Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit Nr. 243

Exploring Complex Data Sets in Multi-Display Environments

Jan Hoffmann

Course of Study:	Informatik
Examiner:	Prof. Dr. Albrecht Schmidt
Supervisor:	Dipl.-Inf. Lars Lischke, Patrick Bader M.Sc., Robert Krüger M.Sc.
Commenced:	June 10, 2015
Completed:	December 10, 2015
CR-Classification:	H.5.2

Kurzfassung

In vielen Disziplinen besteht die Notwendigkeit, große Datensätze zu explorieren und zu verstehen. Um diesen Prozess zu unterstützen ist es erforderlich, neue Interaktionstechniken zu entwickeln. Große und hochauflösende Displays ermöglichen die Darstellung vieler Informationen auf einmal. Dies ist besonders nützlich im Zusammenhang mit räumlichen Datensätzen. Allerdings gibt es keine etablierten Benutzerschnittstellen für solche großen Display-Anordnungen. Ein vielversprechender Ansatz ist die Verbindung eines großen hochauflösenden Displays mit zusätzlichen Bildschirmen, wie Tablets, Smartphones oder Desktop-Computern. In dieser Arbeit werden solche Mehrgeräte-Schnittstellen entworfen, implementiert und evaluiert. Hierzu benutzen wir geolokalisierte Mikroblog-Daten des Dienstes Twitter. Als zusätzliche Geräte betrachten wir Tablets. Zunächst entwerfen wir verschiedene Konzepte für die Interaktion zwischen großen Displays und Tablets. Anschließend werden einige dieser Konzepte implementiert. Um die Nützlichkeit zu bestätigen führen wir eine Benutzerstudie durch, in der wir unser System einer Vergleichsoberfläche für Desktop-Rechner gegenüberstellen. In den Ergebnissen zeigt sich, dass die meisten Benutzer die Tablet-Oberfläche bevorzugen.

Abstract

In many disciplines there is a need for exploring and understanding large data sets. In order to support this process it is required to develop new interaction techniques. Large high-resolution displays offer the opportunity to show a lot of information at once. This is especially useful in relation to spatial data sets. However, there are no established user interfaces for such large-screen setups. A promising approach is to connect a large high-resolution display with additional screens, such as tablets, smartphones or desktop computers. In this thesis, we design, implement and evaluate such a multi-device interface. In order to do this, we use geolocated microblog data from the Twitter service. We use tablet computers as devices for interacting with the system. As a first step, we design a number of different concepts for user interaction involving large displays and tablets. Then, we implement some of the created concepts. In order to confirm the usefulness of our system, we compare it to a reference interface for desktop computers. The results show that most users prefer interacting with the tablet interface.

Contents

List of Figures

List of Tables

1 Introduction

Today, the amount of data that is available and stored has become quite large and is still increasing. Digital information plays a very important role in a wide range of disciplines. Among these are emergency response and uses in businesses, as well as in science. However, due to the large amount of data it is not possible to use it directly. The field of visual analytics provides a set of solution for this issue. This is achieved by combining the capabilities of a computer with the abilities of a human user. An example for data to be analyzed are geolocated microblog posts as available on Twitter. A use case for Twitter data is the analysis of a natural disaster such as a flood or an earthquake.

Another recent development exists in advancement of technology. For example, display technology has evolved considerably. Today, large screens with diagonals of more than a meter and high resolutions of more than eight million pixels are available. At the same time, the prices for such devices have become lower over time. This development has made it much more affordable to build large wall-sized displays. Such a display wall consisting of multiple high resolution screens can be run using a single computer, making it possible to use standard applications on it.

Large wall displays can be used for visualization purposes. There are several advantages from this approach. For example, due to the high resolution, it is possible to show small details on the display. At the same time, the user can get an overview without having to modify the actual screen content by just moving around. A large screen size and the possibility to move around can also be a useful aspects when the system is used by multiple people at once.

While large display walls are well-suited for visualization of data, interacting with such devices is still challenging. Traditional concepts of interaction with computers, such as mouse and keyboard, were designed more than three decades ago. At that time, large displays were not available, so it was not a consideration when these interaction methods were developed. Among the input issues regarding large displays are long travel distances when using a mouse as pointing device. Another issue is the static position of the input devices, as mouse and keyboard are usually not easily portable. This is a concern because the user may move around while interacting with a large display.

Thus, a promising approach is to combine a large display with a mobile device that can be used for interaction. Mobile computers have become increasingly smaller recently. This has lead to new form factors such as smartphones and tablets. The traditionally used interaction methods are no longer relevant on such devices. Instead, touch input has become a central aspect of modern portable devices, allowing users to directly manipulate content on a screen. In addition to touch input, many devices include a variety of sensors such as accelerometers. These sensors open new opportunities for interaction, such as incorporating movement of the device itself.

In the end, a combination of the aforementioned technologies can be useful. In this thesis, a system using a large display for visualization purposes together with mobile devices for user interaction is developed and presented. The system allows to combine the advantages of both technologies in order to support visual analytics, which is an important technique to master the amount of available data and make use of it. By using a mobile device the user can move around and either view details or get an overview. Another aspect is the possibility to add multiple control devices to the system, which facilitates collaborative work. In order to evaluate the usefulness of the system, we conduct a user study and discuss the results. For the implementation and evaluation of the system, geolocated data sets from Twitter are used.

Structure

The chapters of this thesis are structured in the following way:

Chapter 2 – Related Work: This section gives an overview on previous work in the fields of visual analytics and multi-device interaction.

Chapter 3 – Design of the System: The process of creating the prototypes for our system is described here along with detailed descriptions of different interaction models.

Chapter 4 – Implementation: Our software is described here in detail, beginning with an overview of the components and followed by information about the implementations of each of them.

Chapter 5 – Evaluation: A user study was conducted to evaluate our system. This chapter describes the design of the study.

Chapter 6 – Analysis and Results: In this chapter, the results of the study are analyzed and discussed with regards to the usefulness of the system.

Chapter 7 – Conclusion and Future Work: Finally, the last chapter gives a summary of our own work and an outlook on future work related to multi-device interaction with large data sets.

2 Related Work

As the system presented in this thesis is a tool for visual analytics, the first part of the related work chapter is devoted to visual analytics. It starts with a detailed overview on visual analytics in general. Basic ideas and common challenges in the field are described. After that, a few example applications are shown. A more detailed look is also given at collaborative aspects in relation to visual analytics, as that is also a focus for this thesis.

The second part of this chapter deals with large displays and multi-device interaction. For this, several previously developed systems involving large displays and multiple devices are presented. This also gives an overview over basic ideas utilized in multi-device interaction systems. Some of the presented approaches also incorporate collaborative elements, similar to our own system.

2.1 Visual Analytics

The book “Mastering the Information Age: Solving Problems with Visual Analytics” by Keim et al. [KKEM10] offers a detailed introduction to visual analytics. As decribed in the introduction, visual analytics is an important field because of the amount of data available today. This means that existing standard tools for data analysis are not sufficient. In order to solve the requirement of finding relevant and useful data within large data sets, visual analytics can be used. The basic idea is to combine the analysis capabilities of computer programs with human abilities in order to find relevant data and interpret it. In order to do this, data mining and data management techniques are used.

In a general process for visual analytics, the first step is data transformation. This is a preprocessing step that transforms the data into another representation that is more suited for the following tasks. Then, the data can either be visualized and interacted with or one applies data mining. In both cases, new knowledge is created that can be used for further refinement steps. In our thesis, the main focus is on visualization and user interaction.

A common issue for visual analytics is to obtain knowledge relating to data in time and space. This is also what is done in this thesis by analyzing geolocated data sets. There are usually dependencies or relations between nearby objects. A challenge is uncertainty which is caused by the original data quality. An important aspect for the analysis is scale, because some patterns are only visible in an overview while others are very localized. Overplotting can also be an issue, because it hides relevant data within a flood of mostly irrelevant data.

Space can be represented in different ways. A standard approach with a long history is cartography. It can be described as an abstract visualization of a selection of reality. Geographic information systems allow to manage data and perform computational analysis. However, the main goal of such systems is the creation of maps and they are not primarily designed for interactive analysis of large data sets. Geovisualization allows to perform visual exploration, analysis, and presentation of geographic data. The main aspect is to provide dynamic and interactive maps that can be used for exploration.

For the representation of time, different types of time have to be considered. First, there is the distinction between a point in time and a time interval. Second, time can either be linear or circular. Cycles are a common aspect for time, an example for this are seasons or holidays, which repeat more or less regularly.

Pirolli and Card [PC05] describe a process for sensemaking during the analysis of data sets. Their basic model is a system of feedback loops, where the effort and structure is increasing over time. The model starts with external data sources, which are processed by searching and filtering. The result consists of relevant documents and is called shoebox. The next step is reading and extraction, which leads to relevant snippets and low-level inferences that are stored in evidence files. Now, the information is represented in a schematic way. The next task is to build a theory or case from the schema that supports or rejects a hypothesis. Finally, the results are presented or publicized. For all the mentioned steps, there is a corresponding step in the opposite direction that allows to confirm or refine findings, creating feedback loops.

The value of information visualization has been evaluated by Fekete et al. [FVWSN08]. Their goal was to understand the qualities of visual analytics. For this, they considered different points of view leading to arguments that explain the value of the field.

Their first argument is related to cognition and perception. Visual representations help the human brain in understanding things. The reason for this is that visualization reduces the required memory and processing resources. Search effort is reduced and pattern recognition in visualizations helps for understanding. An example for this is Gestalt theory, which allows the human to detect patterns. There are different aspects for this, such as proximity, similarity, continuity, symmetry, closure and relative size.

In general, it is useful to prefer visualization over automatic analysis when there is no clear idea of the model. Because the human vision system is very good at filtering out noise, visual exploration allows to find patterns that could not be easily detected by computers.

Another explanation for the value of visual analytics is to give success stories. One example is the cholera epidemic in London in 1854. John Snow represented the deaths on a map and found that they were clustered around a single point. This lead to the discovery of a bad water pump responsible for the epidemic.

