Abello, Frank van Ham, and Neeraj Krishnan. "ASK-GraphView: A Large Scale Graph Visualization System". In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (Sept. 2006), pp. 669–676. DOI: 10.1109/tvcg.2006.120 (cit. on pp. 65, 66).

[AKK02] James Abello, Jeffrey Korn, and Matthias Kreuseler. "Navigating Giga-Graphs". In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2002, pp. 290–299. DOI: 10.1145/1556262.1556308 (cit. on pp. 64, 66).

[Ahn+15] Youngseok Ahn, Sungjae Hwang, HyunGook Yoon, Junghyeon Gim, and Jung-hee Ryu. "BandSense: Pressure-sensitive Multi-touch Interaction on a Wristband". In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2015, pp. 251–254. DOI: 10.1145/2702613.2725441 (cit. on p. 96).

[AZ03] Pär-Anders Albinsson and Shumin Zhai. "High Precision Touch Screen Interaction". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2003, pp. 105–112. DOI: 10.1145/642611.642631 (cit. on p. 19).

[AAJ20] Abeer Alsaiari, Jillian Aurisano, and Andrew E. Johnson. "Evaluating Strategies of Exploratory Visual Data Analysis in Multi Device Environments". In: *EuroVis 2020 - Short Papers*. The Eurographics Association, 2020. DOI: 10.2312/EVS.20201054 (cit. on p. 37).

[AES05] Robert Amar, James Eagan, and John Stasko. "Low-level Components of Analytic Activity in Information Visualization". In: *IEEE Symposium on Information Visualization*. Piscataway, NJ, USA: IEEE, 2005, pp. 111–117. DOI: 10.1109/infvis.2005.1532136 (cit. on pp. 26, 27, 163, 168).

[Ami+17] Fereshteh Amini, Khalad Hasan, Andrea Bunt, and Pourang Irani. "Data Representations for In-situ Exploration of Health and Fitness Data". In: *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare*. New York, NY, USA: ACM, 2017, pp. 163–172. DOI: 10.1145/3154862.3154879 (cit. on pp. 32, 104).

[AEN10] Christopher Andrews, Alex Endert, and Chris North. "Space to Think: Large High-resolution Displays for Sensemaking". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2010, pp. 55–64. DOI: 10.1145/1753326.1753336 (cit. on pp. 21, 22, 147).

[And+11] Christopher Andrews, Alex Endert, Beth Yost, and Chris North. "Information Visualization on Large, High-resolution Displays: Issues, Challenges, and Opportunities". In: *Information Visualization* 10.4 (Aug. 2011), pp. 341–355. DOI: 10.1177/1473871611415997 (cit. on pp. 21, 33).

[And18] Keith Andrews. *Responsive Visualisation*. CHI 2018 Workshop on Data Visualization on Mobile Devices (MobileVis 2018). 2018 (cit. on pp. 33, 46, 51, 53, 55).

[AS17] Keith Andrews and Aleš Smrdel. "Responsive Data Visualisation". In: *EuroVis 2017 - Posters*. Ed. by Anna Puig Puig and Tobias Isenberg. Aire-la-Ville, Switzerland: Eurographics Association, 2017, pp. 113–115. DOI: 10.2312/eurp.20171182 (cit. on pp. 6, 46, 51).

[Ang+18] Marco Angelini, Giuseppe Santucci, Heidrun Schumann, and Hans-Jörg Schulz. "A Review and Characterization of Progressive Visual Analytics". In: *Informatics* 5.3 (July 2018), p. 31. DOI: 10.3390/informatics5030031 (cit. on p. 50).

[AMC17] Axel Antoine, Sylvain Malacria, and Géry Casiez. "ForceEdge: Controlling Autoscroll on Both Desktop and Mobile Computers Using the Force". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 3281–3292. DOI: 10.1145/3025453.3025605 (cit. on p. 18).

[ATS82] M. D. Apperley, I. Tzavaras, and R. Spence. "A Bifocal Display Technique for Data Presentation". In: *Eurographics Conference Proceedings*. Ed. by D.S. Greenaway and E.A. Warman. The Eurographics Association, 1982. DOI: 10.2312/eg.19821002 (cit. on pp. 65, 82).

[APP11] Daniel Archambault, Helen Purchase, and Bruno Pinaud. "Animation, Small Multiples, and the Effect of Mental Map Preservation in Dynamic Graphs". In: *IEEE Transactions on Visualization and Computer Graphics* 17.4 (Apr. 2011), pp. 539–552. DOI: 10.1109/tvcg.2010.78 (cit. on p. 176).

[Ash+08] Daniel L. Ashbrook, James R. Clawson, Kent Lyons, Thad E. Starner, and Nirmal Patel. "Quickdraw: The Impact of Mobility and On-body Placement on Device Access Time". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2008, pp. 219–222. DOI: 10.1145/1357054.1357092 (cit. on pp. 118, 120).

[Bac+15] Myroslav Bachynskyi, Gregorio Palmas, Antti Oulasvirta, Jürgen Steimle, and Tino Weinkauf. "Performance and Ergonomics of Touch Surfaces: A Comparative Study using Biomechanical Simulation". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2015, pp. 1817–1826. DOI: 10.1145/2702123.2702607 (cit. on p. 49).

[Bad+16] Sriram Karthik Badam, Fereshteh Amini, Niklas Elmqvist, and Pourang Irani. "Supporting Visual Exploration for Multiple Users in Large Display Environments". In: *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*. Piscataway, NJ, USA: IEEE, 2016, pp. 1–10. DOI: 10.1109/vast.2016.7883506 (cit. on pp. 33, 34, 123, 134).

[BE17] Sriram Karthik Badam and Niklas Elmqvist. "Visfer: Camera-based Visual Data Transfer for Cross-device Visualization". In: *Information Visualization* (Aug. 2017). DOI: 10.1177/1473871617725907 (cit. on pp. 23, 36, 119).

[BFE15] Sriram Karthik Badam, Eli Fisher, and Niklas Elmqvist. "Munin: A Peer-to-Peer Middleware for Ubiquitous Analytics and Visualization Spaces". In: *IEEE Transactions on Visualization and Computer Graphics* 21.2 (Feb. 2015), pp. 215–228. DOI: 10.1109/tvcg.2014.2337337 (cit. on pp. 36, 138, 159).

[Bad+19] Sriram Karthik Badam, Andreas Mathisen, Roman Rädle, Clemens N. Klokmose, and Niklas Elmqvist. "Vistrates: A Component Model for Ubiquitous Analytics". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 586–596. DOI: 10.1109/TVCG.2018.2865144 (cit. on pp. 36, 159, 161, 168, 169, 193).

[BN07] Robert Ball and Chris North. "Realizing Embodied Interaction for Visual Analytics Through Large Displays". In: *Computers & Graphics* 31.3 (June 2007), pp. 380–400. DOI: 10.1016/j.cag.2007.01.029 (cit. on pp. 20, 31).

[BNB07] Robert Ball, Chris North, and Doug A. Bowman. "Move to Improve: Promoting Physical Navigation to Increase User Performance with Large Displays". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2007, pp. 191–200. DOI: 10.1145/1240624.1240656 (cit. on pp. 5, 17, 20–22, 123).

[BMG10] Till Ballendat, Nicolai Marquardt, and Saul Greenberg. "Proxemic Interaction: Designing for a Proximity and Orientation-aware Environment". In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2010, pp. 121–130. DOI: 10.1145/1936652.1936676 (cit. on pp. 20, 126).

[Ban+14] Nikola Banovic, Christina Brant, Jennifer Mankoff, and Anind Dey. "ProactiveTasks: The Short of Mobile Device Use Sessions". In: *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA: ACM, 2014, pp. 243–252. DOI: 10.1145/2628363.2628380 (cit. on p. 17).

[BE18] Andrea Batch and Niklas Elmqvist. "The Interactive Visualization Gap in Initial Exploratory Data Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018), pp. 278–287. DOI: 10.1109/tvcg.2017.2743990 (cit. on p. 34).

[BH19] Leilani Battle and Jeffrey Heer. "Characterizing Exploratory Visual Analysis: A Literature Review and Evaluation of Analytic Provenance in Tableau". In: *Computer Graphics Forum* 38.3 (June 2019), pp. 145–159. DOI: 10.1111/cgf.13678 (cit. on pp. 26, 28).

[Bau06] Thomas Baudel. "From Information Visualization to Direct Manipulation: Extending a Generic Visualization Framework for the Interactive Editing of Large Datasets". In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2006. DOI: 10.1145/1166253.1166265 (cit. on pp. 61, 66, 67, 84).

[BC09] Patrick Baudisch and Gerry Chu. "Back-of-device Interaction Allows Creating Very Small Touch Devices". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2009, pp. 1923–1932. DOI: 10.1145/1518701.1518995 (cit. on p. 96).

[BGS01] Patrick Baudisch, Nathaniel Good, and Paul Stewart. "Focus Plus Context Screens: Combining Display Technology with Visualization Techniques". In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2001, pp. 31–40. DOI: 10.1145/502348.502354 (cit. on p. 101).

[BH10] Patrick Baudisch and Christian Holz. "My new PC is a mobile phone". In: *ACM XRDS* 16.4 (June 2010), pp. 36–41. DOI: 10.1145/1764848.1764857 (cit. on p. 4).

[BR03] Patrick Baudisch and Ruth Rosenholtz. "Halo: A Technique for Visualizing Off-screen Objects". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2003, pp. 481–488. DOI: 10.1145/642611.642695 (cit. on pp. 55, 58).

[Bau+06] Patrick Baudisch, Desney S. Tan, Maxime Collomb, et al. "Phosphor: Explaining Transitions in the User Interface using Afterglow Effects". In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2006, pp. 169–178. DOI: 10.1145/1166253.1166280 (cit. on p. 176).

[BLC12] Dominikus Baur, Bongshin Lee, and Sheelagh Carpendale. "TouchWave: Kinetic Multi-touch Manipulation for Hierarchical Stacked Graphs". In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2012, pp. 255–264. DOI: 10.1145/2396636.2396675 (cit. on p. 30).

[BW17] Fabian Beck and Daniel Weiskopf. "Word-Sized Graphics for Scientific Texts". In: *IEEE Transactions on Visualization and Computer Graphics* 23.6 (June 2017), pp. 1576–1587. DOI: 10.1109/tvcg.2017.2674958 (cit. on p. 65).

[Bed00] Benjamin B. Bederson. “Fisheye Menus”. In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2000, pp. 217–225. DOI: 10.1145/354401.354782 (cit. on p. 101).

[Bed+04] Benjamin B. Bederson, Aaron Clamage, Mary P. Czerwinski, and George G. Robertson. “DateLens: A Fisheye Calendar Interface for PDAs”. In: *ACM Transactions on Computer-Human Interaction* 11.1 (Mar. 2004), pp. 90–119. DOI: 10.1145/972648.972652 (cit. on p. 65).

[Beh+16] Michael Behrisch, Benjamin Bach, Nathalie Henry Riche, Tobias Schreck, and Jean-Daniel Fekete. “Matrix Reordering Methods for Table and Network Visualization”. In: *Computer Graphics Forum* 35.3 (June 2016), pp. 693–716. DOI: 10.1111/cgf.12935 (cit. on p. 66).

[BM88] D. Benyon and D. Murray. “Experience with Adaptive Interfaces”. In: *The Computer Journal* 31.5 (Jan. 1988), pp. 465–473. DOI: 10.1093/comjnl/31.5.465 (cit. on p. 45).

[BST19] Philip Berger, Heidrun Schumann, and Christian Tominski. “Visually Exploring Relations between Structure and Attributes in Multivariate Graphs”. In: *Proceedings of the IEEE International Conference on Information Visualization*. Piscataway, NJ, USA: IEEE, 2019. DOI: 10.1109/IV.2019.00051 (cit. on pp. 64, 72).

[BI12] Anastasia Bezerianos and Petra Isenberg. “Perception of Visual Variables on Tiled Wall-Sized Displays for Information Visualization Applications”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2516–2525. DOI: 10.1109/tvcg.2012.251 (cit. on p. 33).

[BB06] Jacob T. Biehl and Brian P. Bailey. “Improving Interfaces for Managing Applications in Multiple-device Environments”. In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2006, pp. 35–42. DOI: 10.1145/1133265.1133273 (cit. on p. 24).

[Bil+02] Daniel Billsus, Clifford A. Brunk, Craig Evans, Brian Gladish, and Michael Pazzani. “Adaptive Interfaces for Ubiquitous Web Access”. In: *Communications of the ACM* 45.5 (May 2002), pp. 34–38. DOI: 10.1145/506218.506240 (cit. on p. 45).

[Bla+21] Tanja Blascheck, Frank Bentley, Eun Kyoung Choe, Tom Horak, and Petra Isenberg. “From Perception to Behavior Change: Characterizing and Evaluating Glanceable Mobile Visualizations”. In: *Mobile Data Visualization*. Ed. by Bongshin Lee, Eun Kyoung Choe, Petra Isenberg, and Raimund Dachselt. AK Peters Visualization Series. (To appear). CRC Press, 2021 (cit. on pp. 17, 32, 104, 109).

[Bla+19] Tanja Blascheck, Lonni Besancon, Anastasia Bezerianos, Bongshin Lee, and Petra Isenberg. “Glanceable Visualization: Studies of Data Comparison Performance on Smartwatches”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 630–640. DOI: 10.1109/tvcg.2018.2865142 (cit. on p. 32).

[BGV10] Florian Block, Hans Gellersen, and Nicolas Villar. "Touch-display keyboards: Transforming Keyboards into Interactive Surfaces". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2010, pp. 1145–1154. DOI: 10.1145/1753326.1753498 (cit. on p. 99).

[Bor+13] Rita Borgo, Johannes Kehrer, David H. S. Chung, et al. "Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications". In: *Eurographics 2013 - State of the Art Reports*. Ed. by M. Sbert and L. Szirmay-Kalos. Aire-la-Ville, Switzerland: Eurographics Association, 2013. DOI: 10.2312/conf/EG2013/stars/039-063 (cit. on p. 78).

[BOH11] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. "D^3: Data-Driven Documents". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (Dec. 2011), pp. 2301–2309. DOI: 10.1109/TVCG.2011.185 (cit. on pp. 85, 108, 131, 153, 161, 184).

[BRR] Nadia Boukhelifa, Jonathan C. Roberts, and Peter .J. Rodgers. "A Coordination Model for Exploratory Multiview Visualization". In: *Proceedings International Conference on Coordinated and Multiple Views in Exploratory Visualization*. Piscataway, NJ, USA: IEEE, p. 2003. DOI: 10.1109/cmv.2003.1215005 (cit. on p. 29).

