Effects of Different Display Form Factors on InfoVis Applications: Exploring Selection Management and Brushing and Linking for Mobile Cross-Device Interaction

Tamara Flemisch

April 17, 2018

Technische Universität Dresden

Faculty of Computer Science
Institute of Software and Multimedia Technology
Interactive Media Lab Dresden (IMLD)

Effects of Different Display Form Factors on InfoVis Applications: Exploring Selection Management and Brushing and Linking for Mobile Cross-Device Interaction

Tamara Flemisch

1. Reviewer Prof. Dr.-Ing. Raimund Dachselt
Faculty of Computer Science
Technische Universität Dresden

2. Reviewer Dr.-Ing. Annett Mitschick
Faculty of Computer Science
Technische Universität Dresden

Supervisors Dipl.-Ing. Ricardo Langner and Prof. Dr.-Ing. Raimund Dachselt

April 17, 2018

Tamara Flemisch
Effects of Different Display Form Factors on InfoVis Applications: Exploring Selection Management and Brushing and Linking for Mobile Cross-Device Interaction
Documentation, April 17, 2018
Reviewers: Prof. Dr.-Ing. Raimund Dachselt and Dr.-Ing. Annett Mitschick
Supervisors: Dipl.-Ing. Ricardo Langner and Prof. Dr.-Ing. Raimund Dachselt

Technische Universität Dresden
Interactive Media Lab Dresden (IMLD)
Institute of Software and Multimedia Technology
Faculty of Computer Science
Nöthnitzer Str. 46
01062 and Dresden

Abstract

This thesis examines how to manage selections and use brushing and linking with multiple coordinated mobile devices. We discuss thoughts for a conceptual framework for selection management in multi-device environments. We then present a concept for managing selections and supporting brushing and linking for co-located mobile devices. Finally, we are providing an overview of and the goals for our proof-of-concept prototype.

More and more mobile devices are used for visualization. However, it is still an open question how to adjust common interaction techniques, such as brushing and linking, for mobile devices. Furthermore, it has not been addressed how to manage the selections that are created through brushing. We explore how brushing and linking can be used in a setting with multiple, co-located mobile devices and how to manage its selections.

Zusammenfassung

Diese Arbeit untersucht wie man mit mehreren, miteinander koordinierten Mobilgeräten Selektionen verwalten und Brushing und Linking unterstützen kann. Im Rahmen der Arbeit wird ein konzeptionelles Framework zur Selektionsverwaltung für mehrere Geräte angedacht und ein Konzept zur Selektionsverwaltung vorstellt, das Brushing und Linking für Mobilgeräte unterstützt. Des Weiteren werden Ziele für den Prototypen erörtert, die zur Umsetzung das Konzept beitragen.

Mobile Geräte werden immer häufiger für Informationsvisualisierungen verwendet. Jedoch wurde bisher noch nicht untersucht, wie gängige Interaktionstechniken, wie Brushing und Linking, für diese Geräte anpassen werden müssen. Während des Brushings, werden zusätzlich Selektionen erstellt, für die es bislang keine Art der Verwaltung für Mobilgeräte gibt. In dieser Arbeit wird untersucht wie man Brushing und Linking mit mehreren Mobilgeräten unterstützen kann und Selektionen gleichzeitig verwalten kann.

Acknowledgement

First and foremost my sincere thanks go to my supervisor Ricardo Langner, who has supported me throughout my thesis with advice, knowledge, and patience. I am profoundly grateful for the many extensive, fruitful discussions that shaped this thesis. I consider myself very lucky to have had him as a supervisor.

My sincere thanks go to Prof. Raimund Dachselt who supported me along the road to this thesis and made it possible for me to study and work in a great environment. I cannot thank him enough for creating the opportunity for me to spend time studying in Canada.

Furthermore, I would like to thank all members of the Interactive Media Lab Dresden who were always supportive and helped whenever there was a problem. Special thanks go to Ulrike Kister to took the time to give me valuable feedback.

I would like to thank Christin Engel and Meinhardt Branig who were there when I was in need of a coffee break that usually resulted in good discussions and valuable input for my thesis.

Heartfelt thanks go to the people from the Kampfsport Akademie Dresden, especially Franziska Szokoli, who provided me with a pleasant change and new strength for my thesis.

I am extremely grateful to my parents who always supported and encouraged me through the years of my study without doubt or question.

Finally, this thesis would have been impossible without the help of David Ledo and Laura Zepner. First, I would like to thank David Ledo for his invaluable advice on academic writing in English. Second, I must express my deep gratitude for their constant encouragement, their persistence, and overall support that made this thesis possible.

Contents

1 Introduction

1.1 Contributions

This thesis examines how to manage selections and use brushing and linking with multiple coordinated mobile devices. First, we[1] propose initial thoughts for a conceptual framework for selection management in multi-device environments. Second, we present a concept for managing selections and supporting brushing and linking for co-located mobile devices. Third, we are providing an overview of and the goals for our proof-of-concept prototype.

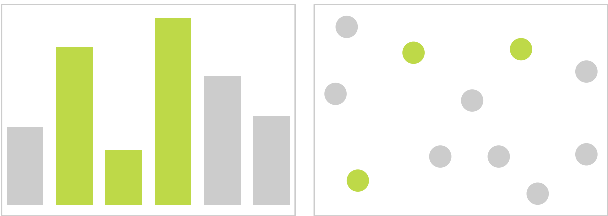


Fig. 1.1 An example for brushing and linking of a bar chart and a scatter plot. The brush are the selected data elements and the link is represented by using the same color for the corresponding data elements in both views.

1.2 Motivation and Background

Information visualization (InfoVis) is a well established instrument for analyzing data. Over the last decade, visualizations have slowly conquered our personal lives due to the rise of fitness trackers, the quantified self movement [WK], and personal visualization [Hua+15]. InfoVis has reached most settings, from personal and educational settings to business settings [Eis08]. Given that information influences us in almost every area of our lives, InfoVis will continue to play an important role in the future.

Many visualization systems, including commercial products (e.g., Tableau [Tab18]) feature visualizations on a single screen, with the expectation that a single user will work on a computing system to view and analyze the data. However, new technologies, such as large interactive displays or mobile devices, make their way into the area of InfoVis [Rob+14; Lee+12]. While large displays, walls and tables, have demonstrated promise for InfoVis tasks [Elm+11; And+11; Bal+07], these devices

[1] Although I am the primary investigator and writer, I am going to use "we" throughout this thesis to acknowledge my supervisor's support and contribution.

are expensive and not commonplace [Wal+17]. In contrast, one can integrate everyday mobile devices into our environments [RK14; RK13; SS14]. Mobile devices come in various form factors, such as phones and tablets.

However, due to their mobility and therefore small size, they offer a limited amount of screen real estate. One way to tackle this challenge is by combining the mobile devices [Chu+14]. Piazza et al. [Pia+13] suggest to unite phones and tablets since they complement each other in most aspects. Combining different aspects to balance out weaknesses, has been done for visualizations as well [Kei02]. *Coordinated and multiple views (CMVs)* propose that it is beneficial for people to interact with their data through multiple, different representations [Rob07]. Thus, one visualization technique is used to balance out the weakness of another visualization technique. First attempts have been made to combine CMVs and mobile devices. Work by Langner et al. [Lan+17] distributes views provided by a CMV application among multiple mobile devices.

When moving from a single-device setting to a multi-device environment, common established interaction techniques might no longer work or need adjustment. An example for a technique that needs to be adjusted is brushing and linking, one of the most commonly implemented interaction techniques in CMV applications [Rob07]. Interactively selecting a set of data items is called *brushing*. Making a connection, through showing links or highlighting, to items in other views that represent the brushed data points is *linking* [Voi02]. An example for brushing and linking is illustrated in Fig. 1.1. When translated to multi-device settings, brushing and linking becomes a major challenge as views are separated. Visually linking data points is hindered by the distance between the devices. Additionally, users are often unaware of the surrounding devices and the opportunities they provide [Mar+12].

When brushing elements in a visualization, users create selections which are a fundamental part of interactive visualizations [SS16b]. We define selections as a collection of data points that are of interest to the interacting person. Current selection techniques often address how to select data elements in visualizations [SS16b; Koy+17]. However, only few attend to how they can be used to further assist the exploration process, for instance through applying operations or using them to filter data points.

1.3 Goals

VisTiles is a starting point for supporting visualizations through the use of spatial interactions and mobile devices. However, it only supports a basic implementation of brushing and linking and no selection management. To further enhance brushing and linking in this setting, there needs to be a consideration of:

Awareness Users need to be aware of their surrounding devices and the actions they offer in a multi-device environment.

Selection management It is important to consider selections as a powerful tool and give the user the option to use them accordingly.

1.4 Research Questions

In the presented thesis, we want to address the following research questions in the context of multi-device environments with mobile devices:

- How might we create awareness using brushing and linking?
- How might we support storing, combining, managing, and interacting with selections?

1.5 Thesis Overview

This thesis unfolds as follows: first, we describe the background literature regarding mobile devices in InfoVis, cross-device interaction, brushing and linking, and selection management in Chapter 2. In Chapter 3, we discuss initial thoughts on a conceptual framework for the use of selections in a multi-device environments and propose a concept for managing selections for co-located mobile devices. Chapter 4 defines goals for our prototype that is going to work as a proof-of-concept in the future. Finally, we conclude with Chapter 5 by discussing the contributions and limitations of the suggested concept.

2 Related Work

This thesis builds upon prior research in areas of information visualization and human-computer interaction. We explore related works that deal with different devices and screen real estate for visualizations. We give an overview over the interaction with CMVs and how they are used in non-desktop environments. We examine brushing and linking, especially for multiple or mobile devices and how selection management ties into these aspects.

Table 2.1 shows relevant papers that present a system or techniques related to the topic of this work. The columns represent the areas of human-computer interaction and InfoVis are connected to this thesis. We sorted this table by visual similarity using Bertifier [Per+14]. Afterwards, we rearranged the groups of similar papers.

2.1 Mobile Devices for InfoVis and their Screen Real Estate Issues

In this section, we explain how natural interaction and mobile devices impacts InfoVis. We give an overview over opportunities and challenges mobile devices create, mostly due to their small form factor.

2.1.1 Natural Interaction for InfoVis

Touch interaction became widely popular and ubiquitous with the introduction of multi-touch displays and it has affected InfoVis as well. Since InfoVis benefits from natural interaction, lots of research focussed on incorporating it into InfoVis [Elm+11; Lee+12; Rob+14]. Natural interaction, that includes touch input, pen input, gestures, etc., enable the user to directly access data and UI elements [Ise+13].

As Lee et al. [Lee+12] stated, it is challenging to translate traditional desktop interaction and operations to a multi-touch interface. One common issue when directly interacting with surfaces is the fat finger problem [Sie+05]. This is crucial for InfoVis applications because they tend to use small objects, such as visual data items, and widgets [SS14]. To adjust to visualizations beyond the desktop, Jansen & Dragicevic [JD13] propose a modification of the InfoVis pipeline. This adjustment includes the process beginning with physical presentation and leading to the insights

■ This paper relates to this field.
◩ This paper relates partially to this field.
□ This paper does not relate or not enough to this field.

NAME	REF.	YEAR	MOBILE DEVICES	CROSS-DEVICE INTERACTION	CMVS	VISUALIZATIONS	BRUSHING AND LINKING	SELECTION MANAGEMENT	
WHEN DAVID MEETS GOLIATH	Hor+18	2018	■	■	■	■	■	■	CMVS + MOBILE DEVICES
VISTILES	Lan+17	2017	■	■	■	■	■		
DESIGNING CMVS FOR TABLETS	SS16a	2016	■		■	■	■	◩	
THADDEUS	Wo14	2014	■	■		■	◩	◩	
VISPORTER	Chu+14	2014	■	■	■	■		◩	
SPATIALLY AWARE OR SPATIALLY AGNOSTIC?	Rä+15	2015	■	■			■		MULTIPLE DEVICES + MOBILE DEVICES
HOLY SMARTPHONES AND TABLETS, BATMAN!	Pia+13	2013	■	■			■		
GRADUAL ENGAGEMENT	Mar+12	2012	■	■			■		
RELATE	Gel+09	2009	■	■			■		
SAVIL	CN17	2017	■	■			■		
SLEED	Zad+14	2014	■	■			◩		
CONDUCTOR	HW14	2014	■	■			◩		
HUDDLELAMP	Rä+14	2014	■	■					
PASS-THEM-AROUND	Luc+11	2011	■	■					
CUBEQUERY	Lan+14	2014	■	■					
PROXEMIC-AWARE CONTROLS	Led+15	2015	■	■					
KINETICA	RK14	2014	■			■			MOBILE VISUALIZATIONS
WEDGE	Gus+08	2008	■			■			
TOUCHVIZ	Dru+13	2013	■			■			
TOUCHWAVE	Bau+12	2012	■			■			
HALO	BR03	2003	■			■			
DESIGNING SCATTERPLOTS FOR TABLETS	SS14	2014	■			■		◩	
MONDRIAN	The02	2002			■	■	■	■	TRADITIONAL CMVS
BRUSHING FUNCTION GRAPHS	Kon+06	2006			■	■	■	■	
BRUSHING ATTRIBUTE CLOUDS	Jä+08	2008			■	■	■	■	
ROLLING THE DICE	Elm+08	2008			■	■	■	■	
CITY'O'SCOPE	BG03	2003			■	■	■	■	
CONNECTED CHARTS	VM12	2012			■	■	■	◩	
VISUALIZATION MOSAICS	ME13	2013			■	■	■	◩	
MYBRUSH	Koy+17	2017			■	■	■	◩	
VISLINK	CC07	2007			■	■	■	◩	
LARK	Tob+09	2009			■	■	■		
MATCHES, MISMATCHES, AND METHODS	Bre+16	2016			■	■	■		
BRUSHING SCATTERPLOTS	BC87	1987			■	■	■		
FLEXIBLE LINKED AXES	CW11	2011			◩	■	■	◩	
PROBING PROJECTIONS	Sta+16	2016				■	■	■	

Tab. 2.1 Relevant works and how they relate to certain fields. The rows of the table are papers or systems. The columns are the fields we consider for this thesis. The basis for this table was done with Bertifier [Per+14].

gained by the user. A physical presentation can be any object, such as a piece of paper or a LCD display.

2.1.2 Using Mobile Devices for InfoVis

As already mentioned in Chapter 1, mobile devices, such as smartphones, tablets, and smartwatches, are part of our everyday life. While discovering multi-touch, InfoVis research inevitably tapped into the area of mobile devices. Their size, their input, and output options create opportunities and challenges alike, such as the changing physical environment or the user's attention span [Chi06]. Furthermore, mobile visualizations need to adjust to the limited screen real estate [Rob+14].

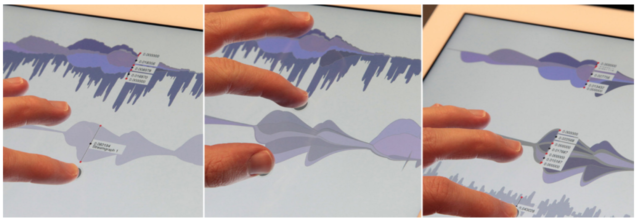


Fig. 2.1 Interacting with stacked graphs using TouchWave by Baur et al. [Bau+12].

Drucker et al. [Dru+13] stated that when interfaces are designed with the device modalities in mind, relying on natural interaction leads to a better task performance. However, creating one set of touch interactions for general purpose, is not an option because of the amount of different visualizations [Bau+12]. Therefore, Baur et al. [Bau+12] created a special set for interacting with stacked graphs, as seen in Fig. 2.1.

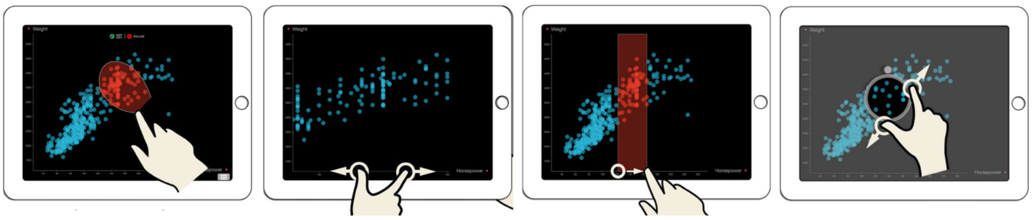


Fig. 2.2 Interaction techniques for scatter plots on tablets [SS14].

Sadana & Stasko [SS14] consider two types of interaction for their scatter plot implementation for mobile devices: WIMP-derived and gesture-driven interaction. WIMP elements occupy a great amount of space and feel "less than optimal for these types of devices" [SS14]. They suggest using gesture-driven interaction for mobile devices to facilitate direct access to UI elements and for freeing up screen space. Figure 2.2 shows some of the implemented gesture-driven interactions for their scatter plot application for tablets.

Kinetica [RK14] and TouchViz [RK13] by Rzeszotarki & Kittur are using a multi-variate scatter plot visualization on tablets similar to Sadana & Stasko's [SS14]

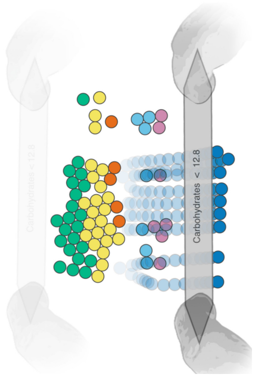


Fig. 2.3 Using a semi-permeable filter in Kinetica by Rzeszotarki & Kittur [RK14].

approach. However, they base their interaction concept on physics, such as force and gravity. This physics-driven interaction, as illustrated in Fig. 2.3, facilitates playfulness and incorporates the tablets' form factor. They decided to use a tablet because it "occupied the physical space of the user, could be twisted and turned, and was responsive to touch" [RK14].

Visualization research has found some solutions on how to tackle smaller form factors, such as off-screen visualization which has been explored by Baudisch & Rosenholtz [BR03] (see Fig. 2.4) and Gustafson et al. [Gus+08] among others. Another way of dealing with the space issues of mobile devices in InfoVis is combining multiple devices.

Fig. 2.4 Using Halo by Baudisch & Rosenholtz [BR03] as an off-screen visualization technique.

2.1.3 Combining Multiple Devices for InfoVis

Combining devices and cross-device interaction leads to new possibilities for visualization: the screen real estate can be increased, data can be spread across devices, and visualization tasks can be distributed across independent users. [Ise+13]

Jokela et al. [Jok+15] state that connecting devices seems simple but is a complex procedure. The devices need to discover other devices first and second, the users need to clarify which devices are supposed to be included in the group. Pass-them-around by Lucero et al. [Luc+11] is one example for dealing with the space issue of mobile devices by combining them. They use side-by-side positioning and user's gestures to connect screens and extend them. However, if the devices are not next to each other, the user might lose the overview. Therefore, maintaining awareness for the devices in the setting is an important goal. Marquart et al. [Mar+12] use proxemics and visual representations of devices on the screen's edges to create awareness. Expanding further on awareness, Ledo et al. [Led+15] adjust an application's features and functionality depending on the user's spatial proximity to the devices. The closer the user is to a device, the more detailed is the functionality of the devices.

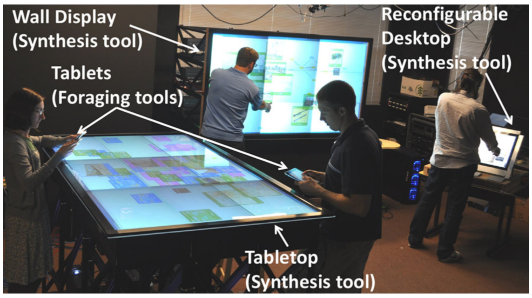


Fig. 2.5 A setting of multiple devices for visual analysis in VisPorter by Chung et al. [Chu+14].

How devices, especially tablets and smartphones, can be combined is explored by Piazza et al. [Pia+13]. They analyze the differences in interaction between phones and tablets and reveal that their attributes in input, output, and interaction style are complementary. Thus, they suggest to combine them instead of isolating them, such as using a phone as a secondary, controlling device and a tablet as the primary device. Conductor by Hamilton & Wigdor [HW14] addresses the problem of sharing information between devices via visual representations around the screens edges, similar to the ones found by Marquart et al. [Mar+12]. Additionally, they offer the user feedback on multiple devices at a time.

Isenberg et al. [Ise+13] mention that for combining devices the relationships between devices is of importance. This inevitably leads to spatial awareness of a device. Rädle et al. [Rä+15] argue that users associate cross-device interaction with spatial awareness, i.e., they assume a device is spatially aware if it interacts with other devices. They advocate for using spatially aware interactions with care but stress that they can lead to less mental demand and effort for the user. HuddleLamp [Rä+14] implements spatially aware and spatially agnostic techniques for a collaborative setting with multiple mobile devices.

Researchers started to combine multiple devices for the purpose of visualizing data. Isenberg et al. [Ise+13] discuss questions about roles of visualizations, the relationships between them, how their data is coordinated, and how the design has to be adjusted for different screens. Chung et al. [Chu+15] call this synthesis of multiple devices a display ecology. Similar to Isenberg et al. [Ise+13], they offer key considerations for display ecologies: how displays are combined, how visualizations and data can be moved among displays, how users can associate corresponding, scattered information, and how different devices can be added or removed.

Fig. 2.6 Interacting with a SleeD by von Zadow et al. [Zad+14] on an interactive display wall.

Some works have already tackled these questions and developed display ecologies for InfoVis. VisPorter [Chu+14] is a text analytics tool for multiple displays that include tabletops, tablets, smartphones, and wall-sized displays. Figure 2.5 shows an example of a VisPorter setup. SleeD [Zad+14], shown in Fig. 2.6, combines a wall-sized displays with an arm-mounted display which gives the user access to the controls. Furthermore, CubeQuery [Lan+14] uses multiple small devices to form database queries by using their relative spatial information. Thaddeus by Woźniak et al. [Woz+14] is a system that combines a smartphone and a tablet to explore visualizations. They use spatial movement to control a multi device setup, as illustrated in Fig. 2.7.

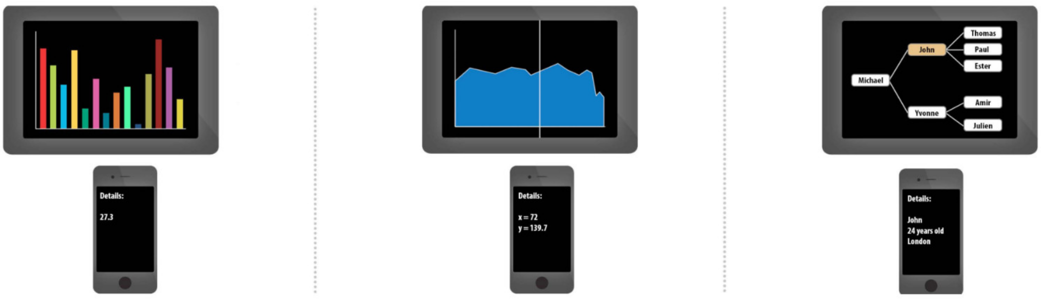


Fig. 2.7 Scenarios in Thaddeus by Wońiak et al. [Woz+14] that use spatial input of a phone to control a visualization on a tablet.

Overall, Blumenstein et al. [Blu+15] summarizes that for collaborative cross-device data visualization, we need to consider three important parts: Multi user, multi screen, and multi device.

2.2 Interacting with CMVs

CMVs have played an important role in InfoVis since they first appeared in the late 1990s [Rob98]. Lots of research has focussed on them over the years and they are used in popular commercial applications, such as Tableau [Tab18]. CMV stands for *coordinated & multiple views* and represents a concept for InfoVis applications where the users are presented with multiple visualization views, such as scatter plots, bar charts, or parallel coordinate plots, that are combined and coupled. It relies on the idea that “users understand their data better if they interact with the presented information and view it through different representations” [Rob07].

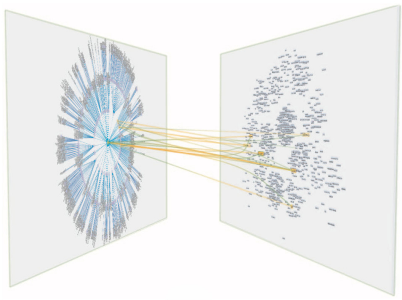


Fig. 2.8 Showing the link between two 2D visualizations in 3D in VisLink by Collins & Carpendale [CC07].