2.1.1 Applications

There are many applications for visual analytics. Examples named by Keim et al. [KKEM10] are astrophysics, climate and weather monitoring, emergency management, biology and medicine, as well as business intelligence.

One example for an application is disaster response in relation to critical infrastructure. Mittelstädt et al. [MWE+15] present an approach for this application. The power grid, communication networks and streets are part of critical infrastructure, because today's world depends on them. There are dependencies between these different networks, and thus the results of a failure in one of them are difficult to discover or even predict. In their system, Mittelstädt et al. use real-time analysis of social media data, as well as people that are in the field. By using a simulation model, some of the effects in infrastructure can be predicted and visualized on screen. In addition to that, human sensor information is incorporated. This means that unstructured data from social media is used for detection of anomalies and visual exploration. For this aspect, a large display can be used to aid in the exploration. As a third source of insight, some users are working in the field. They can communicate with the others using a mobile visualization application.

A system for emergency response is also described by Kim et al. [KJM+07]. They use interactive visualization techniques on mobile devices for reacting to the situation. As a part of that, visualizations of sensor data, locations and video data are presented to the helpers on site. The benefit of this approach is the improved ability to make decisions. It also enhances situational awareness, by showing locations of helpers and areas of danger.

2.1.2 Collaborative Work

As described by Heer and Agrawala [HA08], visual analytics often has collaborative aspects. Reasons for this can be the desire to get a deeper understanding of the data or simply being able to cope with the large size of a data set. An essential distinction to be made regarding collaborative work is between synchronous and asynchronous collaboration.

The paper describes some general aspects of collaboration that are shown in the following. A fundamental aspect of collaborative work is the division of labor among multiple people. Finding common ground and awareness of others' activities are the basis for communication between collaborators. In order to achieve this, references are an important aspect. References allow to identify and refer to specific objects within the visualization. There exist a variety of incentives for experts to take part in the analysis. Among these are personal relevance of the subject, unique tasks and aspects of gameplay. The interaction between users in a collaborative system is affected by identity, trust and reputation. Aspects of group dynamics also have to be considered for collaborative work, as the size and diversity of groups affects the results. Mechanisms for group formation are required, if the groups are not already established. As a result of collaborative visual analytics, a consensus has to be reached by aggregation of individual decisions.

Based on another paper by Heer et al. [HVHC$^+$08], we take a more detailed look on aspects of collaboration in visual analytics. There are two different types of collaborative work. On the one hand it is possible to work together asynchronously by using a internet connection to communicate. On the other hand, it is possible for the users to be co-located, for example by using wall-sized displays or tabletop displays in a single room. Because our own system provides an approach for co-located collaboration, we focus on describing this aspect here. The benefit of co-located collaboration depends on the actual implementation of the visualization tool. In some cases, users may instead prefer to work independently.

First, we look at the hardware for a collaborative visualization environment. The display size is a very important aspect. It is not sufficient if the display is large enough to show all the information, but it needs to be large enough to be shared and viewed by all users of the system at the same time. Different display configurations can be used. The main differentiation here is between connected individual displays and a single large interactive display that is shared by the collaborators. The tradeoff to be made is between providing private views and encouraging to work together. The resolution of the system needs to be high enough for the content to be legible. Also, the accuracy of the input devices should be considered in relation to the display resolution. Regarding interaction with the system, every user should have a means of input. However, a common issue is to handle multiple users interacting with the same object at a time, which may cause conflicts.

In a collaborative environment, communication is a key aspect. On the one hand it can appear in an explicit form, for example by speaking or by using gestures. On the other hand, implicit communication also plays a role. It occurs in the form of body language or digital actions on the screen. However, an issue is that such means of communication may be unnoticed by others, especially for digital actions. Coordination is also an important aspect. Some users may want to have a division into different areas for personal and group work as well as for storage. Thus, it can be useful to provide sharing spaces and private spaces within the system.

The paper also shows some common design challenges. Data can be represented in different ways, using a different manner of encoding. This affects the efficiency for a task, and users' preferences may differ. The presentation is given by the way in which the encoded data is laid out in space. The display size needs to be taken into account here. In order to support the needs of multiple collaborators, it should be possible to focus on multiple aspects at the same time in the visualization. All users should have their own view, which can also change when they move. For interaction with the system it is important that the response rate of the system is adequate. Another aspect is that changes should require few actions and no switch between input modalities in order to provide fluid interaction. Also, there should be a kind of history to help understand the actions of others.

The alternative to co-located collaboration is an asynchronous system. Possible advantages are broader discussions, more complete reports and longer solutions. There are also distinct challenges compared to co-located collaboration, such as ensuring coordination and awareness.

2.2 Multi-Device Interaction

Chapuis et al. [CBF14] have developed an input system for wall displays called Smarties. Their system connects a wall display with a mobile application that can run on a tablet. The tablet application displays small icons called pucks. Each puck corresponds to an object on the wall display. The pucks can be manipulated on the tablet, for example by moving them. The changes to the pucks are synchronized with the corresponding objects on the main display. An example application is a map where the pucks correspond to magnification lenses that allow to show some parts of the map in detail. The Smarties system provides a freely available software framework that includes the tablet application. The general approach of Chapuis et al. is similar to our approach. However, for a map application other interaction methods than pucks are better suited for the tablet interface.

A quite different approach is to use the position of an object to interact with a system. Spindler et al. [SMD12] have developed a system for interacting using position above a table. In their system an object is moved above a tabletop and the displayed content varies depending on the position. This means that there are different levels depending on the distance from the table. The moved object can be described as a Magic Lens. There are multiple ways to implement such a Magic Lens. One is to use a device with a built-in screen, such as a tablet or a smartphone. The other is to use a project an image onto an object without a display such as a piece of cardboard.

When multiple electronic devices are used in a system together, a way for establishing the connection between devices is necessary. Aumi et al. [AGG+13] created a system called DopLink that makes use of the doppler effect. The system allows to detect relative positions of multiple devices and to select a device to interact with. They use the built-in audio hardware of mobile devices to play and receive ultrasonic noises, which allows to use the doppler effect to measure relative movements of a device to another one. Among the applications of the system are pairing of devices and synchronization of multiple displays.

The UbiCicero system by Ghiani et al. [GPSS09, GPS09] provides a location-aware museum guide using multiple devices. The system uses a Portable Digital Assistant (PDA) that shows supplementary information to an exhibition. The range of possible features goes from showing information on a piece of art to providing fitting mini-games such as quizzes. In order to determine the position of mobile devices, the system uses Radio Frequency Identification (RFID) technologies. The system also supports stationary displays that can be operated independently using mouse and keyboard. When the user is nearby one of the stationary displays, it can be used as an extension of the PDA. For example, in a quiz corresponding images may be moved to the larger stationary screen, which provides more space for the input controls on the PDA. For editing purposes, the UbiCicero system contains a graphical tool that allows to modify the presented content.

In order to facilitate the creation of content for multi-device interfaces, Braun et al. [BHKM04] have developed an authoring system. The system allows to distribute the application onto different devices automatically at runtime. The different parts of the interface are distributed in a way that reduces the effort for interaction. An example for this is a music player that shows detailed information and a playlist on a large screen and control buttons on a mobile device. Technically, the user interface is described using a language based on XForms and XHTML. A component called dialog manager is used as an intermediate layer between the actual application and the display devices.

A system that takes the location of the user into account was developed by Nacenta et al. [NSY+07] under the name E-conic. In multi-display environments, the user does not necessarily look at the display at a perpendicular angle. Due to perspective, this can lead to distorted text that is difficult to read. To solve this, the system detects the user's head position and transforms the content such that user perceives it as if he was looking at the display at a perpendicular angle.

A similar approach that uses transformations to simulate the position of display devices was created by Baur et al. [BBF12]. As a metaphor for interaction, the system simulates projection of contents from a device without an optical projector. It combines a mobile computing device with a larger stationary display. For this, the content of the mobile device is replicated on the stationary display. There are two different modes, for one of them the content is simply shown on the larger screen in a natural orientation. For the other one, the content is distorted to look as if the mobile device was projecting it onto the screen. The projected content can show more information than the mobile screen, for example part of an image could be displayed on the mobile device and the full image on the large screen. Interaction with the content is possible on the mobile device, such as moving it around or zooming. Ideas from this approach could also be used for a map application, such as for navigation within a map. Another possibility is to reverse the projection process and use the mobile device to select content from the large screen.

2.2.1 Multi-User Systems

Cheng et al. [CLMT12] designed a system to support collaboration on large displays using tablets. On a large shared display, the system shows an overview of the workspace, in the example a two-dimensional graph. On the tablets, either the whole workspace or a viewport containing a subset of the workspace can be viewed. Interaction for navigation and manipulation purposes is possible on the tablet using multitouch gestures. Using this approach, multiple users can work together on the same workspace at a time. Technically, the system uses the Virtual Network Computing (VNC) protocol for transferring the visual contents. Touch events are sent using a modified version of the TUIO protocol [KBBC05], which is designed for creating interactive multitouch surfaces. The approach of Cheng et al. could also be applied to collaborative work on a map application with data overlays.

An approach for creating multi-user systems is to use an existing single-user application as a basis. Forlines et al. [FES+06] have done this to develop a multi-user system for a multi-display environment. Their system uses multiple wall displays, a table with integrated touch display and tablet computers. The screens display a three-dimensional map application. This is implemented using multiple communicating instances of Google Earth as a basis. The system allows to set up multiple camera positions that are used on different screens. It also allows to create layers and annotations. Interaction with the system is possible using the touch table and the tablet computers. The wall displays are used for display purposes only. The multi-user system of Forlines et al. provides a concept that could also be adapted for a two-dimensional map application and exploration of geolocated data sets.