[Bra+13] Lauren Bradel, Alex Endert, Kristen Koch, Christopher Andrews, and Chris North. "Large High Resolution Displays for Co-located Collaborative Sensemaking: Display Usage and Territoriality". In: *International Journal of Human-Computer Studies* 71.11 (Nov. 2013), pp. 1078–1088. DOI: 10.1016/j.ijhcs.2013.07.004 (cit. on pp. 21, 34, 119).

[Bra+11] Andrew Bragdon, Rob DeLine, Ken Hinckley, and Meredith Ringel Morris. "Code Space: Touch + Air Gesture Hybrid Interactions for Supporting Developer Meetings". In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2011, pp. 212–221. DOI: 10.1145/2076354.2076393 (cit. on p. 22).

[Bre+20] Matthew Brehmer, Bongshin Lee, Petra Isenberg, and Eun Kyoung Choe. "A Comparative Evaluation of Animation and Small Multiples for Trend Visualization on Mobile Phones". In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (Jan. 2020), pp. 364–374. DOI: 10.1109/tvcg.2019.2934397 (cit. on p. 56).

[Bre+19a] Matthew Brehmer, Bongshin Lee, Petra Isenberg, and Eun Kyoung Choe. "Visualizing Ranges over Time on Mobile Phones: A Task-Based Crowdsourced Evaluation". In: *IEEE Transactions on Visualization and Computer Graphics* 25 (1 Jan. 2019), pp. 619–629. DOI: 10.1109/tvcg.2018.2865234 (cit. on pp. 32, 38, 48).

[BM13] Matthew Brehmer and Tamara Munzner. "A Multi-Level Typology of Abstract Visualization Tasks". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2376–2385. DOI: 10.1109/tvcg.2013.124 (cit. on pp. 4, 8, 26–28, 38, 119, 123, 124, 163, 166, 168).

[Bre+19b] Nathalie Bressa, Kendra Wannamaker, Henrik Korsgaard, Wesley Willett, and Jo Vermeulen. "Sketching and Ideation Activities for Situated Visualization Design". In: *Proceedings of the ACM Designing Interactive Systems Conference*. New York, NY, USA: ACM, 2019, pp. 173–185. DOI: 10.1145/3322276.3322326 (cit. on p. 188).

[Bro+07] Barry Brown, Alex S. Taylor, Shahram Izadi, et al. "Locating Family Values: A Field Trial of the Whereabouts Clock". In: *UbiComp 2007: Ubiquitous Computing*. Springer Berlin Heidelberg, 2007, pp. 354–371. DOI: 10.1007/978-3-540-74853-3_21 (cit. on p. 104).

[Bro+14] Eli T Brown, Alvitta Ottley, Helen Zhao, et al. "Finding Waldo: Learning about Users from their Interactions". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 1663–1672. DOI: 10.1109/tvcg.2014.2346575 (cit. on pp. 189, 190).

[Bru+18] Frederik Brudy, Joshua Kevin Budiman, Steven Houben, and Nicolai Marquardt. "Investigating the Role of an Overview Device in Multi-Device Collaboration". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, 300:1–300:13. DOI: 10.1145/3173574.3173874 (cit. on p. 23).

[Bru+19] Frederik Brudy, Christian Holz, Roman Rädle, et al. "Cross-Device Taxonomy: Survey, Opportunities and Challenges of Interactions Spanning Across Multiple Devices". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, 562:1–562:28. DOI: 10.1145/3290605.3300792 (cit. on pp. 5, 9, 20, 21, 24, 25, 36, 152).

[Bru+16] Frederik Brudy, Steven Houben, Nicolai Marquardt, and Yvonne Rogers. "CurationSpace: Cross-Device Content Curation Using Instrumental Interaction". In: *Proceedings of the ACM Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2016, pp. 159–168. DOI: 10.1145/2992154.2992175 (cit. on pp. 22, 116, 121).

[Bru+14] Frederik Brudy, David Ledo, Saul Greenberg, and Andreas Butz. "Is Anyone Looking? Mitigating Shoulder Surfing on Public Displays Through Awareness and Protection". In: *Proceedings of The International Symposium on Pervasive Displays*. New York, NY, USA: ACM, 2014, 1:1–1:6. DOI: 10.1145/2611009.2611028 (cit. on p. 34).

[Bru+20] Frederik Brudy, David Ledo, Michel Pahud, et al. "SurfaceFleet: Exploring Distributed Interactions Unbounded from Device, Application, User, and Time". In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2020, pp. 7–21. DOI: 10.1145/3379337.3415874 (cit. on p. 193).

[BR05] Thorsten Büring and Harald Reiterer. "ZuiScat: Querying and Visualizing Information Spaces on Personal Digital Assistants". In: *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA: ACM, 2005, pp. 129–136. DOI: 10.1145/1085777.1085799 (cit. on p. 31).

[BSV15] Jesse Burstyn, Paul Strohmeier, and Roel Vertegaal. "DisplaySkin: Exploring Pose-Aware Displays on a Flexible Electrophoretic Wristband". In: *Proceedings of the International Conference on Tangible, Embedded, and Embodied Interaction*. New York, NY, USA: ACM, 2015, pp. 165–172. DOI: 10.1145/2677199.2680596 (cit. on pp. 96, 98, 104, 114).

[Büs+17] Wolfgang Büschel, Patrick Reipschläger, Ricardo Langner, and Raimund Dachselt. "Investigating the Use of Spatial Interaction for 3D Data Visualization on Mobile Devices". In: *Proceedings of the ACM Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2017, pp. 62–71. DOI: 10.1145/3132272.3134125 (cit. on p. 31).

[Bus45] Vannevar Bush. "As We May Think". In: *The Atlantic Monthly* 176.1 (1945), pp. 101–108 (cit. on p. 20).

[Car+19] Steven P Cardinali, Katherine E Tong, Trevor J Ness, and William C Lukens. *Indicators for Wearable Electronic Devices*. US Patent App. 16/151,199. 2019 (cit. on p. 115).

[Cha83] John Chambers. *Graphical Methods for Data Analysis*. Belmont, Calif. Boston: Wadsworth International Group Duxbury Press, 1983 (cit. on p. 69).

[CBF14] Olivier Chapuis, Anastasia Bezerianos, and Stelios Frantzeskakis. "Smarties: An Input System for Wall Display Development". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 2763–2772. DOI: 10.1145/2556288.2556956 (cit. on p. 22).

[Che+21] Xi Chen, Wei Zeng, Yanna Lin, et al. "Composition and Configuration Patterns in Multiple-View Visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pp. 1514–1524. DOI: 10.1109/tvcg.2020.3030338 (cit. on pp. 28, 38, 160, 162).

[Che+14] Xiang Chen, Tovi Grossman, Daniel J. Wigdor, and George Fitzmaurice. "Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 159–168. DOI: 10.1145/2556288.2556955 (cit. on pp. 23, 116).

[Che17] Yang Chen. "Visualizing Large Time-series Data on Very Small Screens". In: *Short Paper Proceedings of the IEEE VGTC/Eurographics Conference on Visualization*. Aire-la-Ville, Switzerland: Eurographics Association, 2017, pp. 37–41. DOI: 10.2312/eurovisshort.20171130 (cit. on pp. 32, 38).

[Chi11] Luca Chittaro. "Designing Visual User Interfaces for Mobile Applications". In: *Proceedings of the ACM Symposium on Engineering Interactive Computing Systems*. New York, NY, USA: ACM, 2011, pp. 331–332. DOI: 10.1145/1996461.1996550 (cit. on p. 46).

[Chi06] Luca Chittaro. "Visualizing Information on Mobile Devices". In: *Computer* 39.3 (Mar. 2006), pp. 40–45. DOI: 10.1109/mc.2006.109 (cit. on pp. 31, 46, 55).

[Cho+19] Eun Kyoung Choe, Raimund Dachselt, Petra Isenberg, and Bongshin Lee. “Mobile Data Visualization (Dagstuhl Seminar 19292)”. en. In: *Dagstuhl Reports* 9.7 (2019), pp. 78–93. DOI: 10.4230/DAGREP.9.7.78 (cit. on pp. 44, 51).

[Chu+15] Haeyong Chung, Chris North, Sarang Joshi, and Jian Chen. “Four Considerations for Supporting Visual Analysis in Display Ecologies”. In: *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*. Piscataway, NJ, USA: IEEE, 2015, pp. 33–40. DOI: 10.1109/vast.2015.7347628 (cit. on pp. 5, 37, 39, 166).

[Chu+14] Haeyong Chung, Chris North, Jessica Zeitz Self, Sharon Lynn Chu, and Francis K. H. Quek. “VisPorter: Facilitating Information Sharing for Collaborative Sensemaking on Multiple Displays”. In: *Personal and Ubiquitous Computing* 18.5 (June 2014), pp. 1169–1186. DOI: 10.1007/s00779-013-0727-2 (cit. on pp. 23, 24, 35, 138).

[CKB09] Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. “A Review of Overview+Detail, Zooming, and Focus+Context Interfaces”. In: *ACM Computing Surveys* 41.1 (Jan. 2009), pp. 1–31. DOI: 10.1145/1456650.1456652 (cit. on pp. 29, 32, 67).

[CH18] Matt Conlen and Jeffrey Heer. “Idyll: A Markup Language for Authoring and Publishing Interactive Articles on the Web”. In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2018, pp. 977–989. DOI: 10.1145/3242587.3242600 (cit. on p. 161).

[DSo+17] Trevor D’Souza, Padmalata V. Nistala, Swapna Bijayinee, et al. “Patterns for Interactive Line Charts on Mobile Devices”. In: *Proceedings of the 22nd European Conference on Pattern Languages of Programs*. New York, NY, USA: ACM, 2017, 1:1–1:13. DOI: 10.1145/3147704.3147727 (cit. on p. 56).

[DB09] Raimund Dachselt and Robert Buchholz. “Natural Throw and Tilt Interaction between Mobile Phones and Distant Displays”. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2009, pp. 3253–3258. DOI: 10.1145/1520340.1520467 (cit. on p. 19).

[DP08] David Dearman and Jeffery S. Pierce. “It’s on My Other Computer!: Computing with Multiple Devices”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2008, pp. 767–776. DOI: 10.1145/1357054.1357177 (cit. on p. 4).

[Dru+13] Steven M. Drucker, Danyel Fisher, Ramik Sadana, Jessica Herron, and m.c. schraefel m.c. “TouchViz: A Case Study Comparing Two Interfaces for Data Analytics on Tablets”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2013, pp. 2301–2310. DOI: 10.1145/2470654.2481318 (cit. on pp. 30, 38, 56).

[Ear+18a] Rachel Eardley, Anne Roudaut, Steve Gill, and Stephen J. Thompson. "Designing for Multiple Hand Grips and Body Postures within the UX of a moving Smartphone". In: *Proceedings of the ACM Designing Interactive Systems Conference*. New York, NY, USA: ACM, 2018, pp. 611–621. DOI: 10.1145/3196709.3196711 (cit. on p. 49).

[Ear+18b] Rachel Eardley, Anne Roudaut, Steve Gill, and Stephen J. Thompson. "Investigating How Smartphone Movement is Affected by Body Posture". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, 202:1–202:8. DOI: 10.1145/3173574.3173776 (cit. on p. 49).

[Ear+17] Rachel Eardley, Anne Roudaut, Steve Gill, and Stephen J. Thompson. "Understanding Grip Shifts: How Form Factors Impact Hand Movements on Mobile Phones". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 4680–4691. DOI: 10.1145/3025453.3025835 (cit. on p. 49).

[Eic+20] Philipp Eichmann, Darren Edge, Nathan Evans, et al. "Orchard: Exploring Multivariate Heterogeneous Networks on Mobile Phones". In: *Computer Graphics Forum* 39.3 (June 2020), pp. 115–126. DOI: 10.1111/cgf.13967 (cit. on p. 55).

[Eic+16] Christian Eichner, Stefan Gladisch, Heidrun Schumann, and Christian Tominski. "Direct Visual Editing of Node Attributes in Graphs". In: *Informatics* 3.4 (Oct. 2016). DOI: 10.3390/informatics3040017 (cit. on p. 66).

[Elm+08a] Niklas Elmqvist, Thanh-Nghi Do, Howard Goodell, Nathalie Henry, and Jean-Daniel Fekete. "ZAME: Interactive Large-Scale Graph Visualization". In: *Proceedings of the IEEE Pacific Symposium on Visualization*. IEEE, Mar. 2008. DOI: 10.1109/pacificvis.2008.4475479 (cit. on pp. 64–66).

[Elm+08b] Niklas Elmqvist, Nathalie Henry, Yann Ri he, and Jean-Daniel Fekete. "Mélange: Space Folding for Multi-Focus Interaction". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2008, pp. 1333–1342. DOI: 10.1145/1357054.1357263 (cit. on p. 65).

[EI13] Niklas Elmqvist and Pourang Irani. "Ubiquitous Analytics: Interacting with Big Data Anywhere, Anytime." In: *IEEE Computer* 46.4 (Apr. 2013), pp. 86–89. DOI: 10.1109/mc.2013.147 (cit. on p. 30).

[Elm+11] Niklas Elmqvist, Andrew Vande Moere, Hans-Christian Jetter, et al. "Fluid Interaction for Information Visualization". In: *Information Visualization* 10.4 (Aug. 2011), pp. 327–340. DOI: 10.1177/1473871611413180 (cit. on p. 30).

[EW14] Stef van den Elzen and Jarke J. van Wijk. "Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 2310–2319. DOI: 10.1109/tvcg.2014.2346441 (cit. on p. 63).

[EFK95] J. L. Encarnação, M. Frühauf, and T. Kirste. "Mobile Visualization: Challenges and Solution Concepts". In: *Computer Applications in Production Engineering*. Springer US, 1995, pp. 725–737. DOI: 10.1007/978-0-387-34879-7_75 (cit. on p. 46).

[End+17] A. Endert, W. Ribarsky, C. Turkay, et al. "The State of the Art in Integrating Machine Learning into Visual Analytics". In: *Computer Graphics Forum* 36.8 (Mar. 2017), pp. 458–486. DOI: 10.1111/cgf.13092 (cit. on pp. 189, 190).

[End+11] Alex Endert, Christopher Andrews, Yueh Hua Lee, and Chris North. "Visual Encodings that Support Physical Navigation on Large Displays". In: *Proceedings of Graphics Interface*. Canadian Human-Computer Communications Society. 2011, pp. 103–110 (cit. on pp. 31, 33).

[Est+15] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. "Orbits: Gaze Interaction for Smart Watches Using Smooth Pursuit Eye Movements". In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2015, pp. 457–466. DOI: 10.1145/2807442.2807499 (cit. on p. 96).

[Fek+19] Jean-Daniel Fekete, Danyel Fisher, Arnab Nandi, and Michael Sedlmair. "Progressive Data Analysis and Visualization (Dagstuhl Seminar 18411)". en. In: *Dagstuhl Reports* 8.10 (2019), pp. 1–40. DOI: 10.4230/DAGREP.8.10.1 (cit. on p. 50).