Wang Baldonado et al. [WB+00] define guidelines for creating CMV applications. In addition, Roberts presents several interaction techniques that are commonly implemented by CMV applications: Changing data processing, filtering data, selecting what is displayed, changing the mapping, navigating information, and changing the placement of the windows.

Apart from the more traditional CMVs, there are some unique takes to them. For instance, VisLink [CC07] takes 2D visualizations, arranges them in a 3D space, and links corresponding data elements in the visualizations, as illustrated in Fig. 2.8. Javed & Elmqvist [JE12] review existing work in composite visualizations and create a design space for describing the relationship between two visualizations in particular.

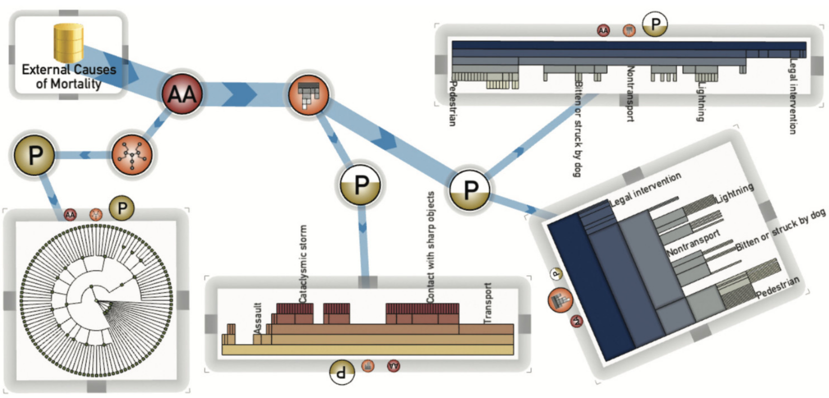


Fig. 2.9 Using four views in combination with the meta-visualization in Lark by Tobiasz et al. [Tob+09].

Since CMVs first emerged, the field has been thoroughly explored and some considered it a ‘solved problem’ [Rob07]. However, there are still more recent works [ME13; Bre+16] and new challenges [Rob07]. One of the new challenges for CMVs is to

Fig. 2.10 Sadana & Stasko's [SS16a] CMV implementation for tablets.

apply them to non-desktop devices, such as large interactive screens or mobile devices. Tobiasz et al. [Tob+09] developed a system called Lark that brings CMVs to a tabletop. The views are shown on the large interactive display and can be resized and rearranged. The additional space provided by the tabletop display is used to visualize a representation of the InfoVis pipeline to enable the user to always manipulate the activation of certain views. An example setting for Lark is shown in Fig. 2.9.

Fig. 2.11 Combining multiple mobile devices using VisTiles [Lan+17].

Fig. 2.12 Using a smartwatch in combination with a large display wall [Hor+18].

Sadana & Stasko [SS16a] focus on smaller devices, i.e., tablets. They created a CMV application that supports up to three different charts and is specifically designed for the form factor of a small device. For instance, they aim to maximize the display space that is used for data representation in contrast to controls, which can be seen in Fig. 2.10. Rather than combining the views on one tablet, Langner et al. [Lan+17] distribute their views among multiple devices in VisTiles. This helps to overcome the problem of screen real estate, that mobile devices tend to have and offers opportunities for new interaction techniques, similar to Thaddeus [Woz+14].

The VisTiles system is shown in Fig. 2.11. Furthermore, Horak et al. [Hor+18] use a large interactive display to represent a CMV setting. They combine the large screen with a smartwatch to explore the benefits of an additional small device for data exploration. An exemplary setup is displayed in Fig. 2.12.

2.3 Brushing and Linking

In this section, we give an overview on existing brushing and linking techniques, starting with a definition and some conventional techniques. Moving further, we explore how brushing and linking is translated to other devices. Finally, we examine how applications that focus on brushing and linking deal with selection management.

2.3.1 Conventional Brushing and Linking

Hearst [Hea99] defines brushing and linking as "the connecting of two or more views of the same data, such that a change to the representation in one view affects the representation in the other views as well." Voigt [Voi02] states that interactively selecting a set of data items is called *brushing*, whereas making a connection, through showing links or highlighting, to items in other views that represent the brushed data points is *linking*.

Brushing and linking can be used to combine visualization techniques and to balance out the disadvantages of single visualizations [Kei02]. Therefore, it is not surprising that it is one of the most commonly implemented interaction techniques in CMV applications [Rob07]. It is also part of popular toolkits, such as D3 [Bos+11] and Tableau [Tab18]. Becker & Cleveland [BC87] introduced brushing and linking and used it for scatter plot matrices in 1987. Various takes on brushing and linking have been developed over the years. Koytek et al. [Koy+17] offer a collection of works that deal with brushing and linking in the context of InfoVis. They explain, that there are different versions of brushing and linking, such as highlighting elements in other views by hovering over data points. Drawing a shape, such as a rectangle or a lasso, to select data points is considered brushing and linking as well.

The most common implementations of brushing and linking do not explicitly depict the links. They are rather displayed in highlighting corresponding data points (e.g., [BC87; Elm+08; Bre+16]). However, some researchers have explored the benefit of using explicit links. VisLink [CC07], shown in Fig. 2.8, connects two 2D visualizations in a 3D space by drawing the connections between the data points as lines. Depending on the amount of links, they are combining the lines into edge bundles. Explicit links are also used in ConnectedCharts by Viau & McGuffin [VM12]. They facilitate creating new charts by combining visualizations. In contrast to

VisLink [CC07], they use multiple 2D visualizations and arrange them inside a 2D space. As mentioned in Section 2.2, Lark [Tob+09] takes a less automatic approach. It allows the users to choose how the information is linked and where in the InfoVis pipeline it is linked by interacting with a meta-visualization (see Fig. 2.9).

Claessen & Wijk [CW11] establish flexible linked axes. Their system does not connect different views but different axes by brushing and linking. They support multiple axes to be connected to create new charts, as seen in Fig. 2.13. They further support selecting a part of the axes to highlight them through their links. MyBrush by Koytek et al. [Koy+17] deconstructs brushing and linking into three components: source, link, and target. They are incorporating personal agency to offer the user to customize all three parts of the interaction. An example scenario can be seen in Fig. 2.14. They further define a design space for brushing and linking, that includes the temporality of a brush, if a system supports group selection, and if logical combinations can be executed among other aspects. Further newer applications of brushing and linking include Show me the invisible [Gey+14], Domino [Gra+14], and Entourage [Lex+13].

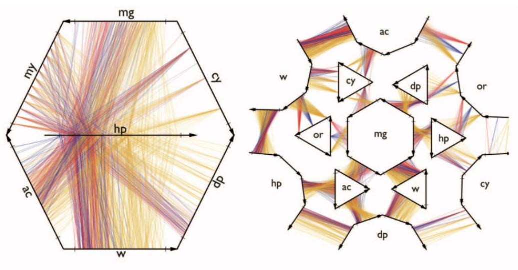


Fig. 2.13 Linking multiple axes to each other using flexible linked axes by Claessen & Wijk [CW11].

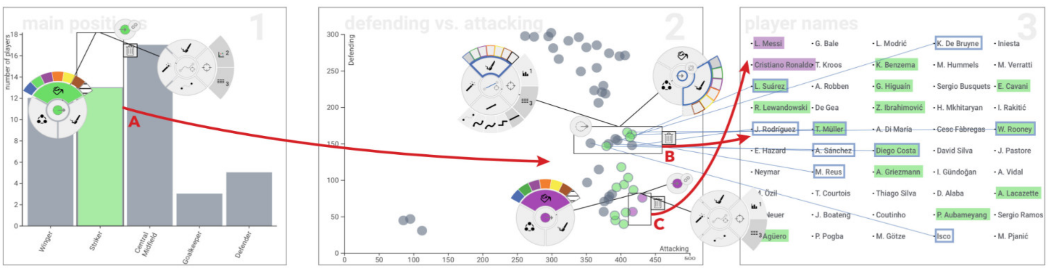


Fig. 2.14 Using deconstructed brushing and linking to explore multiple visualizations [Koy+17].

2.3.2 Brushing and Linking with Multiple or Mobile Devices

There are only very few works that deal with brushing and linking on mobile devices or using multiple devices. However, systems in the area of cross-device interaction exist that tackle the challenge of linking devices. We examine the related works for brushing and linking in InfoVis and in cross-device interaction.

Usage in InfoVis

We only know of three works that relate to brushing and linking on mobile devices in the InfoVis community. There are some more works that relate slightly to brushing and linking and deal with a combination of devices [Zad+14; Woz+14]. However, they either combine the mobile device with large screens [Zad+14] or they do not support brushing and linking in multiple visualization views [Woz+14].

Sadana & Stasko [SS16a] present a CMV application for tablet computers, that allows brushing and linking over three different views. They support one tablet and try to use the provided space to its full potential (see Fig. 2.10). However, they do not explicitly focus on brushing and linking. They incorporate it in their CMV application and it works “as one would typically expect for visualization applications” [SS16a]. When David Meets Goliath by Horak et al. [Hor+18] uses a large interactive display in combination with a smartwatch to interact with CMVs, as shown in Fig. 2.12. In this context, they implemented brushing and linking without focussing on it. It is included in their CMV application on the large display. Using both, mobile and multiple devices, for brushing and linking, is done by VisTiles by Langner et al. [Lan+17]. They combine multiple mobile devices to interact with a CMV application that is distributed among the screens. Brushing and linking is supported in a basic way, where the user can create one selection, that is overwritten when they create a new selection. Furthermore, they support brushing of axes and data elements.

Usage in Cross-Device Interaction

In the context of our work, we investigate possibilities to create brushing and linking for multiple mobile devices. Therefore, we broadened our definition of brushing and linking and extended it to include cross-device interaction. People often deal with linking devices and creating awareness for surrounding devices in the field of cross-device interaction.

Fig. 2.15 Relate Gateway by Gellersen et al. [Gel+09].

Most of the applications try to create awareness for other devices by using marks around the edges similar to off-screen visualization [BR03; Gus+08]. Gellersen et al. [Gel+09] developed the Relate Gateways, shown in Fig. 2.15. The gateways display nearby services, such as a printer, and are arranged around the screen's edges where they indicate the direction the service or device is located. Furthermore, they serve as access points to services. Users can execute drag and drop operations to use a service or they can click on the gateway as they would on a button.

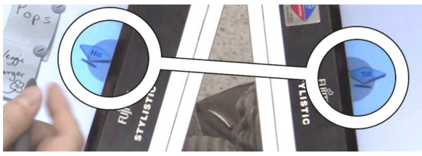


Fig. 2.16 Showing the visual indicators used by Marquart et al. [Mar+12]

Building on the Relate Gateways [Gel+09], Marquart et al. [Mar+12] add visual indicators around the screen as well. However, the interaction possibilities with this indicator depend on the relative distance of the user to the device. Users can either create a connection to the device or receive more information by moving closer. These visual indicators can be seen in Fig. 2.16. Rädle et al. [Rä+15] also make use of visual indicators around the edges. They call them Edge Bubbles and use them for interacting with remote devices. They enable dragging and dropping of files to another devices or copying the text on a remote device. The dragging and dropping operation can be seen in Fig. 2.17. Conductor by Hamilton & Wigdor [HW14] use the visual indicator differently. They use it as a notification for incoming information from other devices.

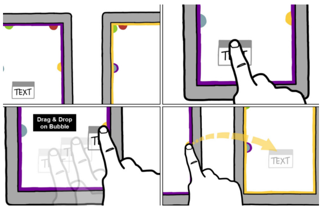


Fig. 2.17 Edge Bubbles by Rädle et al. [Rä+15].

Piazza et al. [Pia+13] combine a tablet and a smartphone for various interaction techniques. One of them includes brushing and linking. While reading a text on a tablet that includes a link, the user can tap the link and it opens on the secondary device, a phone.

Chung & North created SAViL, a system that synthesizes distributed information across multiple devices by linking the information, as depicted in Fig. 2.18. SAViL includes brushing and linking for visual text analysis. Apart from using multiple devices, such as large screens and mobile devices, they developed different linking styles, where they highlight various aspects of a connection, bundle the links for one view, or for one device.

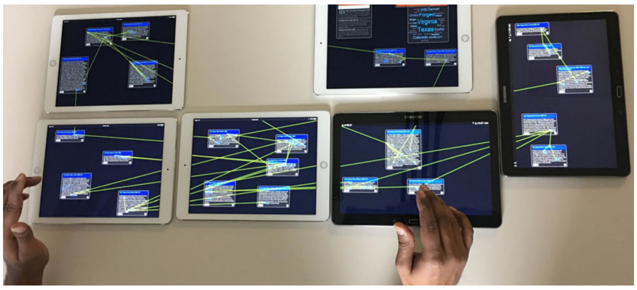


Fig. 2.18 Setup with multiple mobile devices using SAViL by Chung & North [CN17] to link the devices.

2.3.3 Selection Management for Brushing and Linking

How selections are created is often an important part of brushing and linking, as shown in the literature review provided by Koytek et al. [Koy+17]. Recently, even more specific cases, such as selecting data elements on tablets [SS16b], were addressed. However, taking selections a step further and exploring their potential and how users could use them is rarely done.

We reviewed a pool of papers regarding how selections are treated after they were created and if they are stored. Our pool consisted of the previous mentioned papers in this chapter and chosen papers from the literature review by Koytek et al. [Koy+17]. We chose the papers that Koytek et al. [Koy+17] considered *persistent* in their temporality, i.e. that they keep selections until they are explicitly deleted. From out total pool of papers, we found that if selections are mentioned, it is about how elements are selected [CN17; CW11; CC07; Koy+17; Lan+17; ME13; SS14; SS16a; VM12; Zad+14; Woz+14]. If selections are further used or if they are kept was rarely talked about.

Some applications do not focus on selection management but include some of it nonetheless. For instance, Jänicke et al. [Jä+08] offer a rectangle, a circle, and a lasso selection. The selection's color can be changed through the menu evoked by right-clicking on a selected point. Additionally, they support annotations that can be accessed through the same menu. City'o'scope by Brodbeck & Girardin [BG03] defines and supports three types of selections: probing, selecting, and painting which vary in temporality. Painting data elements is a permanent selection until it is reset. However, they only include a small part of selection management by supporting to change a selection's color by pressing a button. Konyha et al. [Kon+06] support permanent selections that can be dragged, resized, and combined by operations. In addition, a tabular display shows detailed numeric information about the data elements that are currently brushed.

Horak et al. [Hor+18] use a separate device, a smartwatch, as personal storage for selections, which they call sets, illustrated in Fig. 2.19. Their system is shown in Fig. 2.12. They support merging selections by applying operations, such as union or intersection. The selections they support are permanent since they are automatically stored on the smartwatch. After storing a selection, they can be temporarily linked to another view. By executing a gesture, they can be pushed to the large display and the link is permanent. Here, selection management is moved to a separate device that holds all selections.

Fig. 2.19 Using a smartwatch as a personal storage for sets [Hor+18].

Elmqvist et al. [Elm+08] support a number of queries for their visualization. These queries are groups of data points, thus they are what we call selections. Their queries are stored in a window that is inspired by the photoshop layers (see Fig. 2.20). Queries can be selected, named, and deleted. The user can drag one query onto another one to perform a union or an intersection operation. Furthermore, they show the size of the query through a visual indication of the percentage of items it has selected.

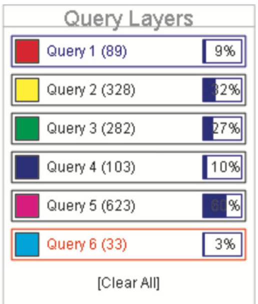


Fig. 2.20 The query window by Elmqvist et al. [Elm+08] that is used to manage queries.

In Stahnke et al.'s [Sta+16] scatter plot visualization, the user can drag the mouse around points or shift-click to select them to create a temporary selection. They are automatically grouped and can be stored as a new permanent selection. After storing it, the user can name the selection. The part of their UI for the selection management can be seen in Fig. 2.21. Every selection has a different color and the management window shows the selection's name, the number of data items, and a thumbnail of their location in the scatter plot.

Mondrian by Theus [The02] specifically deals with selection techniques and how to manage the selections. Mondrian [The02] keeps a list of every selection that belongs to the data set, where selections can be active (shown in black) or inactive (shown in

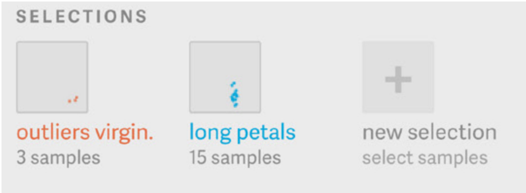


Fig. 2.21 Stahnke et al. [Sta+16] use thumbnails of the visualization to store selections.

grey). He states that there are three ways of creating selections. The standard way where a new selection automatically replaces the old one. The advanced way that supports applying operations to selections to drill them down. The third way, he calls it storing the sequence of selections, enables the user to change the selection at any time. According to his categorization, only three of the papers we found [Elm+08; Sta+16; Hor+18] operate in the third way. Most other work around brushing and linking belongs to the standard way and some fall in the second category.

2.4 Summary

Overall, the related work for selection management in the context of brushing and linking for multiple, mobile devices is a combination of InfoVis literature and work in cross-device interaction that we used to inspire our exploration of the topic. Table 2.1 gives an overview over the presented works and how they relate. In summary, we can say that especially looking at the first four columns there are four distinct groups of related works:

- CMV applications that use mobile devices
- Systems that combine multiple and mobile devices
- InfoVis on mobile devices
- Traditional CMV applications that do not use mobile devices

There is also one outlier by Stahnke et al. [Sta+16]. This paper relates because of how it deals with selections and its management component. However, it does not fit in any of the other categories.

In the following chapters, we keep the established guidelines from the InfoVis community around CMVs and try to incorporate parts from cross-device interaction to translate brushing and linking from traditional interfaces to mobile devices.

3 Selection Management and Brushing and Linking for Mobile Cross-Device Interaction

In this chapter, we present our concept which uses VisTiles by Langner et al. [Lan+17] as a basis and extends it with support for brushing and linking and selection management. Overall, our UI concept consists of five main parts:

Visualization View Creating selections within a visualization

Selection Menu Interacting with selections within a visualization

Selection List Managing multiple selections

Detail View Managing a single selection

Portal View Linking selections to other devices

The following sections further explain these components and the role they play in the UI. First, we present initial thoughts on the selection management framework and explain how it can be used. Second, we explain VisTiles as the basis for our concept and mention the changes we made. Third, we describe aspects that we considered during our design process. Section 3.4 deals with creating selections inside a visualization whereas Section 3.5 shows how to further interact with selections inside a visualization. Section 3.6 describes the managing of multiple selections. In contrast, Fig. 3.31 attends to managing a single selection. Linking selections to other devices is described in Fig. 3.39. Sections 3.9 and 3.10 further explain features of the UI. Finally, Section 3.1 describes summarizes this chapter by applying the selection management framework to our own concept.

3.1 The Selection Management Framework

During our iterative design process, we structured our ideas into single UI concepts. We unified them into one concept that is part of this chapter. The single parts of our concept address different aspects. We noticed the following categories for the single elements of the UI:

Number of Devices Is one single device used or are multiple devices combined?

Number of Selections Is only one selection at a time supported or can multiple selections be used at the same time?

Temporality of Selections Are the selections temporary or permanent?

We define temporary selections as selections that are not part of a selection management system, i.e. they can neither be stored nor can operations be applied to them. Permanent selections, however, are stored selections that can be further worked with.

We realized that we created a conceptual framework. It presents a way to categorize the interaction with selections and situate them in the context of multiple mobile devices. It describes a design space that can be used to inform and create future designs for managing selections in a multi-device environment. Table 3.1 shows how our UI elements fit into the framework.

		TEMPORARY SELECTIONS	PERMANENT SELECTIONS
SINGLE SELECTION	SINGLE DEVICE	SELECTION MENU	SELECTION MENU DETAIL VIEW
	MULTIPLE DEVICES		PORTAL VIEW
MULTIPLE SELECTION	SINGLE DEVICE	SELECTION LIST	SELECTION LIST
	MULTIPLE DEVICES	VISUALIZATION VIEW	VISUALIZATION VIEW PORTAL VIEW

Tab. 3.1 The five parts of the UI categorized by number of devices, number of selections, and temporality of selections.

The framework describes different dimensions of selection management. It further extends the design space introduced by Koytek et al. [Koy+17]. They deconstruct brushing and linking and reveal the parts: source, link, and target of the brushing and linking technique. In our case, the framework does not focus on brushing and linking in particular but rather on selections itself even though it originated from brushing and linking.

Koytek et al.'s [Koy+17] design space lists the number of selections as "Multiple Selections". They use a dimension called temporality as well. However, it differs from our perspective on it. Koytek et al. [Koy+17] define three cases of temporality:

transient, temporary, and persistent. Brushes are transient when they are only active as long as a cursor points at a data item. Temporary brushes stay active until a new brush is generated. Persistent brushes are active until they are removed. Relating them to our framework, Koytek et al.'s [Koy+17] transient and temporary brushes belong to our temporary selections. Their persistent brushes, however, can be either part of our temporary category or the permanent category depending on the existence of a selection management system. Therefore, we extend Koytek et al.'s [Koy+17] temporality category by one attribute, permanent selections.

The framework can be used to classify parts of an interface regarding managing selections in multi-device environments, as done in Table 3.1. Moreover, we could use it to categorize existing work to putz it into perspective considering selection management.

3.2 Fundamental Information about VisTiles

Before explaining the parts of our concept, we briefly describe how VisTiles [Lan+17] deal with device relations

The relations between devices can be determined by their locations. However, VisTiles [Lan+17] established workspaces as a way to support collaboration and coordinate between different views. Thus, they create a relation between devices that is independent from their location. Overall, we focus on three different device relations that are illustrated in Fig. 3.1. The "relation" between two devices that shows the least proximity is when two devices are *not related* at all, i.e., the devices are not spatially close and do not belong to the same workspace. If devices share a workspace their overall proximity increases and they share attributes that are exclusive to their workspace. The highest proximity achievable in our case is the *side-by-side* state. Devices in this state belong to the same workspace and are arranged *side-by-side*. A device or view that is *side-by-side* to another view is called an associated view to the other view. These relations have previously been established by Langner et al. [Lan+17] in VisTiles. For the relations included in this work, absolute tracking is a benefit but they can be implemented without. For instance, the *side-by-side* state can be achieved by executing a gesture, such as a pinch, to connect the two devices that are placed next to each other.

Since the presented concept relies on VisTiles, parts of it were adopted, such as device relations. Other parts were altered to support new features. A component that we decided to change is the use of color. In VisTiles, visualizations have a color scheme that is part of the visual encoding. Additionally, the colored border around the edges of the screen represents the workspace with a distinct color for every workspace. For our concept, we decided to reserve the color usage for selections in

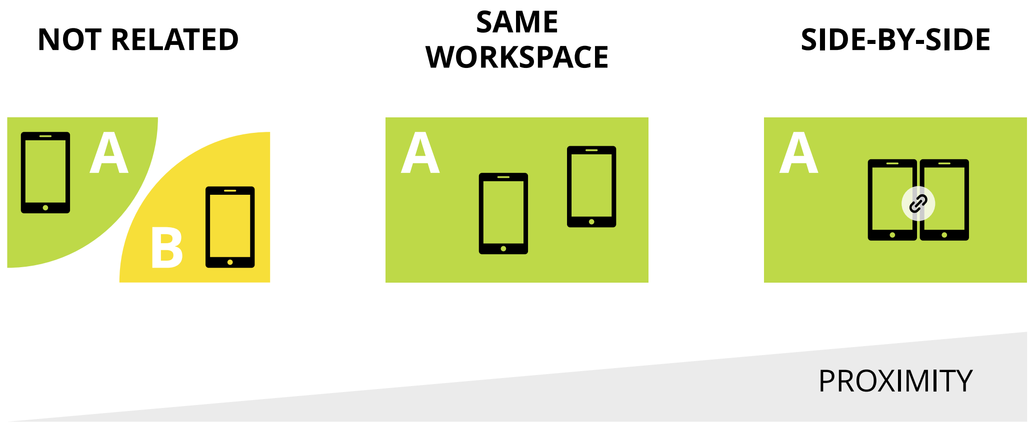


Fig. 3.1 Three device relations: Devices that are not related, devices sharing the same workspace, and two devices sharing the same workspace and are arranged side-by-side. (from left)

Fig. 3.2 Triangular shape in the top left corner of every view showing the letter A. This view belongs to workspace A.

general. Every selection gets an individual, bright color. Other, non-selected data elements use different shades of grey or muted colors to not interfere with selections. Workspaces are no longer represented with a border but with a triangular shape in the top left corner of the screen which is shown in Fig. 3.2. It depicts a letter that corresponds with the name of the workspace. The first workspace is *Workspace A*, the second is *Workspace B* and so forth. For now the workspace's names are fixed but in the future their names could be changed by the users. Color is reserved for the interaction with selections, as previously seen by Stahnke et al. [Sta+16].