Another system for combining a wall display and a touch table was implemented by Everitt et al. [ESRF06]. The rationale for using a touch table is to improve collaboration, as users can sit around it. In addition to the shared touch table and wall display, personal devices such as laptops can be integrated. Interaction is possible both on the touch table and the wall display. For the wall display, a mouse and keyboard, as well as a stylus are used. Using the system, individual objects such as text windows or images can be moved and rotated. At the edges and corners of the screens, the system shows portals. These portals allow to move objects to another screen or to a USB storage stick, as well as copying or deleting objects.

Rekimoto [Rek98] developed an approach for whiteboard-based interaction. In addition to the interactive whiteboard, a tablet computer is used. The system allows to show existing data on a digital whiteboard easily. This can be achieved by copying data from the tablet computer by using a stylus to select an item on the tablet and then tapping at the whiteboard. Another use case is to utilize the tablet as a toolbox. As an example, this allows to select color and line width on the tablet and then draw using the stylus on the whiteboard.

A different approach for interaction is to use hand gestures. Lee et al. [LJL+08] have created an interface for multiple devices that is controlled using gestures. The system, which consists of a table display, multiple wall displays and laptops, is described as an intelligent meeting room. Using the system, information can be shared between different screens. This can be done by first selecting the object on a screen and then pointing the hand towards a target screen to transfer it.

3 Design of the System

As described previously, good user interfaces are still a challenge for wall-sized displays. Existing user interfaces for desktop computers are not well suited for large displays. In the section on related work, multiple approaches to do this were presented, such as direct touch input or additional devices. Another possibility could be to use a gesture based interface. The solution presented in this thesis is to add another device for interacting with the main screen. In the following, we design, implement (chapter 4) and evaluate (chapter 5) a system that allows to interact with a large display using a tablet as input device. Our system is primarily focused on geolocated data sets displayed as a map on a wall-sized display. While the design of our user interface focuses on interaction with this map, the underlying basic interaction concepts could also be used for other applications.

3.1 Basic Concepts and Design Procedure

Some general interaction concepts that are relevant for the design include navigation and selection. When selecting data on a map, the traditional input device on desktop computers is the mouse. Single points can be selected using a mouse by pointing at them and pressing a button. A mouse also allows to select rectangular areas by moving the mouse while pressing a button and taking the start and end points as corners.

Navigation in the context of a map application consists of panning and zooming. As with selection of content, navigation on desktop computers is also mainly focused on the mouse. The map is panned by moving the mouse while pressing a button, and zooming can often be done by rotating the scroll wheel. Another approach to zoom a map using the mouse can be implemented by means of an on-screen slider that is moved. It is also possible to navigate within a map by using a keyboard, which allows to move around using directional keys. However, panning and zooming are not the only aspects of navigation within a map. A further aspect could be to search for places in order to jump there directly. For entering the search term a keyboard can be used.

Taking into account that a map can also show additional data on top of the geographic plan of an area, additional requirements for interaction become apparent. When performing visual analytics on a data set, the user needs to apply filters to the data set in order find more details about specific aspects. Our designs also provides solutions in this area.

For designing our tablet user interface for a multi-display environment, the first step is to gather ideas and concepts for interaction. Thus, we create drafts in the form of low-fidelity prototypes. These prototypes consist of sketch drawings showing the user interface and the interaction with it by the user. The advantage of such a process is that it allows to express ideas quickly without spending too much time focusing on unnecessary details.

This section will discuss the main concepts and ideas for the user interface in the following. These concepts are illustrated by sketches of the user interface. The interaction prototypes for our multi-display environment can be classified into two conceptually different categories:

In the first category, the user interaction happens solely between the user and the visual content presented on a screen. This interaction can be executed using direct touch or by using a mouse and keyboard. However, using existing interaction concepts, the user is only interacting using their hands. The physical positions of the user or any objects do not matter at all in this model.

In the second category, the users interact also by movement of themselves or by movement of other physical objects. In this concept, the user may move towards the display and the actual screen content changes according to the user's position. Another type of interaction could be through a mobile display device, where the screen content changes depending on the device's position.

For our design prototypes, we consider two types of geolocated data sets. The first one consists of individual independent data points that have some kind of content associated with them. An example for this are geolocated microblog messages such as they are available from Twitter posts when users share their location. For this kind of data set, each point contains a message, as well as the date and time and the coordinates of the location where it was posted. An important aspect of such data sets is that they contain both location data and textual data.

Another type of data that is considered for our prototypes are trajectories. In these data sets, multiple geolocated data points are linked in the order of acquisition time. This results in a path that shows the temporal development of an item. An example for this type of data set is the logged position of a taxi, that shows which routes it has taken. The data points could also be grouped, and then the groups can have additional parameters. In the case of taxi positions, each fare could be one group and have assigned the cost of the ride.

3.2 Map Navigation on Tablet

The most important aspect that is required for our system is an ability to navigate within the map. Because of this, our concept is to utilize the majority of the display area on our tablet application for an interactive map. A visual representation of this concept is shown in Figure 3.1 In comparison with traditional interaction using a mouse, the touchscreen of the tablet allows the user to interact with a replica of the original map directly using his hands.

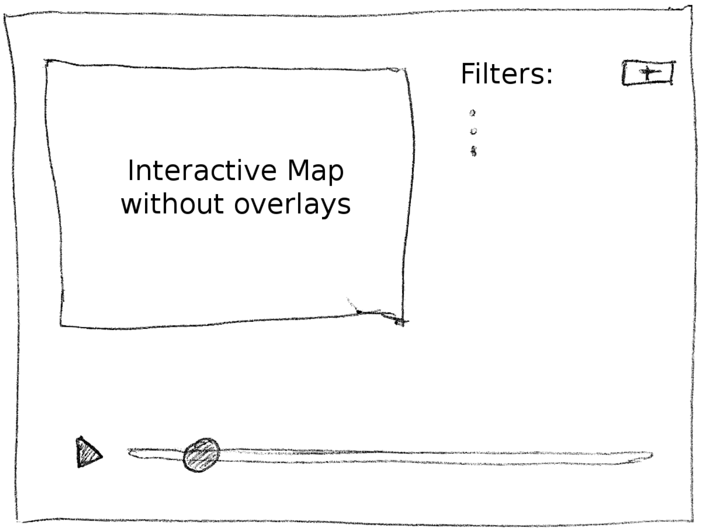


Figure 3.1: Interface concept for map navigation and content filtering. The main part is an interactive map. To the right a list of textual filters is shown and can be edited. The area on the bottom allows filtering within the data set by time.

This interactive map can be moved around by sliding a single finger on the touchscreen. For zooming, two fingers can be used in a pinch gesture as this is already an established interaction model on mobile devices containing a touchscreen. The idea for the map is that the displayed area is at all times synchronized to the main display. Thus, if multiple users are using a tablet to control the application the visible map area is updated in real time on all devices.

In addition to navigation on the map our design supports creating and editing filters. Beside the interactive map, a list of textual filters can be shown. When the user clicks on a button, they can enter the an additional filter text using a virtual keyboard on the tablet screen. As with navigation on the map, the filter item is then synchronized with the main application and applied to the map on the large screen.

Another element of our concept is the ability to filter the data set by time. For this, a slider is shown below the map that allows selecting a timeframe. One possible implementation of the slider could be to have two controls where one is used for the start date and another one is used for the end date. Another idea is to add an additional slider or a drop-down field that allows selection the duration of the filter. Then, a normal slider can be used to choose the central point in time.

An additional feature of the time filter is to automatically move the time forward. Using a button beside the slider, this mode can be activated. For exiting automatic playback and switching to manual mode, it could be possible to dynamically change the button into a stop

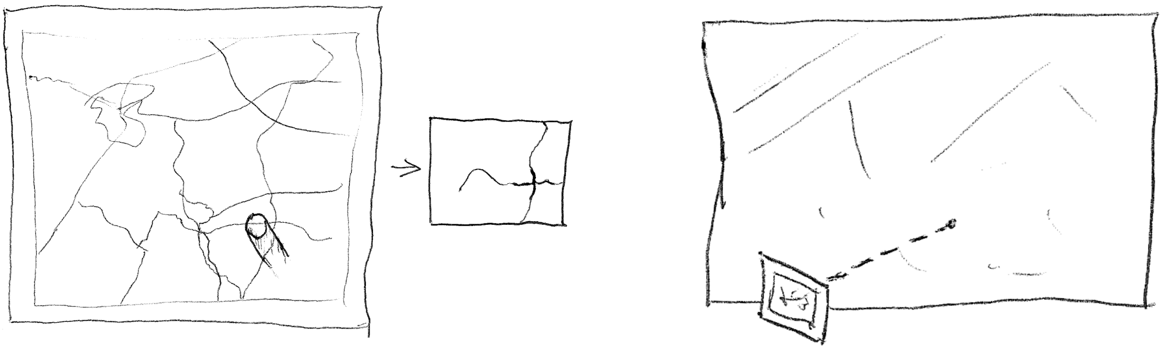


Figure 3.2: Two different concepts for selecting points on the tablet. In the left concept, the user selects items by touching the tablet screen. In the right concept, the user selects contents by pointing at the screen with the tablet.

button. An alternative could be to allow moving the slider which implicitly stops automatic playback.

3.3 Selection of Points on Tablet

For selection of points using a tablet interface, we designed two different concepts. Figure 3.2 shows the first concept on the left. In this concept, the user selects content by touching the screen. For this to work, it is required that the map is fixed and can't be moved anymore. Thus, the user has to switch to the selection screen by pressing a button on the main screen of the application.

The idea of this approach is that the system finds information about the touched position and shows it on the large screen. As an alternative it would also be possible to show information on the tablet screen, which would be preferential when multiple users are working in parallel. However, the shown information could be obstructed by the hand of the user. The approach is well-suited for trajectory data. In that case it could be possible to highlight all trajectories that intersect the point touched by the user. It can also be useful for other data sets, an example is a kind of zoom lens that increases the size within an area about the touched point.