[FBW17] Andreas Rene Fender, Hrvoje Benko, and Andy Wilson. "MeetAlive: Room-Scale Omni-Directional Display System for Multi-User Content and Control Sharing". In: *Proceedings of the ACM Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2017, pp. 106–115. DOI: 10.1145/3132272.3134117 (cit. on pp. 22, 23).

[Fer+14] Denzil Ferreira, Jorge Goncalves, Vassilis Kostakos, Louise Barkhuus, and Anind K. Dey. "Contextual Experience Sampling of Mobile Application Micro-usage". In: *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA: ACM, 2014, pp. 91–100. DOI: 10.1145/2628363.2628367 (cit. on p. 17).

[FM08] Leah Findlater and Joanna McGrenere. "Impact of Screen Size on Performance, Awareness, and User Satisfaction with Adaptive Graphical User Interfaces". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2008, pp. 1247–1256. DOI: 10.1145/1357054.1357249 (cit. on p. 45).

[Fit+03] George W. Fitzmaurice, Azam Khan, William Buxton, Gordon Kurtenback, and Ravin Balakrishnan. "Sentient Data Access via a Diverse Society of Devices". In: *Queue* 1.8 (Nov. 2003), pp. 52–62. DOI: 10.1145/966712.966721 (cit. on p. 5).

[For+07] Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. "Direct-touch vs. Mouse Input for Tabletop Displays". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2007, pp. 647–656. DOI: 10.1145/1240624.1240726 (cit. on p. 19).

[Fri12] Mathias Frisch. *Visualization and Interaction Techniques for Node-link Diagram Editing and Exploration*. München: Verlag Dr. Hut, 2012 (cit. on pp. 30, 186).

[FHD09] Mathias Frisch, Jens Heydekorn, and Raimund Dachselt. “Investigating Multi-touch and Pen Gestures for Diagram Editing on Interactive Surfaces”. In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2009, pp. 149–156. DOI: 10.1145/1731903.1731933 (cit. on pp. 31, 89).

[FSS18] Alexandra Fuchs, Miriam Sturdee, and Johannes Schöning. “Foldwatch: Using Origami-inspired Paper Prototypes to Explore the Extension of Output Space in Smartwatches”. In: *Proceedings of the ACM Nordic Conference on Human-Computer Interaction*. New York, NY, USA: ACM, 2018, pp. 47–59. DOI: 10.1145/3240167.3240173 (cit. on p. 95).

[Fuc11] Georg Fuchs. “Task-based Adaptation of Graphical Content in Smart Visual Interfaces”. PhD thesis. University of Rostock, 2011. DOI: 10.18453/ROSDOK_ID00000937 (cit. on p. 46).

[Fuc+17] Johannes Fuchs, Petra Isenberg, Anastasia Bezerianos, and Daniel Keim. “A Systematic Review of Experimental Studies on Data Glyphs”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.7 (July 2017), pp. 1863–1879. DOI: 10.1109/tvcg.2016.2549018 (cit. on p. 65).

[Fun+14] Markus Funk, Alireza Sahami, Niels Henze, and Albrecht Schmidt. “Using a Touch-sensitive Wristband for Text Entry on Smart Watches”. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 2305–2310. DOI: 10.1145/2559206.2581143 (cit. on p. 96).

[Fur86] G. W. Furnas. “Generalized Fisheye Views”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 1986, pp. 16–23. DOI: 10.1145/22339.22342 (cit. on pp. 65, 101).

[GJ13] Peter S. Games and Alark Joshi. “Visualization of Off-screen Data on Tablets using Context-providing Bar Graphs and Scatter Plots”. In: *Visualization and Data Analysis 2014*. Ed. by Pak Chung Wong, David L. Kao, Ming C. Hao, and Chaomei Chen. SPIE, 2013, pp. 132–146. DOI: 10.1117/12.2038456 (cit. on pp. 55, 58).

[GJ81] M. R. Garey and D. S. Johnson. “Approximation Algorithms for Bin Packing Problems: A Survey”. In: *Analysis and Design of Algorithms in Combinatorial Optimization*. Vienna: Springer Vienna, 1981, pp. 147–172. DOI: 10.1007/978-3-7091-2748-3_8 (cit. on p. 171).

[GFC05] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. “On the Readability of Graphs Using Node-Link and Matrix-Based Representations: A Controlled Experiment and Statistical Analysis”. In: *Information Visualization* 4.2 (May 2005), pp. 114–135. DOI: 10.1057/palgrave.ivs.9500092 (cit. on pp. 64, 81).

[Gla+14] S. Gladisch, H. Schumann, M. Ernst, G. Füllen, and C. Tominski. “Semi-Automatic Editing of Graphs with Customized Layouts”. In: *Computer Graphics Forum* 33.3 (June 2014), pp. 381–390. DOI: 10.1111/cgf.12394 (cit. on pp. 66, 84).

[Gla+15a] Stefan Gladisch, Ulrike Kister, Christian Tominski, Raimund Dachselt, and Heidrun Schumann. *Mapping Tasks to Interactions for Graph Exploration and Graph Editing on Interactive Surfaces*. Apr. 2015 (cit. on pp. 66, 84).

[Gla+15b] Stefan Gladisch, Heidrun Schumann, M Luboschik, and Christian Tominski. *Toward using Matrix Visualizations for Graph Editing*. Poster Program of the 2015 IEEE Conference on Information Visualization (InfoVis). 2015 (cit. on pp. 66, 84, 150).

[Gle+11] Michael Gleicher, Danielle Albers, Rick Walker, et al. “Visual Comparison for Information Visualization”. In: *Information Visualization* 10.4 (Sept. 2011), pp. 289–309. DOI: 10.1177/1473871611416549 (cit. on pp. 27, 71, 78, 148, 162).

[GMG18] Alix Goguey, Sylvain Malacria, and Carl Gutwin. “Improving Discoverability and Expert Performance in Force-Sensitive Text Selection for Touch Devices with Mode Gauges”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, 477:1–477:12. DOI: 10.1145/3173574.3174051 (cit. on pp. 18, 19).

[GTV15] Antonio Gomes, Tristan Trutna, and Roel Vertegaal. “DisplayCover: A Tablet Keyboard with an Embedded Thin-Film Touchscreen Display”. In: *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA: ACM, 2015, pp. 531–535. DOI: 10.1145/2785830.2785843 (cit. on p. 99).

[Gon+18a] Jun Gong, Zheer Xu, Qifan Guo, et al. “WrisText: One-handed Text Entry on Smartwatch Using Wrist Gestures”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, 181:1–181:14. DOI: 10.1145/3173574.3173755 (cit. on p. 96).

[Gon+18b] Jun Gong, Xin Yang, Teddy Seyed, Josh Urban Davis, and Xing-Dong Yang. “Indutivo: Contact-Based, Object-Driven Interactions with Inductive Sensing”. In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2018, pp. 321–333. DOI: 10.1145/3242587.3242662 (cit. on p. 96).

[Gou+16] Rúben Gouveia, Fábio Pereira, Evangelos Karapanos, Sean A. Munson, and Marc Hassenzahl. “Exploring the Design Space of Glanceable Feedback for Physical Activity Trackers”. In: *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. New York, NY, USA: ACM, 2016, pp. 144–155. DOI: 10.1145/2971648.2971754 (cit. on pp. 32, 104).

[Gre+11] Saul Greenberg, Nicolai Marquardt, Till Ballendat, Rob Diaz-Marino, and Miaosen Wang. “Proxemic Interactions: The New Ubicomp?” In: *Interactions* 18.1 (Jan. 2011), pp. 42–50. DOI: 10.1145/1897239.1897250 (cit. on p. 20).

[Grø+20] Jens Emil Grønbæk, Mille Skovhus Knudsen, Kenton O'Hara, et al. "Proxemics Beyond Proximity: Designing for Flexible Social Interaction Through Cross-Device Interaction". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, 252:1–252:14. DOI: 10.1145/3313831.3376379 (cit. on pp. 25, 152).

[Gru+15] Jens Grubert, Matthias Heinisch, Aaron Quigley, and Dieter Schmalstieg. "MultiFi: Multi Fidelity Interaction with Displays On and Around the Body". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2015, pp. 3933–3942. DOI: 10.1145/2702123.2702331 (cit. on p. 96).

[GF04] Carl Gutwin and Chris Fedak. "Interacting with Big Interfaces on Small Screens: A Comparison of Fisheye, Zoom, and Panning Techniques". In: *Proceedings of Graphics Interface 2004*. Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society, 2004, pp. 145–152 (cit. on p. 31).

[HNC14] Jonathan Haber, Miguel A. Nacenta, and Sheelagh Carpendale. "Paper vs. Tablets: The Effect of Document Media in Co-located Collaborative Work". In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2014, pp. 89–96. DOI: 10.1145/2598153.2598170 (cit. on p. 144).

[HSS11] S. Hadlak, Hans-Jorg Schulz, and H. Schumann. "In Situ Exploration of Large Dynamic Networks". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (Dec. 2011), pp. 2334–2343. DOI: 10.1109/tvcg.2011.213 (cit. on p. 65).

[HSS15] Steffen Hadlak, Heidrun Schumann, and Hans-Jörg Schulz. "A Survey of Multi-faceted Graph Visualization". In: *Eurographics Conference on Visualization (EuroVis) - STARs*. Aire-la-Ville, Switzerland: Eurographics Association, 2015. DOI: 10.2312/eurovisstar.20151109 (cit. on p. 81).

[Hal66] Edward T. Hall. *The Hidden Dimension*. Garden City, NY: Anchor Books, 1966. DOI: 10.2307/1572461 (cit. on p. 20).

[HW14] Peter Hamilton and Daniel J. Wigdor. "Conductor: Enabling and Understanding Cross-device Interaction". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 2773–2782. DOI: 10.1145/2556288.2557170 (cit. on pp. 23, 24, 163).

[HZ07] Jie Hao and Kang Zhang. "A Mobile Interface for Hierarchical Information Visualization and Navigation". In: *IEEE International Symposium on Consumer Electronics*. Piscataway, NJ, USA: IEEE, June 2007. DOI: 10.1109/isce.2007.4382214 (cit. on p. 31).

[HBW11] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. "OmniTouch: Wearable Multitouch Interaction Everywhere". In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2011, pp. 441–450. DOI: 10.1145/2047196.2047255 (cit. on p. 96).

[HA06] Jeffrey Heer and Maneesh Agrawala. "Multi-Scale Banking to 45 Degrees". In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (Sept. 2006), pp. 701–708. DOI: 10.1109/tvcg.2006.163 (cit. on pp. 51, 167, 177).

[HKA09] Jeffrey Heer, Nicholas Kong, and Maneesh Agrawala. "Sizing the Horizon: The Effects of Chart Size and Layering on the Graphical Perception of Time Series Visualizations". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2009, pp. 1303–1312. DOI: 10.1145/1518701.1518897 (cit. on p. 56).

[HR07] Jeffrey Heer and George Robertson. "Animated Transitions in Statistical Data Graphics". In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (Nov. 2007), pp. 1240–1247. DOI: 10.1109/tvcg.2007.70539 (cit. on p. 176).

[HS12] Jeffrey Heer and Ben Shneiderman. "Interactive Dynamics for Visual Analysis: A Taxonomy of Tools that Support the Fluent and Flexible use of Visualizations". In: *Queue* 10.2 (Feb. 2012). DOI: 10.1145/2133416.2146416 (cit. on pp. 4, 8, 26, 27, 29, 38).

[HFM07] Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. "NodeTrix: a Hybrid Visualization of Social Networks". In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1302–1309. DOI: 10.1109/tvcg.2007.70582 (cit. on p. 64).

[Hin03] Ken Hinckley. "Synchronous Gestures for Multiple Persons and Computers". In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2003, pp. 149–158. DOI: 10.1145/964696.964713 (cit. on pp. 24, 25, 152).

[Hin+16] Ken Hinckley, Seongkook Heo, Michel Pahud, et al. "Pre-Touch Sensing for Mobile Interaction". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2016, pp. 2869–2881. DOI: 10.1145/2858036.2858095 (cit. on p. 19).

[Hin15] Bill Hinderman. *Building Responsive Data Visualization for the Web*. Wiley, 2015. 448 pp. (cit. on pp. 46, 57).

[HLL20] Jane Hoffswell, Wilmot Li, and Zhicheng Liu. "Techniques for Flexible Responsive Visualization Design". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, pp. 1–13. DOI: 10.1145/3313831.3376777 (cit. on pp. 6, 33, 45, 46, 51, 53–55, 59).

[Hol+13] David Holman, Jesse Burstyn, Ryan Brotman, Audrey Younkin, and Roel Vertegaal. "Flexkit: A Rapid Prototyping Platform for Flexible Displays". In: *Adjunct Publication of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2013, pp. 17–18. DOI: 10.1145/2508468.2514934 (cit. on pp. 48, 108, 192).

[HB13] Christian Holz and Patrick Baudisch. “Fiberio: A Touchscreen That Senses Fingerprints”. In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2013, pp. 41–50. DOI: 10.1145/2501988.2502021 (cit. on p. 139).

[HB10] Christian Holz and Patrick Baudisch. “The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2010, pp. 581–590. DOI: 10.1145/1753326.1753413 (cit. on p. 19).

[Hor+21a] Tom Horak, Wolfgang Aigner, Matthew Brehmer, Alark Joshi, and Christian Tominski. “Responsive Visualization Design for Mobile Devices”. In: *Mobile Data Visualization*. Ed. by Bongshin Lee, Eun Kyoung Choe, Petra Isenberg, and Raimund Dachselt. AK Peters Visualization Series. (To appear). CRC Press, 2021 (cit. on pp. 12, 44).

[Hor+18a] Tom Horak, Sriram Karthik Badam, Niklas Elmqvist, and Raimund Dachselt. “Demonstrating David Meets Goliath: Combining Smartwatches with a Large Vertical Display for Visual Data Exploration”. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, D414:1–D414:4. DOI: 10.1145/3170427.3186497 (cit. on p. 129).

[Hor+18b] Tom Horak, Sriram Karthik Badam, Niklas Elmqvist, and Raimund Dachselt. “When David Meets Goliath: Combining Smartwatches with a Large Vertical Display for Visual Data Exploration”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, 19:1–19:13. DOI: 10.1145/3173574.3173593 (cit. on pp. 12, 22, 25, 35, 89, 118, 164).

[Hor+21b] Tom Horak, Philip Berger, Heidrun Schumann, Raimund Dachselt, and Christian Tominski. “Responsive Matrix Cells: A Focus+Context Approach for Exploring and Editing Multivariate Graphs”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pp. 1644–1654. DOI: 10.1109/TVCG.2020.3030371 (cit. on pp. 12, 62, 85).