3.3 Design Considerations

In the context of distributed CMV applications on multiple, co-located mobile devices, two aspects are responsible for making brushing and linking a challenge in contrast to traditional settings: Combining multiple, heterogeneous devices in one system and having various people interact with the system at the same time. Starting with these two issues and taking inspiration from related works [Ise+13; Chu+15; Blu+15], we explored their implications and challenges for designing the concept. We derived six Design Considerations (DC) that served as guidelines for our design process and form the foundation for all parts of the concept presented hereinafter:

DC 1 **Novice users versus expert users**
Is the concept tailored towards users with different expertise levels?

DC 2 **Collaboration**
Does the concept promote collaboration and supports the user in collaborating with others?

DC 3 **Relations between devices**
Are the relations and connections between individual devices visualized and communicated?

DC 4 **Spatially aware devices versus spatially agnostic devices**
Are the devices aware of their own position in relation to the other devices?

DC 5 **Distributed views versus views on the device**
Are related views on one device or are they distributed onto several devices?

DC 6 **Device attributes**
What are the devices properties, such as its form factor, size, resolution, and input and output modalities?

The first two considerations (**DC 1** and **DC 2**) relate to having multiple users, the last three (**DC 4**, **DC 5**, and **DC 6**) deal with multiple mobile devices, and the third consideration (**DC 3**) is connected to users and devices at the same time.

3.3.1 Multi User: Novice Users Versus Expert Users and Collaboration

Different people come with varying expertise levels regarding the presented system. People who work with visualizations regularly and people who use mobile devices every day. At the same time people who might not own a mobile device or have never seen a CMV application before could come in touch with the system. Our concept reflects these stages of competence especially in the way we designed and adapted the interface for novice and expert users (**DC 1**). Apart from different skill levels, users have various goals while using and exploring a visualization system. People can work together and are encouraged to do so (**DC 2**) but they should have the option to have their personal space where they work on their own. If one person has the possibility to affect a device that another person is working with, informing people about changes becomes essential. Arising questions are: What is changing? Where does it come from? Who is responsible for the change? Can I prevent the change from happening? If not, can I revert the change? Considering these issues is part of our concept (**DC 3**).

3.3.2 Multi Device: Spatially Awareness, Distributed Views, and Device Attributes

Mobile devices are heterogeneous and appear in many shapes, such as tablets and smaller phones. For instance, they vary in overall size, bezel size and provided sensors (**DC 6**). In a system like VisTiles [Lan+17] that encourages people to bring their own devices and incorporate them into the system, social issues could emerge. Some people might not want their devices to be used by other people and consider their device as private. If the device is viewed as private by the owner is another attribute that influences process. Regarding the devices, working with devices that know their position in relation to other devices is an opportunity. This information can be used to create spatially aware interactions. However, currently spatially awareness requires a tracking system (see VisTiles) which is not always available. Therefore, our concept is tailored towards both cases: spatially aware and agnostic devices (**DC 4**). The devices' locations are essential information to the users especially in a case like ours where an action on one device can affect another device. Keeping the user informed about the change and the device's state in the current workflow is beneficial, i.e., how the different devices are related and connected (**DC 3**). This is crucial when views, that manipulate and control other views, are distributed, such as setting menus. In case of having views that interact with each other on different devices, the user needs awareness for the connection between the views (**DC 5**).

3.4 Creating Selections Within a Visualization

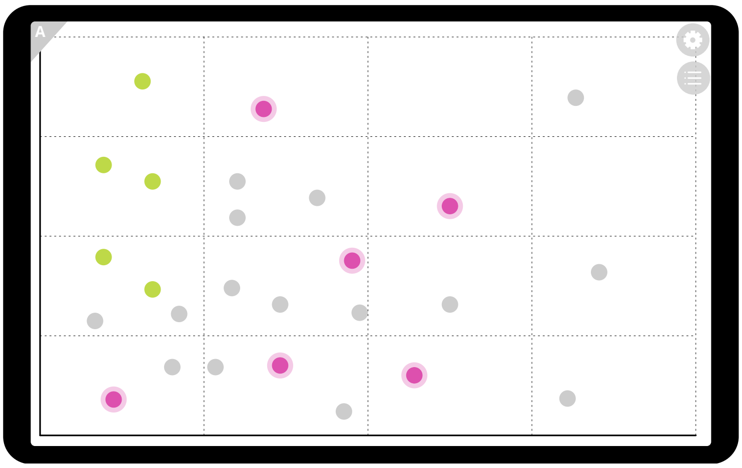


Fig. 3.3 The Visualization View with a scatter plot and two selections, a green one and a pink one. The pink selection is currently in the Editing Mode. In the top right corner there are two buttons. The top one opens a menu to access settings and workspaces. The bottom one opens the Selection List (see Section 3.6)

Starting with a visualization, one of the first actions people want to do is getting an overview of the data. Their next actions usually include looking for subsets of data that they are interested in. Generating selections helps with this step and structures the items into subsets. We define selections as a number of specific elements from a data set that are of interest to the user. Usually, the user creates selections but selections can also be the product of an automated process or be provided by the system itself. For instance after brushing elements in a visualization they are considered a selection. Thus, selections can represent a foundation for all visualization tasks that require examining subsets. Figure 3.3 illustrates a Visualization View with a scatter plot and several selections. Other possible visualization types are bar charts, line charts, parallel coordinate plots, and stream graphs.

3.4.1 Selection Techniques

As soon as the person found data items of interest, they are encouraged to use the selection techniques our interface provides (see Fig. 3.4), that were inspired by Sadana & Stasko [SS14]:

- Tapping on single elements to select and deselect them.
- Using a lasso to enclose multiple elements by executing a double tap and dragging subsequently.
- Brushing an axis to specify a range that contains elements by dragging over an axis.

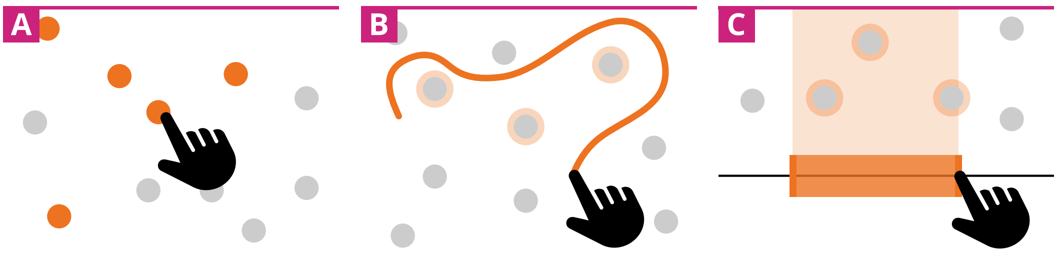


Fig. 3.4 Methods for selecting items or areas in a visualization: *(a)* Tapping to select items, *(b)* drawing a Lasso to select items, and *(c)* Brushing over an axis to select a range.

At first, we included a rectangular selection like in MyBrush [Koy+17]. We decided against it since the Lasso provides all the features the rectangular selection offers and more. To support creating selections some basic functions are available, such as zooming and panning. The user can zoom by using a pinch gesture and pan by dragging the view. Zooming addresses the issue of identifying small data points, e.g., in a scatter plot, mentioned by Sadana & Stasko [SS14]. They also argued that the Lasso technique and the Axis Pan technique (equivalent to our Axis Brush) are suitable for scatter plot visualizations on mobile devices. Furthermore, a button in the top right corner opens the Selection List, where all generated selections are stored in. The Selection List is further explained in Section 3.6. How selections are created in detail is different for novice and expert users since they have different needs and expertise levels to work with. If the system is in novice or expert mode is defined in the settings menu.

3.4.2 Creating Selections for Novice Users and the Editing Mode

At the beginning, novice users can use one of the aforementioned selection techniques to create a new selection. Tapping an element, brushing an axis, or drawing a lasso generates a new selection which opens the Selection Menu at the same time (see Section 3.5). The user can manually evoke the Selection Menu by double tapping an item in the selection. The selection starts in a so called Editing Mode where the user can modify the selection. The Editing Mode is the only way for a novice user to modify their selection. A selection's items that are in the Editing Mode have an additional light outline as illustrated in Fig. 3.5. The user can modify the items in the selection by using the selection techniques. They can use them in the described way. Tapping a single item that is already in the selection deselects the item. Tapping an item that is not part of the selection adds it. The Lasso and the Axis Brush always add the chosen items to the selection and *never* deselect them. After making the desired changes, there are three ways for exiting the Editing Mode and storing the changes made: Double tap on the visualization background, switching off the Editing Mode in the Selection Menu (see Section 3.5) or in the Selection List (see Section 3.6). If no selection is in Editing Mode, executing a selection technique creates a new selection. As mentioned before the Editing Mode is highlighted by a light colored outline around the data items (see Fig. 3.5). It is not a global mode and is restricted to one device, i.e., a selection in Editing Mode on one device appears normal on other devices that show the same selection.

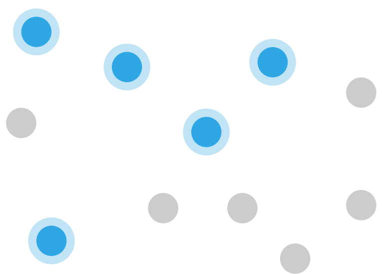


Fig. 3.5 A blue selection in a scatter plot visualization in Editing Mode. The light colored outline around the circles show that the visualization is currently in the Editing Mode.

3.4.3 Creating Selections for Expert Users

In contrast to novice users, expert users have a more flexible way of creating selections which allows for parallel interactions at the same time. For experts the Editing Mode plays a smaller role and is not primarily used when modifying selections. However, every executed selection technique creates a new selection by default. To add or remove items the person needs to use one hand to hold an item that is part of the selection to signify that they want to modify this selection. With the other hand, they can perform a selecting technique. If they brush an axis, draw a lasso, or tap a single unselected item while still holding a selection item, the newly selected items are added to the selection. Tapping an item that is part of the selection while holding another item, removes it. For experts, the Selection Menu does not appear automatically when a selection is created. It has to be evoked by double tapping on an item in the selection.

3.4.4 Temporary and Permanent Selections

The concept of brushing and linking is involved in the creation process of a selection. At the moment of creation, a selection is automatically linked to all views in the same workspace. However, our concept implements two types of selections: temporary selections and permanent selections. Temporary selections are quick and easy to work with but less powerful than permanent selections. Every selection is a temporary selection when it is created. It can be turned into a permanent selection by storing it (see Section 3.5). A temporary selection on the one hand is always linked to all devices in the current workspace, i.e., it is highlighted on all views. It has a limited amount of features and settings. Because it can be deleted quickly it does not lead to unwanted clutter. On the other hand, a permanent selection can be deactivated instead of deleted. The user can choose to which views they want to link a permanent selection instead of having to link them to all views. It can be linked to specific views only instead of all views. It can also be used as a filter and it can be combined with other permanent selections. In case a person wants to quickly see where some data items are located in another visualization temporary selections are sufficient and convenient. However, turning a temporary selection into a permanent selection can be beneficial. If the person wants to keep the selection but not have it visible all the time, permanent selections solve the problem. The same goes for combining selections by creating an intersection, sharing them with other workspaces, or viewing them only on selected devices to not disturb people using the other devices. When we started developing the concept we only used permanent selections which were powerful and customizable because we wanted to supply the user with features that enrich the brushing and linking process. However, we discovered that quickly selecting items and immediately seeing their connection to other views is a core feature of CMVs. Restricting the use to permanent selections would prevent this rapid interaction. Therefore, we established temporary selections that work in a similar way as traditional brushing and linking interfaces, as seen in VisTiles [Lan+17].

Furthermore, we added an interaction technique that is available from every view in the system: Shaking the device "highlights" all devices belonging to the workspace. This means that the target devices' screens show a bright colored overlay that presents the workspace's name and the corresponding letter for a few moments. Figure 3.6 depicts this screen. Additionally to the visual signal, the device starts to vibrate for a short amount of time. This is especially helpful if many devices are positioned on the table.

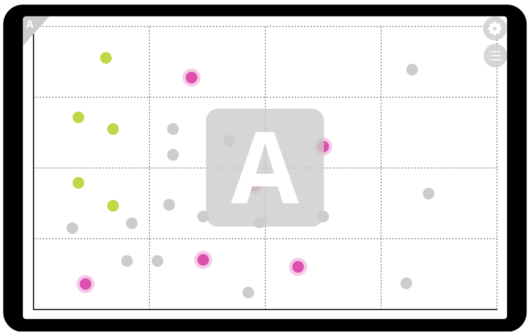


Fig. 3.6 A device showing the overlay that appears when the user highlights the devices belonging to the workspace by shaking the device. The overlay shows the letter that corresponds to the workspace.

3.4.5 Highlighting Workspace Devices

This possibility of highlighting devices ties into the awareness aspect that brushing and linking includes. Creating and preserving awareness for the devices belonging to the interactive surroundings of the user is crucial in a distributed setting. Aside from our proposed solution, we identified other techniques that are suitable for highlighting and drawing attention to distant devices and belong to different groups. It is important to notice that all techniques work for spatially aware and spatially agnostic devices alike.

Visual cues offer the largest range of different techniques. Static visual cues are, for instance, filling the whole screen with a bright color (Fig. 3.6), flashing the screen once or twice (Fig. 3.7b), or showing a colored frame around the edges of the screen (Fig. 3.7a). They can range from more subtle to bold and depending on the situation they can all be useful. Dynamic visual cues involve animation which is usually more distracting and intense, especially if a person is currently working with the device. The view itself or the elements within the view can be used for animation such as pulsing or shaking. Audio cues provide techniques ranging from subdued to very dominant depending on their pitch, volume, and the actual sound.

Fig. 3.7 Two types of visual cues: *(a)* a colored frame around the edges of the screen and *(b)* a colored flashing of the screen.

They do not disturb the person interacting with the device but they attract other people to the device. However, they are less intrusive than some visual cues. The last group of techniques is haptic cues which is rather restricted due to the technologies. Examples for haptic cues are vibration or force feedback. Compared to vibration, force feedback is very hard to achieve in current mobile devices. Additional hardware is needed to be attached to the device. Potentially, it could be used to change the position of a device to convey some kind of direction. However, vibration is available for all mobile devices and is a feature that is rarely used. By vibrating, the device shows that a change occurred.

We decided to use a combination of visual and haptic cues since the visual channel is the one that is most used by interacting with mobile devices. Adding vibration seemed like an unusual choice but an interesting one that we want to investigate more in the future.

3.4.6 Further Interaction

Another kind of technique we previously introduced is a Side-by-side Interactions. In the case of Visualization Views, we support one that works implicitly for two Visualization Views in a side-by-side state. If the two views show the same active selections, links will be drawn between the selections' elements. This is an explicit form of the brushing and linking in a distributed setting. Chung & North [CN17] have a setting of devices that is similar in appearance. However, our links are drawn when the devices are close to each other. Figure 3.8 shows this technique

To further support the fast and convenient way of using selections, we decided to include a function that clears the Visualization View, i.e., all temporary selections are deleted and permanent selections are deactivated. To clear the Visualization View the person performs a hold on the visualization's background. Regarding the interaction in the Visualization View, temporary and permanent selections do not differ much since every selection starts as a temporary one. Although, the Selection

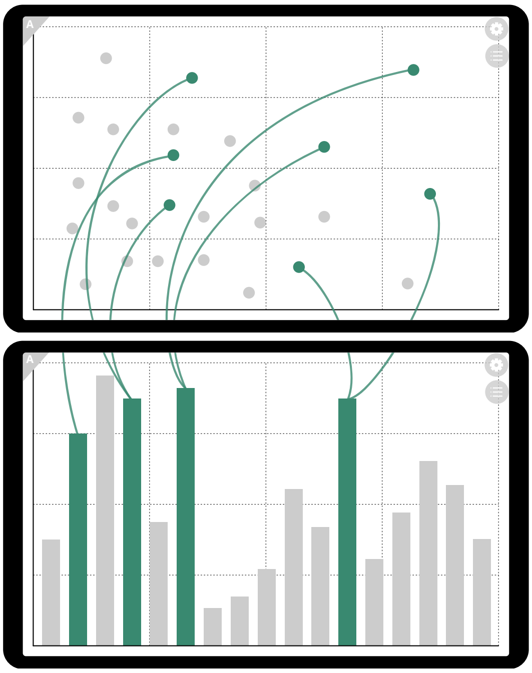


Fig. 3.8 Two devices side-by-side showing Visualization Views. The connections between the same data elements in both visualizations are shown through drawn links.

Menu that provides the user with essential features is different for both selection types which we explain in the following section.

Summary: Visualization View

Selecting Techniques

- *Create Lasso.* Double tap and drag
- *Create Axis Brush.* Drag over axis
- *Select and deselect element.* Tap data element

Zoom. Pinch

Pan. Drag

Open Selection List. Tap on button in the top right corner (see Fig. 3.3)

Open Selection Menu. Double tap on data element belonging to the selection

Clear view. Hold on the visualization background (= Delete all temporary selections and deactivate all permanent selections)

Exit Editing Mode.

- Double tap on the visualization background
- Switching off the Editing Mode in the Selection Menu (see Section 3.5)

- Switching off the Editing Mode in the Selection List (see Section 3.6)

Highlight workspace devices. Shake the device

Draw links between corresponding selection elements. Links are automatically drawn between corresponding selections if side-by-side with a Visualization View

Creating Selections for Novice Users

- *Create a new selection.* Exit Editing Mode if necessary and execute a selection technique. The new selection starts in Editing Mode.
- *Edit a selection.* Enter Editing Mode. Use any selection technique to add. Use tapping an item to remove it from the selection.

Creating Selections for Expert Users

- *Create a new selection.* Every execution of a selection technique creates a new selection.
- *Edit a selection.* Hold an item of the selection to be edited. While holding, use any selection technique to add and use tapping an item to remove it from the selection.

3.5 Interacting with Selections within a Visualization

Creating a selection is a fundamental, first step in interacting with a visualization. However, since our concept focuses on implementing extended brushing and linking that allows for selection management there are more actions people want to perform with selections. Thus, we created the Selection Menu that lets people interact with a selection within a visualization. The Selection Menu plays an important role. It controls and is always associated with one selection. It gives access to essential features and operations that the user can apply to a selection. This section explains the basic functions and explains the menu's visuals and positioning. We discuss the decisions we made as well as the final menu design.

As mentioned before, we differentiate between two kinds of selections: temporary and permanent selections. Since permanent selections offer an extended set of features, their Selection Menu is different from the one for temporary selections. Figure 3.9 shows the Selection Menu for temporary selections in its version for novice

Fig. 3.9 Selection Menu for temporary selections in *(a)* the version for novice users and *(b)* the version for expert users.

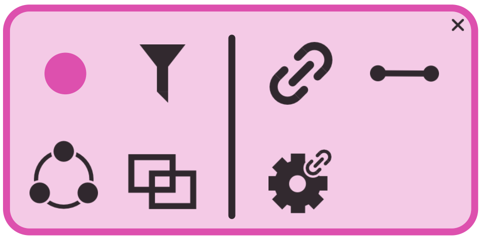


Fig. 3.10 Selection Menu for permanent selections

users and for expert users. Figure 3.10 displays the Selection Menu for permanent selections. Figure 3.11 illustrates how the two menus look in a Visualization View.

Overall, the menu is designed as a tool for easy and quick access to features. It is represented as a rectangle with rounded corners. Its color matches the selection's color to indicate its belonging. The menu opens automatically when a user creates a new selection in mode for novice users which we explained in Section 3.4. Any user can evoke the Selection Menu by executing a double tap on a data item that belongs to the selection. Multiple people can open multiple menus at the same time and in the same view if they wish to. When opening the menu, it is automatically positioned close to the selection's elements. The interacting person can move the menu by touching an area of the menu without icons and dragging it with one finger. They can rotate it by using two fingers. The menu's items show icons that represent executable actions. The most important features are included in the Selection Menu whereas an extensive list of possible actions can be found in the Detail View (see Section 3.7). Menu items can be activated by tapping on them. Some menu items trigger another level of items (e.g., the icon for color), which can be selected by simple tapping as well. In the case of a second menu level, there is an alternative to just tapping the menu items: the person can start dragging their finger from the original icon position to the subsidiary icon to execute the function, as illustrated in Fig. 3.12. This works for all secondary menus mentioned in this section and is especially suited for experts. These second level menus can be closed by tapping on the primary item. For closing the whole Selection Menu, it offers a small x mark on the top right corner. The menu can not be closed by tapping outside the menu.

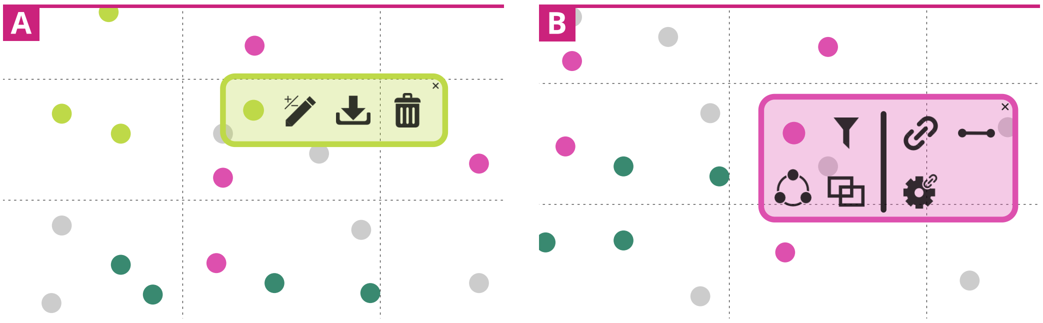


Fig. 3.11 The *(a)* temporary Selection Menu and the *(b)* permanent Selection Menu in a Visualization View.

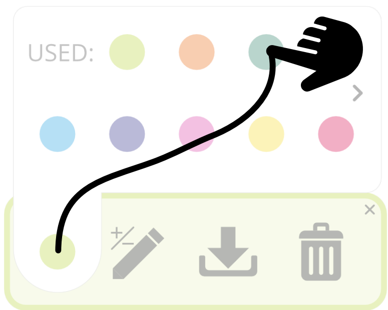


Fig. 3.12 Dragging the finger towards the chosen sub-item from the original item to select the item.

3.5.1 Selection Menu for Temporary Selections

Figure 3.9 shows the two variants of the Selection Menu for temporary selections. Temporary selections are supposed to be short-lived and for quick brushing and linking to other devices. Therefore, the features included for them are limited compared to the full set of features for permanent selections. We explain the menu items starting from the left.

The first item is the *Color* item. Activating it expands another menu level that shows the available colors (see Fig. 3.13). The space is divided into two parts. The upper part shows a selection of colors that are currently unused in the system. The bottom part shows the colors that are in use. By tapping the arrow on the right side or swiping from the right, the user can scroll through more colors. Picking a new color, changes the Selection Menu's color. Furthermore, the *Color* menu item represents the currently selected color. Color is an essential part in most interfaces and having multiple color choices is an important accessibility component, especially for visually impaired people and people with color vision deficiency. Hence, we provide a number of colors to choose from. The second item is the *Editing Mode* item which is only available for novice users. Expert users can manipulate selections by directly interacting with them, as explained in Fig. 3.3. The user can toggle the item to activate and deactivate the Editing Mode for the current selection. An activated item has a lighter background as illustrated in Fig. 3.14. The *Store* item is the next item. It is responsible for transforming a temporary selection into a permanent selection by storing it in the Selection List. As soon as the user activated the item, the menu switches to the menu for permanent selections. The last item is *Delete*. It will delete the selection, remove the color from the data elements, and close the Selection Menu.