The second approach deals with pointing using the tablet itself. Figure 3.2 shows this on the right-hand side. With this concept, the user points with the back surface of the tablet towards the wall display. Then, the tablet can show details about the part of the map where the user is pointing. An example of this could be implemented by having a circle in the center of the display that is used to allow focusing a specific point. In order to aid the user either a map can be shown that changes to reflect the movement or the rear camera of the tablet could be used to give an orientation. The additional details can then be shown around the circle in the

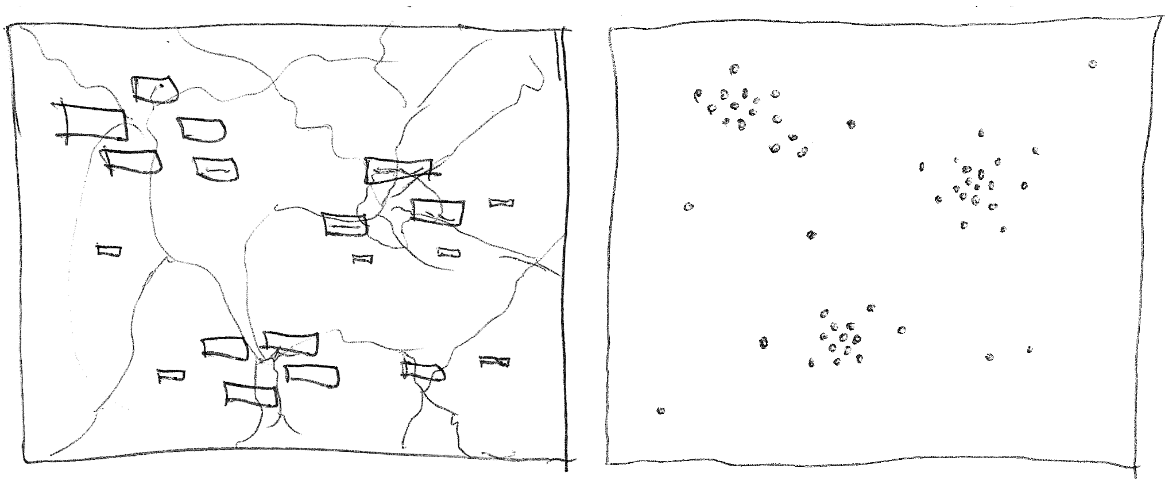


Figure 3.3: Different concepts for displaying content from a data set on a map. In the left example, textual content is represented as a tag map. The concept on the right side shows the frequency of content as a heat map.

center of the screen. An example for geolocated microblog data would be to show the actual text content. As an additional feature a means of stopping the tracking of the movement could be added. The user can then continue to work with the selected content without having to point at the screen.

3.4 Map Overlays

This section focuses on visualization of point data on a map. There are two different concepts for showing such data on the map. Figure 3.3 show both of them side by side.

In the first concept, which is shown on the right, only the location and frequency of data points is considered. We consider two ways of doing this. First, the simplest way of showing the data points is to show a small dot or circle for every item. However, when multiple items are located at the same point, this may not be visible on the map when this approach is used. Thus, the second concept is to use a heat map which shows the distribution of data points more clearly. This means that areas with a lot of data points are shown as a bright and opaque color. On the other hand, areas with less data points have a more dampened color.

In the second concept, the content of the data points is considered in addition to the location. This can be done by showing a tag map, which is created from textual content from the data set. In such a tag map, some of the words that occur in the data set are shown on the map. The selection of these words can be done according to the number of times it occurs in the data set. A given word can appear on the map multiple times, and multiple locally close occurrences of a word can be grouped as a cluster. In this case, the font size can be adjusted, so if a word is very frequent within a small area, it can be highlighted by being larger than others. This allows the user to quickly see relevant patterns within the data set and to consequently refine their search.

3.5 Multi-User Interaction

An aspect that has not been addressed in detail in our previous concepts is multi-user interaction. While evaluating the prototypes, it became apparent that more needs to be done regarding this aspect. When multiple users are using the system at the same time, they may not want to view the same filters.

An example situation may be that multiple experts are analyzing a data set together. First, everyone works on the data set individually and analyzes it with respect to their field of knowledge. When the individual users are finished, they discuss their findings and find relationships between their results.

In order to support such an approach, we add the concept of layers. A layer is a filtered instance of the data set that is shown on the screen. Each user can select a different layer on their tablet and work on it. Then, filters are only applied to the layer that is currently active for a user. In order to distinguish the layers on the large display, they can be displayed in different colors.

This concept may be confusing when multiple users are operating the system at the same time and there are lots of changes on the screen. Another disadvantage is that the users can only view a single map section at a time. To solve this, one possibility is to allow working locally on the tablets. Thus, the map overlays are visible only on the tablet and the users can optionally select layers to be visible on the large screen for sharing.

4 Implementation

In this chapter, we describe the implementation of our system. We implemented a subset of the concepts shown in the previous chapter. The first section describes the data sets we use for our system. After that we look in detail on the implementation of the individual components of the system. There are four different components. The server is responsible for the communication between the other three components. The main application running on the large display is our map application. For controlling the map application, the tablet application is used. It can run on multiple devices at the same time. Similarly, the desktop control application offers the same feature set, but optimized for mouse and keyboard.

4.1 Datasets

In the previous chapter, we described two different types of data sets. On the one hand there are data sets that contain independent data points such as geolocated microblog posts. On the other hand, there are connected data points such as trajectories from a vehicle.

We decided to use geolocated microblog data for implementing our system. More specifically, we use Twitter messages that users have posted with location sharing turned on. This microblog data offers some advantages over trajectory-based data. They contain textual data in form of the message content, which offers more opportunities to visualize and interact with. Also, there is a large amount of such data available. This allows selecting data sets that contain interesting data, which is a positive aspect for the user study. This is especially true when large events occur, because there are a lot of messages in the same area.

We use multiple data sets to test our system. The largest of these data set contains more than 1.25 million messages. This specific data set contains data from all over the world without any regional filtering. In contrast, the other data sets are pre-filtered to a given region such as a country in order to remove unneeded data. Thus, the largest data set we actually used for the study contains a bit less than 250 thousand entries and has a size of 40 megabytes. The individual messages are stored in text files with the individual fields separated by tabs. For each message, the full text, the date and time as well as the latitude and longitude are stored. In addition to that, a user identifier and some other unused fields are contained in the files. More details on the contents of the data sets are given in the following chapter on the evaluation procedure.

4.2 Communication Server

There are multiple existing tools that support communication in multi-device environments. One example is the Smarties framework by Chapuis et al. [CBF14]. In addition to the communication protocol between devices, it also includes a mobile interface and allows integration into existing application for wall displays. The central aspect of this system are pucks which are shown on the mobile interface and correspond to objects on the large screen. Another example is the TUIO framework by Kaltenbrunner et al. [KBBC05]. This framework defines a protocol and programming interface for multitouch interaction and is available for a wide variety of programming languages.

However, we decided to design and implement our own communication system. This allows us to have full control over the communication and to keep the implementation simple. Our communication system provides a basic way for applications to exchange messages. Primarily, the system is designed to allow transporting messages in string format. However, the messages exchanged using our system could be of any type, including binary data. The only requirement is that the line delimiter is not contained within the data, meaning that the application would have to take care of escaping it. Each message sent through our system has a type that is given as a string. Clients that are connected to the server can register to receive messages of a specific type. Then only messages that have one of the registered types are sent to this client. Using this feature, a client has to deal only with messages it actually needs.

When a client application is started, it connects to the server, and the server then assigns an identifier to the client. This connection stays alive during the entire runtime of the application. Thus, all messages are sent over this single connection. When the server sends a message to a client, it includes the identifier of the sender. This allows the receiver to know the sender which can be useful when sending a reply to the original message. When a client sends a message, it can include a list of receivers. This list can contain one or multiple identifiers of clients or a special identifier that refers to all clients. In addition, a specific client can be excluded from receiving a message by prefixing the identifier with a keyword. The receiver identifiers in a message are then used by the server for filtering the list of possible receivers that is given by the type registrations.

The server itself is implemented in the programming language Go[1]. The decision fell for Go because it provides a good standard library for network transfer. Each client gets assigned a receiving and a sending goroutine that can run concurrently. The individual goroutines are connected using channels that also serve as queues for messages.

The client library for our communication system is implemented in Java. This is also the language we use for the other three applications. The decision for Java allows portability of our applications to different platforms.

[1] https://golang.org/

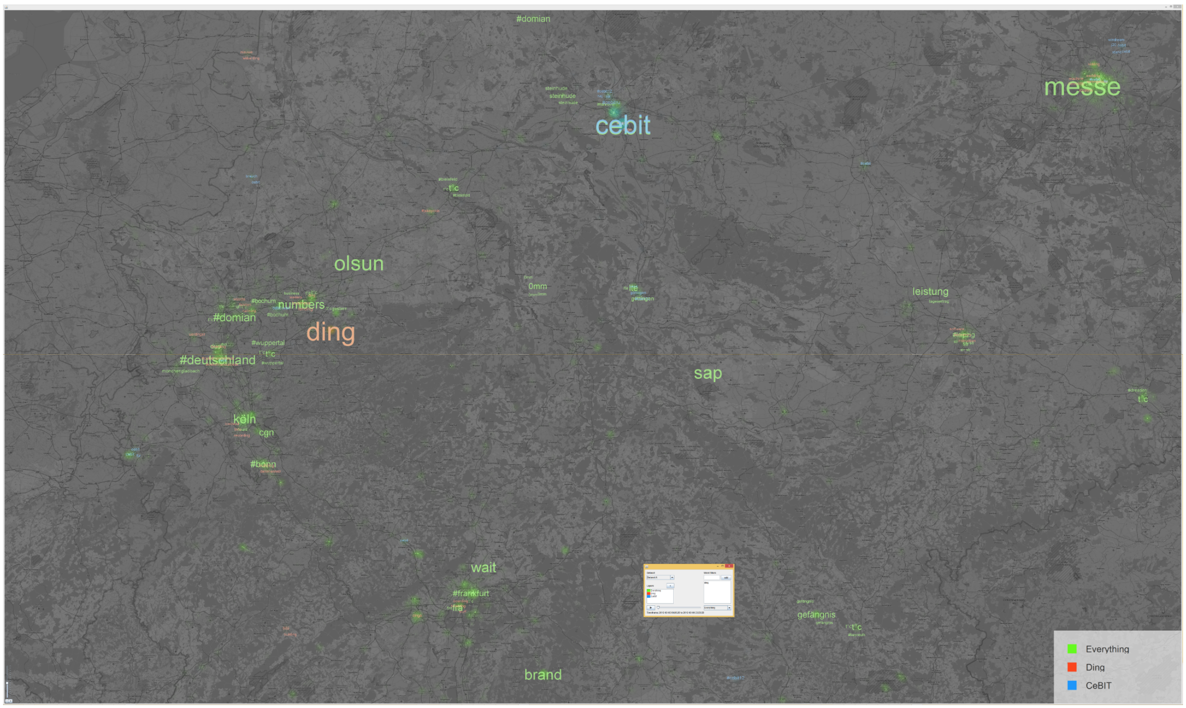


Figure 4.1: Screenshot of the map application. This example shows three different layers, each which is shown in a different color and consists of a heat map and a tag map. The small window on top is part of our separate desktop control application.