[HD18] Tom Horak and Raimund Dachselt. “Hierarchical Graphs on Mobile Devices: A Lane-based Approach”. In: *Proceedings of the CHI 2018 Workshop on Data Visualization on Mobile Devices*. MobileVis ’18. Montreal, QC, Canada, 2018 (cit. on p. 68).

[HKD17] Tom Horak, Ulrike Kister, and Raimund Dachselt. “Improving Value Driver Trees to Enhance Business Data Analysis”. In: *Poster Program of the 2017 IEEE Conference on Information Visualization (InfoVis)*. Phoenix, Arizona, USA, Oct. 2017 (cit. on pp. 70, 85).

[HKD16] Tom Horak, Ulrike Kister, and Raimund Dachselt. “Presenting Business Data: Challenges During Board Meetings in Multi-Display Environments”. In: *Proceedings of the ACM Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2016, pp. 319–324. DOI: 10.1145/2992154.2996774 (cit. on p. 23).

[HLD20] Tom Horak, Ricardo Langner, and Raimund Dachselt. "Towards Visualizing and Exploring Multivariate Networks on Mobile Devices". In: *Companion Proceedings of the 2020 Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2020, pp. 5–8. DOI: 10.1145/3380867.3426201 (cit. on p. 62).

[Hor+19] Tom Horak, Andreas Mathisen, Clemens N. Klokmose, Raimund Dachselt, and Niklas Elmqvist. "Vistribute: Distributing Interactive Visualizations in Dynamic Multi-Device Setups". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, 616:1–616:13. DOI: 10.1145/3290605.3300846 (cit. on pp. 13, 37, 158).

[Hor+16] Tom Horak, Ulrich von Zadow, Matthias Kalms, and Raimund Dachselt. "Discussing the State of the Art for "in the wild" Mobile Device Localization". In: *ISS 2016 Workshop on Interacting with Multi-Device Ecologies "'in the wild"'*. Niagara Falls, ON, Canada, Nov. 2016 (cit. on pp. 25, 152).

[HF01] Kasper Hornbæk and Erik Frøkjær. "Reading of Electronic Documents: The Usability of Linear, Fisheye, and Overview+Detail Interfaces". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2001, pp. 293–300. DOI: 10.1145/365024.365118 (cit. on pp. 29, 101).

[HM15] Steven Houben and Nicolai Marquardt. "WATCHCONNECT: A Toolkit for Prototyping Smartwatch-Centric Cross-Device Applications". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2015, pp. 1247–1256. DOI: 10.1145/2702123.2702215 (cit. on pp. 159, 187).

[Hus+18] Maria Husmann, Alfonso Murolo, Nicolas Kick, Linda Di Geronimo, and Moira C. Norrie. "Supporting Out of Office Software Development Using Personal Devices". In: *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA: ACM, 2018, 27:1–27:11. DOI: 10.1145/3229434.3229454 (cit. on pp. 4, 22, 23, 159).

[HHN85] Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. "Direct Manipulation Interfaces". In: *Human–Computer Interaction* 1.4 (Dec. 1985), pp. 311–338. DOI: 10.1207/s15327051hci0104_2 (cit. on pp. 18, 30).

[ILL07] Maria Indrawan, Sea Ling, and Seng Loke. "Device Ecology: A Micro Digital Ecosystem". In: *2007 Inaugural IEEE-IES Digital EcoSystems and Technologies Conference*. IEEE, 2007, pp. 192–197. DOI: 10.1109/dest.2007.371969 (cit. on p. 5).

[Ise+13] Petra Isenberg, Pierre Dragicevic, Wesley Willett, Anastasia Bezerianos, and Jean-Daniel Fekete. "Hybrid-Image Visualization for Large Viewing Environments". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2346–2355. DOI: 10.1109/tvcg.2013.163 (cit. on pp. 31, 33).

[Ise+11] Petra Isenberg, Niklas Elmqvist, Jean Scholtz, et al. “Collaborative Visualization: Definition, Challenges, and Research Agenda”. In: *Information Visualization* 10.4 (July 2011), pp. 310–326. DOI: 10.1177/1473871611412817 (cit. on pp. 34, 186).

[ITC08] Petra Isenberg, Anthony Tang, and Sheelagh Carpendale. “An Exploratory Study of Visual Information Analysis”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2008, pp. 1217–1226. DOI: 10.1145/1357054.1357245 (cit. on p. 144).

[Isl+20] Alaul Islam, Anastasia Bezerianos, Bongshin Lee, Tanja Blascheck, and Petra Isenberg. “Visualizing Information on Watch Faces: A Survey with Smartwatch Users”. In: *Short Papers of the IEEE Conference on Visualization (VIS)*. Oct. 2020. DOI: 10.1109/VIS47514.2020.00038 (cit. on pp. 32, 104, 110).

[Jac+08] Robert J. K. Jacob, Audrey Girouard, Leanne M. Hirshfield, et al. “Reality-based Interaction: A Framework for Post-WIMP Interfaces”. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2008, pp. 201–210. DOI: 10.1145/1357054.1357089 (cit. on p. 17).

[Jak+13] Mikkel R. Jakobsen, Yonas Sahlemariam Haile, Soren Knudsen, and Kasper Hornbaek. “Information Visualization and Proxemics: Design Opportunities and Empirical Findings”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2386–2395. DOI: 10.1109/tvcg.2013.166 (cit. on p. 31).

[JH13] Mikkel R. Jakobsen and Kasper Hornbæk. “Interactive Visualizations on Large and Small Displays: The Interrelation of Display Size, Information Space, and Scale”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2336–2345. DOI: 10.1109/tvcg.2013.170 (cit. on pp. 20, 167).

[JH15] Mikkel R. Jakobsen and Kasper Hornbæk. “Is Moving Improving?: Some Effects of Locomotion in Wall-Display Interaction”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2015, pp. 4169–4178. DOI: 10.1145/2702123.2702312 (cit. on p. 20).

[JH14] Mikkel R. Jakobsen and Kasper Hornbæk. “Up Close and Personal: Collaborative Work on a High-resolution Multitouch Wall Display”. In: *ACM Transactions on Computer-Human Interaction* 21.2 (2014), pp. 1–34. DOI: 10.1145/2576099 (cit. on pp. 34, 123).

[Jan+14] T. J. Jankun-Kelly, Tim Dwyer, Danny Holten, et al. “Scalability Considerations for Multivariate Graph Visualization”. In: *Multivariate Network Visualization*. Cham, Switzerland: Springer International Publishing, 2014, pp. 207–235. DOI: 10.1007/978-3-319-06793-3_10 (cit. on p. 68).

[Jan14] Yvonne Jansen. “Physical and Tangible Information Visualization”. PhD thesis. Université Paris Sud, 2014 (cit. on p. 31).

[JE12] Waqas Javed and Niklas Elmqvist. “Exploring the Design Space of Composite Visualization”. In: *Proceedings of the IEEE Pacific Symposium on Visualization*. Piscataway, NJ, USA: IEEE, 2012, pp. 1–8. DOI: 10.1109/pacificvis.2012.6183556 (cit. on pp. 28, 29, 65, 71, 162).

[Jeh14] Scott Jehl. *Responsible Responsive Design*. New York: A Book Apart, 2014 (cit. on p. 46).

[Jen+19] Walther Jensen, Ashley Colley, Jonna Häkkilä, Carlos Pinheiro, and Markus Löchtefeld. “TransPrint: A Method for Fabricating Flexible Transparent Free-Form Displays”. In: *Advances in Human-Computer Interaction* (May 2019), pp. 1–14. DOI: 10.1155/2019/1340182 (cit. on p. 98).

[Jet+20] Hans-Christian Jetter, Roman Rädle, Tiare Feuchtner, et al. “"In VR, everything is possible!": Sketching and Simulating Spatially-Aware Interactive Spaces in Virtual Reality”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, 523:1–523:16. DOI: 10.1145/3313831.3376652 (cit. on p. 188).

[Jia+19] Yue Jiang, Ruofei Du, Christof Lutteroth, and Wolfgang Stuerzlinger. “ORC Layout: Adaptive GUI Layout with OR-Constraints”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, 413:1–413:12. DOI: 10.1145/3290605.3300643 (cit. on p. 57).

[JHH15] Haojian Jin, Christian Holz, and Kasper Hornbæk. “Tracko: Ad-hoc Mobile 3D Tracking Using Bluetooth Low Energy and Inaudible Signals for Cross-Device Interaction”. In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2015, pp. 147–156. DOI: 10.1145/2807442.2807475 (cit. on p. 192).

[Jo+17] Jaemin Jo, Sehi L'Yi, Bongshin Lee, and Jinwook Seo. “TouchPivot: Blending WIMP & Post-WIMP Interfaces for Data Exploration on Tablet Devices”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, May 2017, pp. 2660–2671. DOI: 10.1145/3025453.3025752 (cit. on p. 31).

[Kan+11] Sean Kandel, Jeffrey Heer, Catherine Plaisant, et al. “Research Directions in Data Wrangling: Visualizations and Transformations for Usable and Credible Data”. In: *Information Visualization* 10.4 (Sept. 2011), pp. 271–288. DOI: 10.1177/1473871611415994 (cit. on pp. 26, 61).

[KR18] Jan-Frederik Kassel and Michael Rohs. “Valletto: A Multimodal Interface for Ubiquitous Visual Analytics”. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, LBW0005:1–LBW005:6. DOI: 10.1145/3170427.3188445 (cit. on p. 57).

[Kat+16] Keiko Katsuragawa, Krzysztof Pietroszek, James R. Wallace, and Edward Lank. “Watchpoint: Freehand Pointing with a Smartwatch in a Ubiquitous Display Environment”. In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2016, pp. 128–135. DOI: 10.1145/2909132.2909263 (cit. on pp. 22, 122).

[Kay+05] Joseph 'Jofish' Kaye, Mariah K. Levitt, Jeffrey Nevins, Jessica Golden, and Vanessa Schmidt. "Communicating Intimacy One Bit at a Time". In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2005, pp. 1529–1532. DOI: 10.1145/1056808.1056958 (cit. on p. 104).

[Kei+06] D.A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. "Challenges in Visual Data Analysis". In: *International Conference on Information Visualisation*. Piscataway, NJ, USA: IEEE, 2006, pp. 9–16. DOI: 10.1109/iv.2006.31 (cit. on p. 26).

[Kei02] Daniel A. Keim. "Information Visualization and Visual Data Mining". In: *IEEE Transactions on Visualization and Computer Graphics* 8.1 (2002), pp. 1–8. DOI: 10.1109/2945.981847 (cit. on p. 29).

[KPW14] Andreas Kerren, Helen C. Purchase, and Matthew O. Ward, eds. *Multivariate Network Visualization*. Springer International Publishing, 2014. DOI: 10.1007/978-3-319-06793-3 (cit. on pp. 62, 63, 67).

[Ker+17] E. Kerzner, A. Lex, C.L. Sigulinsky, et al. "Graffinity: Visualizing Connectivity in Large Graphs". In: *Computer Graphics Forum* 36.3 (June 2017), pp. 251–260. DOI: 10.1111/cgf.13184 (cit. on p. 63).

[Kim+19] Hwiyeon Kim, Juyoung Oh, Yunha Han, et al. "Thumbnails for Data Stories: A Survey of Current Practices". In: *2019 IEEE Visualization Conference (VIS)*. Piscataway, NJ, USA: IEEE, 2019, pp. 116–120. DOI: 10.1109/visual.2019.8933773 (cit. on p. 54).

[Kim+07] Jungsoo Kim, Jiasheng He, Kent Lyons, and Thad Starner. "The Gesture Watch: A Wireless Contact-free Gesture based Wrist Interface". In: *Proceedings of the IEEE Symposium on Wearable Computers*. Piscataway, NJ, USA: IEEE, Oct. 2007, pp. 15–22. DOI: 10.1109/iswc.2007.4373770 (cit. on p. 96).

[Kir95] David Kirsh. "The Intelligent Use of Space". In: *Artificial Intelligence* 73.1-2 (Feb. 1995), pp. 31–68. DOI: 10.1016/0004-3702(94)00017-u (cit. on pp. 25, 144, 147, 151).

[Kis+17] Ulrike Kister, Konstantin Klamka, Christian Tominski, and Raimund Dachselt. "GraSp: Combining Spatially-aware Mobile Devices and a Display Wall for Graph Visualization and Interaction". In: *Computer Graphics Forum* 36 (June 2017), pp. 503–514. DOI: 10.1111/cgf.13206 (cit. on pp. 6, 23, 33, 35, 38, 65, 66, 84, 138, 150, 164, 192).

[KRD16] Ulrike Kister, Patrick Reipschläger, and Raimund Dachselt. "MultiLens: Fluent Interaction with Multi-Functional Multi-Touch Lenses for Information Visualization". In: *Proceedings of the ACM Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2016, pp. 139–148. DOI: 10.1145/2992154.2992168 (cit. on pp. 30, 31, 65, 70).

[Kis+15] Ulrike Kister, Patrick Reipschläger, Fabrice Matulic, and Raimund Dachselt. “BodyLenses: Embodied Magic Lenses and Personal Territories for Wall Displays”. In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2015, pp. 117–126. DOI: 10.1145/2817721.2817726 (cit. on pp. 33, 34, 119).

[KD17] Konstantin Klamka and Raimund Dachselt. “IllumiPaper: Illuminated Interactive Paper”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 5605–5618. DOI: 10.1145/3025453.3025525 (cit. on p. 98).

[KHD20] Konstantin Klamka, Tom Horak, and Raimund Dachselt. “Watch+Strap: Extending Smartwatches with Interactive StrapDisplays”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, 72:1–72:15. DOI: 10.1145/3313831.3376199 (cit. on pp. 12, 48, 94, 97, 106).

[Klo+15] Clemens N. Klokmose, James R. Eagan, Siemen Baader, Wendy Mackay, and Michel Beaudouin-Lafon. “Webstrates: Shareable Dynamic Media”. In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2015, pp. 280–290. DOI: 10.1145/2807442.2807446 (cit. on pp. 36, 159, 168).

[Kör16] Christoph Körner. *Learning Responsive Data Visualization*. Packt Publishing, 2016. 258 pp. (cit. on p. 46).

[Koy+18] Philipp Koytek, Charles Perin, Jo Vermeulen, Elisabeth Andre, and Sheelagh Carpendale. “MyBrush: Brushing and Linking with Personal Agency”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 605–615. DOI: 10.1109/tvcg.2017.2743859 (cit. on p. 29).

[KNS04] Matthias Kreuseler, Thomas Nocke, and Heidrun Schumann. “A History Mechanism for Visual Data Mining”. In: *IEEE Symposium on Information Visualization*. Piscataway, NJ, USA: IEEE, 2004, pp. 49–56. DOI: 10.1109/infvis.2004.2 (cit. on p. 89).