Fig. 3.13 Expanded second menu level for the *Color* item which shows the colors to choose from for the selection.

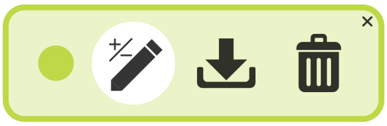


Fig. 3.14 Activated *Editing Mode* item in the Selection Menu.

3.5.2 Selection Menu for Permanent Selections

Figure 3.10 displays the Selection Menu for permanent selections. Permanent selections offer a rich variety of features. They are thought to be used as a storage for data items of interest that are used regularly. The menu for permanent selections is divided into two parts. The four items in the first part are related to brushing and the selection itself. The three items in the second part control the linking to other devices. In the same way as before, we explain the menu items starting with the brushing part.

Menu Items for Brushing

The *Color* item works in the same manner as in the menu for temporary selections. The second menu item in the first row is the *Filter* item which can be toggled. Activating it, filters all items that do not belong to the selection which means that they are hidden. Filtering a selection is global and affects all devices. The *Aggregation* item is in the second row. Aggregation means that separate data items are combined and displayed as one item [Koy+17]. Especially de-aggregation can offer detailed insight for visualization techniques (see Fig. 3.15). For instance, in bar charts

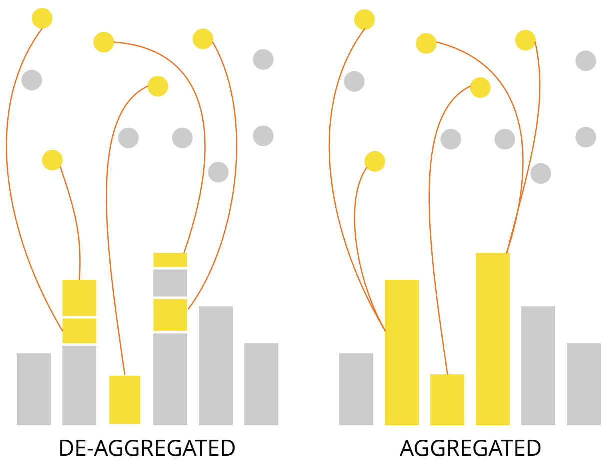


Fig. 3.15 The de-aggregated state on the left shows that the links of each element lead to a separated section in the bar. In the aggregated state on the right, the links lead to the bar that contains the data element.

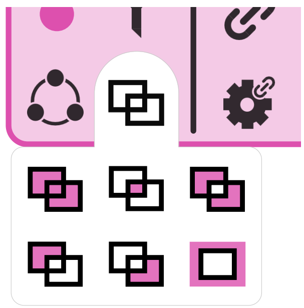


Fig. 3.16 Expanded second menu level for the *Join Operations* item which shows the possible operations to apply to the selection.

one bar usually consists of several separate data items. In the aggregated state, the bar is represented as one continuous bar. In the de-aggregated state, the bar is separated into multiple bars. The individual data items are displayed with an outline to distinguish them from the other items in the bar. The *Aggregation* can be switched on and off in the same way as the Editing Mode. When a selection is first created, aggregation is switched on. The last item in the menu's brushing section is *Join Operations* which shows two overlapping rectangles. Join Operations are regularly used for selections [Hor+18; Kon+06; Elm+08; The02]. By activating the item, a secondary menu opens and shows the six operations to choose from: five operations that use two selections and one operation that uses only one selection. The operations are the following:

- Union of two selections
- Intersection of two selections
- Symmetric difference of two selections

- Difference of two selections in both directions
- Complement of a selection

To learn more about how each of them works and how to choose the second selection for an operation, see Section 3.9.

Menu Items for Linking

The linking section of the Selection Menu starts with the *Activate* item. This item is responsible for activating the linking to other devices. The link serves as inspiration for the icon design. The item can be toggled and is switched on by default. If a selection is active, it is linked to all devices specified in the Portal View (by default all devices), which we explain in Section 3.8. The selection is therefore visible in other views that belong to the workspace. If the user decides to deactivate the selection, it is no longer visible on other devices. After deactivating a selection, it is still shown on the current device that the interaction was executed from until the Selection Menu is closed. The next item is the *Link Style* item. Link Styles determine how the link is represented in the data items and the view themselves. Activating it expands another menu level that shows the possible Link Styles, illustrated in Figs. 3.17 to 3.19:

Color only The data items are displayed in the selection's color (see Fig. 3.17).

Links A link is drawn as a line between two corresponding data elements (see Fig. 3.18).

Edge Marks Edge Marks are positioned on the edge of the screen that represent a connection between two devices (see Fig. 3.19).

Glyphs Glyphs are displayed at the edge of the screen. They show a miniature representation of a Visualization View. They can be used in addition to all other Link Styles (see Figs. 3.17 to 3.19a).

The menu item for using glyphs can be toggled and used combined with the other three styles. Additionally, all Link Styles use the color to highlight the data elements. The *Link Style* menu item always represents the currently selected Link Style.

Visualizing links benefits greatly from spatial awareness because we can use the information to convey directional cues to the user. However, as mentioned before it is not always possible to have a spatially aware setting. Hence, the Link Styles we chose work in spatially aware and agnostic settings as shown in Figs. 3.17 to 3.19. Edge Marks have their origin in cross-device interaction. They are used to create awareness for surrounding devices [Gel+09; Mar+12; Rä+15] and are similar to off-screen visualization [BR03; Gus+08]. Edge Marks can be positioned showing the direction of the other device. They can also be positioned at the top edge without hinting at a direction. They are still valuable to the user because they show how many devices a selection is linked to. The Link Style *Links* is a special case since it can only be used under certain circumstances in a spatially agnostic setting: the devices need to be side-by-side to show the links. Otherwise, information about the direction is not available and therefore, links cannot be drawn in the correct direction. Furthermore, to avoid cluttering, we recommend using *Links* only to visualize links to one or two other views.

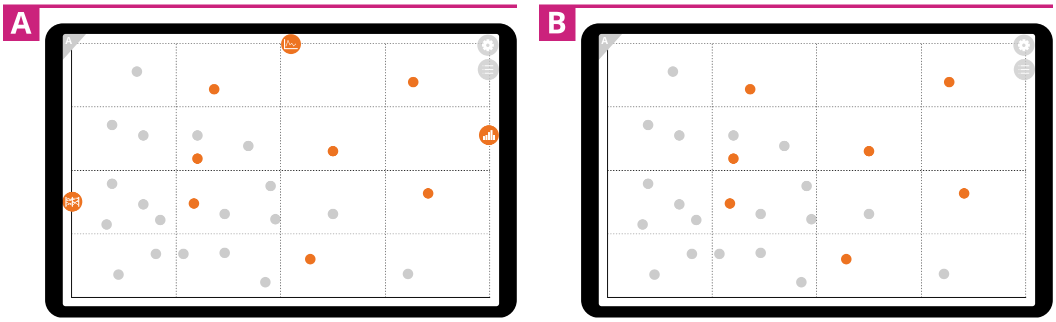


Fig. 3.17 Link Style *Color only* in different setting *(a)* in a spatially aware setting with glyphs and *(b)* in a spatially agnostic setting without glyphs.

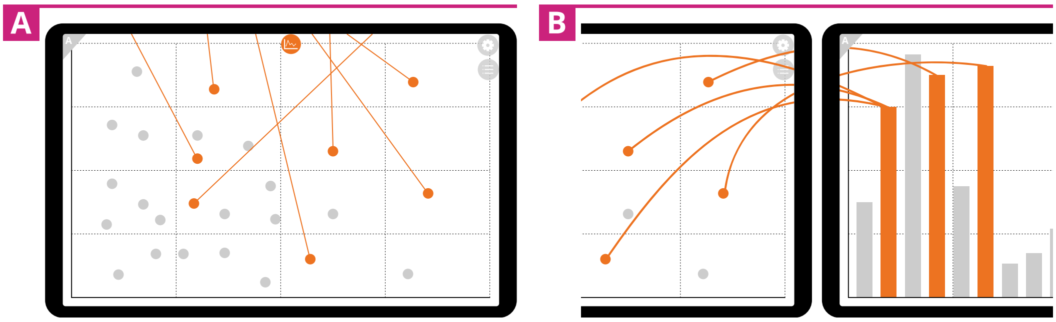


Fig. 3.18 Link Style *Links* in different setting *(a)* in a spatially aware setting with glyphs and *(b)* in a spatially agnostic setting without glyphs.

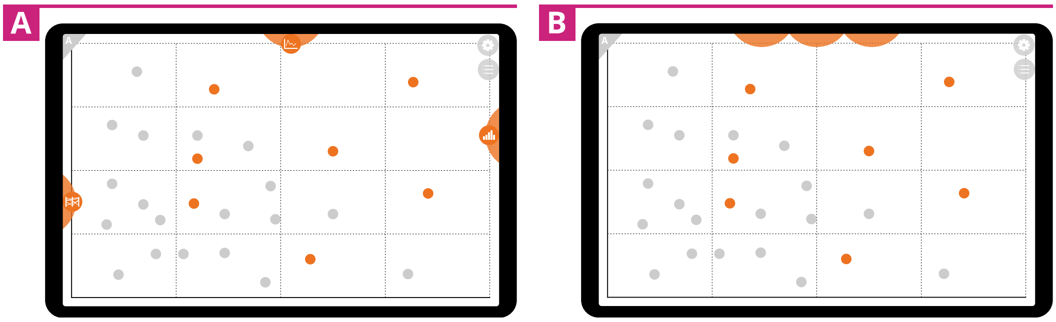


Fig. 3.19 Link Style *Edge Marks* in different setting *(a)* in a spatially aware setting with glyphs and *(b)* in a spatially agnostic setting without glyphs.

Visualizing links ties into the awareness aspect of brushing and linking. By visualizing the link between selections, the person is more aware of the connection. Visualizing links can be done explicitly and implicitly. Explicitly drawing lines is the most common and established way of showing the connection between two selections.

The lines can be drawn as straight lines, curved lines, or dashed lines. Using implicit cues shows that there is a connection to another device but there is no clear indication of the connection between single data elements. Implicit cues can be represented as visual cues or animation cues. Visual cues show a connection to another device or view. Examples for this category are the Edge Marks, colored circles increasing in size (see Fig. 3.20a), and colored rays (see Fig. 3.20b). Animation cues are an ambivalent technique because they are prominent and dominant. An example of how to use animation as a cue for links is shown in Fig. 3.20c. The chosen selection moves as a less opaque copy into the direction where the linked views are. This technique needs to be used carefully but can be of use in some situations.

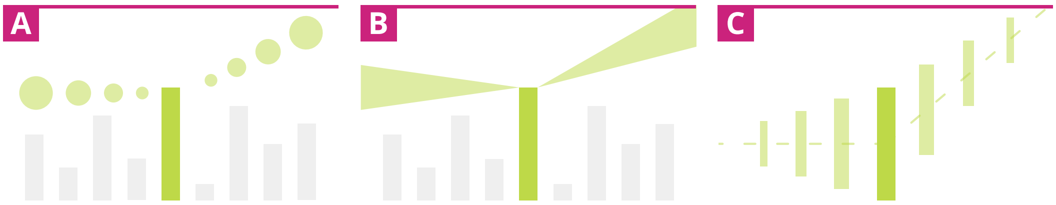


Fig. 3.20 Three implicit cues that can be used for link visualization: *(a)* Colored circles, *(b)* colored rays, and *(c)* using movement as an animation cue to represent links. The movement is along the dashed lines.

The last menu item is the *Link to other devices* item. It leads to the Portal View where the user can make changes to the devices that they want to link the selection to. How the Portal View works and what it looks like, is described in Section 3.8.

3.5.3 Design Process

Through our design process, we explored different styles and positions for the menu. In the beginning, the idea was to implement a radial menu like the one shown in MyBrush [Koy+17] (see Fig. 3.21b). Since it was used in MyBrush for a purpose similar to ours, it is an obvious consideration. The other option was keeping the design consistent with VisTiles by using the rectangular menus with rounded corners, as seen in Fig. 3.21a. MyBrush's menu shows a lot of hierarchy and multiple levels whereas ours is rather flat with items predominantly on one level. This was a minor aspect that spoke against using a radial menu because we thought that we could not use the radial menu to its full potential if our menu does not have multiple levels. Although it could have worked, keeping the consistency with VisTiles seemed more important since the projects are closely related. Hence, we decided to use the visual design from VisTiles with some inspiration from MyBrush

For the components included in the menu, our first iteration started with the MyBrush [Koy+17] menu as well. Since it breaks brushing and linking into its basic parts and gives the user full control, it serves as a good starting point. The first draft consisted of the components of MyBrush, the source, link, and target, in a

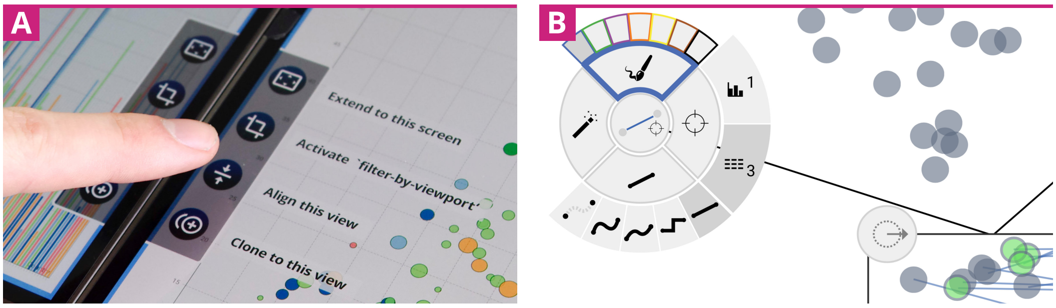


Fig. 3.21 Menu options that were discussed in the beginning of the design process: *(a)* the menu from VisTiles [Lan+17] and *(b)* the menu from MyBrush [Koy+17]

VisTiles inspired menu design. From that state onwards, menu items were refined, omitted, or augmented. We tried to keep the notion of personal agency that MyBrush established by deconstructing brushing and linking. However, we did not want to overwhelm the user by offering too much customization in addition to the selection management tools that we wanted to incorporate. Therefore, the compromise was reducing the number of personalization tools for brushing and linking in favor of adding instruments for selection management. We merged the settings for source and target because in most cases it is sufficient and even desired for the source and the target to have the same visual appearance. We also chose to omit the outline color. In case an outline is needed (e.g., for de-aggregation), a neutral colored outline is provided.

For the menu's location, attaching it onto the upper edge of the screen was the initial idea. (see Fig. 3.22a) It again aims for consistency with the menus used in VisTiles. These are attached to the edges of the screen, rectangular and have rounded corners. However, the screen's edges can be far away from the selection it is representing which can lead to problems depending on the size of the screen. Usually, it is easier and more intuitive to have the menu closer to the element it is associated with. A further problem are multiple selections with several menus. The screen's edges could hold a few menus but the screen real estate is used up quickly. Therefore, a menu attached to the selection was a plausible idea to come up with, shown in Fig. 3.22b. Even though the menu is very close to the selection itself and multiple selections are no issue, it presents some problems. For instance, if the selection is small and only consists of a few elements, it is difficult to find the space for a menu with more than one or two items. Even more so, if the menu has a lot of items to choose from. Because the menu does not change its position, a small or big part depending on the amount of menu items is always occluded. The person needs to pan the view in order to access the occluded parts again. Another issue that developed during the design process is the shape that was drawn to create the selection. At the beginning, we kept the Lassos and the Axis Brushes, that were drawn by the user, visible and connected the menu to the Lasso with a line. After omitting the drawn shapes, visually connecting the menu to the selection bears more problems because it now consists of single data elements instead of a drawn shape.

The final design is a compromise with elements from both ideas. We designed a menu that is separate from the selection shape but still visually connected to it. The small rectangular menu that holds the menu items pops up close to the selection when evoked. The menu's color always shows its connection to the selection it represents. The advantage of this positioning over the other concepts is that it is local but moveable. The menu and the corresponding selection are in the same area but there is less restriction than with the previous designs. A benefit is that having multiple selections at the same time does not cause any obvious issues, such as occlusion. Naturally, having a large amount of selections with their menus on a small screen is not feasible but that applies to all presented concepts.

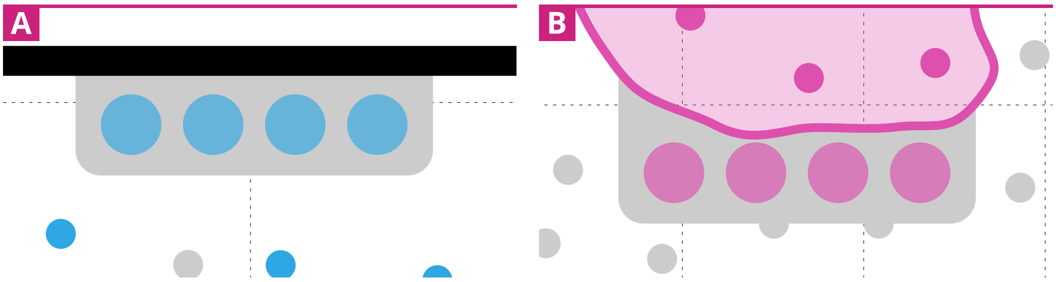


Fig. 3.22 Different menu placements that were discussed during the design process: *(a)* Menu at the top of the view and *(b)* a menu attached to the Lasso.

Summary: Selection Menu

Open Selection Menu. Double tap on data element belonging to the selection

Close Selection Menu. Tap on x mark in the right top corner of the menu

Move Selection Menu. Drag starting on an area of the menu without icons

Rotate Selection Menu. Drag with two fingers

Selection Menu for Temporary Selections

- Change color
- Enter/exit Editing Mode (only for novices)
- Store selection
- Delete selection

Selection Menu for Permanent Selections

- Brushing
  - Change color

- Activate and deactivate filter
- Activate and deactivate aggregation
- Apply Join Operation

- Linking
  - Activate and deactivate selection
  - Change Link Style
  - Link to other devices

3.6 Managing Multiple Selections

When a person creates a selection, it is stored in the Selection List where each selection receives its own entry. This list, which is shown in Fig. 3.23, is a central part of the selection management. The list is shared by all devices in the same workspace, which means that all devices have access to the same set of selections. Thus, lists are automatically shared with other devices and people working with the same data. A user can access the Selection List via tapping the list button in the top right corner of a Visualization View, which we described in Section 3.4. To close the list, the user can press the x mark on the top right corner of the list or execute a swipe gesture to the right on top of the list's title.

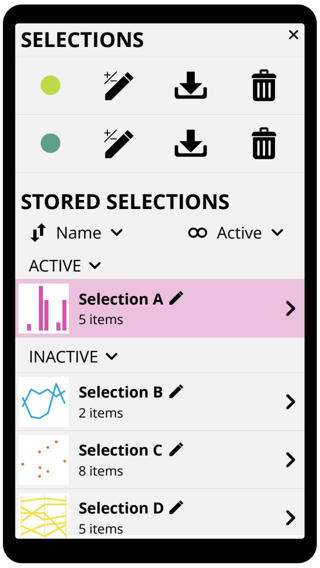


Fig. 3.23 Selection List with two temporary, one active permanent selection and 3 inactive ones.

Figure 3.23 shows the Selection List with all its features and some sample selections. Naturally, the user can scroll up and down the Selection List, which consists mostly of Selection Entries. The top part of the list is reserved for temporary selections. The bottom part which is titled *Stored Selections* contains the permanent selections. The

Selection Entries for temporary selections focus on functionality and practicability whereas the entries for permanent selections concentrate on conveying information and helping the user recognize the selections. Although, the two sections are different they support some common interactions. If the user taps and holds a Selection Entry, the data elements of the chosen selections start to shake in the Visualization Views. We call this action *Highlighting a selection*. The shaking is used to draw attention to data elements. This is analogous to highlighting workspace devices by shaking the device described in Section 3.4. Highlighting a selection has a global impact since it affects all workspace devices that the selection is active in at the moment. If the selection is inactive, the data items will not shake. Additionally, there is a Side-by-side Interaction that works for temporary and permanent selections. If the device with the Selection List is side-by-side to a device with a Visualization View, the person can double tap on a Selection Entry. This evokes a Selection Menu in the Visualization View. The interaction also works for Selection Lists that are on the same device as a Visualization View. It is analogous to the double tap on a data element which opens the Selection Menu as well (see Section 3.5).

By default, the Selection List opens on the right side of the screen for landscape mode and on the bottom part of the screen for portrait mode, illustrated in Fig. 3.24. For devices with smaller screen sizes, splitting the view into a part for the visualization and an overlay for the Selection List can be difficult. In that case, the list is shown as a separate view that occupies the whole screen. When pressing the list button, a new view that shows the Selection List opens. The separate view is illustrated in Fig. 3.23. The list can be positioned either on the same device as a Visualization View or it can be transferred to a separate device for increased functionality and usability benefits by using a Side-by-side Interaction.

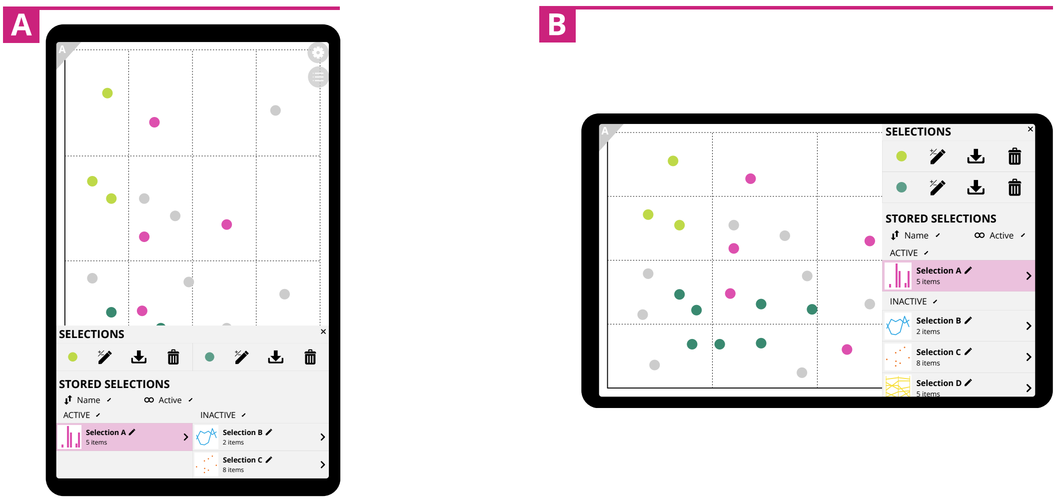


Fig. 3.24 The Selection List *(a)* opens on the bottom part of the screen if the device is in portrait mode, and it occupies the *(b)* right part of the screen in landscape mode.

3.6.1 Transfer the Selection List to a Separate Device

To transfer the list to a separate device, the two devices need to be in a side-by-side state. Smaller devices, like smartphones, are especially suited for holding a Selection List. They can be moved easily and their size is fitting for a list and options interface. Once the devices are arranged side-by-side, the so called Edge Menu with possible options appears. It includes an option for transferring the Selection List to the other device, as shown in Fig. 3.25. After relocating, the Selection List is shown on the separate device and the original device displays the visualization. This kind of Side-by-side Interaction is called explicit because the user explicitly starts and controls the interaction. Yet, implicit Side-by-side Interactions do not require the user's confirmation. They are triggered automatically by a side-by-side state and adjust the interface or enable certain interactions. Later in this section we explain some implicit Side-by-side Interactions. Relocating the Selection List to a separate device instead of having it on screen has several benefits. First, the list can be moved around and is not bound to one device. Hence, several people can use it at the same time while primarily interacting with other devices. Second, it offers people to change the list's position, since some people might prefer having the list on the left side of the Visualization View instead of the right side. Third, the list serves as an overview device for the workspace. It is a central hub from where the workspace can be controlled. Using a separate device to hold selections, was done by Horak et al. [Hor+18]. They used a smartwatch to show the selections. However, they did not include a view that provides an overview, such as a list.