4.3 Map Viewer

The map application is used for displaying content on the large application. It is also a central part of the system, as it is the only one that has access to the raw data from the data sets. The task of the map application is to manage the data sets and execute filter operations as well as rendering the results on the screen. A screenshot of the map application is shown in Figure 4.1.

The map application is written in Java, which allows it to run on multiple platforms without modification. Another aspect is that there were already existing Java applications that work with the used data sets. For the background map that is shown in the application, multiple mapping frameworks were considered. Many of these frameworks are available for Java, which is also a factor for the decision.

We use tiles from the OpenStreetMap project[2] for the background map in our application. OpenStreetMap is a collection of open geographic data that is edited by volunteers. One

[2] https://www.openstreetmap.org/

example for a map framework that supports the OpenStreetMap tiles is JMapViewer[3]. This library is originally based on an editor for OpenStreetMap data. However, ifist supports only very basic features.

Another map framework is JXMapViewer which is part of SwingX[4], a collection of components for graphical user interfaces implemented in Java Swing. JXMapViewer provides a variety of features, but it is no longer actively developed. However, there are existing applications that work with our data sets which is helpful for the development. For this reason, we decided to use JXMapViewer for our application. A modified version of JXMapViewer exists under the name JXMapViewer2[5]. It has a lower number of dependencies and supports a few additional features in addition to some fixed bugs. We used JXMapViewer2 as a reference to implement some of the features in our application.

The map application is responsible for managing and rendering of the data set overlays. Each data layer contains a textual as well as a temporal filter. The textual filter keeps only messages that contain all of the words in the filter list. The temporal filter specifies the start and end date for the layer. Every time a filter is changed using one of the control interfaces, the map application creates a corresponding filtered instance of the data set. In addition to that there is a feature for automatic temporal playback. The periodic update is also implemented within the map application.

For each layer there are three actual overlays that are drawn onto the map. The simplest one is the point overlay. This type of overlay is responsible for drawing a single pixel for every message from the data set. The second overly is the heat map which shows the density of the messages. For this, a semi-transparent gradient is drawn for every message, which highlights regions with a lot of messages. Finally there is the tag map overlay which is the most complex one of the three. As a first step, the messages are split into individual words and a clustering process is started in order to merge nearby occurrences of the same word. When the clustering is finished, the resulting clusters are arranged on the map. For this task, the clusters are sorted by the number of occurrences and drawn at a descending size in order to highlight most important ones.

The most challenging aspect of the map application is the rendering of the map tiles and the overlays at an adequate speed. In order to achieve this goal, each overlay has it's own worker thread. This thread constantly waits for changes in the filtered data set. When a change occurs, the updated overlay can be rendered in the background and is drawn onto the screen afterwards. While it may still take some time to render the overlays, this approach ensures that the application stays responsive.

[3] https://wiki.openstreetmap.org/wiki/JMapViewer

[4] https://swingx.java.net/

[5] https://github.com/msteiger/jxmapviewer2

For the tag map, the most expensive step is the clustering, which has to be done before rendering the data. Especially with large data sets can take up to 30 seconds or more to render. As a measure for improvement the result of the clustering is cached for the unfiltered data set. As a result, a significant improvement in rendering time is achieved for the common case where a layer doesn't have any filter set up. Another aspect that has turned out to be very costly is an iteration over all pixels on the screen. This was an issue in the original implementation of the heat map, but a simplification of the rendering process resulted in noticeable improvements.

An issue that still remains regarding the rendering performance is the rendering of the background tiles, which seems to be inherent to JXMapViewer. When a zoom level is viewed for the first time, it is visible that the tiles are loaded one after another, beginning from the outer parts of the screen. We tried to improve upon this situation by using offline tiles, but the performance did not change significantly. Also, when the application is running on a high resolution display the rendering time for already loaded map tiles is noticeably higher than on a smaller display. However, the rendering performance still allows interactive usage.

4.4 Tablet Application

As a main control application for our system we developed a tablet application. The design of the user interface is based on the concept for map navigation in chapter 3. An example of the tablet application in use along with the large display is shown in Figure 4.2.

The application is implemented as an Android application. For the rendering of the interactive map, we consider different libraries. First, there is Google Maps for Android[6], which is the first-party solution on the platform and provides fast vector-based maps. However, it is a proprietary software and requires registration for an API key. A major disadvantage is also that it doesn't provide a consistent map with the desktop application where we already used OpenStreetMap tiles.

As an alternative, we look at OSMDroid[7] which is a drop-in replacement for an older version of Google Maps that supports OpenStreetMap tiles. Finally, we choose to use the MapsForge[8] library with OpenStreetMap tiles. This library is developed primarily for Android and supports vector data in addition to pre-rendered tiles, however only at a slow speed. Nonetheless, the application is a good choice, as it supports all the features we need and is under active development.

As described above, the tablet application is based on our original concept for map navigation, and looks similar to it. The main part of the screen is taken up by the map, that is synchronized

[6] https://developers.google.com/maps/documentation/android-api/

[7] https://github.com/osmdroid/osmdroid

[8] https://github.com/mapsforge/mapsforge

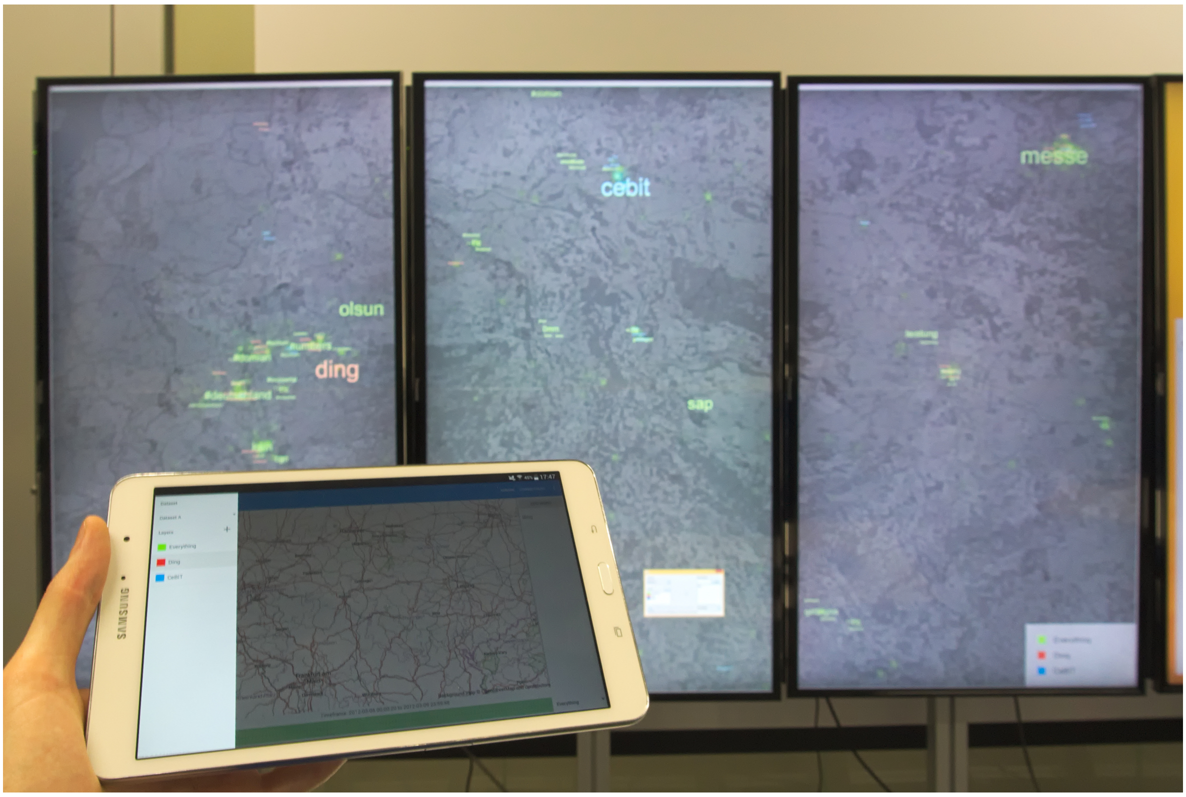
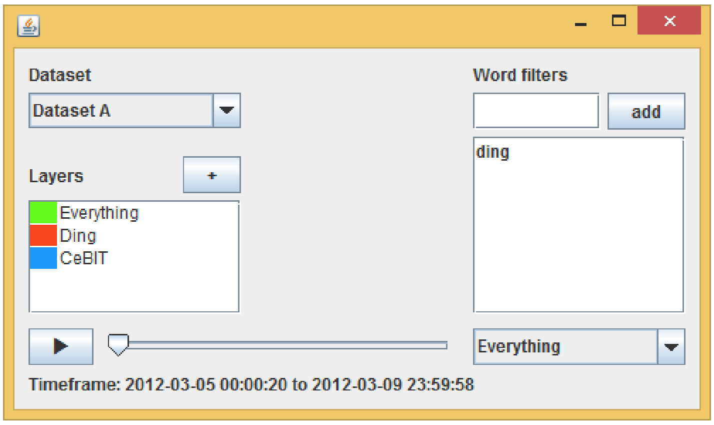


Figure 4.2: Photo of our tablet application in use. Most of the area is occupied by the map. The slide-out menu on the left provides a list of layers. On the right, text filters for the current layer are edited. The bar at the bottom allows filtering by time.

to the desktop application in realtime. On the right side of the screen, a list of word filters is displayed. Additional terms can be added to the filter by tapping on a button and entering the word. Existing terms are easily deleted by tapping on the list item.