[Lam08] Heidi Lam. “A Framework of Interaction Costs in Information Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (Nov. 2008), pp. 1149–1156. DOI: 10.1109/tvcg.2008.109 (cit. on p. 189).

[LHD18a] Ricardo Langner, Tom Horak, and Raimund Dachselt. “Demonstrating VisTiles: Visual Data Exploration Using Mobile Devices”. In: *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2018, 69:1–69:3. DOI: 10.1145/3206505.3206583 (cit. on pp. 13, 144, 152).

[LHD15] Ricardo Langner, Tom Horak, and Raimund Dachselt. “Information Visualizations with Mobile Devices: Three Promising Aspects”. In: *Workshop on Data Exploration for Interactive Surfaces DEXIS 2015*. 2015 (cit. on p. 152).

[LHD16] Ricardo Langner, Tom Horak, and Raimund Dachselt. “Towards Combining Mobile Devices for Visual Data Exploration”. In: *Poster Program of the 2016 IEEE Conference on Information Visualization (InfoVis)*. Baltimore, MD, USA, Oct. 2016 (cit. on p. 152).

[LHD18b] Ricardo Langner, Tom Horak, and Raimund Dachselt. “VisTiles: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018), pp. 626–636. DOI: 10.1109/tvcg.2017.2744019 (cit. on pp. 13, 23, 25, 36, 144, 152, 153, 160, 163, 164).

[LKD19] Ricardo Langner, Ulrike Kister, and Raimund Dachselt. “Multiple Coordinated Views at Large Displays for Multiple Users: Empirical Findings on User Behavior, Movements, and Distances”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 608–618. DOI: 10.1109/TVCG.2018.2865235 (cit. on pp. 5, 6, 17, 22, 33, 35, 119, 123, 160, 164).

[Lan+15] Ricardo Langner, Ulrike Kister, Christin Engel, Tom Horak, and Raimund Dachselt. “Towards a Fluid Interaction Concept Bringing Star Plots to Interactive Displays”. In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2015, pp. 349–354. DOI: 10.1145/2817721.2823505 (cit. on p. 30).

[Lan+21] Ricardo Langner, Marc Satkowski, Wolfgang Büschel, and Raimund Dachselt. “MARVIS: Combining Mobile Devices and Augmented Reality for Data Analysis”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. (To appear). New York, NY, USA: ACM, 2021. DOI: 10.1145/3411764.3445593 (cit. on p. 191).

[Lan+16] Ricardo Langner, Ulrich von Zadow, Tom Horak, Annett Mitschick, and Raimund Dachselt. “Content Sharing Between Spatially-Aware Mobile Phones and Large Vertical Displays Supporting Collaborative Work”. In: *Collaboration Meets Interactive Spaces*. Cham: Springer International Publishing, 2016, pp. 75–96. DOI: 10.1007/978-3-319-45853-3_5 (cit. on pp. 22, 23, 192).

[Lap+14] Gierad Laput, Robert Xiao, Xiang ’Anthony’ Chen, Scott E. Hudson, and Chris Harrison. “Skin Buttons: Cheap, Small, Low-powered and Clickable Fixed-icon Laser Projectors”. In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2014, pp. 389–394. DOI: 10.1145/2642918.2647356 (cit. on p. 96).

[Led+15] David Ledo, Saul Greenberg, Nicolai Marquardt, and Sebastian Boring. “Proxemic-Aware Controls: Designing Remote Controls for Ubiquitous Computing Ecologies”. In: *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA: ACM, 2015, pp. 187–198. DOI: 10.1145/2785830.2785871 (cit. on pp. 20, 25, 164).

[Lee+18] Bongshin Lee, Matthew Brehmer, Petra Isenberg, et al. “Data Visualization on Mobile Devices”. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, W07:1–W07:8. DOI: 10.1145/3170427.3170631 (cit. on pp. 32, 51).

[Lee+21] Bongshin Lee, Eun Kyoung Choe, Petra Isenberg, and Raimund Dachselt, eds. *Mobile Data Visualization*. AK Peters Visualization Series. (To appear). CRC Press, 2021 (cit. on pp. 31, 33).

[Lee+12] Bongshin Lee, Petra Isenberg, Nathalie Henry Riche, and Sheelagh Carpendale. "Beyond Mouse and Keyboard: Expanding Design Considerations for Information Visualization Interactions". In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2689–2698. DOI: 10.1109/tvcg.2012.204 (cit. on pp. 3, 4, 7, 30).

[Lee+06] Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. "Task Taxonomy for Graph Visualization". In: *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*. New York, NY, USA: ACM, 2006, pp. 1–5. DOI: 10.1145/1168149.1168168 (cit. on pp. 61, 66).

[LK12] Ming Li and Leif Kobbelt. "Dynamic Tiling Display: Building an Interactive Display Surface Using Multiple Mobile Devices". In: *Proceedings of the ACM International Conference on Mobile and Ubiquitous Multimedia*. New York, NY, USA: ACM, 2012, pp. 1–4. DOI: 10.1145/2406367.2406397 (cit. on pp. 23, 149).

[Liu+17] Can Liu, Olivier Chapuis, Michel Beaudouin-Lafon, and Eric Lecolinet. "CoReach: Cooperative Gestures for Data Manipulation on Wall-sized Displays". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 6730–6741. DOI: 10.1145/3025453.3025594 (cit. on pp. 34, 117).

[Liu+14] Can Liu, Olivier Chapuis, Michel Beaudouin-Lafon, Eric Lecolinet, and Wendy E. Mackay. "Effects of Display Size and Navigation Type on a Classification Task". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 4147–4156. DOI: 10.1145/2556288.2557020 (cit. on pp. 20, 22, 167).

[LHJ11] Andrés Lucero, Jussi Holopainen, and Tero Jokela. "Pass-them-around: Collaborative Use of Mobile Phones for Photo Sharing". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2011, pp. 1787–1796. DOI: 10.1145/1978942.1979201 (cit. on p. 149).

[Lyo+12] Kent Lyons, David Nguyen, Daniel Ashbrook, and Sean White. "Facet: A Multi-segment Wrist Worn System". In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2012, pp. 123–130. DOI: 10.1145/2380116.2380134 (cit. on p. 96).

[Mac04] Alan MacEachren. *How Maps Work: Representation, Visualization, and Design*. New York: Guilford Press, 2004 (cit. on p. 54).

[MB19] Timothy Major and Rahul C. Basole. "Graphicle: Exploring Units, Networks, and Context in a Blended Visualization Approach". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 576–585. DOI: 10.1109/tvcg.2018.2865151 (cit. on p. 63).

[Mar11] Ethan Marcotte. *Responsive Web Design*. New York: A Book Apart/Jeffrey Zeldman, 2011 (cit. on pp. 33, 45, 48).

[Mar+12] Nicolai Marquardt, Till Ballendat, Sebastian Boring, Saul Greenberg, and Ken Hinckley. "Gradual Engagement: Facilitating Information Exchange between Digital Devices as a Function of Proximity". In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2012, pp. 31–40. DOI: 10.1145/2396636.2396642 (cit. on pp. 20, 25).

[MG12] Nicolai Marquardt and Saul Greenberg. "Informing the Design of Proxemic Interactions". In: *IEEE Pervasive Computing* 11.2 (Feb. 2012), pp. 14–23. DOI: 10.1109/mprv.2012.15 (cit. on p. 126).

[MHG12] Nicolai Marquardt, Ken Hinckley, and Saul Greenberg. "Cross-device Interaction via Micro-mobility and F-formations". In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2012, pp. 13–22. DOI: 10.1145/2380116.2380121 (cit. on p. 25).

[Mar+18] Kim Marriott, Falk Schreiber, Tim Dwyer, et al., eds. *Immersive Analytics*. Springer International Publishing, 2018. DOI: 10.1007/978-3-030-01388-2 (cit. on p. 30).

[MC12] Juha Matero and Ashley Colley. "Identifying Unintentional Touches on Handheld Touch Screen Devices". In: *Proceedings of the ACM Designing Interactive Systems Conference*. New York, NY, USA: ACM, 2012, pp. 506–509. DOI: 10.1145/2317956.2318031 (cit. on p. 19).

[Mat+19] Andreas Mathisen, Tom Horak, Clemens Nylandsted Klokmose, Kaj Grønbæk, and Niklas Elmqvist. "InsideInsights: Integrating Data-Driven Reporting in Collaborative Visual Analytics". In: *Computer Graphics Forum* 38.3 (June 2019). DOI: 10.1111/cgf.13717 (cit. on pp. 26, 28, 57, 89).

[Mat+02] K. Matkovic, H. Hauser, R. Sainitzer, and M.E. Groller. "Process Visualization with Levels of Detail". In: *IEEE Symposium on Information Visualization*. Piscataway, NJ, USA: IEEE, 2002, pp. 67–70. DOI: 10.1109/infvis.2002.1173149 (cit. on pp. 65, 76).

[McG+12] Will McGrath, Brian Bowman, David McCallum, et al. "Branch-explore-merge: Facilitating Real-time Revision Control in Collaborative Visual Exploration". In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2012, pp. 235–244. DOI: 10.1145/2396636.2396673 (cit. on pp. 23, 34, 123, 164).

[MJ09] M.J. McGuffin and I. Jurisica. "Interaction Techniques for Selecting and Manipulating Subgraphs in Network Visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (Nov. 2009), pp. 937–944. DOI: 10.1109/tvcg.2009.151 (cit. on p. 66).

[McI+19] Jess McIntosh, Paul Strohmeier, Jarrod Knibbe, Sebastian Boring, and Kasper Hornbæk. “Magnetips: Combining Fingertip Tracking and Haptic Feedback for Around-Device Interaction”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, 408:1–408:12. DOI: 10.1145/3290605.3300638 (cit. on p. 96).

[McL+08] Peter McLachlan, Tamara Munzner, Eleftherios Koutsofios, and Stephen North. “LiveRAC: Interactive Visual Exploration of System Management Time-series Data”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2008, pp. 1483–1492. DOI: 10.1145/1357054.1357286 (cit. on pp. 65, 66, 76).

[MA14] Silvia Miksch and Wolfgang Aigner. “A Matter of Time: Applying a Data–Users–Tasks Design Triangle to Visual Analytics of Time-oriented Data”. In: *Computers & Graphics* 38 (Feb. 2014), pp. 286–290. DOI: 10.1016/j.cag.2013.11.002 (cit. on p. 59).

[Mor+14] Robert Morawa, Tom Horak, Ulrike Kister, Annett Mitschick, and Raimund Dachselt. “Combining Timeline and Graph Visualization”. In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2014, pp. 345–350. DOI: 10.1145/2669485.2669544 (cit. on p. 56).

[Mor+19] Dominik Moritz, Chenglong Wang, Greg L. Nelson, et al. “Formalizing Visualization Design Knowledge as Constraints: Actionable and Extensible Models in Draco”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 438–448. DOI: 10.1109/tvcg.2018.2865240 (cit. on p. 177).

[Mun09] Tamara Munzner. “A Nested Model for Visualization Design and Validation”. In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (Nov. 2009), pp. 921–928. DOI: 10.1109/tvcg.2009.111 (cit. on pp. 38, 58).

[Mun14] Tamara Munzner. *Visualization Analysis and Design*. Taylor & Francis Ltd., Dec. 1, 2014. 428 pp. (cit. on pp. 3, 25, 29, 38, 46, 53).

[Mur+12] Sundar Murugappan, Vinayak, Niklas Elmqvist, and Karthik Ramani. “Extended Multitouch: Recovering Touch Posture and Differentiating Users Using a Depth Camera”. In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2012, pp. 487–496. DOI: 10.1145/2380116.2380177 (cit. on p. 139).

[NC14] Mathieu Nancel and Andy Cockburn. “Causality: A Conceptual Model of Interaction History”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 1777–1786. DOI: 10.1145/2556288.2556990 (cit. on p. 89).

[Neb17] Michael Nebeling. “XDBrowser 2.0: Semi-Automatic Generation of Cross-Device Interfaces”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 4574–4584. DOI: 10.1145/3025453.3025547 (cit. on p. 159).

[ND16] Michael Nebeling and Anind K. Dey. “XDBrowser: User-Defined Cross-Device Web Page Designs”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2016, pp. 5494–5505. DOI: 10.1145/2858036.2858048 (cit. on p. 159).

[Nob+19] Carolina Nobre, Miriah Meyer, Marc Streit, and Alexander Lex. “The State of the Art in Visualizing Multivariate Networks”. In: *Computer Graphics Forum* 38.3 (June 2019), pp. 807–832. DOI: 10.1111/cgf.13728 (cit. on pp. 62–64, 68).

[NSL19] Carolina Nobre, Marc Streit, and Alexander Lex. “Juniper: A Tree+Table Approach to Multivariate Graph Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 544–554. DOI: 10.1109/tvcg.2018.2865149 (cit. on p. 63).

[Nob+20] Carolina Nobre, Dylan Wootton, Lane Harrison, and Alexander Lex. “Evaluating Multivariate Network Visualization Techniques Using a Validated Design and Crowdsourcing Approach”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, pp. 1–12. DOI: 10.1145/3313831.3376381 (cit. on p. 64).

[ND86] Donald Norman and Draper. *User Centered System Design: New Perspectives on Human-computer Interaction*. Hillsdale, NJ, USA: L. Erlbaum Associates, 1986 (cit. on p. 18).

[OL14] Ian Oakley and Doyoung Lee. “Interaction on the Edge: Offset Sensing for Small Devices”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 169–178. DOI: 10.1145/2556288.2557138 (cit. on p. 98).

[OO05] Ian Oakley and Sile O’Modhrain. “Tilt to Scroll: Evaluating a Motion Based Vibrotactile Mobile Interface”. In: *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. Piscataway, NJ, USA: IEEE, 2005, pp. 40–49. DOI: 10.1109/whc.2005.138 (cit. on p. 19).

[OTI15] Masa Ogata, Ryosuke Totsuka, and Michita Imai. “SkinWatch: Adapting Skin As a Gesture Surface”. In: *SIGGRAPH Asia 2015 Emerging Technologies*. New York, NY, USA: ACM, 2015, 22:1–22:2. DOI: 10.1145/2818466.2818496 (cit. on p. 96).

[OT12] Takashi Ohta and Jun Tanaka. “Pinch: An Interface That Relates Applications on Multiple Touch-Screen by ‘Pinching’ Gesture”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 320–335. DOI: 10.1007/978-3-642-34292-9_23 (cit. on pp. 24, 149, 152).

[OJK19] Mershack Okoe, Radu Jianu, and Stephen Kobourov. “Node-Link or Adjacency Matrices: Old Question, New Insights”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.10 (Oct. 2019), pp. 2940–2952. DOI: 10.1109/tvcg.2018.2865940 (cit. on pp. 64, 81).