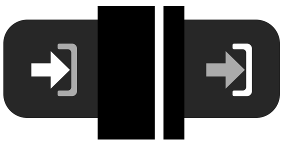


Fig. 3.25 An Edge Menu with the option to transfer the Selection List to a separate device appears when another device is next to the current device.

3.6.2 Section for Temporary Selections

The section for temporary selections occupies the top part of the list and shows one entry for each temporary selection. Theses list entries visually resemble their respective Selection Menus because the available items and functionality is the same. The user has the following options through buttons: change the selection's color, enter and exit the Editing Mode, store the selection, and delete it. The option to enter and exit the Editing Mode is only available for novices. Additionally, the option only appears if the Selection List is on the same screen as a Visualization View. Alternatively, the Selection List can be side-by-side to a Visualization View. The reason for combining the two views is that the Editing Mode always needs a visualization that it can be used in. It works as an implicit Side-by-side Interaction. An example of a Selection Entry in expert mode is shown in Fig. 3.26.

The Selection List also supports an explicit Side-by-side Interaction. In case the Selection List is put next to a Visualization View, the Edge Menu appears and includes an item for storing a selection, i.e., convert it to a permanent selection. There is one menu item for each temporary selection that is currently in the Visualization View. Figure 3.27 illustrates an exemplary Edge Menu for two temporary selections.

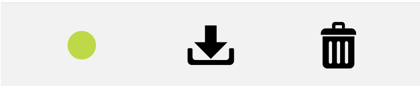


Fig. 3.26 Selection Entry for a temporary selection in expert mode.

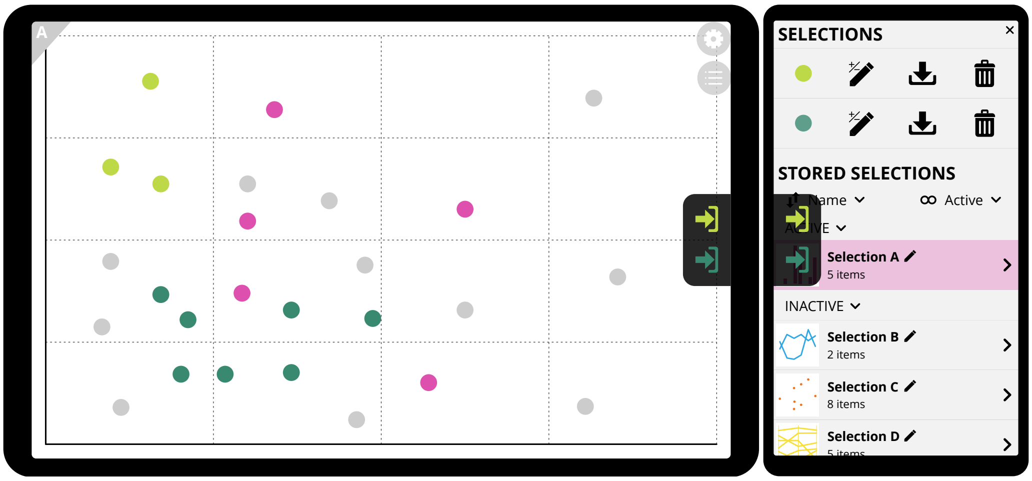


Fig. 3.27 Edge Menu for a Selection List is side-by-side to a Visualization View. It includes a *Store selection* button for every temporary selection in the Visualization View. In this case two buttons for a green selection and a blue selection.

3.6.3 Section for Permanent Selections

The section for the permanent selection is located below the part for the temporary selections. This part is more elaborated and offers more features than the temporary section.

Sorting and Grouping

The two drop-down lists below the heading are for sorting and grouping the selections, which can be seen in Fig. 3.23. The right one presents the sorting options whereas the left shows the grouping. Currently, the list in the figure is sorted by the selection's names and grouped by their activity or inactivity. By tapping on one of the areas, a drop-down list appears showing the user possible alternatives. They can choose to sort the list by attributes, including the following:

- *Name.* Sort by name of the selection
- *Number of elements.* Sort by number of elements in the selection
- *Newest first.* From the last created selection to the first created selection
- *Oldest first.* From the first created selection to the last created selection
- *Last use.* Descending from the most recently used selection

The attributes *Name* and *Number of elements* are available in an ascending and in a descending way. All attributes' point of reference is the current workspace. Grouping works in a similar manner with a drop-down list. The Selection List can be grouped by one attribute at a time. At the moment, the list can be grouped by active and inactive selections or the user can choose not split the list into groups. Further attributes to group the selections can be added.

Selection Entry

The Selection Entry is the entry in the Selection List representing a single selection. As mentioned before, it focuses on providing information rather than offering features. We assume, for permanent selections it is more important to recognize the selections based on their attributes, like name or number of elements, rather than accessing all features right away. Therefore, it is appropriate to emphasize the provided attributes and information. Each selection has its own entry, which shows a visualization glyph on the left side. Beside it, there is the selection's name and a small pen icon. The interacting person can tap the pen icon or the selection's name to change the name. The number of elements belonging to the selection is located beneath the name. The last element of the Selection Entry is a small arrow pointing to the right. The person can tap the arrow to reach the Detail View for the chosen selection. An alternative way to open the Detail View is swiping from the right to the left on the Selection Entry. The Detail View and its contents are explained in Section 3.7. The most important action the user can execute in the Selection List is activating the selection. They can do so by tapping on a part of the entry that is not occupied. This activates the selection and links it to the chosen views which results in making the selection visible on the views in the workspace. An active selection's entry takes on the selection's color as illustrated in Fig. 3.28.

On the left side of every entry, a glyph shows the view the selection was created from. It is a miniature representation of the original visualization and shows the selection's color, size, and location in the view. It adds a visual component to an otherwise theoretical amount of data objects. It is supposed to help the user recognize their previously stored selections. These glyphs were inspired by Stahnke et al. [Sta+16]. They are part of an implicit Side-by-side Interaction with a Visualization View. If a Visualization View is side-by-side with the Selection List, the glyphs adopt the visualization that is shown in the Visualization View. For instance, if the Visualization View is a bar chart, all glyphs transform into the same bar charts and show the selection as part of it. This helps to give the user information about where and how selections are positioned in the Visualization View without needing to activate it. Furthermore, it can assist the user in deciding which selection they want to activate and explore in the Visualization View.

Fig. 3.28 Selection Entry of an activated selection. Its background has the same color as the selection itself.

3.6.4 Working with Multiple Selections in the Selection List

Another interaction technique that can be used with the Selection List is performing a hold on a Selection Entry. This selects the target selection and highlights its entry. In this state, the user can select multiple selections by tapping on their entries and deselect them by tapping again, as demonstrated in Fig. 3.29. This procedure is commonly implemented in UIs for mobile devices to select multiple items from a list. On the screen's top edge, a menu bar shows the number of selected elements and possible actions the person can choose to perform.

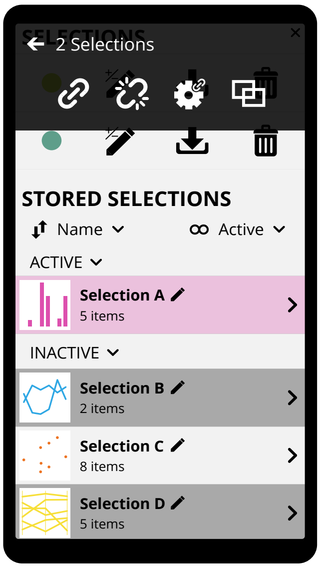


Fig. 3.29 A Selection List with multiple selections selected with options to activate, deactivate, link to other selections, or apply Join Operations.

The possible actions are: Activate selection, deactivate selection, link to other devices, and apply Join Operations. The *Link to other devices* option opens the Portal View (see Section 3.8) where the selections can be collectively linked to and unlinked from other views. The option called *Apply Join Operations* lets the user choose an operation to execute with the selections. If a single selection is chosen, the icon for the *Complement* (one rectangle with everything outside of the rectangle colored) option is shown because it only needs one selection. If two selections are picked, the icon with two overlapping rectangles for the combination operations is displayed. After choosing a Join Operation and executing it, a notification with feedback about the procedure appears and the newly generated selection gets added to the Selection List, as further described in Section 3.9. If the user selects more than two selections, the option *Apply Join Operations* is not available because we are currently limiting Join Operations to two parameters.

The above described way of applying Join Operations is quick but with expert users in mind we decided to include an alternative: dragging and dropping of a selection. This has been done before by Elmqvist et al. [Elm+08] By performing a hold on a Selection Entry and starting to drag, the entry will be dragged along with the finger. The dragged selection is the first parameter for the Join Operation. Stopping and releasing the drag on another Selection Entry chooses the second parameter and triggers the aforementioned menu bar on the screen's top edge. However in this case, it shows the icons for five binary Join Operations (see Fig. 3.30). The system's reactions are the same as described above.

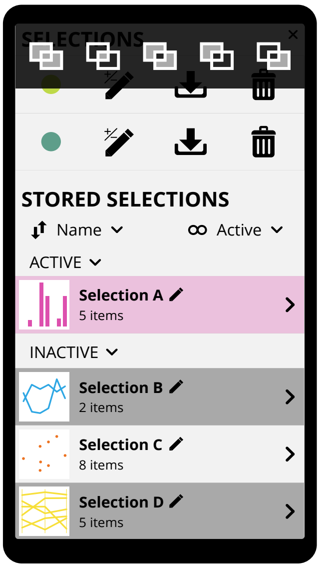


Fig. 3.30 After dragging and dropping a selection onto another selection, the menu bar on the edge of the screen offers five Join Operations.

Summary: Selection List

Open Selection List. Tap on button in the top right corner in the Visualization View

Close Selection List (if on screen with a Visualization View).

- Tap on x mark in the top right corner
- Swipe from left to right on the title

Transfer Selection List to a separate device. Choose option from the Edge Menu if side-by-side with any device

Open Selection Menu. Double tap on Selection Entry if side-by-side with a Visualization View

Highlight selection. Hold on Selection Entry

Highlight workspace devices. Shake the device

Section for Temporary Selections

- Change color
- Enter/exit Editing Mode (only for novices)
- Delete selection
- *Store selection.*
  - Tap on button in the Selection Entry
  - Choose option from the Edge Menu if side-by-side with a Visualization View

Section for Permanent Selections

- Sort Selection List
- Group Selection List
- Rename selection
- *Activate selection.* Tap Selection Entry
- *Open Detail View.*
  - Tap on arrow in the Selection Entry
  - Swipe from right to left on the Selection Entry
- *Apply Join Operations.* Drag and drop a selection on another one
- *Select multiple selections.* Hold one selection and release, then tap other selection entries
- *Adapt glyphs.* Glyphs adapt automatically to the visualization if side-by-side with a Visualization View

3.7 Managing a Single Selection

The Detail View represents one selection. It exists for every permanent selection in the Selection List. It holds all information regarding the selection and can execute every operation possible with the selection. The interacting person can reach the Detail View via tapping on a little arrow icon in a Selection Entry in the Selection List or using the swipe gesture on the Selection Entry. Figure 3.31 provides an illustration of the Detail View.

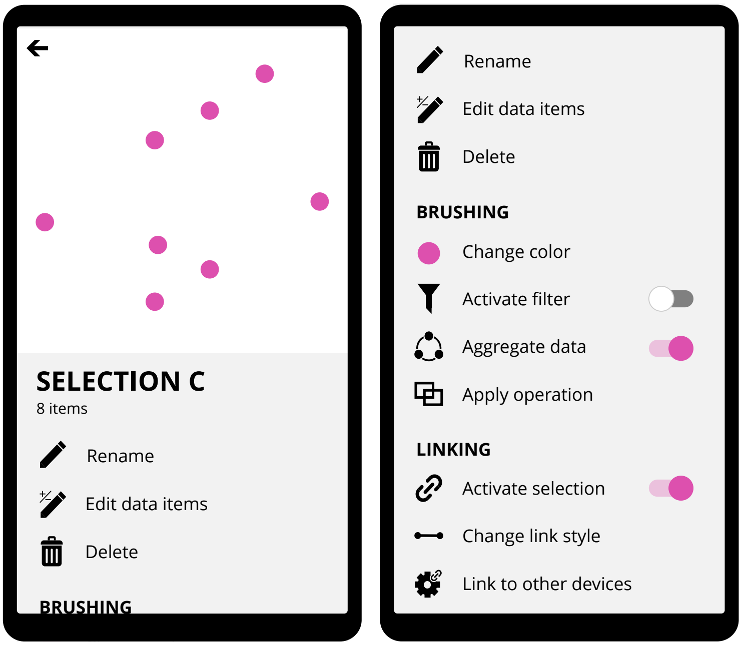


Fig. 3.31 Representation of the Detail View.

3.7.1 Visual Appearance

At first, the idea was to include the Detail View in the Selection List. That by tapping the arrow in the Selection Entry, the Detail View opens as a drop-down view. Tapping the icon again, closes the Detail View again. This version of the Detail View is illustrated in Fig. 3.32a. We decided not to pursue this path since it makes Side-by-side Interactions harder to implement than having a separate view for the Detail View. Furthermore, it would overload the list with possibilities and it would get long if several Detail Views were open. Therefore, the decision was made in favor for separating the Detail View from the Selection List.

The top and main part of the view is occupied by a visual representation of the selection. It is the same representation that was previously used in the Selection Entry of the Selection List to show a depiction of the current selection. It shows a miniature representation of the visualization where the selection was first generated. It conveys the selection's color, size, and location in the visualization. Overall, it works as a reminder for the interacting person of the selection since it adds a visual component to an otherwise abstract set of data items. The selection's name and number of elements are located below the visualization image. Moving downwards, the person can find all actions that can be executed with the selection. As mentioned before we thought about adding a visual part to the abstract data, thus we considered giving it a prominent place in the view, as demonstrated in Fig. 3.32b. Comparing the sketch to the final Detail View in Fig. 3.31, shows that it had a big impact on the ultimate design choices.

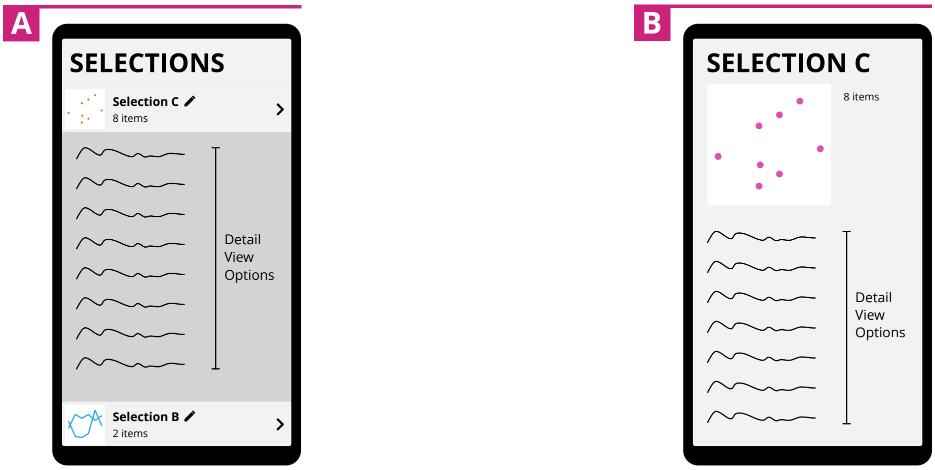


Fig. 3.32 First drafts for the Detail View. One idea was that it *(a)* opens inside the Selection List as a drop-down view and another idea was *(b)* to make it an own view.

3.7.2 Actions to Interact with a Single Selection

Below the visual representation, the user can find information about the selection, such as its name and the number of elements in the selection. Actions that can be executed with the selection are listed next and grouped in the same way as the actions in the Selection Menu Section 3.5, by actions related to brushing and actions related to linking. The actions are the following:

- Rename selection
- Edit data items
- Delete selection
- Brushing
  - Change color
  - Activate and deactivate filter
  - Activate and deactivate aggregation
  - Apply Join Operation
- Linking
  - Activate and deactivate selection

- Change Link Style
- Link to other devices

If an action is not a toggle but has several options to pick from, like the *Change color option*, the item expands and shows the available choices in a second level, as illustrated in Fig. 3.33. Second level menus can be closed by tapping on the original button again. Dragging the finger from the menu item to to a second-level item instead of tapping it is supported in the same way as in the Selection Menu.

General Actions

The first action is renaming the selection. Choosing it, the user can enter a new name for the current selection. *Edit data items* always opens a Data Item List that shows all data elements in the selection as a list. Some of their data is shown in the list and the user has the possibility to delete items. They can perform a hold on entries, select multiple items, and delete them collectively, similar to the usage of the press and hold in the Selection List. Additionally, *Edit data items* enters the Editing Mode in case the device with the Detail View or the Data Item List respectively is side-by-side to a device with a Visualization View and the selection is active. Then, the person can edit the data elements as described in Section 3.4 on the device with the Visualization View. After returning from the Data Item List to the Detail View, the Editing Mode is exited. If the device is not side-by-side to a Visualization View, only the Data Item List is displayed. *Delete selection* deletes the current selection from all views and from the Selection List. After deleting, the device will show the Selection List.

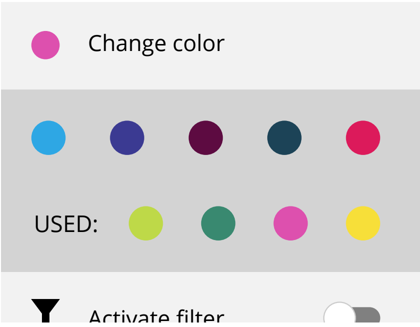


Fig. 3.33 Expanded item for picking a color that shows the available options.

Brushing Actions

Most of the following actions are part of the Selection Menu as well. They all work in the same way as the menu items and are described in detail in Section 3.5. The first three general actions are followed by the actions related to brushing. These

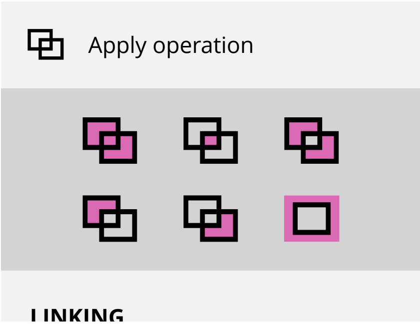


Fig. 3.34 Expanded item for applying Join Operations that shows the six possible operations to choose from.

are: Change color, activate and deactivate filter, activate and deactivate aggregation, and apply Join Operations. *Change color* expands itself to show the colors to pick from. This part of the interface is shown in Fig. 3.33. *Activate and deactivate filter* is a toggle that hides all items that are not contained in the current selection in all views belonging to the workspace. *Activate and deactivate aggregation* is another toggle which controls if aggregation is applied to the data items. Choosing *Apply Join Operations* expands the item and offers the six available operations, illustrated in Fig. 3.34. To learn more about how they are applied in detail, see Section 3.9.

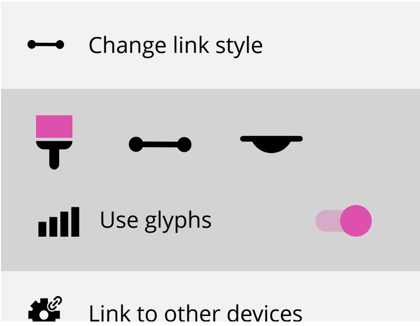


Fig. 3.35 Expanded item for choosing a Link Style that shows the available options.

Linking Actions

The last three actions that can be executed with a selection are connected to linking a selection. The first item *Activate and deactivate selection* activated the selection in all views that are defined in the Portal View (see Section 3.8). *Change Link Style* opens an expanded menu that offers the user to pick a style out of three: Color only, Links, and Edge Marks. In addition, using glyphs can be switched on. The expanded menu is depicted in Fig. 3.35. The last item in the Detail View is *Link to other devices* which leads the user to the Portal View that is explained in Section 3.8.

Side-by-side Interactions

Apart from applying different actions to the selection, the Detail View provides interaction techniques that use the side-by-side state of two devices. If the Detail View is side-by-side to a device holding a Visualization View, the visual representation in the Detail View adapts to the visualization shown on the other view. This process is the same as the Side-by-side Interaction between a Selection List and a Visualization View described in Section 3.6.3. This interaction is an implicit interaction since it happens automatically and does not involve the user.

If a device with a Detail View is side-by-side to another device showing a Detail View, the interacting person can choose to apply Join Operations to these two selections. Once in side-by-side state, the Edge Menu appears and shows a button with the icon for Join Operations. Choosing it, evokes the five Join Operations that demand two selections, as illustrated in Fig. 3.36. After choosing an operation to apply, the action is executed and a new selection is generated. The selection's name is created from the Join Operation itself and the original selections' names, e.g. *Union of Selection A and Selection B*. Both devices show a notification with feedback about whether the procedure was successful and include the new selection's name accordingly. The devices still display the original selections. The generated selection can be accessed via the Selection List.

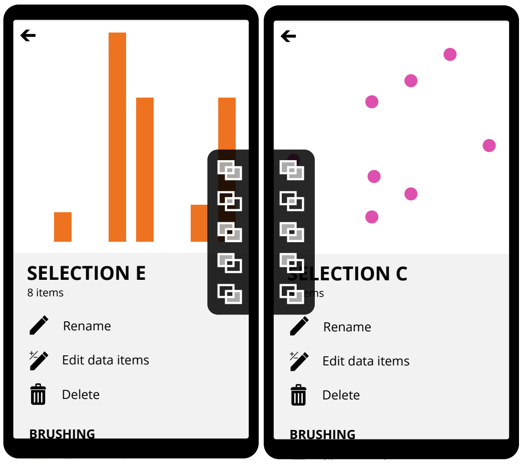


Fig. 3.36 Edge Menu showing the five possible Join Operations to choose from on the two devices in a side-by-side state.

Another supported interaction with two Detail Views is copying the current settings from one selection to another selection. Analogous to the last Side-by-side Interaction, the Edge Menu appears and shows two additional buttons that enable the user to transfer the settings from the left to the right device or from the right to the left device (see Fig. 3.37). This action copies the color, the Link Style, if filtering is active or inactive, if the aggregation is active or inactive, and if the selection itself is active and if it is in which views.

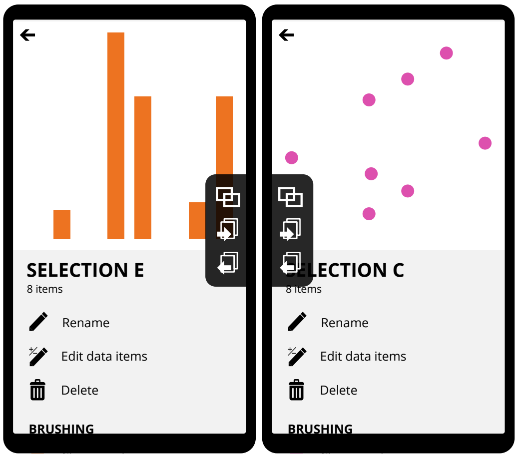


Fig. 3.37 Edge Menu showing the two options to copy settings from one selection to the other in a side-by-side state.

In case the interacting people store many selections and the Selection List gets long, we added another Side-by-side Interaction. If a Detail View is side-by-side with a Selection List, the Edge Menu is evoked and offers a button to find the Detail View's selection in the Selection List. By choosing the option, the Selection List scrolls to the selection's position and visually highlights it.

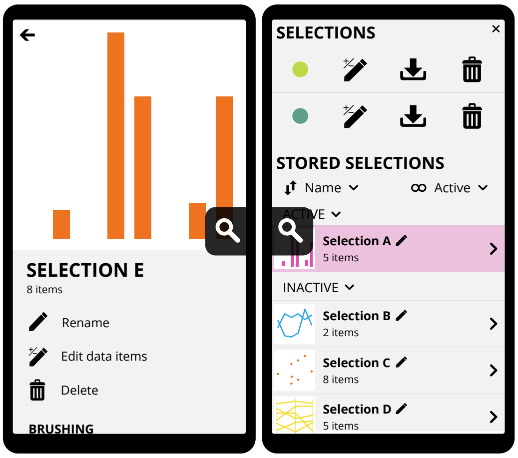


Fig. 3.38 Edge Menu showing the option to find the selection from the Detail View in the Selection List.