On the bottom side of the screen, a slider for the time filter is displayed. We decided to implement a slider with a single control that moves the time frame. The slider is a custom implementation and the scales in the background show the boundaries of days for better orientation. In addition to that there is a drop-down menu in the bottom right corner to select the duration of the filter timeframe. In order for the user to see the exact filtered timeframe, the start and end date are shown above the slider. In the bottom left corner, a button allows to switch between automatic playback mode and manual control. While in automatic playback mode, interaction with the slider or the drop-down menu is not possible.

An aspect that is not in the original prototype for the tablet application is the ability to manage multiple layers. The list can be shown by swiping from the left edge of the screen. Then, layers



Dataset
Dataset A

Layers
🟩 Everything
🟥 Ding
🟦 CeBIT

Word filters
ding

Everything

Timeframe: 2012-03-05 00:00:20 to 2012-03-09 23:59:58

Figure 4.3: Screenshot of the desktop control application. It allows to edit different layers on the left side. On the right side, the textual filters can be modified. The slider at the bottom allows to filter temporally.

can be added, switched and removed. The word filters apply only to the currently active layer. When multiple tablets are connected to the system, each one can edit a different layer.

4.5 Desktop Control Application

We developed the desktop control application in order to have a reference point for the evaluation of our system in the user study. The features of the desktop control application are practically identical to the features of the tablet application. Thus, multiple layers can be managed and for each layer word filters can be set up. In a similar way to the tablet application, the layer manager on the left side of the window offers options for every layer after right-clicking on an item. Then, the color can be changed and it can be renamed or deleted.

Time filters can be set up on the bottom of the screen. It is implemented using a standard slider for moving the timeframe in conjunction with a drop-down menu for choosing the duration. The main difference of the desktop control application to the tablet application is the missing map. However, the map can be panned and zoomed directly on the map application using the mouse.

Just as the map application, the desktop control application uses the Java Swing toolkit to render the user interface. It is set up to be always on top of other windows in order to provide fast access while using the map application. Figure 4.3 shows a screenshot of the application.

5 Evaluation

In order to evaluate the practical usefulness of our system, we conduct a user study. In the study, two different user interfaces for the map application are compared. The focus is on evaluating the tablet interface we developed. As a baseline for comparison we use our alternative interface that is operated using traditional mouse and keyboard controls. As mentioned previously, collaborative work is also part of the motivation for our tablet interface. To get information about the value of our interface for collaborative work, the participants work in groups of two people.

5.1 Setup

Our setup for the study consists of the components that are described in the previous chapter. The user-visible parts of the system consist of our main map application, the tablet interface and the desktop control interface. An image of the setup is shown in Figure 5.1.

As a main display for the map application, we use three Panasonic TX-50AXW804 monitors. At a diagonal size of 50 inches, these screens can show content in 4K resolution. The actual screen size of the whole setup is approximately 2.02×1.13 meters. This leads to an actual on-screen size of 6480×3840 pixels. At the chosen size, it is possible to view the screen content as an overview and see some details as well. The aspect ratio of our setup is similar to the common $16 : 10$ aspect ratio, and allows two users to comfortably use the system at the same time.

In addition to the three main screens, there are two more screens. The fourth monitor is used for question forms and allows the participants to enter answers during the study. The fifth screen is used for diagnostic purposes, and is not used by the participants of the study. All screens are connected to a single computer running Windows 8.1. This computer runs the communication server, as well as the map application and the desktop control application.

In front of the wall display, we have set up a table. The users sit in chairs at a distance of approximately one meter from the display. In order to control the desktop application and the question forms, a mouse and a keyboard are supplied on the table. There is only a single set of input devices that the users have to share when working together using the desktop interface.

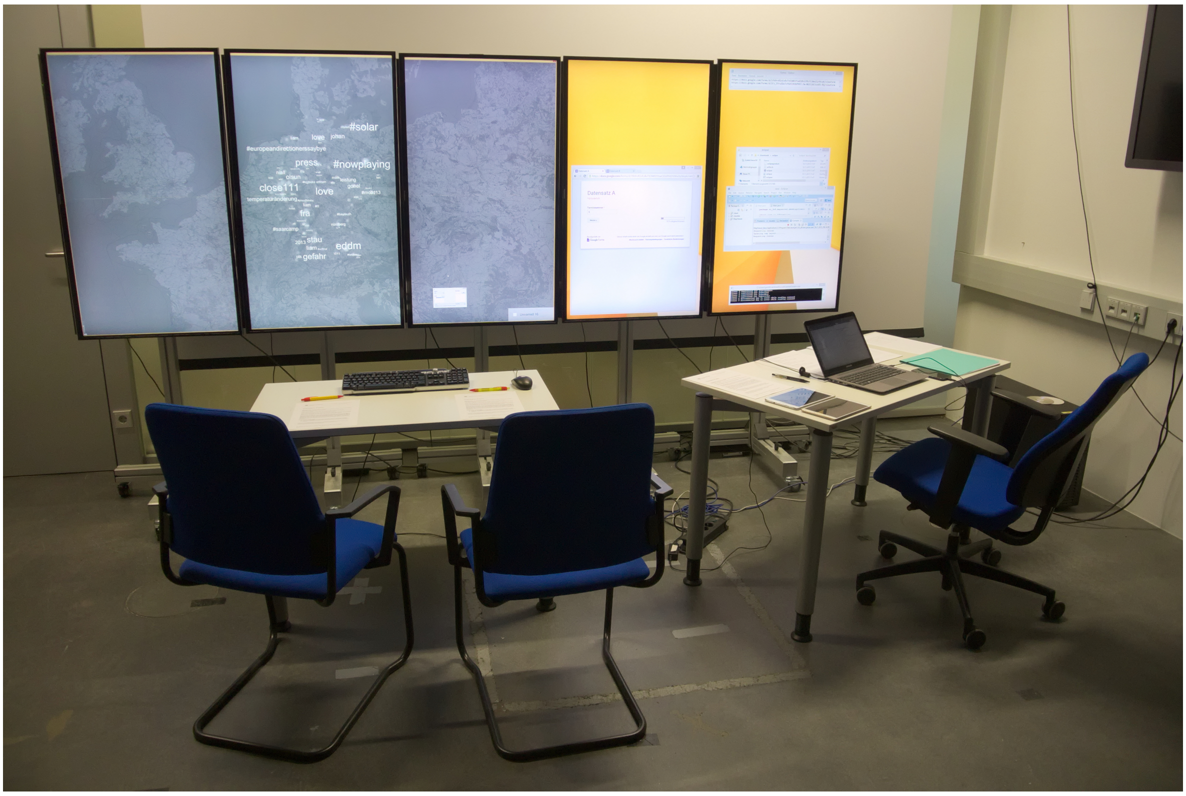


Figure 5.1: The setup for our user study. The three left-most screens are used for the map application. The fourth monitor allows the user to answer questions during the study, the right-most screen is used for diagnostic purposes only. At the left table, the users take place and operate the system.

As an alternative to the traditional input devices, there are two 8.4-inch Android tablets. The screens of these tablets have a resolution of 2560×1600 pixels. During the study, each participant uses one of the tablets. Both tablets run our mobile control application, and are connected to the main computer using a wireless network.

5.2 Design

During the study, the participants use our system on their own in order to solve the given tasks. In order to evaluate the collaborative aspects of our system, the users work in teams of two people. There are two different data sets containing geolocated Twitter messages. For each of them we created 9 questions as tasks to solve. An example for such a task is to find the main event in a data set. Another one is to find the location or date for an event. Some questions also deal with specific details on a given event. A few more details on the topics of the questions are given in the following section on our data sets. The full list of questions is presented in Appendix A. All participating teams work with both of our user interfaces and both data sets. In order minimize the influence of learning effects on the results, the combination of data set and interface that users start with is changed for every session.

While the participants find the solutions to the given tasks, we measure the completion time for each question. As there is no direct support for such a feature in our software, the measurement is done manually. The answers to the questions are recorded using an online form, which is displayed on a separate screen. The participants enter the questions using the mouse and keyboard on the table in front of them. In addition to that, we take notes while the users are working with the system. The goal of taking notes is to analyze how participants use the system. Also, some users may speak about their opinion on the system while they are using it.

After the participants are finished with all the questions on one of the data sets, we collect their feedback on specific aspects of the system. For this purpose, we use two different questionnaires. The first one is the NASA Task Load Index (TLX) [HS88, Har06]. It is used to measure the workload that the system poses for the users, and thus the effectiveness of the system. Specifically, we use the Raw TLX variant. In this questionnaire, there are 7 different items. Each of them is to be answered on a 21-point Likert scale.

The second questionnaire is based on the Networked Minds measure that was developed by Biocca et al. [BHG01] The task of this form is to measure aspects related to collaborative work. It consists of several questions in the categories of co-presence, psychological involvement and behavioral engagement. Each of these questions comes in a pair with another one. One of them is about rating the feeling of the users themselves. In the corresponding question every participant rates their impression of the other participant. The questionnaire consists of a total of 38 unique questions, and each of them is answered on a 7-point Likert scale.