[OWS14] Simon Olberding, Michael Wessely, and Jürgen Steimle. “PrintScreen: Fabricating Highly Customizable Thin-film Touch-displays”. In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2014, pp. 281–290. DOI: 10.1145/2642918.2647413 (cit. on pp. 95, 98).

[Olb+13] Simon Olberding, Kian Peen Yeo, Suranga Nanayakkara, and Jurgen Steimle. “AugmentedForearm: Exploring the Design Space of a Display-enhanced Forearm”. In: *Proceedings of the International Conference on Augmented Human*. New York, NY, USA: ACM, 2013, pp. 9–12. DOI: 10.1145/2459236.2459239 (cit. on p. 96).

[Olw18] Alex Olwal. “Hybrid Watch User Interfaces: Collaboration Between Electro-Mechanical Components and Analog Materials”. In: *Adjunct Publication of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2018, pp. 200–202. DOI: 10.1145/3266037.3271650 (cit. on p. 95).

[OGW19] Alvitta Ottley, Roman Garnett, and Ran Wan. “Follow The Clicks: Learning and Anticipating Mouse Interactions During Exploratory Data Analysis”. In: *Computer Graphics Forum* 38.3 (June 2019), pp. 41–52. DOI: 10.1111/cgf.13670 (cit. on pp. 189, 190).

[Par+18] Seonwook Park, Antti Oulasvirta, Otmar Hilliges, et al. “AdaM: Adapting Multi-User Interfaces for Collaborative Environments in Real-Time”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, 184:1–184:14. DOI: 10.1145/3173574.3173758 (cit. on pp. 159, 161, 164, 165, 176, 177).

[PSS11] Jerome Pasquero, Scott J. Stobbe, and Noel Stonehouse. “A Haptic Wristwatch for Eyes-free Interactions”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2011, pp. 3257–3266. DOI: 10.1145/1978942.1979425 (cit. on p. 121).

[Pav+21] Leonardo Pavanatto, Chris North, Doug Bowman, Richard Stoakley, and Carmen Badea. “Do We Still Need Physical Monitors? An Evaluation of the Usability of AR Virtual Monitors for Productivity Work”. In: *2021 IEEE Conference on Virtual Reality and 3D User Interfaces*. (To appear). Piscataway, NJ, USA, 2021 (cit. on p. 191).

[PDF14] Charles Perin, Pierre Dragicevic, and Jean-Daniel Fekete. “Revisiting Bertin Matrices: New Interactions for Crafting Tabular Visualizations”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 2082–2091. DOI: 10.1109/tvcg.2014.2346279 (cit. on p. 66).

[PF93] Ken Perlin and David Fox. “Pad: An Alternative Approach to the Computer Interface”. In: *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: ACM, 1993, pp. 57–64. DOI: 10.1145/166117.166125 (cit. on pp. 65, 76).

[Per+13] Simon T. Perrault, Eric Lecolinet, James Eagan, and Yves Guiard. “WatchIt: Simple Gestures and Eyes-free Interaction for Wristwatches and Bracelets”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2013, pp. 1451–1460. DOI: 10.1145/2470654.2466192 (cit. on pp. 96, 103).

[Piz+16] Stefania Pizza, Barry Brown, Donald McMillan, and Airi Lampinen. “Smartwatch in Vivo”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2016, pp. 5456–5469. DOI: 10.1145/2858036.2858522 (cit. on p. 17).

[Pla+17] Thomas Plank, Hans-Christian Jetter, Roman Rädle, et al. “Is Two Enough?! Studying Benefits, Barriers, and Biases of Multi-Tablet Use for Collaborative Visualization”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 4548–4560. DOI: 10.1145/3025453.3025537 (cit. on pp. 23, 37, 164).

[PD15] Bernhard Preim and Raimund Dachselt. *Interaktive Systeme: Band 2: User Interface Engineering, 3D-Interaktion, Natural User Interfaces*. Vol. 2. Springer Berlin Heidelberg, 2015. DOI: 10.1007/978-3-642-45247-5 (cit. on p. 16).

[PPS14] Johannes Pretorius, Helen C. Purchase, and John T. Stasko. “Tasks for Multivariate Network Analysis”. In: *Multivariate Network Visualization*. Cham, Switzerland: Springer International Publishing, 2014, pp. 77–95. DOI: 10.1007/978-3-319-06793-3_5 (cit. on pp. 61, 63, 66).

[PBC17] Arnaud Prouzeau, Anastasia Bezerianos, and Olivier Chapuis. “Evaluating Multi-User Selection for Exploring Graph Topology on Wall-Displays”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.8 (2017), pp. 1936–1951. DOI: 10.1109/tvcg.2016.2592906 (cit. on pp. 33, 34).

[PBC16] Arnaud Prouzeau, Anastasia Bezerianos, and Olivier Chapuis. “Towards Road Traffic Management with Forecasting on Wall Displays”. In: *Proceedings of the ACM Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2016, pp. 119–128. DOI: 10.1145/2992154.2992158 (cit. on p. 33).

[QFZ08] Lin Qiao, Ling Feng, and Lizhu Zhou. “Information Presentation on Mobile Devices: Techniques and Practices”. In: *Progress in WWW Research and Development*. Springer Berlin Heidelberg, 2008, pp. 395–406. DOI: 10.1007/978-3-540-78849-2_40 (cit. on p. 46).

[QH18] Zening Qu and Jessica Hullman. “Keeping Multiple Views Consistent: Constraints, Validations, and Exceptions in Visualization Authoring”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018), pp. 468–477. DOI: 10.1109/tvcg.2017.2744198 (cit. on pp. 59, 160, 162, 165, 177).

[Räd+14] Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. “HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration”. In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2014, pp. 45–54. DOI: 10.1145/2669485.2669500 (cit. on pp. 19, 23, 36, 159, 164).

[Räd+15] Roman Rädle, Hans-Christian Jetter, Mario Schreiner, et al. “Spatially-aware or Spatially-agnostic?: Elicitation and Evaluation of User-Defined Cross-Device Interactions”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2015, pp. 3913–3922. DOI: 10.1145/2702123.2702287 (cit. on p. 24).

[Räd+17] Roman Rädle, Midas Nouwens, Kristian Antonsen, James R. Eagan, and Clemens N. Klokmose. “Codestrates: Literate Computing with Webstrates”. In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2017, pp. 715–725. DOI: 10.1145/3126594.3126642 (cit. on pp. 36, 168).

[Rad+14] Axel Radloff, Christian Tominski, Thomas Nocke, and Heidrun Schumann. “Supporting Presentation and Discussion of Visualization Results in Smart Meeting Rooms”. In: *The Visual Computer* 31.9 (July 2014), pp. 1271–1286. DOI: 10.1007/s00371-014-1010-x (cit. on p. 22).

[RC94] Ramana Rao and Stuart K. Card. “The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 1994, pp. 318–322. DOI: 10.1145/191666.191776 (cit. on pp. 65, 66, 101).

[RJS01] Uwe Rauschenbach, Stefan Jeschke, and Heidrun Schumann. “General Rectangular Fisheye Views for 2D Graphics”. In: *Computers & Graphics* 25.4 (Aug. 2001), pp. 609–617. DOI: 10.1016/s0097-8493(01)00089-9 (cit. on p. 65).

[Red+15] Khairi Reda, Andrew E. Johnson, Michael E. Papka, and Jason Leigh. “Effects of Display Size and Resolution on User Behavior and Insight Acquisition in Visual Exploration”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2015, pp. 2759–2768. DOI: 10.1145/2702123.2702406 (cit. on p. 139).

[RRF20] Joshua Reibert, Patrick Riehmann, and Bernd Froehlich. “Multitouch Interaction with Parallel Coordinates on Large Vertical Displays”. In: *Proceedings of the ACM on Human-Computer Interaction* 4.ISS (Nov. 2020), pp. 1–22. DOI: 10.1145/3427327 (cit. on pp. 30, 32, 33).

[RFD21] Patrick Reipschläger, Tamara Flemisch, and Raimund Dachselt. “Personal Augmented Reality for Information Visualization on Large Interactive Displays”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pp. 1182–1192. DOI: 10.1109/TVCG.2020.3030460 (cit. on p. 191).

[Rek01] Jun Rekimoto. “GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices”. In: *Proceedings of the IEEE International Symposium on Wearable Computers*. Piscataway, NJ, USA: IEEE, 2001, pp. 21–27. DOI: 10.1109/iswc.2001.962092 (cit. on pp. 118, 120, 122).

[Ren+17] Donghao Ren, Matthew Brehmer, Bongshin Lee, Tobias Hollerer, and Eun Kyoung Choe. “ChartAccent: Annotation for Data-driven Storytelling”. In: *Proceedings of the IEEE Pacific Symposium on Visualization*. Piscataway, NJ, USA: IEEE, 2017, pp. 230–239. DOI: 10.1109/pacificvis.2017.8031599 (cit. on p. 54).

[Rie+20] Patrick Riehmann, Gabriela Molina León, Joshua Reibert, Florian Echtler, and Bernd Froehlich. “Short-Contact Touch-Manipulation of Scatterplot Matrices on Wall Displays”. In: *Computer Graphics Forum* 39.3 (2020), pp. 265–276. DOI: 10.1111/cgf.13979 (cit. on pp. 30, 33, 38).

[Rob07] Jonathan C. Roberts. “State of the Art: Coordinated & Multiple Views in Exploratory Visualization”. In: *Proccedings of the IEEE Conference on Coordinated and Multiple Views in Exploratory Visualization*. Piscataway, NJ, USA: IEEE, July 2007, pp. 61–71. DOI: 10.1109/cmv.2007.20 (cit. on pp. 29, 38, 67, 119, 145, 163).

[RHR16] Jonathan C. Roberts, Chris Headleand, and Panagiotis D. Ritsos. “Sketching Designs Using the Five Design-Sheet Methodology”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), pp. 419–428. DOI: 10.1109/tvcg.2015.2467271 (cit. on p. 59).

[Rob+19] Jonathan C. Roberts, Hayder Al-maneea, Peter W. S. Butcher, et al. “Multiple Views: Different Meanings and Collocated Words”. In: *Computer Graphics Forum* 38.3 (June 2019), pp. 79–93. DOI: 10.1111/cgf.13673 (cit. on pp. 28, 29, 38, 160, 162).

[Rob+14] Jonathan C. Roberts, Panagiotis D. Ritsos, Sriram Karthik Badam, et al. “Visualization beyond the Desktop—the Next Big Thing”. In: *IEEE Computer Graphics and Applications* 34.6 (Nov. 2014), pp. 26–34. DOI: 10.1109/mcg.2014.82 (cit. on pp. 3, 4, 7, 30).

[RL04] Yvonne Rogers and Siân Lindley. “Collaborating Around Vertical and Horizontal Large Interactive Displays: Which Way is Best?” In: *Interacting with Computers* 16.6 (Dec. 2004), pp. 1133–1152. DOI: 10.1016/j.intcom.2004.07.008 (cit. on p. 21).

[Rom+19] Hugo Romat, Nathalie Henry Riche, Ken Hinckley, et al. “ActiveInk: (Th)Inking with Data”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, 42:1–42:13. DOI: 10.1145/3290605.3300272 (cit. on pp. 31, 89).

[RM15] Puripant Ruchikachorn and Klaus Mueller. “Learning Visualizations by Analogy: Promoting Visual Literacy through Visualization Morphing”. In: *IEEE Transactions on Visualization and Computer Graphics* 21.9 (Sept. 2015), pp. 1028–1044. DOI: 10.1109/tvcg.2015.2413786 (cit. on p. 56).

[RLL11] Jaime Ruiz, Yang Li, and Edward Lank. “User-defined Motion Gestures for Mobile Interaction”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2011, pp. 197–206. DOI: 10.1145/1978942.1978971 (cit. on p. 19).

[RK14] Jeffrey M. Rzeszotarski and Aniket Kittur. "Kinetica: Naturalistic Multi-touch Data Visualization". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 897–906. DOI: 10.1145/2556288.2557231 (cit. on p. 30).

[SS14] Ramik Sadana and John Stasko. "Designing and Implementing an Interactive Scatterplot Visualization for a Tablet Computer". In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2014, pp. 265–272. DOI: 10.1145/2598153.2598163 (cit. on pp. 30, 31, 38, 89).

[SS16] Ramik Sadana and John Stasko. "Designing Multiple Coordinated Visualizations for Tablets". In: *Computer Graphics Forum* 35.3 (June 2016), pp. 261–270. DOI: 10.1111/cgf.12902 (cit. on pp. 31, 32, 156).

[SSS20] Ayshwarya Saktheeswaran, Arjun Srinivasan, and John Stasko. "Touch? Speech? or Touch and Speech? Investigating Multimodal Interaction for Visual Network Exploration and Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 26.6 (June 2020), pp. 2168–2179. DOI: 10.1109/tvcg.2020.2970512 (cit. on p. 31).

[Sam18] Cedric Sam. "Ai2html and Its Impact on the News Graphics Industry". In: *Proceedings of the CHI Workshop on Data Visualization on Mobile Devices*. 2018 (cit. on p. 45).

[SW13] Stephanie Santosa and Daniel Wigdor. "A Field Study of Multi-device Workflows in Distributed Workspaces". In: *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. New York, NY, USA: ACM, 2013, pp. 63–72. DOI: 10.1145/2493432.2493476 (cit. on pp. 4, 16).

[Sar+19] Alper Sarikaya, Michael Correll, Lyn Bartram, Melanie Tory, and Danyel Fisher. "What Do We Talk About When We Talk About Dashboards?" In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 682–692. DOI: 10.1109/TVCG.2018.2864903 (cit. on pp. 28, 57, 160, 162).

[Sar+93] Manojit Sarkar, Scott S. Snibbe, Oren J. Tversky, and Steven P. Reiss. "Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens". In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 1993, pp. 81–91. DOI: 10.1145/168642.168650 (cit. on p. 65).

[Sat+17] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. "Vega-Lite: A Grammar of Interactive Graphics". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (Jan. 2017), pp. 341–350. DOI: 10.1109/tvcg.2016.2599030 (cit. on pp. 161, 184).

[SWH14] Arvind Satyanarayan, Kanit Wongsuphasawat, and Jeffrey Heer. "Declarative Interaction Design for Data Visualization". In: *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2014, pp. 669–678. DOI: 10.1145/2642918.2647360 (cit. on p. 161).

[SBH17] Léa Saviot, Frederik Brudy, and Steven Houben. “WRISTBAND.IO: Expanding Input and Output Spaces of a Smartwatch”. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 2025–2033. DOI: 10.1145/3027063.3053132 (cit. on pp. 95, 96, 98, 103).