Summary: Detail View

Highlight workspace devices. Shake the device

Return to previous view. Tap on back button in the top left corner.

Adapt visual representation. The visual representation adapts automatically to the visualization if side-by-side with a Visualization View

Apply Join Operations. Choose option from the Edge Menu if side-by-side with a Detail View, then choose a Join Operation to apply

Copy settings. Choose option from the Edge Menu if side-by-side with a Detail View

Find selection. Choose option from the Edge Menu if side-by-side with a Selection List

Actions provided by the view itself

- Rename selection
- Edit data items
- Delete selection
- Brushing
  - Change color
  - Activate and deactivate filter
  - Activate and deactivate aggregation
  - Apply Join Operation
- Linking
  - Activate and deactivate selection
  - Change Link Style
  - Link to other devices

3.8 Linking Selections to Other Devices

An essential component of our concept is linking selections to devices and highlighting them on other views. Basic linking to all devices in the workspace is supported through the UI on several occasions, like the Selection Menu (see Section 3.5) and the Detail View (see Section 3.7). However, we want to offer the users an extended version of linking that is customizable and tailored towards collaboration. This version can be found in the Portal View which represents a way to interact with devices in the same workspace. It opens the opportunity to link a selection to other devices or to chosen single devices. The Portal View is accessible through several UI items. It can be accessed from the following points in the UI:

- *Selection Menu.* Choosing the *Link to other devices* menu item.
- *Selection List.* Selecting multiple selections by executing a hold evokes a menu bar that contains a *Link to other devices* option.
- *Detail View.* Picking the *Link to other devices* action.

Therefore, a Portal View is either individually created for each selection, which is the case for most situations. It can represent multiple selections at once if it was evoked from the Selection List by choosing multiple selections, as seen in Fig. 3.29. Figure 3.39 shows the Portal View in its two versions: the version for spatially aware devices (Fig. 3.39a) and for spatially agnostic devices (Fig. 3.39b). For both versions, to return to the previous view, the Visualization View with a Selection Menu or the Detail View, the interacting person can use the back button in the top left corner.

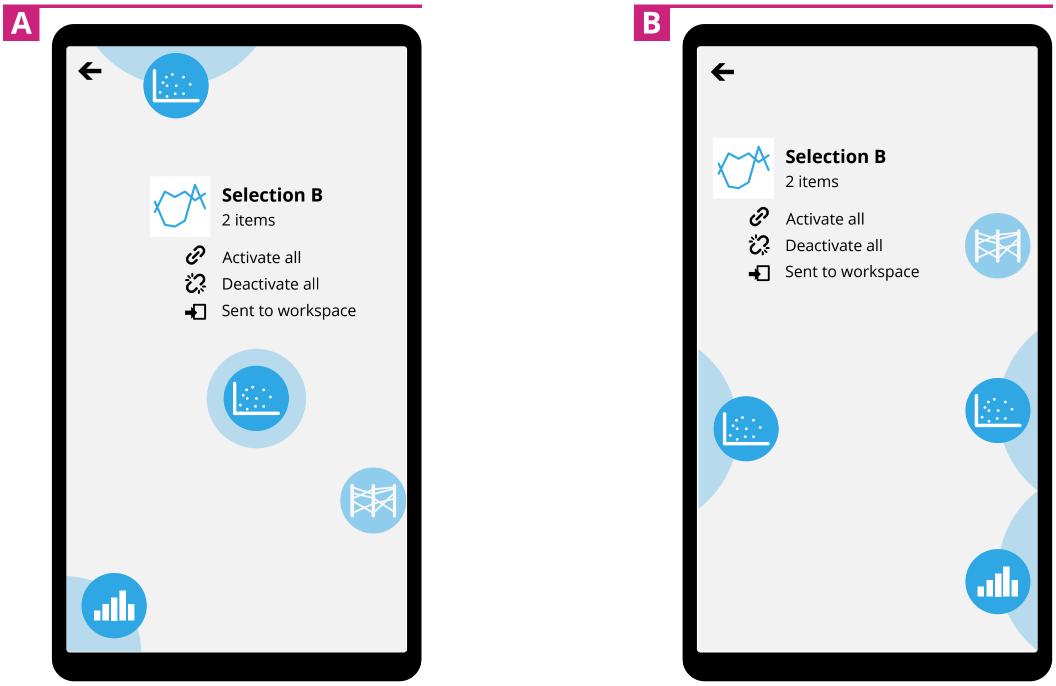


Fig. 3.39 The two versions of the Portal View: *(a)* on spatially aware devices and *(b)* on spatially agnostic devices. *(a)* The portals for the devices are positioned around the edges using spatial information. The portal for the Visualization View located on the same device is positioned in the view's center. *(b)* The portals for the devices are positioned on the right side of the screen. The portal for the Visualization View located on the same device is positioned on the left side.

The Portal View's central element is a representation of the selection or the group of selections that the view is currently handling. For one selection, it shows a miniature representation of the selection and some information which appeared in the Selection List and the Detail View before. For multiple selections, the number of selections is shown. Beneath this element, there are three buttons for executing actions. The first two are *Activate all* and *Deactivate all* which activates or deactivates the chosen single or multiple selections in all views belonging to the workspace. The third option is *Send to workspace* which opens a list with all workspaces. By choosing a workspace, the chosen selection or selections are send to the respective workspace's Selection List. This part is the same for spatially aware and spatially agnostic settings.

3.8.1 Portals and Their Functionality

So far, the user could either activate a selection or deactivate it which affected all views in the workspace. However, in some cases it is useful to activate a selection only for a few selected views and deactivate it for others. For instance, when person A is working with a device and person B is using another device that belongs to the same workspace and wants to examine a selection with numerous elements. In this case, it is beneficial to deactivate the selection for the device that Person A is using to not disturb them in their workflow.

Therefore, we decided to add portals to the view which gave the view its name. Portals are connections and representations of Visualization Views. They can be seen in Fig. 3.39. They resemble the Edge Marks we introduces to visualize links (see Section 3.5.2). Therefore also inspired by cross-device interaction [Gel+09; Mar+12; Rä+15] and off-screen visualization [BR03; Gus+08]. The Portal View contains one portal for every Visualization View in the workspace. It shows the visualization type, such as bar chart, scatter plot, or parallel coordinate plot, and the current state of the view, i.e., active or inactive. The state is represented by a colored highlight surrounding the portal, that resembles the Edge Marks used for visualizing links (see Section 3.5). By tapping on a portal, the user activates or deactivates the chosen selection for the view represented by the portal. This visual design of the Portal View is adjusted if the devices are spatially aware or agnostic. In case the devices are spatially aware (see Fig. 3.39a), the portals are arranged around the edges of the screen based on the position of the devices that hold the Visualization Views in relation to the device with the Portal View. In some cases, the device with the Portal View holds a Visualization View as well, e.g., if the Portal View was evoked by using the Selection Menu. The portal for this particular Visualization View is located in the center of the view, as seen in Fig. 3.39a. Figure 3.40 illustrates a possible scenario of devices and the Portal View that shows the visualization icons according to their relative position to the Portal View device. In case the devices are spatially agnostic, the portals for distributed views are placed on the right side, illustrated in Fig. 3.39b. The portal for the Visualization View that is on the same device as the current Portal View is located on the left side of the screen.

3.8.2 Further Interactions

In addition to activating and deactivating a selection for a view by using a portal, we included a Side-by-side Interaction. In order to perform the interaction, the device needs to be side-by-side to a Visualization View. As soon as the condition is achieved, the Edge Menu appears. It shows an additional button with a chain link icon that is either open or closed, as depicted in Fig. 3.41. Tapping on the button with the open chain link will deactivate the selection for the Visualization View that is in the

Fig. 3.40 Shows a possible arrangement of devices and the Portal View that has its icons placed according to the devices' positions.

side-by-side state. Tapping on the button with the closed chain link will activate the selection for the chosen Visualization View. This can be helpful in situations where the user is already holding both devices or interacting with them. Thus, it can be slower to use the Side-by-side Interaction instead of using the portals.

By performing a hold on a portal, the user can highlight a view in case they do not remember where the device is located or they want to draw another person's attention to it. The device starts to vibrate for as long as the user holds the portal. This function is analogous to highlighting a selection in the Selection List by holding a Selection Entry and highlighting workspace devices by shaking the device, described in Section 3.4.

Summary: Portal View

Return to previous view. Tap on back button in the top left corner

Activate selection. Tap the button *Activate all*

Deactivate selection. Tap the button *Deactivate all*

Activate or deactivate selection for one Visualization View.

- Tap on portal that represents the chosen Visualization View
- Choose option from the Edge Menu if side-by-side with the chosen Visualization View

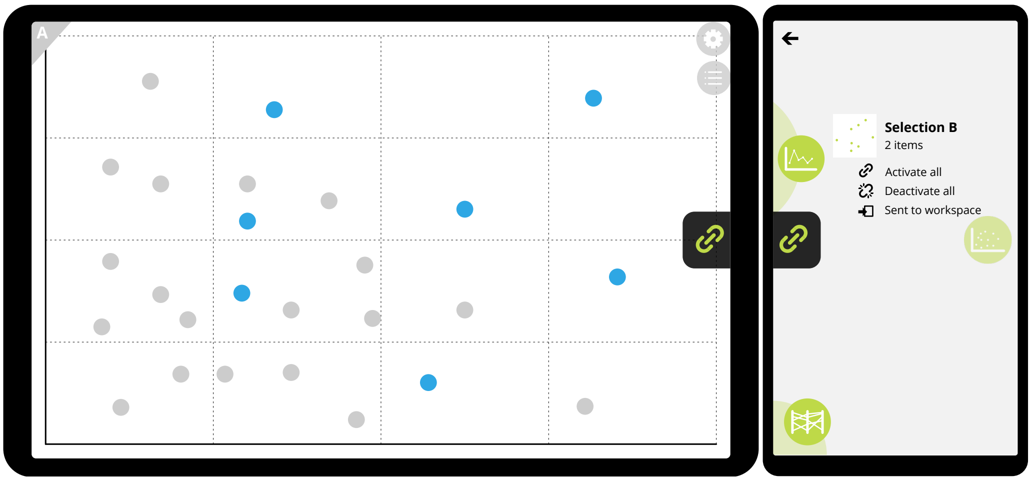


Fig. 3.41 Two devices in a side-by-side state. The left device is showing a Visualization View, the right device the Portal View. The Edge Menu on both devices offers the user to activate or deactivate the selection for the Visualization View on the right device.

Highlight view. Hold on portal.

Highlight workspace devices. Shake the device

3.9 Incorporating Join Operations

Join Operations are an important part of selection management and have been an essential component in the concept from the beginning. They offer to combine, divide, and separate parts from selections.

3.9.1 The Operations

The idea to incorporate combinations first emerged by contemplating what a person might want to do with a selection. It started with wanting to merge selections and turned into creating intersections of two selections and more advanced options. Obviously, this lead to set theory and left us with six different operations in the end. Five of them need two parameters, i.e., selections, and one needs only one parameter. Figure 3.42 provides the icons for the six operations which are the following:

- Union of selection A and selection B: $A \cup B$
- Intersection of selection A and selection B: $A \cap B$
- Symmetric difference of selection A and selection B: $A \triangle B = (A \cup B) \setminus (A \cap B)$
- Difference of selection A and selection B (also called the relative complement of selection B in selection A): $A \setminus B$

- Difference of selection B and selection A (also called the relative complement of selection A in selection B): $B \setminus A$
- Complement of selection A: $\overline{A}$

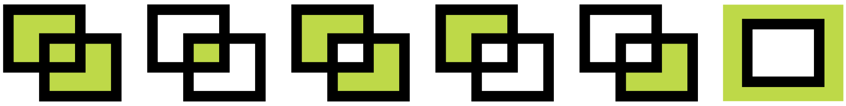


Fig. 3.42 Icons for all six Join Operations, starting from the left: Union, intersection, symmetric difference, the two relative complements, and the complement.

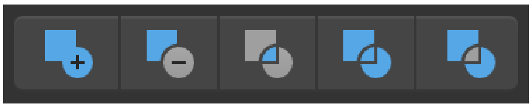


Fig. 3.43 Icons from the *Join Operations* in Affinity Designer.

In theory, it is possible to use more than two parameters for one operation. However, we focus on only two selections at a time. Linking more than two devices in a side-by-side state can be challenging and opens up new issues and promising interaction techniques, which we talk more about in Section 5.2. The icons was inspired by vector graphics editors, like Adobe Illustrator[1] or Affinity Designer[2]. These tools support similar features for vector-based shapes (see Fig. 3.43). After choosing all parameters for the operation, the system asks if the person wants to delete the original selections that are used for the operation. They can answer *Yes* or *No* and have the option to tell the system to remember the decision which is then stored in the Settings and can be changed there.

3.9.2 Applying Join Operations

The UI has five ways of eliciting Join Operations:

1. *Selection Menu.* The Selection Menu contains a menu for Join Operations. (see Section 3.5)
2. *Selection List.* After selecting one or two selections by performing a hold, the option for Join Operations is available in the menu bar. (see Section 3.6)
3. *Selection List.* By dragging one selection onto another selection, the person evokes a prompt for choosing a Join Operation. (see Section 3.6)
4. *Detail View.* The Detail View includes a component for Join Operations. (see Section 3.7)
5. *Detail View.* If a Detail View is put in a side-by-side state with another Detail View, the Edge Menu appears and shows a button for Join Operations. (see Section 3.7)

[1] Adobe Illustrator website
[2] Affinity Designer website

Five out of the six provided Join Operations need two parameters, i.e., selections, to work with which means there are three steps to execute an operation with two parameters: choosing the first parameter, choosing the second parameter, and choosing the operation. The aforementioned list items differ in the order the steps are carried out.

For versions 2, 3, and 5 the user picks the two parameters first and then chooses an operation to apply to these parameters. The person has to choose the two selections they want to combine either by performing a hold (Version 2), dragging and dropping (Version 3), or putting two devices side-by-side (Version 5). Subsequently, a view with the possible Join Operations prompts the person to select an operation. After choosing an operation, it is executed, the new selection generated, and the person receives a notification about the operation's outcome.

This procedure happens differently for version 1 and 4, where the user starts by picking a selection, mostly implicitly, and the operation. The Selection Menu in version 1 belongs to one selection, the operation's first parameter. The user selects an operation from the provided ones in the menu. It works similarly for the Detail View in version 4. The first parameter is the selection that is represented by the current Detail View and the user chooses an operation to execute. Because of the different order of steps, the user needs to select the second parameter after choosing an operation. An small overlay on the top edge of the screen appears on all devices that belong to the workspace and ask the user if they want to use a selection from this device. We decided to use an overlay over the top of the screen because it offers the users of other devices in the collaborative setting to ignore the notification prompt and continue with their work on the device if they are not interested in executing the operation. The overlay differs depending on the view it is being shown at. For the Detail View and Portal View, it asks if this particular selection represented by the view should be used for the operation. For Selection Lists or Visualization Views however, it asks if one of the selections should be used since these views are able to represent more than one selection at a time. If the user answer is positive, the overlay asks the user to pick one selection by tapping on it.

3.10 Overview over Side-by-side Interactions

As described at the beginning of the chapter, Side-by-side Interactions use the device's location to another device. Either the system can automatically detect if two devices are next to each other which makes it spatially aware or the user has to tell the system about their state which is the case in a spatially agnostic setting. Two devices being next to each other, has different effects on the UI and the possible interactions in the context. Therefore, we created an overview that contains all Side-by-side Interactions that were described in the previous sections. They can either be explicit or implicit. Explicit Side-by-side Interactions always use the Edge Menu to provide functions. The user can select a function to execute by tapping a button. Implicit Side-by-side Interactions do not require the user's attention or action. They are executed automatically and provide smaller and mostly visual changes in the UI. For actions that actively change the workflow or the appearance of selections, we decided to use explicit Side-by-side Interactions.

Our concept provides the following explicit Side-by-side Interactions:

- *Selection List + any view.*
  Transfer the Selection List to the other device.
- *Selection List + Visualization View.*
  Store temporary selections from the Visualization View and transform them into permanent selections.
- *Detail View + Selection List.*
  Show selection from Detail View in the Selection List and scroll to its position.
- *Detail View + Detail View.*
  Apply Join Operations to the two selections from the two Detail Views.
- *Detail View + Detail View.*
  Copy the settings from one selection to the other. This Side-by-side Interaction exists twice, once for each direction.
- *Portal View + Visualization View.*
  Activate the selection or group of selections from the Portal View for the Visualization View.

In addition, the concept supports these implicit Side-by-side Interactions:

- *Visualization View + Visualization View.*
  Draw links between selections that are active in both Visualization Views.
- *Selection List + Visualization View.*
  Adapt glyphs in the Selection List to match the visualization in the Visualization View.
- *Selection List + Visualization View.*
  Add item *Enter/exit Editing Mode* to the entries for temporary selections if in novice mode.
- *Detail View + Visualization View.*
  The button *Edit data items* additionally enters the Editing Mode for the selection if it is active in the Visualization View.
- *Detail View + Visualization View.*
  Adapt the visual representation of the selection in the Detail View to match the visualization in the Visualization View.

3.11 Summary: Reviewing the Selection Management Framework

In this chapter, we presented initial thoughts for our conceptional framework for selection management. We discussed considerations for the design process, that lead to the concept we described after it.

To conclude this conceptual chapter, we want to review Table 3.1, which we presented in Section 3.1. It shows how our UI concept is distributed in the space we created with the framework. The Visualization View accesses multiple devices by using links, it displays temporary and permanent selections alike and at the same time. The Selection Menu presents a special case. The small line between that divides the blue rectangle represents the fact that a single Selection Menu can either support a temporary or a permanent selection. The Selection List covers multiple temporary and permanent selections. However, it is restrained to its own device and does not communicate with other devices. The Detail View is a specialized view. It is available only to permanent selections and works on a single device. The Portal View is the only element in our concept that can be used either for single or for multiple selections, but temporary selections do not have access to the Portal View, which connects multiple devices. We notice only one blank spot in the table, which a special single-selection part for temporary selections that involves with multiple devices. It is no surprise, we do not provide this part of managing selections. We made the design decision, that temporary selections cannot be selectively linked to other devices, i.e. they do not have access to the Portal View.

Through applying the framework and reviewing our own concept, we now realize what pieces are missing and can evaluate on whether we want to address this part of selection management in the future or in another iteration of the design process.

Prototype

4

In this chapter, we describe the prototype that our implementation is based on. We further explain the current state of the prototype and plans on extending it. We would like to note that the figures used in this chapter do not represent the actual prototype but a vision of its future state.

4.1 Technical Setup

Our implementation extends the VisTiles prototype by Langner et al. [Lan+17] and is fundamentally and technically based on it.

The prototype currently uses several Android devices, such as the 8.4" Dell Venue 8 tablets and the 5.15" Huawei Honor 9 smartphones. Additionally, we use the external tracking system OptiTrack by NaturalPoint.[1] The IR cameras are located on the ceiling above a table. This table functions as the space to use the devices in. Furthermore, the devices are marked by small pieces of IR-reflective foil to facilitate the tracking.

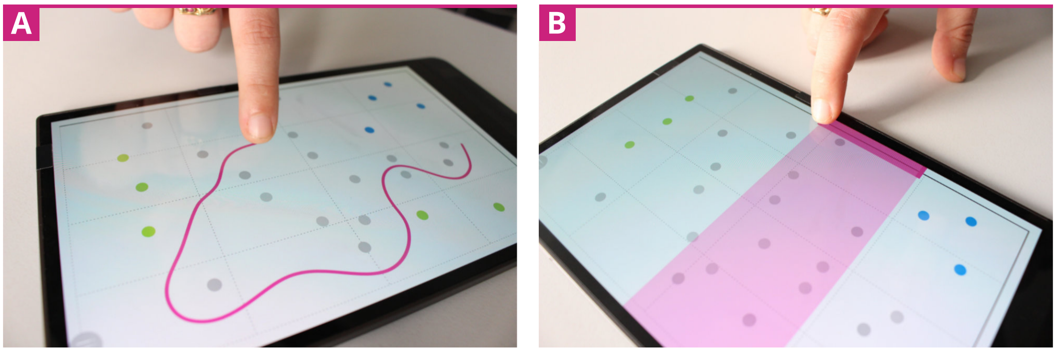


Fig. 4.1 *(a)* A person is drawing a Lasso to select elements. *(b)* A person is creating an Axis Brush by dragging across an axis to select a range.

The prototype relies on a web technologies and a client-server architecture for managing multiple devices and communication. The server side uses Node.js[2] and WebSockets. It receives the tracking data, processes it and transmits events to the client. The client side uses D3.js[3] for the visualizations and Materialize CSS[4] for the

[1] OptiTrack website
[2] Node.js website
[3] D3.js website
[4] Materialize CSS website

UI elements. The client's interface can be accessed through a browser. The presented data is a uses the World Development Indicators[5] data set by The World Bank.

4.2 Techniques

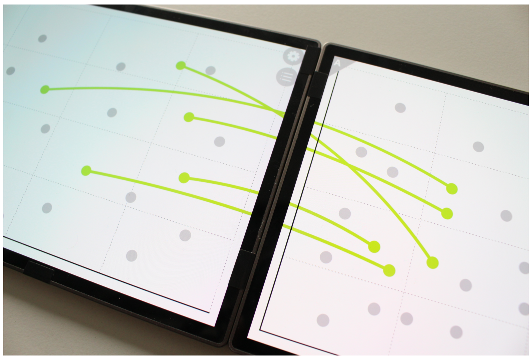


Fig. 4.2 Showing the links between corresponding data elements in a selection when the two Visualization Views are next to each other.