5.3 Study Procedure

Before starting with the study, we welcome the participants and ask them to read and sign the consent form. We also collect some statistical data about the participants. This data consists of age, gender and profession.

Then, the users are introduced to our system. We start by explaining the general idea and some basic facts about the type of the data set. Afterwards, we tell the users how to control the map application. Either the tablet application or the desktop application is explained in this first stage. For every team, we change the order of the interfaces in order to rule out learning effects in the results. When the users know about the basic usage, they are allowed to try out the system for a few minutes on their own to get familiar. If there are additional questions on the way the system works, we give further explanations. For this first step, there is a special test data set. It contains data of a short timeframe of two days only, and there are no events that could give any clues for the later tasks.

As soon as the users had some time to get familiar with the system, we start with the first round of questions. For this round, the participants use the same interface that they already know, and we select one of the two main data sets. The data set to start with is chosen differently for each team. After the first round of questions, the users answer the two questionnaires.

Then, we give a short introduction to the other type of user interface. For this, we explain primarily the differences to the previous interface type. Again, the users are allowed to try out the system for a short time using the test data set. When they are familiar with it, the second round of questions is started with the remaining main data set. As with the first round, the second round is finished with the NASA TLX and Networked Minds questionnaires.

Finally, we discuss our system with the participants in order to get some qualitative feedback. Specifically, we ask them which interface they preferred, and why it is like this. We also talk about possible suggestions for improvement and aspects they like about the system. After a session of the study is finished, the users are given chocolate as a reward.

5.4 Datasets

For the study, we use two main data sets with questions that are to be answered by the study participants. The data sets contain geolocated messages extracted from Twitter. In addition to the main data sets, there is a small test data set. It contains unfiltered data for a timeframe of two days, and contains no special events. All data sets that are used for our study contain only data that is geolocated within Germany.

The first of the main data set is focused on the events around a trade fair. Specifically, the main event is the CeBIT 2012, a trade fair on information technology that takes place each year in Hanover. With this data set, detailed information on the CeBIT fair is visible from the visualization of the map. This includes the start and end date of the fair, and even the positions of some individual booths. Also, some more specific topics that were a subject at the fair in 2012 can be seen.

In addition to the CeBIT, there is also another fair in the data set. In Berlin, the ITB (Internationale Tourismus-Börse Berlin), which is a fair on tourism, took place at the same time. Compared to the CeBIT, this fair is not as large, but the date can still be approximated from the data set.

Apart from these trade fairs, some other things can also be seen in the data set. For example, there is a tweeting church bell, that posts the current time and the sound of the bell (“ding”) every few minutes. Another pattern that is contained are job offers of the software company SAP at multiple places.

The main feature of the second data set is the flood which occurred in some parts of Germany in 2013. This data set allows discovery of several features that are related to the flood. At a coarse scale, the temporal and spatial development of the flood can be extracted. But there are also more detailed aspects. For example at one place a transformer station that is endangered by the flood can be seen. At a nearby place, a dam has been washed out and the calls for help at rebuilding it are shown on the map.

In addition to the flood, the data set also contains some other events. The first one is the "Rock am Ring" music festival, where the data can be found in the data set. Another one is the "Blockupy" protest that took place in Frankfurt on one day during the timeframe of the data set. Lastly, messages about the broadcast of the German TV show "Wetten, dass..?" can be found in the data set.

For both of our main data sets, some pre-filtering is required. This is the case, because some rather unimportant tweets would clutter the view on more significant ones otherwise. An example is a radio station in Berlin that tweets every single song that is played. It is removed, because the other events in Berlin are very difficult to see if the tag cloud is dominated by the radio station. Another example are traffic news that are posted by a single account all over the southern part of Germany. Also, there is an account that posts only about current trends on Twitter. This causes unnecessary duplication, because the original data is already visible on the map. It can even lead to false conclusions, because the location of the tweets about the trends is not accurate.

6 Analysis and Results

In this chapter, the results of the user study are discussed. The results are categorized depending on the kind of feedback. We start by evaluating how successful the participants were with answering the given questions. In this context, we also take a look at the completion times for the answers. Afterwards, the questionnaires are analyzed. Beginning with the results of the NASA TLX questionnaire, we report our findings about the workload. Then, the data from the Networked Minds measure are presented, which gives information about the qualities of our system for collaborative work. Finally, the participants' qualitative feedback is discussed, including their preferred interface and the reasons for the preference. In this part, the qualitative feedback is also compared to the quantitative measurements.

6.1 Study Participants

For the study, we recruited participants from the university mailing list. As a result, a total number of 12 persons (2 female) participated. The participants were aged between 20 and 30 years ($\mathrm{M} = 22.5$, $\mathrm{SD} = 3.37$). All of them were students in the subjects of computer science or software engineering. Due to the fact that every session of our study requires two participants, a larger number of participants would have been preferred. A possible reason for the low number is the duration of the study, which was between one and one and a half hour depending on the participants' speed. In the end, a total of six sessions with two users for each one were carried out.

6.2 General Results and Completion Time

In general, the users were able to use our system in order to solve most of the questions successfully. However, when looking at the results, it also becomes apparent that there was a lot of variance between individual questions and teams. This is for example shown in the completion times for single questions which varied between 11 seconds up to 8 minutes. To find out why this is the case, we take a closer look in the following.

While designing the tasks for the study, all questions were originally intended to have a unique solution. During the study it became apparent that this is not the case for some of the questions.

	Dataset		Interface		Round		
	CeBIT	**Flood**	**PC**	**Tablet**	**First**	**Second**	**Overall**
Time (min)	25:23	30:40	27:09	28:54	31:16	24:47	28:01
Standard error	2.08	3.32	1.97	3.74	2.88	2.39	2.03

Table 6.1: Average completion times for all questions depending on different conditions.

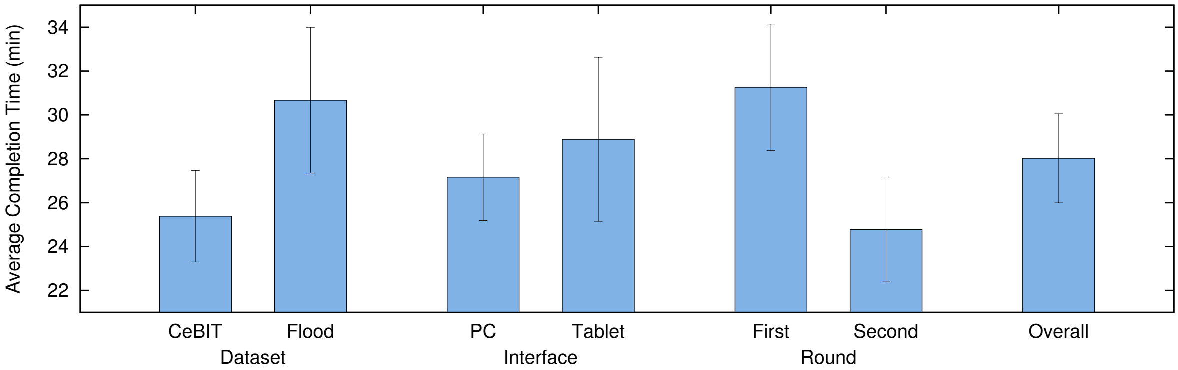


Figure 6.1: The average completion times for all questions. The times are shown with standard error. The y-axis starts at a larger value to show the differences better.

However, this does not affect the results of the study negatively, since the main goal was to confirm the system's usefulness and to compare the different user interfaces. As described below, the users were able to find answers to the questions.

One case with multiple possible solutions for a question occurred in the CeBIT data set. In this question, we asked for the location and name of the company which is posting job offers on Twitter. While there is only one company that posts such messages in the data set, it actually contains 6 different locations. For most places the number of messages is quite small. Nonetheless, some users did not find the location with the highest number of messages. However, all participants found one of the locations and could name the place or the name of a nearby city.

Another case where multiple possible solutions existed was a question related to the flood data set. We asked for the place where sandbags are filled, with the constraint of being nearby a given city. In addition to the intended answer, there is also another town on the opposite direction of the city where messages about sandbags are posted. Some of the participants found both towns and then they decided for one of them. In the end, two teams picked the alternative solution. Because the following questions depended on knowing the originally intended answer, the users were told about this.

During the study, we also found that in general tasks to find temporal information appeared to be difficult to solve. For three of our questions, the task was to find an event that occurred on a single day. All users were able to find the exact date for these events. However, when we asked participants for a data range of an event, or for the start data of a longer event, this was different. There were a lot of different answers for such questions. For one of the questions, no team found the exact solution. In the best case, the answer missed just one day for this question. In another case, the start date of the event was 2 days off the correct solution.

One question appeared to be particularly difficult, as three of the teams did not find the correct answer at all. We asked for the main natural event in that data set. The intended solution would have been a flood. When viewing the map in the overview, i.e. zoomed out to show all content, the German term for flood "hochwasser" was visible on the map. However, many users did not see this term on the map. Instead, they tried entering their own ideas on which natural event could be asked for. An example for this which was often entered is the term "wind". As a result, three teams found the term "wirbelwind", which is German for whirlwind, and gave it as their answer. An investigation in the data set showed that the term is not related to a weather phenomenon at all. Actually, it is the name of a user that is mentioned many times in a conversation. This example shows that it may be useful to add a filter for removing such cases to the application.

Other than the mentioned examples, all users found the solutions to the given questions. In one case, a team even exceeded the intended answer. When asked for the hall where a company's booth is located on a trade fair, they entered the correct number of the stand as their answer. For the given data set, the stand number comes from the message contents, i.e. it is displayed on the tag map. The hall on the other hand can be extracted using the location of the tweets. On the background map we use in our application, the numbers of the halls are shown.