[Sch+20] Alexander Schiewe, Andrey Krekhov, Frederic Kerber, Florian Daiber, and Jens Krüger. “A Study on Real-Time Visualizations During Sports Activities on Smartwatches”. In: *19th International Conference on Mobile and Ubiquitous Multimedia*. New York, NY, USA: ACM, 2020, pp. 18–31. DOI: 10.1145/3428361.3428409 (cit. on p. 32).

[SS04] B.N. Schilit and U. Sengupta. “Device Ensembles”. In: *Computer* 37.12 (Dec. 2004), pp. 56–64. DOI: 10.1109/mc.2004.241 (cit. on p. 5).

[SOA16] Stefan Schneegass, Sophie Ogando, and Florian Alt. “Using On-body Displays for Extending the Output of Wearable Devices”. In: *Proceedings of the ACM International Symposium on Pervasive Displays*. New York, NY, USA: ACM, 2016, pp. 67–74. DOI: 10.1145/2914920.2915021 (cit. on p. 96).

[Sch+15] Mario Schreiner, Roman Rädle, Hans-Christian Jetter, and Harald Reiterer. “Connichiwa: A Framework for Cross-Device Web Applications”. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2015, pp. 2163–2168. DOI: 10.1145/2702613.2732909 (cit. on p. 159).

[Sch+13] Hans-Jorg Schulz, Thomas Nocke, Magnus Heitzler, and Heidrun Schumann. “A Design Space of Visualization Tasks”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2366–2375. DOI: 10.1109/tvcg.2013.120 (cit. on p. 26).

[Sch19] Michail Schwab. *Scalable Scalable Vector Graphics*. 2019. DOI: 10.17605/OSF.IO/GE8WP (cit. on p. 108).

[Sch+21] Michail Schwab, David Saffo, Yixuan Zhang, et al. “VisConnect: Distributed Event Synchronization for Collaborative Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pp. 347–357. DOI: 10.1109/tvcg.2020.3030366 (cit. on pp. 37, 159, 193).

[Sch+14] Julia Schwarz, Robert Xiao, Jennifer Mankoff, Scott E. Hudson, and Chris Harrison. “Probabilistic Palm Rejection Using Spatiotemporal Touch Features and Iterative Classification”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 2009–2012. DOI: 10.1145/2556288.2557056 (cit. on p. 19).

[SS91] Andrew Sears and Ben Shneiderman. “High Precision Touchscreens: Design Strategies and Comparisons with a Mouse”. In: *International Journal of Man-Machine Studies* 34.4 (Apr. 1991), pp. 593–613. DOI: 10.1016/0020-7373(91)90037-8 (cit. on p. 19).

[Sel+06] Abigail Sellen, Rachel Eardley, Shahram Izadi, and Richard Harper. “The Whereabouts Clock: Early Testing of a Situated Awareness Device”. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2006, pp. 1307–1312. DOI: 10.1145/1125451.1125694 (cit. on p. 104).

[SYV16] Teddy Seyed, Xing-Dong Yang, and Daniel Vogel. “Doppio: A Reconfigurable Dual-Face Smartwatch for Tangible Interaction”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2016, pp. 4675–4686. DOI: 10.1145/2858036.2858256 (cit. on p. 95).

[Shi+09] Lei Shi, Nan Cao, Shixia Liu, et al. “HiMap: Adaptive Visualization of Large-scale Online Social Networks”. In: *Proceedings of the IEEE Pacific Symposium on Visualization*. IEEE, Apr. 2009, pp. 41–48. DOI: 10.1109/pacificvis.2009.4906836 (cit. on pp. 65, 66).

[SC17] Mariana Shimabukuro and Christopher Collins. “Abbreviating Text Labels on Demand”. In: *Proc. of IEEE Conf. on Information Visualization (InfoVis)*. 2017 (cit. on p. 52).

[Shn83] Ben Shneiderman. “Direct Manipulation: A Step Beyond Programming Languages”. In: *IEEE Computer* 16.8 (Aug. 1983), pp. 57–69. DOI: 10.1109/mc.1983.1654471 (cit. on pp. 18, 30, 66).

[Shn96] Ben Shneiderman. “The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations”. In: *Proceedings of the IEEE Symposium on Visual Languages*. Piscataway, NJ, USA: IEEE, 1996, pp. 336–343. DOI: 10.1109/VL.1996.545307 (cit. on pp. 3, 119, 128).

[Shu+21] Xinhuan Shu, Aoyu Wu, Junxiu Tang, et al. “What Makes a Data-GIF Understandable?” In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pp. 1492–1502. DOI: 10.1109/tvcg.2020.3030396 (cit. on p. 56).

[Spe01] Robert Spence. *Information Visualization*. Harlow: Addison-Wesley, 2001 (cit. on pp. 3, 25, 62, 76, 84).

[Spi+13] Martin Spindler, Wolfgang Büschel, Charlotte Winkler, and Raimund Dachselt. “Tangible Displays for the Masses: Spatial Interaction with Handheld Displays by Using Consumer Depth Cameras”. In: *Personal and Ubiquitous Computing* 18.5 (Nov. 2013), pp. 1213–1225. DOI: 10.1007/s00779-013-0730-7 (cit. on pp. 23, 30).

[SCD13] Martin Spindler, Victor Cheung, and Raimund Dachselt. “Dynamic Tangible User Interface Palettes”. In: *Human-Computer Interaction – INTERACT 2013*. Springer Berlin Heidelberg, 2013, pp. 159–176. DOI: 10.1007/978-3-642-40498-6_12 (cit. on p. 22).

[SMD12] Martin Spindler, Marcel Martsch, and Raimund Dachselt. “Going Beyond the Surface: Studying Multi-layer Interaction above the Tabletop”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2012, pp. 1277–1286. DOI: 10.1145/2207676.2208583 (cit. on p. 31).

[Spi+14] Martin Spindler, Martin Schuessler, Marcel Martsch, and Raimund Dachselt. “Pinch-Drag-Flick vs. Spatial Input: Rethinking Zoom & Pan on Mobile Displays”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 1113–1122. DOI: 10.1145/2556288.2557028 (cit. on pp. 19, 31, 57).

[Spi+10] Martin Spindler, Christian Tominski, Heidrun Schumann, and Raimund Dachselt. “Tangible Views for Information Visualization”. In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2010, pp. 157–166. DOI: 10.1145/1936652.1936684 (cit. on pp. 25, 31, 35, 38).

[Sri+17] Srinath Sridhar, Anders Markussen, Antti Oulasvirta, Christian Theobalt, and Sebastian Boring. “WatchSense: On- and Above-Skin Input Sensing through a Wearable Depth Sensor”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 3891–3902. DOI: 10.1145/3025453.3026005 (cit. on p. 96).

[Sri+20] Arjun Srinivasan, Bongshin Lee, Nathalie Henry Riche, Steven M. Drucker, and Ken Hinckley. “InChorus: Designing Consistent Multimodal Interactions for Data Visualization on Tablet Devices”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, 653:1–653:13. DOI: 10.1145/3313831.3376782 (cit. on pp. 31, 186).

[SLS20] Arjun Srinivasan, Bongshin Lee, and John T. Stasko. “Interweaving Multimodal Interaction with Flexible Unit Visualizations for Data Exploration”. In: *IEEE Transactions on Visualization and Computer Graphics* (2020). DOI: 10.1109/tvcg.2020.2978050 (cit. on pp. 31, 57, 186).

[SS18] Arjun Srinivasan and John Stasko. “Orko: Facilitating Multimodal Interaction for Visual Exploration and Analysis of Networks”. In: *IEEE Transactions on Visualization and Computer Graphics* (Jan. 2018), pp. 511–521. DOI: 10.1109/tvcg.2017.2745219 (cit. on pp. 31, 89).

[SBV15] Paul Strohmeier, Jesse Burstyn, and Roel Vertegaal. “Effects of Display Sizes on a Scrolling Task using a Cylindrical Smartwatch”. In: *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. New York, NY, USA: ACM, 2015, pp. 846–853. DOI: 10.1145/2786567.2793710 (cit. on pp. 96, 100).

[SA19] Hariharan Subramonyam and Eytan Adar. “SmartCues: A Multitouch Query Approach for Details-on-Demand through Dynamically Computed Overlays”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 597–607. DOI: 10.1109/tvcg.2018.2865231 (cit. on p. 30).

[Sun+17] Ke Sun, Yuntao Wang, Chun Yu, et al. "Float: One-Handed and Touch-Free Target Selection on Smartwatches". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 692–704. DOI: 10.1145/3025453.3026027 (cit. on p. 96).

[TGH12] Justin Talbot, John Gerth, and Pat Hanrahan. "An Empirical Model of Slope Ratio Comparisons". In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2613–2620. DOI: 10.1109/tvcg.2012.196 (cit. on p. 51).

[TC05] James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005 (cit. on pp. 26, 34, 45).

[TH01] Alvin R. Tilley and Henry Dreyfuss Associates. *The Measure of Man and Woman: Human Factors in Design*. WILEY, Dec. 2001. 98 pp. (cit. on p. 102).

[Tom15] Christian Tominski. "Interaction for Visualization". In: *Synthesis Lectures on Visualization* 3.1 (June 2015), pp. 1–107 (cit. on pp. 3, 25, 30, 46, 56).

[TAS09] Christian Tominski, James Abello, and Heidrun Schumann. "CGV—An Interactive Graph Visualization System". In: *Computers & Graphics* 33.6 (Dec. 2009), pp. 660–678. DOI: 10.1016/j.cag.2009.06.002 (cit. on pp. 65, 66).

[Tom+16] Christian Tominski, Stefan Gladisch, Ulrike Kister, Raimund Dachselt, and Heidrun Schumann. "Interactive Lenses for Visualization: An Extended Survey". In: *Computer Graphics Forum* 36.6 (May 2016), pp. 173–200. DOI: 10.1111/cgf.12871 (cit. on pp. 65, 66, 101).

[TS20] Christian Tominski and Heidrun Schumann. *Interactive Visual Data Analysis*. AK Peters Visualization Series. CRC Press, 2020. DOI: 10.1201/9781315152707 (cit. on pp. 3, 4, 25, 26, 28, 50).

[Tuf06] Edward R. Tufte. *Beautiful Evidence*. Cheshire, MA, USA: Graphics Press, 2006 (cit. on pp. 3, 65, 110).

[Tuf01] Edward R. Tufte. *The Visual Display of Quantitative Information*. Cheshire, MA, USA: Graphics Press, 2001 (cit. on p. 75).

[Tuk77] John Tukey. *Exploratory Data Analysis*. Reading, Mass: Addison-Wesley Pub. Co, 1977 (cit. on p. 26).

[Vis+17] Aku Visuri, Zhanna Sarsenbayeva, Niels van Berkel, et al. "Quantifying Sources and Types of Smartwatch Usage Sessions". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 3569–3581. DOI: 10.1145/3025453.3025817 (cit. on p. 17).

[Vog+17] Anita Vogl, Patrick Parzer, Teo Babic, et al. "StretchEBand: Enabling Fabric-based Interactions Through Rapid Fabrication of Textile Stretch Sensors". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 2617–2627. DOI: 10.1145/3025453.3025938 (cit. on p. 96).

[Wal+12] Jagoda Walny, Bongshin Lee, Paul Johns, Nathalie Henry Riche, and Sheelagh Carpendale. “Understanding Pen and Touch Interaction for Data Exploration on Interactive Whiteboards”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2779–2788. DOI: 10.1109/tvcg.2012.275 (cit. on p. 31).

[WR09] Feng Wang and Xiangshi Ren. “Empirical Evaluation for Finger Input Properties in Multi-touch Interaction”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2009, pp. 1063–1072. DOI: 10.1145/1518701.1518864 (cit. on pp. 19, 50).

[War12] Colin Ware. *Information Visualization: Perception for Design*. Boston: Morgan Kaufmann, 2012 (cit. on pp. 25, 49).

[WS15] Benjamin Watson and Vidya Setlur. “Emerging Research in Mobile Visualization”. In: *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. New York, NY, USA: ACM, 2015, pp. 883–887. DOI: 10.1145/2786567.2786571 (cit. on pp. 51, 52, 54).

[Wat06] Martin Wattenberg. “Visual Exploration of Multivariate Graphs”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2006, pp. 811–819. DOI: 10.1145/1124772.1124891 (cit. on p. 63).

[Wei+20] Yating Wei, Honghui Mei, Ying Zhao, et al. “Evaluating Perceptual Bias During Geometric Scaling of Scatterplots”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (Jan. 2020), pp. 321–331. DOI: 10.1109/tvcg.2019.2934208 (cit. on pp. 48, 52, 167).

[Wei91] Mark Weiser. “The Computer for the 21st Century”. In: *Scientific American* 265.3 (1991), pp. 94–104. DOI: 10.1016/b978-0-08-051574-8.50097-2 (cit. on pp. 3, 20, 138).

[WHS16] Dirk Wenig, Florian Heller, and Johannes Schöning. “StrapMaps: Bringing Map-based Navigation to the Straps of Bags”. In: *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. New York, NY, USA: ACM, 2016, pp. 225–228. DOI: 10.1145/2968219.2993411 (cit. on p. 32).

[Wen+15] Dirk Wenig, Johannes Schöning, Brent Hecht, and Rainer Malaka. “StripeMaps: Improving Map-based Pedestrian Navigation for Smartwatches”. In: *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA: ACM, 2015, pp. 52–62. DOI: 10.1145/2785830.2785862 (cit. on p. 32).

[Wen+17] Dirk Wenig, Johannes Schöning, Alex Olwal, Mathias Oben, and Rainer Malaka. “WatchThru: Expanding Smartwatch Displays with Mid-air Visuals and Wrist-worn Augmented Reality”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 716–721. DOI: 10.1145/3025453.3025852 (cit. on p. 95).

[WW11] Daniel Wigdor and Dennis Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann Publishers Inc., Apr. 1, 2011 (cit. on pp. 17, 19).

[Wil+16] Gerard Wilkinson, Ahmed Kharrufa, Jonathan Hook, et al. "Expressy: Using a Wrist-worn Inertial Measurement Unit to Add Expressiveness to Touch-based Interactions". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2016, pp. 2832–2844. DOI: 10.1145/2858036.2858223 (cit. on p. 122).

[Won+13] Pak Chung Wong, P. Mackey, H. Foote, and R. May. "Visual Matrix Clustering of Social Networks". In: *IEEE Computer Graphics and Applications* 33.4 (July 2013), pp. 88–96. DOI: 10.1109/mcg.2013.66 (cit. on p. 64).

[Woź+14] Paweł Woźniak, Lars Lischke, Benjamin Schmidt, Shengdong Zhao, and Morten Fjeld. "Thaddeus: A Dual Device Interaction Space for Exploring Information Visualisation". In: *Proceedings of the ACM Nordic Conference on Human-Computer Interaction*. New York, NY, USA: ACM, 2014, pp. 41–50. DOI: 10.1145/2639189.2639237 (cit. on pp. 6, 23, 25, 35, 36, 38, 150, 164).