The prototype currently supports the techniques mentioned in VisTiles [Lan+17]. Regarding this thesis, we plan on further implementing techniques for the multi-device setup that VisTiles provides. Extending the existing version of VisTiles with our developed conceptual ideas, is going to serve as a proof-of-concept prototype. One of our main goals is to facilitate working with selections in an easy manner. Therefore, we are going to implement the following techniques in the Visualization View (see Section 3.4):

- Creating selections by drawing a Lasso, creating an Axis Brush or tapping single elements
- Showing links between corresponding data elements of a selection if two Visualization Views are side-by-side
- Evoking the Selection Menu for temporary and permanent selections (see Section 3.5)

To create a selection in the Visualization View, the user can double tap and drag to draw a Lasso that creates a selection when released, shown in Fig. 4.1a. Similarly and as depicted in Fig. 4.1b, the user can drag on the axis to create an Axis Brush. To highlight our focus on brushing and linking, we decided to include showing links on two devices. By positioning two devices with Visualization Views next to each other will trigger the links. They are drawn between the corresponding data elements of the displayed selections. A version of how this might look like in the final prototype can be seen in Fig. 4.2. Furthermore, the Selection Menu is an essential part of interacting with selections. Thus, we decided to include it for temporary and

[5]World Development Indicators by The World Bank

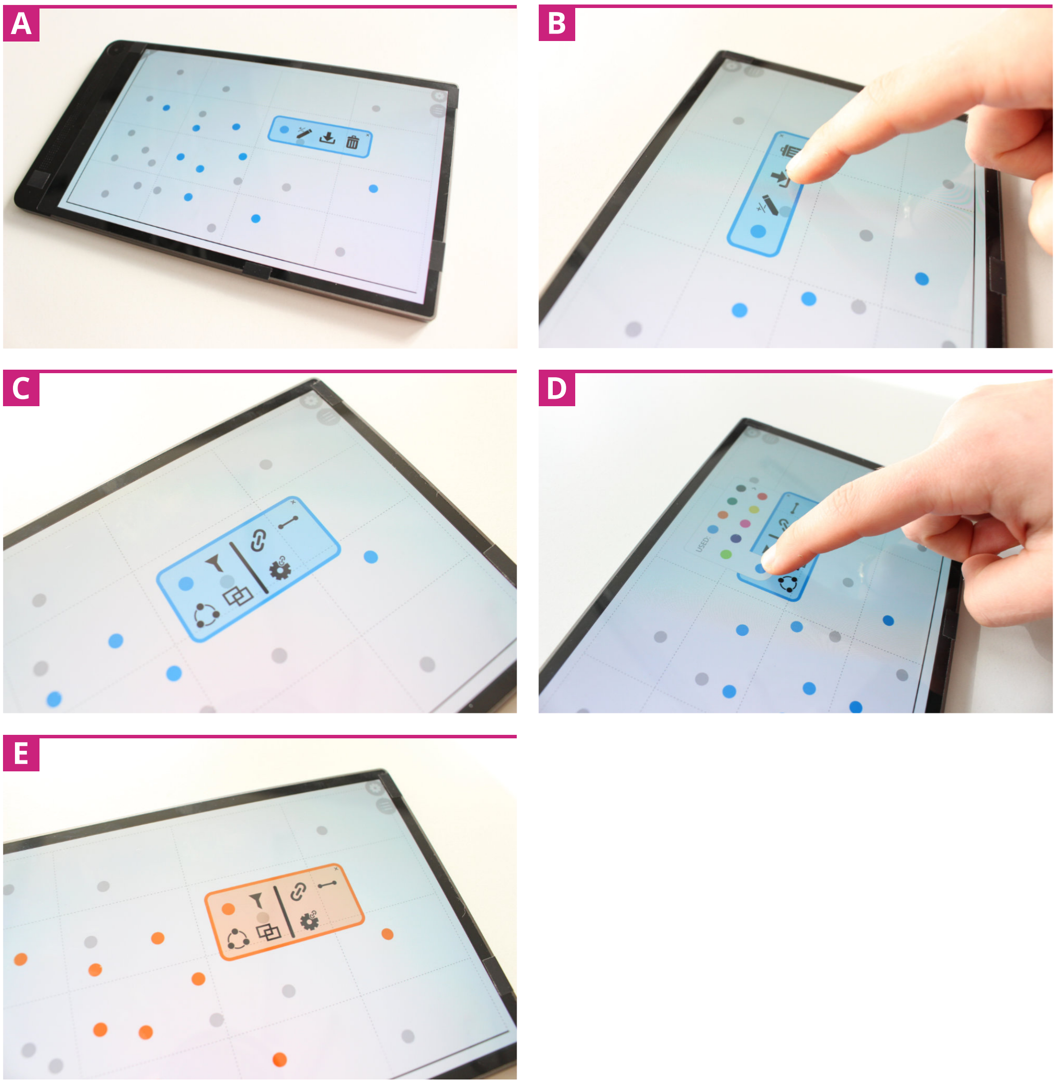


Fig. 4.3 *(a)* The Selection Menu for a temporary selection in novice mode. *(b)* Pressing the icon to store the selection and make it permanent. *(c)* The Selection Menu for the newly created permanent selection. *(d)* Pressing the color icon to pick another color. *(e)* The Selection Menu after the color change.

permanent selections alike (see Fig. 4.3). The interacting person evoked the menu by performing a hold on a data element that belongs to the chosen selection.

For dealing with multiple selections at once, the management tool of a Selection List (see Section 3.6) will be implemented and accompanied by the following actions:

- Managing temporary and permanent selections with the Selection List
- Transferring the Selection List to a separate device (see Section 3.10)
- Adapting a selection's miniature representation using a Side-by-side Interaction (see Section 3.6.3)

The interacting individual accesses the Selection List by tapping the button on the right corner. Additionally, we want to include transferring the Selection List

Fig. 4.4 *(a)* Two Visualization Views where one of them shows the Selection List on screen. *(b)* Putting another device beside it, shows the Edge Menu with the option to transfer the Selection List. *(c)* The transferred Selection List on a separate device.

to another device. The steps of how this will be realized are shown in Fig. 4.4. Moreover, we also decided to implement an implicit Side-by-side Interaction which is adapting the miniature visual representation, which is illustrated in Fig. 4.5.

We are going to implement the Detail View that the user can access from the Selection Entry (see Section 3.6.3) to access the Join operations that can be applied by a Side-by-side Interaction (see Section 3.10). We are planning on implementing merging two selections by applying the Union Operation. This process is illustrated

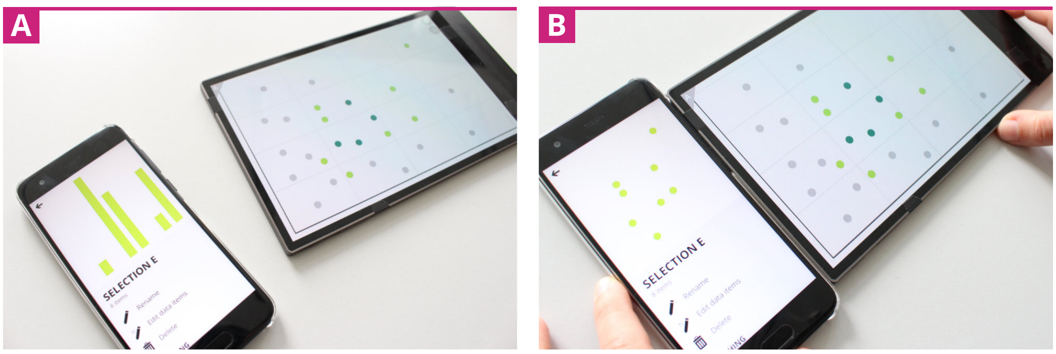


Fig. 4.5 *(a)* The Detail View contains an visual representation of the selection that shows the visualization it was originally created from, here a bar chart. *(b)* When positioned next to a Visualization View, this visual representation adjusts to match the Visualization View's visualization type, in this case it turns to a scatter plot.

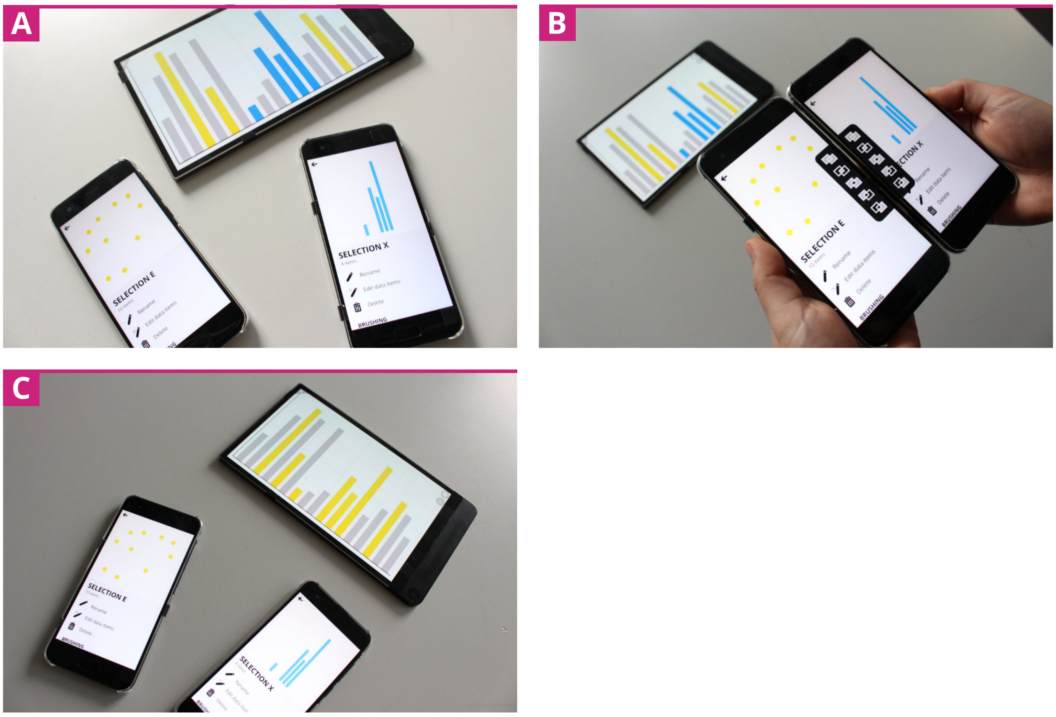


Fig. 4.6 *(a)* A Visualization View showing a bar chart with two selections and the Detail Views of those. *(b)* Juxtaposing the Detail Views shows the Edge Menu with possible Join Operations. *(c)* The final setting after choosing to create the union of the two selections.

in Fig. 4.6. Figure 4.7 shows our last feature that we plan on including: activating and deactivating of selections through the Portal View (see Section 3.8).

We are aware that this is only a portion of the features our concept provides but it will enable a more in-depth exploration of how we can interact with selections in a multi-device environment.

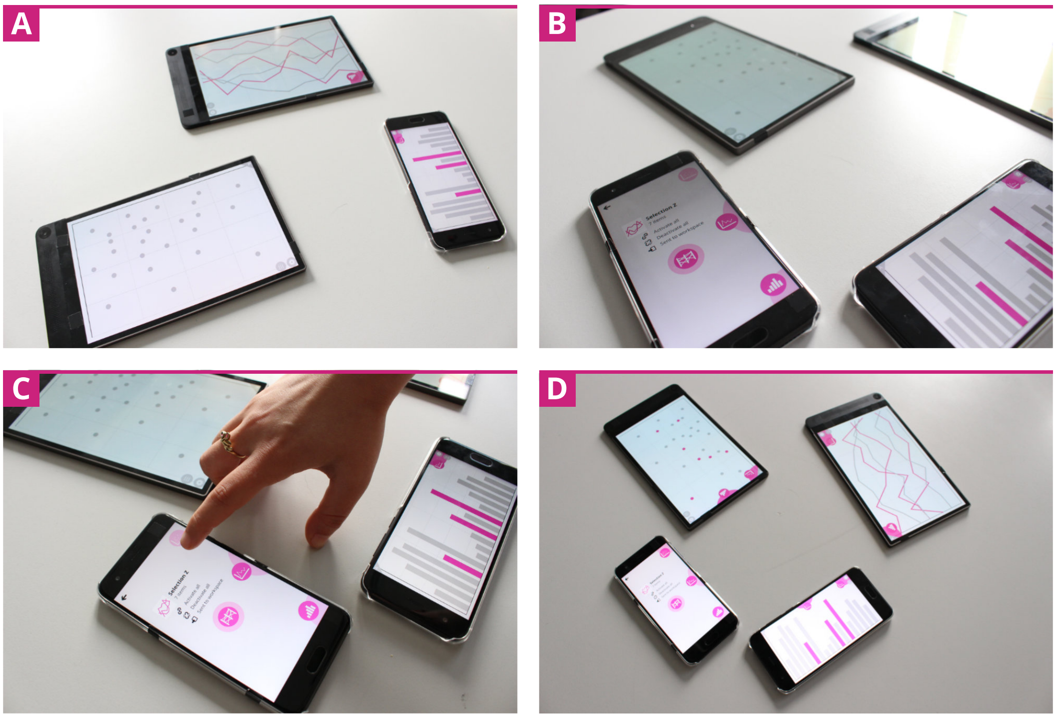


Fig. 4.7 *(a)* Three Visualization Views: A line chart, a bar chart, and a scatter plot. The line chart and the bar chart show a pink selection. *(b)* The Portal View for the pink selection conveys in which views it is active. *(c)* Tapping the scatter plot icon to activate the selection for this view. *(d)* The selection is now active on all Visualization Views.

5 Conclusion and Discussion

We developed a concept for brushing and linking and selection management for a CMV application with multiple, distributed, mobile devices. First, we explained the related work in the fields of InfoVis on mobile devices, cross-device interaction, CMVs, brushing and linking and selection management. Second, we proposed a conceptual framework for selection management in multi-device environments. Third, we identified design consideration for our design process that were the foundation for the concept we developed next. Finally, we planned to extend the prototype by adding extended brushing and linking and selection management functions.

5.1 Discussion

Our UI concept focusses on CMV applications on multiple, distributed, and mobile devices but it can be used on other CMV settings involving remote desktops, tabletops, or display walls. For remote desktops, the devices are located further away and this needs to be reflected in how we present links. Multiple views on tabletops can be used like multiple devices apart from shaking or lifting the devices up. Likewise for display walls, most aspects can be translated from mobile devices to large displays.

First, we briefly discuss aspects that were not included in the final concept. While developing it, we encountered questions which greatly influenced the design process. Some of them are directly reflected in parts of the concept, others did not make an appearance in the final version. Nonetheless, they are worth mentioning since they impacted the process. Second, we are reviewing the Design Considerations we established in Section 3.3. Lastly, we add a brief final conclusion.

5.1.1 Data Foundation and the Difference Between Selections and Filters

During our design process, we often discussed how a selection is defined and if a filter is the same as a selection or something inherently different. We started with the thought that we need to treat them as separate entities. Our first decision was to create separate lists for filters and selections, to have different options for creating them, and even to transform a filter into a selection, and the other way round. When we started to list actions we wanted to perform with filters and selections, we noticed that a big part of these activities are similar or even the same for selections and filters. Thus, the notion that they are not too different started to form. We went from one extreme to another: Selections and filters are the same. Filtering is just one attribute of a selection. After tossing around opinions for a while, one question arose: What if a person wants to filter a region or an attribute range instead of a certain collection of items? Thus, we debated what aspects define a selection. At that point, we decided that a selection is defined by these two aforementioned characteristics: the items that are part of the selection and the area that was defined while creating the selection. From that onwards, it was just a small step to the terms *area based* and *object based*.

The thought was that a selection is defined by two facets: its data items and the attribute ranges it covers. The former is self-explanatory but the latter needs more explanation. Starting at the creating of a selection, the user has three possibilities to generate a selection (see Section 3.4). For instance, if they draw across an axis to select data items, they not only select items but they select a range or area on that same axis. This range was supposed to be stored as well as the respective items. These items were the base of the *object based* Data Foundation and the range/area was the base of the *area based* Data Foundation.

Object based Data Foundation means that the selection is based on and represented by its objects. Every data object has values for the attributes prevalent in the data set. Therefore, the data items itself create a space with their attribute values. The selection's range for an attribute is determined by the selection's items' minimal and maximal values for this particular attribute. If the range is determined by the items themselves, the selection's Data Foundation is *object based* because it relies on the data objects. *Area based* Data Foundation refers to the area that was specified when the selection was first created, drawn by a lasso or brushed over an axis. In this state of the concept, we displayed the lasso and the axis brush in the Visualization View after they were drawn. For selections created by tapping, there is no difference between *object based* and *area based*. In most cases, the drawn area is larger considering the attribute values than the data items' attribute space. Therefore, if the selection's range in attribute values is determined by the area itself, the selection's Data Foundation is *area based*. Examples for the differences in area based Data Foundation and object based Data Foundation are illustrated in.Fig. 5.1

A selection's Data Foundation is irrelevant as long as no data items are hidden by filters and no elements are added or removed from the data set. Hidden data items can be part of an area based selection even though they are not included in the selection's list of items. This case occurs when some items inside the selected area are filtered in the moment the selection is created. The same happens when items are added to the data set and fall into the selection's area. They are not considered part of the selection for an *object based* Data Foundation but for an *area based* Data Foundation. Equally, when items are removed from the data set, they can still be considered part of the *object based* selection but not for the *area based*

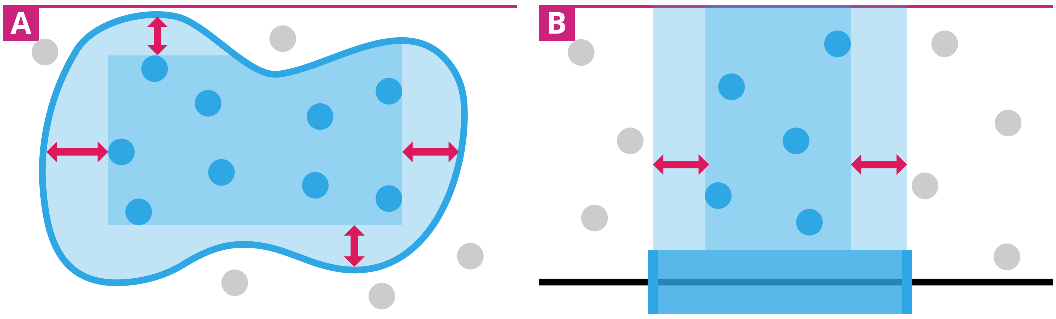


Fig. 5.1 Showing the difference between area based and object based Data Foundation using the example of the *(a)* Lasso and the *(b)* Axis Brush. The lasso's and brush's outline represent the area based Data Foundation whereas the lighter colored rectangle shows the object based Data Foundation. The difference between the two is illustrated by the red arrows.

selection. Further distinction happens when the filter option come into play. With the possibility of changing a selections's Data Foundation, not only items can be filtered and therefore hidden. An *area based* selection could filter a particular attribute range if that is desired by the user. Moreover, we wanted to offer the interacting person to decide which Data Foundation they want to use. Our goal was to be transparent and to not make decisions that do not involve the human interacting with the system. Consequently, we included the manipulation of the Data Foundation into the user interface.

With introducing Data Foundation, we added more features to filters, such as having multiple active filter at once, which lead to complicated processes: Is a filter only active in one view? What happens if elements on one device are filtered but another user wants to interact with them? Can some data elements in an active selection be filtered even though someone is using this selection? How can the user recognize that there are hidden elements in the view? At some point, we made the decision that in this work we want to focus on the selections and the management aspect. Therefore, we omitted the idea of multiple active filters and Data Foundation. However, filtering in a distributed CMV application is still an open problem and adding the ability of Data Foundation offers potential to be explored.

5.1.2 Reviewing the Design Considerations

In Section 3.3, we established Design Considerations that built the foundation for designing the UI concept. In this section, we want to review these considerations and reflect on where they were implemented, how they were implemented, where aspects are missing, and where they can be further improved. Overall, we do not claim that we need to address every consideration in every part of the UI since in particular situations it is not necessary or not appropriate to do so. However, reviewing where the considerations affected the design process and how gives an impression on how meaningful they were. Table 5.1 illustrates the parts of the UI

■ This DC was well addressed in this view.
▧ This DC was partially addressed in this view.
□ This DC was not addressed in this view or not relevant.

	DC 1 Novices vs experts	DC 2 Collaboration	DC 3 Relations between devices	DC 4 Spatially aware vs agnostic	DC 5 Distributed views	DC 6 Device attributes
VISUALIZATION VIEW	■	▧	■	■	▧	■
SELECTION MENU	▧	▧				■
SELECTION LIST	■	■			■	■
DETAIL VIEW	▧	▧				▧
PORTAL VIEW		▧	■	■		▧

Tab. 5.1 Design Considerations and how they were addressed in the parts of the concept.

and how well the Design Considerations were implemented. We want to point out that this assessment is not based on objective criteria but on our subjective opinion. It does not replace a proper evaluation but serves as a reflection. In general, no implementation of a Design Consideration can be considered complete since there is no optimal solution for the problems. Some UI features span more than one view and more than one Design Consideration at the time which makes differentiating them a challenge.

Visualization View

For the Visualization View, **DC 1** is thoroughly implemented. The different needs of novice and expert users are addressed by the two supported ways of creating selections. It is possible to work with and modify two selections at the same time for experts. For novices, editing is more explicit with the Editing Mode which offers a linear approach that is easier to grasp. Collaboration (**DC 2**) is supported to some extent. In general, people can collaboratively interact with one Visualization View at the same time and with multiple Visualization Views on multiple devices. However, when it comes to editing and creating selections, issues arise. Novice users cannot create two selections at a time because of the Editing Mode, which is restricted to one selection at a time. Therefore, after creating a selection, the user needs to exit the Editing Mode before another user can create another selection on the same device. For expert users, the problem is different. More than one selection can be created at the same time by multiple users. The problem arises when two people want to edit two selections at the same time. They hold one part of each selection. There is no way of identifying to what selection the new elements should be added to. This problem could be solved by knowing the person the touch belongs to. However, one person can hold the selection and another person performs a selection technique on the same device to add elements. **DC 3**, which represents visualizing relations between devices, is well supported. In case a Link Style is specified for a selection, the relations between the data elements, thus also devices, are shown.

Furthermore, the Side-by-side Interaction for two Visualization Views draws the links between corresponding data elements on the two devices. The Visualization View adjusts the link representation in case the setting is spatially agnostic by showing the link representations, such as Edge Marks, on top of the screen instead of showing them in the direction of the other device. This addresses **DC 4**. However, **DC 5**, considering distributed views and views on the same screen, is more difficult to define for Visualization Views. The Selection List is accessible via a button in the top right corner and it can be transferred to another device. This is the only way in which **DC 5** is addressed by the Visualization View. For dealing with different device attributes (**DC 6**), the Visualization View supports zooming and panning the visualization for smaller screens. Overall, the Visualization View addresses most of the considerations and is the most extensive part of the UI.

Selection Menu

Regarding novice and expert users (**DC 1**), the Selection Menu is generally more targeted towards novices. It offers the Editing Mode in case the application is in novice mode which makes editing selections easier for them. The Selection Menu supports expert users by allowing them to drag to the item in a subsidiary menu instead of having to tap the top item. However, it lacks more shortcuts for expert users. Allowing the users to open multiple Selection Menus for the same selection on the same device adds to the collaborative setting (**DC 2**). Several people can work on the selection using their own menu. Furthermore, the menus can be rotated and moved to the position the different users desire. However, the Selection Menu could support more features to facilitate collaboration. **DC 3**, which is visualizing the relations between devices, is not addressed in the Selection Menu except for giving access to the linking options. The case is similar for spatially aware and agnostic settings (**DC 4**). Since the Selection Menu does not use the devices's position, there is no need to find a solution for spatially agnostic devices. The Selection Menu does not address distributing views (**DC 5**). Yet, it could be beneficial in some situations to transfer the Selection Menu for one selection to another device. For instance, if the device has a small screen and adding a Selection Menu to the Visualization View would occupy too much space, the Selection Menu could be displayed on another device. This solution also addresses different device attributes (**DC 6**). Furthermore, the Selection Menu offers moving and rotating the menu to accommodate for small form factors. The Selection Menu includes some considerations that are appropriate for a smaller part of the UI but has potential to support more considerations.

Selection List

DC 1 is supported in several ways. The Selection List offers novice users to enter the Editing Mode to modif< selections in a more straightforward way. Experts can select two or more selections by holding and apply actions to them, such as Join Operations. In addition, dragging and dropping of one selection onto another triggers Join Operations which works as a shortcut for more experienced users. Collaboration (**DC 2**) is addressed by allowing Selection Lists on all workspace devices and transferring them onto a separate devices. Thus, more people working with the application can get access to the Selection List. Showing relations to other devices (**DC 3**) is not particularly supported. If the Selection List is on a separate device, the edges of the screen could be use to hint at the positions of other devices in the workspace. This would create more awareness for the other devices in the workspace. For **DC 4**, spatial awareness, the Selection List is similar to the Selection Menu: the devices' positions are not used, thus this consideration does not need to be addressed. The Selection List can be transferred to a separate device by executing a Side-by-side Interaction. This supports distributing views onto separate devices, which is **DC 5**. Furthermore, transferring the view to another device also supports **DC 6**, adapting the UI to device attributes, since it is easier to outsource the views if the devices have little screen real estate. However, as long as the Selection List is located on the same device as a Visualization View, it will fill the whole screen when opened for small devices. In general, the Selection List addresses a lot of considerations but comes short when other devices are involved (**DC 3 + 4**).

Detail View

The Detail View deals the disparate needs of novice and expert users (**DC 1**) on a basic level. In the same way as in the Selection Menu, an experienced user can use dragging to a menu item in a subsidiary menu as a shortcut. Overall, the Detail View is designed for novices in mind since it lists all features and their options. Overall, the Detail View does not particularly focus on the Design Consideration since it serves as a tool to list and access actions. However, the possibility of having multiple Detail Views on multiple devices through transferring a Selection List hints at collaborating (**DC 2**). Relations between devices (**DC 3**) are not explicitly visualized but their visualization can be modified and accessed through the Detail View. Spatial awareness (**DC 4**) is not relevant since the Detail View does not use any directional information about the device. Similarly to the Selection List before, users could benefit from being able to transfer Detail Views to a separate device which considers distributing views (**DC 5**). For **DC 6**, regarding device attributes, the same rules apply as for the Selection List before because they usually share a view. The Detail View lacks in addressing several Design Considerations which shows that there is potential that can be used. Which considerations are appropriate for the Detail View needs to be discussed.

Portal View

The Portal View itself does not have any adjustments for experienced users (**DC 1**) and does not include any special interaction techniques for novices or experts. Collaboration (**DC 2**) is equally considered in the Portal View as in the Detail View. Multiple Portal Views can be used at the same time on multiple devices. Showing relations between devices (**DC 3**) is a main focus for Portal Views. Other workspace devices are displayed in the view as portals that enable the user to interact with other devices. Furthermore, they convey if a selection is active or inactive within a view on a device. The portals are positioned in the relative direction of the device which encapsulates the relation. This means they rely on a spatially aware settings. However, portals are positioned on the screen's edges in case the devices are spatially agnostic (**DC 4**). For distributing views, it could be helpful to be able to transfer a single Portal View to another device which is not included in the concept at the moment (**DC 5**). Adjusting the interface depending on different device attributes (**DC 6**) is done in a basic way similar to the Detail View and Selection List. Overall, the Portal View supports some of the Design Considerations, especially the ones that relate to other devices (**DC 3 + 4**), as expected.

DC 3 is considered to some extend in all views by shaking the device which highlights all workspace devices and thus, shows the relations between devices. In general, Table 5.1 shows that the Design Considerations were well established in the concept since all of them are addressed in at least one part of the interface. It illustrates how different parts of the interface are responsible for different design considerations, e.g., the Visualization View and the Portal View deal with relations between devices (**DC 3**) and spatial awareness (**DC 4**) to an increased extent. Moreover, the table serves as a reminder where improvements are needed, e.g., focussing more on facilitating collaboration in every part, or where the interface has unused potential, e.g., visualizing relations between devices in Selection Menus, Selection Lists, and Detail Views.

5.2 Future Work

For the future, exploring the device's location and the position in relation to other devices is another potential direction for future work which was mentioned by Langner et al. [Lan+17]. Devices can be arranged in various ways: on top of other devices, side-by-side, orthogonal to other devices, only touching a certain part of another device, etc. Additionally, combining more than two devices at once goes hand in hand with exploring new device arrangements. Allowing people to incorporate more than two devices in actions, such as Side-by-side Interactions, can open new design spaces and possibilities for interesting interaction techniques. Introducing different form factors to the setting of distributed CMVs is another interesting step. For instance, combining the current setting with multiple mobile devices with a tabletop, a large display wall, or one or more smartwatches offers compelling scenarios. Large displays could be used for overviews or to assemble multiple visualizations on one device if that is needed. Smartwatches, however, could serve as containers for context aware menus.

Incorporating other device categories not only helps with using the location but they can be used to further enhance the visualization of device and data element relations. Additional technologies that could be of use are tabletops, light sources, or AR glasses like the Microsoft HoloLens. Examples what could be done with this additional hardware can be seen in Figs. 5.2 and 5.3. The colored marks either representing links or highlighting a device can be created by a projector, displayed by an underlying tabletop or represented in AR using a HoloLens. By integrating one of these techniques, we enable the system to draw explicit links between data elements and devices while still having the ability to move devices around or pick them up. A less expensive option for highlighting a device is using separate light source, such as LEDs, that are attached to the devices. Although we cannot show links with this method, we can show direction and highlight the device itself, as illustrated in the sketch in Fig. 5.3. Furthermore, the amount of links that exist when working with more than two selections are not feasible. Therefore, we consider implementing edge bundles or bundled links, as seen in SAViL [CN17].

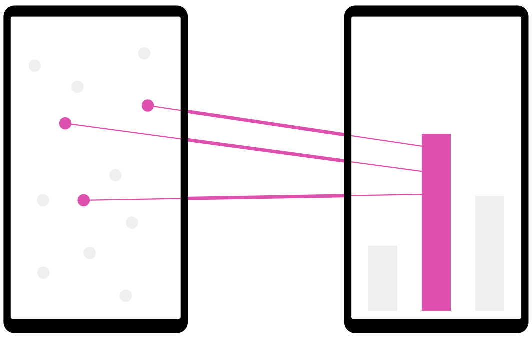


Fig. 5.2 Drawing links between two linked devices on the screen and in between with help of projection or an underlying display.

Conducting an observational study with the prototype should be done to see how people respond to the introduced brushing and linking and selection management. Moreover, the techniques we introduced to create awareness, such as shaking devices to make workspace devices vibrate or the different Link Styles, are worth evaluating as well. Getting people's response on how they perceive them, could be helpful for

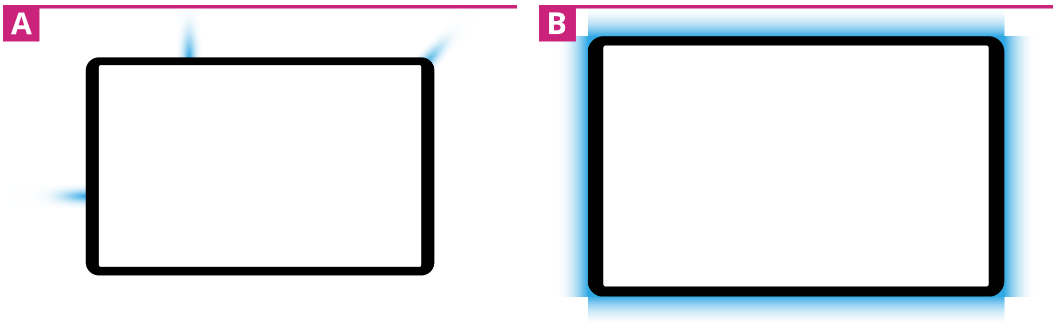
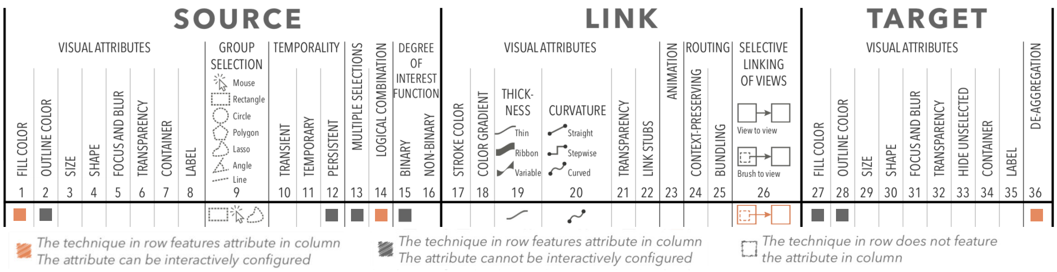


Fig. 5.3 Attaching lights to the devices can either show *(a)* directional light or *(b)* non-directional light that highlights a device.

refining the UI concept in the future, for instance further investigating the use of audio feedback and haptic feedback, i.e., vibration, on a table.

5.3 Conclusion

Since this thesis was partly inspired by MyBrush [Koy+17], to conclude it we want to relate our own concept to the design space created by Koytek et al. [Koy+17]. How the entry for the table they used would look like for our concept, is illustrated in Table 5.2. It shows how this thesis relates to the design space, i.e., it is part of brushing and linking. However, it additionally focusses on other aspects, such as selection management.



Tab. 5.2 The table that shows the design space from Koytek et al. [Koy+17] with the entry that represents our concept.

This thesis was an exploration on how to manage selection and use brushing and linking in the context of multiple coordinated views via mobile devices. We proposed a conceptual framework that informs the design of selection management in multi-device environments, a UI concept for selection management for multiple mobile devices, as well as plans for a proof-of-concept prototype. In doing so, we addressed the following questions:

- *How might we create awareness using brushing and linking?*.
  We examined how brushing and linking can extend beyond a single device and create awareness both for an individual working with multiple devices,

as well as when multiple people are interacting with data. We did so by including multiple styles of links between data points and devices that use spatial awareness. Furthermore, with the Portal View we enable the user to explicitly control the links through a spatially-aware view.

- *How might we support storing, combining, managing, and interacting with selections?*.
  We examined a series of selection techniques to examine storing, combining and management of such selections. In my framework and UI concept, I extend current work using selections so that they can be used at later times, shared between devices, and even shared between different individuals in the workspace. UI elements, like the Selection List, the Selection Menu, and the Detail View, are responsible for managing selections, combining, and storing them. Additionally, various Side-by-side interaction further add awareness and offer opportunities to exploit the spatial setting to interact with selections.

Bibliography

[And+11] Christopher Andrews, Alex Endert, Beth Yost, and Chris North. „Information visualization on large, high-resolution displays: Issues, challenges, and opportunities“. en. In: *Information Visualization* 10.4 (Oct. 2011), pp. 341–355 (cit. on p. 1).

[Bal+07] Robert Ball, Chris North, and Doug A. Bowman. „Move to Improve: Promoting Physical Navigation to Increase User Performance with Large Displays“. In: ACM, Apr. 2007, pp. 191–200 (cit. on p. 1).

[Bau+12] Dominikus Baur, Bongshin Lee, and Sheelagh Carpendale. „TouchWave: Kinetic Multi-touch Manipulation for Hierarchical Stacked Graphs“. In: *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’12. New York, NY, USA: ACM, 2012, pp. 255–264 (cit. on p. 7).

[BC87] Richard A. Becker and William S. Cleveland. „Brushing Scatterplots“. In: *Technometrics* 29.2 (May 1987), pp. 127–142 (cit. on p. 13).

[BG03] D. Brodbeck and L. Girardin. „Design study: using multiple coordinated views to analyze geo-referenced high-dimensional datasets“. In: *Proceedings International Conference on Coordinated and Multiple Views in Exploratory Visualization - CMV 2003* -. July 2003, pp. 104–111 (cit. on p. 17).

[Blu+15] Kerstin Blumenstein, Markus Wagner, and Wolfgang Aigner. „Cross-Platform InfoVis Frameworks for Multiple Users, Screens and Devices: Requirements and Challenges“. In: *Proceedings of the DEXIS 2015 Workshop on Data Exploration for Interactive Surfaces*. Nov. 2015 (cit. on pp. 10, 24).

[Bos+11] M. Bostock, V. Ogievetsky, and J. Heer. „D3: Data-Driven Documents“. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (Dec. 2011), pp. 2301–2309 (cit. on p. 13).

[BR03] Patrick Baudisch and Ruth Rosenholtz. „Halo: A Technique for Visualizing Off-screen Objects“. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’03. New York, NY, USA: ACM, 2003, pp. 481–488 (cit. on pp. 8, 15, 38, 59).

[Bre+16] M. Brehmer, J. Ng, K. Tate, and T. Munzner. „Matches, Mismatches, and Methods: Multiple-View Workflows for Energy Portfolio Analysis“. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), pp. 449–458 (cit. on pp. 11, 13).

[CC07] C. Collins and S. Carpendale. „VisLink: Revealing Relationships Amongst Visualizations“. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (Nov. 2007), pp. 1192–1199 (cit. on pp. 11, 13, 14, 17).

[Chi06] Luca Chittaro. „Visualizing Information on Mobile Devices". In: *Computer* 39.3 (Mar. 2006), pp. 40–45 (cit. on p. 7).

[Chu+14] Haeyong Chung, Chris North, Jessica Zeitz Self, Sharon Chu, and Francis Quek. „VisPorter: Facilitating Information Sharing for Collaborative Sensemaking on Multiple Displays". In: *Personal Ubiquitous Comput.* 18.5 (June 2014), pp. 1169–1186 (cit. on pp. 2, 9, 10).

[Chu+15] H. Chung, C. North, S. Joshi, and Jian Chen. „Four considerations for supporting visual analysis in display ecologies". In: *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*. Oct. 2015, pp. 33–40 (cit. on pp. 9, 24).

[CN17] Haeyong Chung and Chris North. „SAViL: cross-display visual links for sensemaking in display ecologies". en. In: *Personal and Ubiquitous Computing* (Dec. 2017), pp. 1–23 (cit. on pp. 17, 31, 80).

[CW11] J. H. T. Claessen and J. J. van Wijk. „Flexible Linked Axes for Multivariate Data Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (Dec. 2011), pp. 2310–2316 (cit. on pp. 14, 17).

[Dru+13] Steven M. Drucker, Danyel Fisher, Ramik Sadana, Jessica Herron, and m.c. schraefel. „TouchViz: A Case Study Comparing Two Interfaces for Data Analytics on Tablets". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. New York, NY, USA: ACM, 2013, pp. 2301–2310 (cit. on p. 7).

[Eis08] Michael B. Eisenberg. „Information Literacy: Essential Skills for the Information Age". In: *DESIDOC Journal of Library & Information Technology* 28 (Mar. 2008), pp. 39–47 (cit. on p. 1).

[Elm+08] N. Elmqvist, P. Dragicevic, and J. D. Fekete. „Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation". In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (Nov. 2008), pp. 1539–1148 (cit. on pp. 13, 18, 19, 37, 49).

[Elm+11] Niklas Elmqvist, Andrew Vande Moere, Hans-Christian Jetter, et al. „Fluid interaction for information visualization". en. In: *Information Visualization* 10.4 (Oct. 2011), pp. 327–340 (cit. on pp. 1, 5).

[Gel+09] Hans Gellersen, Carl Fischer, Dominique Guinard, et al. „Supporting Device Discovery and Spontaneous Interaction with Spatial References". In: *Personal Ubiquitous Comput.* 13.4 (May 2009), pp. 255–264 (cit. on pp. 15, 16, 38, 59).

[Gey+14] Thomas Geymayer, Markus Steinberger, Alexander Lex, Marc Streit, and Dieter Schmalstieg. „Show Me the Invisible: Visualizing Hidden Content". In: *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*. CHI '14. New York, NY, USA: ACM, 2014, pp. 3705–3714 (cit. on p. 14).

[Gra+14] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit. „Domino: Extracting, Comparing, and Manipulating Subsets Across Multiple Tabular Datasets". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 2023–2032 (cit. on p. 14).

[Gus+08] Sean Gustafson, Patrick Baudisch, Carl Gutwin, and Pourang Irani. „Wedge: Clutter-free Visualization of Off-screen Locations". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. New York, NY, USA: ACM, 2008, pp. 787–796 (cit. on pp. 8, 15, 38, 59).

[Hea99] Marti Hearst. „User Interfaces and Visualization". In: *Modern Information Retrieval*. Ed. by Ricardo Baeza-Yates and Berthier Ribeiro-Neto. Addison Wesley Longman, 1999, pp. 257–323 (cit. on p. 13).

[Hor+18] 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". en. In: *Procedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. New York, NY, USA: ACM, 2018, p. 13 (cit. on pp. 12, 13, 15, 17–19, 37, 45).

[Hua+15] D. Huang, M. Tory, B. Adriel Aseniero, et al. „Personal Visualization and Personal Visual Analytics". In: *IEEE Transactions on Visualization and Computer Graphics* 21.3 (Mar. 2015), pp. 420–433 (cit. on p. 1).

[HW14] Peter Hamilton and Daniel J. Wigdor. „Conductor: Enabling and Understanding Cross-device Interaction". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. New York, NY, USA: ACM, 2014, pp. 2773–2782 (cit. on pp. 9, 16).

[Ise+13] P. Isenberg, T. Isenberg, T. Hesselmann, et al. „Data Visualization on Interactive Surfaces: A Research Agenda". In: *IEEE Computer Graphics and Applications* 33.2 (Mar. 2013), pp. 16–24 (cit. on pp. 5, 8, 9, 24).

[JD13] Y. Jansen and P. Dragicevic. „An Interaction Model for Visualizations Beyond The Desktop". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2396–2405 (cit. on p. 5).

[JE12] W. Javed and N. Elmqvist. „Exploring the design space of composite visualization". In: *2012 IEEE Pacific Visualization Symposium*. Feb. 2012, pp. 1–8 (cit. on p. 11).

[Jok+15] Tero Jokela, Ming Ki Chong, Andrés Lucero, and Hans Gellersen. „Connecting Devices for Collaborative Interactions". In: *interactions* 22.4 (June 2015), pp. 39–43 (cit. on p. 8).

[Jä+08] H. Jänicke, M. Böttinger, and G. Scheuermann. „Brushing of Attribute Clouds for the Visualization of Multivariate Data". In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (Nov. 2008), pp. 1459–1466 (cit. on p. 17).

[Kei02] D. A. Keim. „Information visualization and visual data mining". In: *IEEE Transactions on Visualization and Computer Graphics* 8.1 (Jan. 2002), pp. 1–8 (cit. on pp. 2, 13).

[Kon+06] Z. Konyha, K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser. „Interactive Visual Analysis of Families of Function Graphs". In: *IEEE Transactions on Visualization and Computer Graphics* 12.6 (Nov. 2006), pp. 1373–1385 (cit. on pp. 17, 37).

[Koy+17] P. Koytek, C. Perin, J. Vermeulen, E. André, and S. Carpendale. „MyBrush: Brushing and Linking with Personal Agency“. In: *IEEE Transactions on Visualization and Computer Graphics* PP.99 (2017), pp. 1–1 (cit. on pp. 2, 13, 14, 17, 22, 23, 27, 36, 40, 41, 81).

[Lan+14] Ricardo Langner, Anton Augsburg, and Raimund Dachselt. „CubeQuery: Tangible Interface for Creating and Manipulating Database Queries“. In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’14. New York, NY, USA: ACM, 2014, pp. 423–426 (cit. on p. 10).

[Lan+17] R. Langner, T. Horak, and R. Dachselt. „VISTILES: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration“. In: *IEEE Transactions on Visualization and Computer Graphics* PP.99 (2017), pp. 1–1 (cit. on pp. 2, 12, 15, 17, 21, 23, 26, 30, 41, 67, 68, 79).

[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 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI ’15. New York, NY, USA: ACM, 2015, pp. 187–198 (cit. on p. 9).

[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 (cit. on pp. 1, 5).

[Lex+13] A. Lex, C. Partl, D. Kalkofen, et al. „Entourage: Visualizing Relationships between Biological Pathways using Contextual Subsets“. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2536–2545 (cit. on p. 14).

[Luc+11] Andrés Lucero, Jussi Holopainen, and Tero Jokela. „Pass-them-around: Collaborative Use of Mobile Phones for Photo Sharing“. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’11. New York, NY, USA: ACM, 2011, pp. 1787–1796 (cit. on p. 8).

[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 2012 ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’12. New York, NY, USA: ACM, 2012, pp. 31–40 (cit. on pp. 2, 9, 16, 38, 59).

[ME13] S. MacNeil and N. Elmqvist. „Visualization Mosaics for Multivariate Visual Exploration“. In: *Comput. Graph. Forum* 32.6 (Sept. 2013), pp. 38–50 (cit. on pp. 11, 17).

[Per+14] C. Perin, P. Dragicevic, and J. D. 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 (cit. on pp. 5, 6).

[Pia+13] Tommaso Piazza, Morten Fjeld, Gonzalo Ramos, AsimEvren Yantac, and Shengdong Zhao. „Holy Smartphones and Tablets, Batman!: Mobile Interaction’s Dynamic Duo“. In: *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*. APCHI ’13. New York, NY, USA: ACM, 2013, pp. 63–72 (cit. on pp. 2, 9, 16).

[RK13] Jeffrey M Rzeszotarski and Aniket Kittur. „TouchViz: (Multi)Touching Multivariate Data“. en. In: *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013, pp. 1779–1784 (cit. on pp. 2, 7).

[RK14] Jeffrey M. Rzeszotarski and Aniket Kittur. „Kinetica: Naturalistic Multi-touch Data Visualization“. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. New York, NY, USA: ACM, 2014, pp. 897–906 (cit. on pp. 2, 7, 8).

[Rob+14] J. C. Roberts, P. D. Ritsos, S. K. Badam, et al. „Visualization beyond the Desktop–the Next Big Thing“. In: *IEEE Computer Graphics and Applications* 34.6 (Nov. 2014), pp. 26–34 (cit. on pp. 1, 5, 7).

[Rob07] J. C. Roberts. „State of the Art: Coordinated Multiple Views in Exploratory Visualization“. In: *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*. July 2007, pp. 61–71 (cit. on pp. 2, 11, 13).

[Rob98] J. C. Roberts. „On encouraging multiple views for visualization“. In: *Proceedings. 1998 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics (Cat. No.98TB100246)*. July 1998, pp. 8–14 (cit. on p. 10).

[Rä+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 Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ITS '14. New York, NY, USA: ACM, 2014, pp. 45–54 (cit. on p. 9).

[Rä+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 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. New York, NY, USA: ACM, 2015, pp. 3913–3922 (cit. on pp. 9, 16, 38, 59).

[Sie+05] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. „Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs“. en. In: *Human-Computer Interaction - INTERACT 2005*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Sept. 2005, pp. 267–280 (cit. on p. 5).

[SS14] Ramik Sadana and John Stasko. „Designing and Implementing an Interactive Scatterplot Visualization for a Tablet Computer“. In: *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*. AVI '14. New York, NY, USA: ACM, 2014, pp. 265–272 (cit. on pp. 2, 5, 7, 17, 27).

[SS16a] Ramik Sadana and John Stasko. „Designing Multiple Coordinated Visualizations for Tablets“. en. In: *Computer Graphics Forum* 35.3 (June 2016), pp. 261–270 (cit. on pp. 12, 15, 17).

[SS16b] Ramik Sadana and John Stasko. „Expanding Selection for Information Visualization Systems on Tablet Devices“. In: *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*. ISS '16. New York, NY, USA: ACM, 2016, pp. 149–158 (cit. on pp. 2, 17).

[Sta+16] J. Stahnke, M. Dörk, B. Müller, and A. Thom. „Probing Projections: Interaction Techniques for Interpreting Arrangements and Errors of Dimensionality Reductions“. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), pp. 629–638 (cit. on pp. 18, 19, 24, 47).

[Tab18] Tableau. *Tableau Public*. en. 2018 (cit. on pp. 1, 10, 13).

[The02] Martin Theus. „Interactive Data Visualization using Mondrian“. In: *Journal of Statistical Software* 7.11 (2002), pp. 1–12 (cit. on pp. 18, 37).

[Tob+09] Matthew Tobiasz, Petra Isenberg, and Sheelagh Carpendale. „Lark: Coordinating Co-located Collaboration with Information Visualization“. In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (Nov. 2009), pp. 1065–1072 (cit. on pp. 11, 12, 14).

[VM12] C. Viau and M. J. McGuffin. „ConnectedCharts: Explicit Visualization of Relationships between Data Graphics“. en. In: *Computer Graphics Forum* 31.3pt4 (June 2012), pp. 1285–1294 (cit. on pp. 13, 17).

[Voi02] Robert Voigt. „An Extended Scatterplot Matrix and Case Studies in Information Visualization.“ PhD thesis. Oct. 2002 (cit. on pp. 2, 13).

[Wal+17] James R. Wallace, Steven Houben, Craig Anslow, et al. „The Disappearing Tabletop: Social and Technical Challenges for Cross-Surface Collaboration“. In: *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. ISS ’17. New York, NY, USA: ACM, 2017, pp. 482–487 (cit. on p. 2).

[WB+00] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. „Guidelines for Using Multiple Views in Information Visualization“. In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI ’00. New York, NY, USA: ACM, 2000, pp. 110–119 (cit. on p. 11).

[WK] Gary Wolf and Kevin Kelly. *Quantified Self - Self Knowledge Through Numbers* (cit. on p. 1).

[Woz+14] Pawel Wozniak, Lars Lischke, Benjamin Schmidt, Shengdong Zhao, and Morten Fjeld. „Thaddeus: A Dual Device Interaction Space for Exploring Information Visualisation“. In: *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. NordiCHI ’14. New York, NY, USA: ACM, 2014, pp. 41–50 (cit. on pp. 10, 12, 15, 17).

[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 Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’14. New York, NY, USA: ACM, 2014, pp. 129–138 (cit. on pp. 10, 15, 17).

List of Figures

List of Tables

Colophon

This thesis was typeset with $\LaTeX\ 2_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.

Declaration

I, Tamara Flemisch, declare that I have developed and written the enclosed Master's Thesis with the title *Effects of Different Display Form Factors on InfoVis Applications: Exploring Selection Management and Brushing and Linking for Mobile Cross-Device Interaction* completely by myself, and have not used sources or means without declaration in the text. Any thoughts from others, literal quotations, or images are clearly marked. The Master's Thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

Dresden, April 17, 2018

Tamara Flemisch