[Wu+21] Aoyu Wu, Wai Tong, Tim Dwyer, et al. "MobileVisFixer: Tailoring Web Visualizations for Mobile Phones Leveraging an Explainable Reinforcement Learning Framework". In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pp. 464–474. DOI: 10.1109/tvcg.2020.3030423 (cit. on pp. 46, 58).

[Wu+13] Yingcai Wu, Xiaotong Liu, Shixia Liu, and Kwan-Liu Ma. "ViSizer: A Visualization Resizing Framework". In: *IEEE Transactions on Visualization and Computer Graphics* 19.2 (Feb. 2013), pp. 278–290. DOI: 10.1109/tvcg.2012.114 (cit. on p. 52).

[WT08] Yingxin Wu and Masahiro Takatsuka. "Visualizing Multivariate Networks: A Hybrid Approach". In: *Proceedings of the IEEE Pacific Symposium on Visualization*. Piscataway, NJ, USA: IEEE, 2008, pp. 223–230. DOI: 10.1109/pacificvis.2008.4475480 (cit. on p. 63).

[Wyb+14] Michael Wybrow, Niklas Elmqvist, Jean-Daniel Fekete, et al. "Interaction in the Visualization of Multivariate Networks". In: *Multivariate Network Visualization*. Cham, Switzerland: Springer International Publishing, 2014, pp. 97–125. DOI: 10.1007/978-3-319-06793-3_6 (cit. on p. 66).

[Xia+18] Robert Xiao, Teng Cao, Ning Guo, et al. "LumiWatch: On-Arm Projected Graphics and Touch Input". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, 95:1–95:11. DOI: 10.1145/3173574.3173669 (cit. on p. 96).

[XL15] Cheng Xu and Kent Lyons. "Shimmering Smartwatches: Exploring the Smartwatch Design Space". In: *Proceedings of the International Conference on Tangible, Embedded, and Embodied Interaction*. New York, NY, USA: ACM, 2015, pp. 69–76. DOI: 10.1145/2677199.2680599 (cit. on p. 95).

[YW14] Jishuo Yang and Daniel Wigdor. “Panelrama: Enabling Easy Specification of Cross-device Web Applications”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 2783–2792. DOI: 10.1145/2556288.2557199 (cit. on p. 159).

[YEL10] Ji Soo Yi, Niklas Elmqvist, and Seungyoon Lee. “TimeMatrix: Analyzing Temporal Social Networks Using Interactive Matrix-Based Visualizations”. In: *International Journal of Human-Computer Interaction* 26.11-12 (Nov. 2010), pp. 1031–1051. DOI: 10.1080/10447318.2010.516722 (cit. on pp. 64–66).

[YKS07] Ji Soo Yi, Youn ah Kang, and John Stasko. “Toward a Deeper Understanding of the Role of Interaction in Information Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (Nov. 2007), pp. 1224–1231. DOI: 10.1109/tvcg.2007.70515 (cit. on pp. 4, 8, 26, 27, 29, 38, 119, 123, 124, 163, 168).

[Yi+17] Xin Yi, Chun Yu, Weijie Xu, Xiaojun Bi, and Yuanchun Shi. “COMPASS: Rotational Keyboard on Non-Touch Smartwatches”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, pp. 705–715. DOI: 10.1145/3025453.3025454 (cit. on p. 96).

[Yu+16] Lingyun Yu, Konstantinos Efstathiou, Petra Isenberg, and Tobias Isenberg. “CAST: Effective and Efficient User Interaction for Context-Aware Selection in 3D Particle Clouds”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), pp. 886–895. DOI: 10.1109/tvcg.2015.2467202 (cit. on p. 82).

[Zad18] Ulrich von Zadow. “Personalized Interaction with High-Resolution Wall Displays”. PhD thesis. Technische Universität Dresden, 2018 (cit. on p. 117).

[Zad+14] Ulrich von Zadow, Wolfgang Büschel, Ricardo Langner, and Raimund Dachselt. “SleeD: Using a Sleeve Display to Interact with Touch-sensitive Display Walls”. In: *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2014, pp. 129–138. DOI: 10.1145/2669485.2669507 (cit. on pp. 22, 23, 25, 35, 96, 120–122, 138, 164).

[Zad+16] Ulrich von Zadow, Patrick Reipschläger, Daniel Bösel, Anita Sellent, and Raimund Dachselt. “YouTouch! Low-Cost User Identification at an Interactive Display Wall”. In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2016, pp. 144–151. DOI: 10.1145/2909132.2909258 (cit. on pp. 139, 186).

[ZZD14] Emanuel Zgraggen, Robert Zeleznik, and Steven M. Drucker. “PanoramicData: Data Analysis through Pen & Touch”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 2112–2121. DOI: 10.1109/tvcg.2014.2346293 (cit. on p. 31).

[Zha+16] Cheng Zhang, AbdelKareem Bedri, Gabriel Reyes, et al. “TapSkin: Recognizing On-Skin Input for Smartwatches”. In: *Proceedings of the ACM Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2016, pp. 13–22. DOI: 10.1145/2992154.2992187 (cit. on p. 96).

[Zha+19] Yang Zhang, William Buxton, Ken Hinckley, et al. “Sensing Posture-Aware Pen+Touch Interaction on Tablets”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, 55:1–55:14. DOI: 10.1145/3290605.3300285 (cit. on p. 49).

[ZFÜ18] Kening Zhu, Morten Fjeld, and Ayça Ünlüer. “WristOrigami: Exploring Foldable Design for Multi-Display Smartwatch”. In: *Proceedings of the ACM Designing Interactive Systems Conference*. New York, NY, USA: ACM, 2018, pp. 1207–1218. DOI: 10.1145/3196709.3196713 (cit. on p. 95).

Webpages

[@Ais13] Gregor Aisch. *Chroma.js*. 2013. URL: https://gka.github.io/chroma.js/ (visited on Nov. 16, 2020) (cit. on p. 85).

[@Aut14] Automattic. *Socket.IO*. 2014. URL: https://socket.io/index.html (visited on Jan. 13, 2021) (cit. on p. 191).

[@Bal11] Baltimore Police Department. *BPD Part 1 Victim Based Crime Data*. 2011. URL: https://data.baltimorecity.gov/Public-Safety/BPD-Part-1-Victim-Based-Crime-Data/wsfq-mvij (visited on June 17, 2020) (cit. on pp. 119, 172).

[@BS18] Dominikus Baur and Moritz Stefaner. *Everything Except the Chart: Responsiveness + Mobile Support*. IEEE VIS Tutorial. 2018. URL: http://webvis-do.minik.us/#/responsiveness (visited on Sept. 7, 2020) (cit. on pp. 51, 53, 56, 57).

[@Bre19] Nadieh Bremer. *Techniques for Data Visualization on both Mobile & Desktop*. Blog Post. 2019. URL: https://tinyurl.com/bremer-responsive (visited on Sept. 7, 2020) (cit. on pp. 52, 53, 57).

[@Bre15] Nadieh Bremer. *Using a D3 Voronoi Grid to Improve a Chart’s Interactive Experience*. 2015. URL: https://www.visualcinnamon.com/2015/07/voronoi.html (visited on Nov. 19, 2020) (cit. on p. 57).

[@BHT13] Cynthia Brewer, Mark Harrower, and The Pennsylvania State University. *ColorBrewer: Color Advice for Maps*. 2013. URL: https://colorbrewer2.org/ (visited on June 17, 2020) (cit. on p. 154).

[@Cam19] Jorge Camões. *Efficient Screen Real Estate Management: Improving Data Visualization for Small Screens*. Presentation at the European Commission Conference on New Techniques & Technologies for Statistics (NTTS). 2019. URL: https://tinyurl.com/camoes-ntts (visited on Sept. 8, 2020) (cit. on p. 55).

[@Cot19] Andy Cotgreave. *5 Questions to Ask when Designing a Mobile Dashboard*. Blog post. 2019. URL: https://tinyurl.com/cotgreave-tableau (visited on Sept. 8, 2020) (cit. on p. 58).

[@Esp16] Espressif Systems Co., Ltd. *ESP8266 Overview | Espressif Systems*. 2016. URL: https://www.espressif.com/en/products/hardware/esp8266ex/overview (visited on Dec. 12, 2019) (cit. on p. 107).

[@Fos19] Fossil Group, Inc. *Hybrid Smartwatch HR Collider*. 2019. URL: https://www.fossil.com/en-us/products/hybrid-smartwatch-hr-collider-tan-leather/FTW7007.html (visited on Jan. 27, 2021) (cit. on p. 95).

[@Gla14] Glance Team. *Kickstarter Campagain: Glance: World's first smart accessory for your watch*. 2014. URL: https://www.kickstarter.com/projects/1742184757/glance-worlds-first-smart-accessory-for-your-watch (visited on Dec. 12, 2019) (cit. on p. 95).

[@Gol17] Russell Goldberg. *Responsive Scrollytelling*. *The Pudding*. 2017. URL: https://tinyurl.com/pudding-responsive (visited on Sept. 8, 2020) (cit. on pp. 56, 58).

[@Gro17] Lena Groeger. *Data GIFs*. NICAR Tutorial. 2017. URL: http://lenagroeger.com/datagifs/#/ (visited on Sept. 8, 2020) (cit. on p. 56).

[@Hor+] Tom Horak, Andreas Mathisen, Clemens N. Klokmose, Raimund Dachselt, and Niklas Elmqvist. *Vistribute Prototype*. Source Code. URL: https://github.com/tomhorak21/vistribute (visited on Dec. 18, 2020) (cit. on p. 168).

[@Kai14] Kairos Team. *Indiegogo Campagain: Kairos T-Band*. 2014. URL: https://igg.me/at/kairostband/x%5C#/ (visited on Dec. 12, 2019) (cit. on p. 95).

[@KHD] Konstantin Klamka, Tom Horak, and Raimund Dachselt. *Watch+Strap Research Platform*. Project Page. URL: https://imld.de/watch-strap (visited on Sept. 9, 2020) (cit. on pp. 106, 188).

[@Lbe18] L!ber8 Technology. *TAGO ARC*. 2018. URL: https://liber8tech.com (visited on Dec. 12, 2019) (cit. on p. 95).

[@LHD] Ricardo Langner, Tom Horak, and Raimund Dachselt. *VisTiles Prototype*. Source Code. URL: https://github.com/imldresden/vistiles (visited on Sept. 11, 2020) (cit. on p. 153).

[@Len20] Lenovo. *ThinkPad X1 Fold*. 2020. URL: https://www.lenovo.com/us/en/thinkpad-x1-fold (visited on Jan. 14, 2021) (cit. on p. 190).

[@LG19] LG. *LG Rollable OLED TV R*. 2019. URL: https://www.lg.com/global/lg-signature/rollable-oled-tv-r (visited on Jan. 14, 2021) (cit. on p. 190).

[@LG20] LG. *LG WING*. 2020. URL: https://www.lg.com/global/lgwing (visited on Jan. 14, 2021) (cit. on p. 190).

[@Mat14] Materialize. *Documentation*. 2014. URL: http://materializecss.com (visited on June 17, 2020) (cit. on p. 153).

[@Mic16] Microsoft. *Microsoft HoloLens*. 2016. URL: https://www.microsoft.com/en-us/hololens (visited on Nov. 20, 2020) (cit. on pp. 59, 190).

[@Mic14] Microsoft. *Microsoft Power BI*. 2014. URL: https://powerbi.microsoft.com/ (visited on Dec. 3, 2020) (cit. on p. 28).

[@Mic20] Microsoft. *New Surface Duo – Dual-Screen Mobile Productivity, Do One Better*. 2020. URL: https://www.microsoft.com/en-us/surface/devices/surface-duo (visited on Nov. 20, 2020) (cit. on pp. 59, 190).

[@Mom14] Momentum Labs LLC. *Kickstarter Campagain: Moment Smartwatch: world's first wrap around smart watch*. 2014. URL: http://kck.st/VebX7r (visited on Dec. 12, 2019) (cit. on p. 95).

[@Nat20] NaturalPoint, Inc. *OptiTrack - Motion Capture Systems*. 2020. URL: https://optitrack.com/ (visited on June 17, 2020) (cit. on p. 152).

[@Ope09] OpenJS Foundation. *Node.js*. 2009. URL: http://nodejs.org/ (visited on June 17, 2020) (cit. on p. 153).

[@Sam19] Samsung Electronics. *Reshape Your Future with the Samsung Galaxy Fold*. 2019. URL: https://www.samsung.com/us/mobile/galaxy-fold/ (visited on Nov. 20, 2020) (cit. on pp. 59, 190).

[@Sam15] Samsung Electronics. *Samsung Gear S2*. 2015. URL: https://www.samsung.com/global/galaxy/gear-s2/ (visited on Dec. 12, 2019) (cit. on pp. 106, 131, 134).

[@Sam16] Samsung Electronics. *Samsung Gear S3*. 2016. URL: https://www.samsung.com/global/galaxy/gear-s3/ (visited on Dec. 12, 2019) (cit. on p. 106).

[@Sem13] Semantic UI LLC. *Semantic UI*. 2013. URL: https://semantic-ui.com/ (visited on Nov. 16, 2020) (cit. on p. 85).

[@Son18] Sony. *Sony FES Watch U E-paper Fashion Watch*. 2018. URL: https://www.sony.co.uk/electronics/fes-watch-u-collection/fes-wa1 (visited on Dec. 12, 2019) (cit. on p. 95).

[@Tab17a] Tableau Software, LLC. *Identify best-performing products and predict changes in trends. Customer Surveys Results*. 2017. URL: https://www.tableau.com/solutions/workbook/track-sales-interactive-dashboard (visited on June 17, 2020) (cit. on p. 160).

[@Tab17b] Tableau Software, LLC. *Monitor sales performance and distance to quota. Sales Summary*. 2017. URL: https://www.tableau.com/solutions/workbook/track-sales-interactive-dashboard (visited on June 17, 2020) (cit. on p. 160).

[@Tab03] Tableau Software, LLC. *Tableau Desktop*. 2003. URL: https://www.tableau.com/products/desktop (visited on Dec. 3, 2020) (cit. on p. 28).

[@The20] The World Bank Group. *World Development Indicators | DataBank*. 2020. URL: https://databank.worldbank.org/source/world-development-indicators (visited on June 17, 2020) (cit. on p. 154).

[@TBH20] Christian Tominski, Philip Berger, and Tom Horak. *Responsive Matrix Cells Software Prototype*. 2020. URL: https://vcg.informatik.uni-rostock.de/~ct/software/RMC (visited on Sept. 7, 2020) (cit. on pp. 85, 86).