The task completion time was measured for every single question during the study. However, the significance of the single measurements is limited because of the low number of data points. This is due to the reason that this data is only measured once for each session and question. Instead, it makes more sense to look at the completion time for a whole data set in order to get a coarse overview. The average numbers are shown in Table 6.1 and in Figure 6.1. We always compare two conditions against each other, such as the data sets, the interfaces, or the first to the second run. For these comparisons, all other conditions are equally distributed.

On average, participants took 28 minutes to complete the answers for a single data set. When comparing the CeBIT data set to the flood data set, a difference becomes apparent. Most participants were quicker in solving the questions for the CeBIT data set, on average the difference is around 5 minutes. A comparison of the first to the second round shows an even larger difference. The users were on average 6.5 minutes faster in the second round. This is probably due to a learning effect.

The goal of this study is to evaluate the difference between the tablet and desktop interface. However, the analysis of the completion time gives no clear result here. On average the users

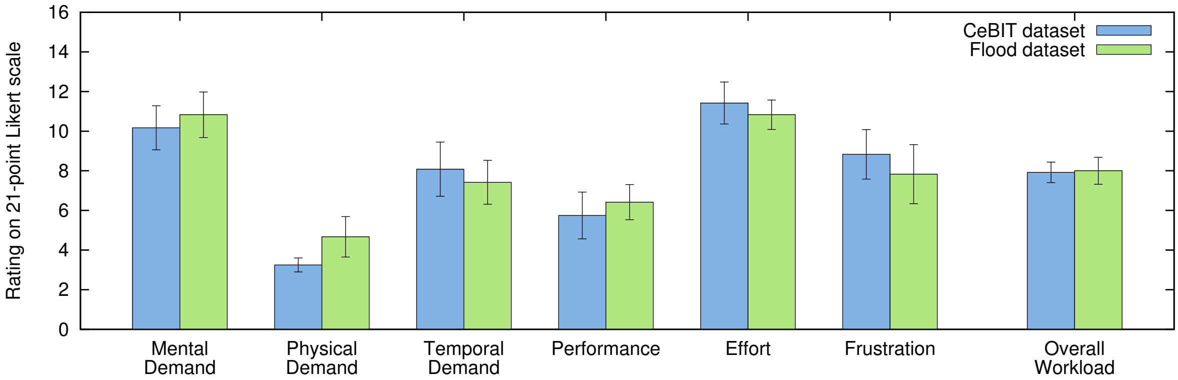


Figure 6.2: The results of the NASA TLX questionnaires. In this graph the differences between the data sets are shown. The y-axis displays only a part of the value range.

were a bit more than 1.5 minutes faster when using the desktop interface, which is not a large amount of time considering the total average completion time of 28 minutes. The measurement could also possibly be affected by the fact that users had to change to mouse and keyboard to enter the solution when they were using the tablet interface.

6.3 NASA TLX

In order to analyze the results of the NASA TLX questionnaire, we look at each of the subscales individually, as well as the overall workload. The value range of the subscales ranges from 0 to 20. For most questions, the value 0 corresponds to a “very low” demand and 20 corresponds to a “very high” demand. For the performance subscale, a 0 means “perfect” and 20 means “failure”. Thus, for all subscales lower numbers are better than larger values. In the original NASA TLX variant, the individual subscores are weighted in order to get the overall workload. Since we are using the raw TLX variant, the overall workload is calculated as the average of all subscales.

First, we look at the results depending on the data set in order to find out whether there is a difference as in the temporal analysis. The results comparing the data sets are shown in Figure 6.2. When looking at the data, it can be seen that there is practically no difference in the overall workload between the two data sets. There are some small differences in the individual subscales, but these do not show that any of the data sets was easier to solve.

Analyzing the changes between the first and the second round gives more useful results, as shown in Figure 6.3. Again, the overall workload is practically the same for both conditions. However, the changes in the individual subscales are more interesting. For the subscales of effort, frustration and the mental demand, the difference is very small. There is a large difference in the performance, though. On one hand, users rated their performance significantly

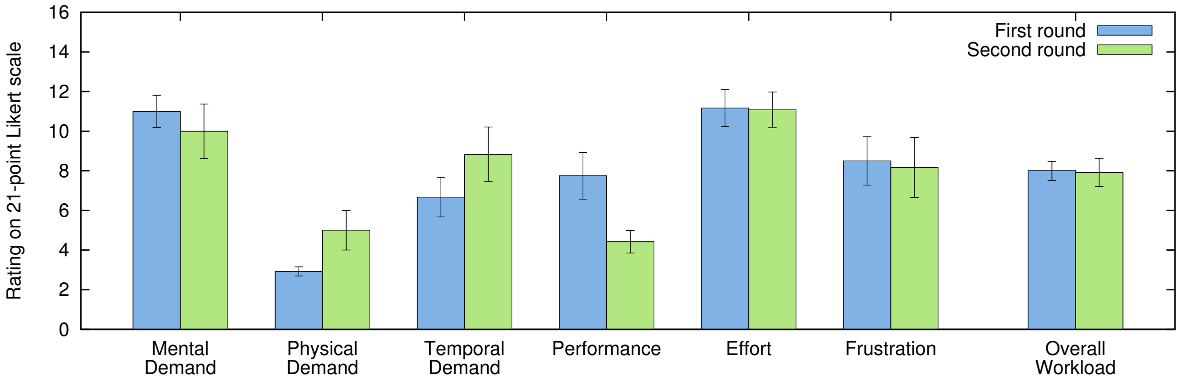


Figure 6.3: This graph shows the differences between the first and second round in the results of the NASA TLX questionnaire. The y-axis displays only a part of the value range.

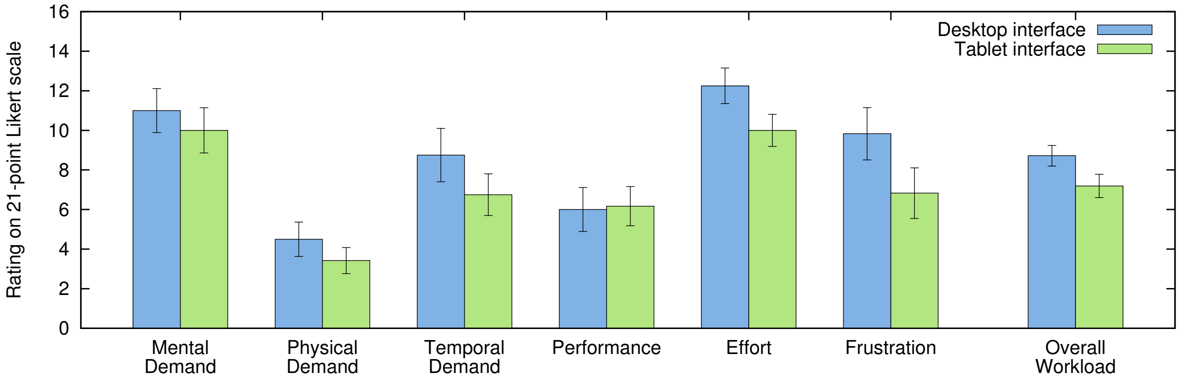


Figure 6.4: The differences between the two types of user interface, as seen from the results of the NASA TLX questionnaires. Only a part of the whole value range is shown on the y-axis.

better in the second round compared to the first round. Thus, the results of the NASA TLX questionnaire seem to support a learning effect. On the other hand, the users' ratings regarding the physical and temporal demand show higher numbers for the second round. This appears to contradict to the better rating that is seen in the perceived performance. A possible explanation from this could be the long duration of the user study. Because of a reduced motivation and fatigue effects, the perceived demand may have increased.

As the main goal of our study is to evaluate the effects of the user interface, we compare the results between our two interfaces. This comparison is shown in Figure 6.4. Here, the overall workload shows a slight advantage for the tablet user interface. The same is also reflected in all of the subscales, except for the level of perceived performance which remains practically independent of the interface. The largest change between the interfaces is shown in the reduced level of frustration for the tablet interface. Along with that, the amount of

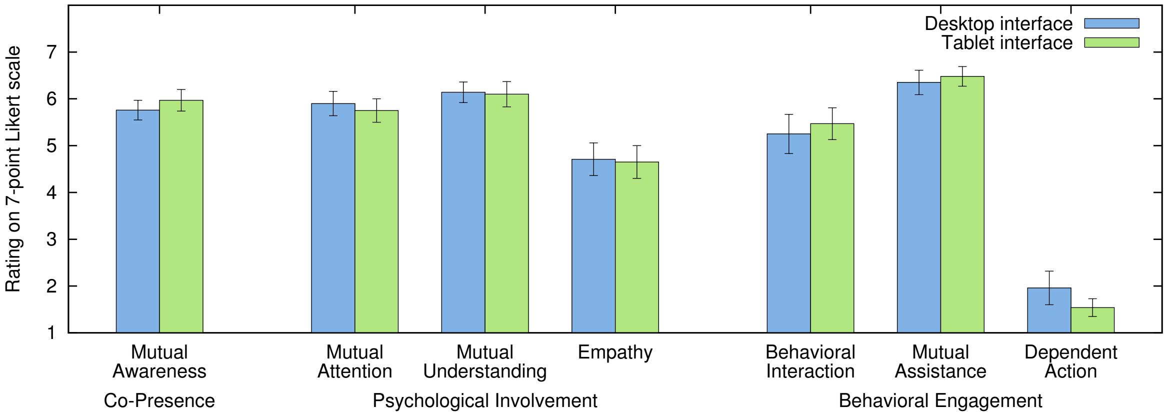


Figure 6.5: Results of the Networked Minds questionnaires, depending on the used interface. The values are shown including the standard error.

effort is reduced, which may also be a reason for the lower level of frustration. Another relatively large difference can be seen in the temporal demand. For the tablet interface, the users rated the temporal demand lower as for the desktop interface. This is despite the actual measurements which show that there was practically no difference, in fact even a very slight increase. The reason for this may be that the subjective perception of time is linked to the level of frustration.

6.4 Networked Minds

In order to evaluate the results of the Networked Minds questionnaires, we group the 38 individual questions and aggregate the results into seven scales. We do not separate the users' own rating from their rating of the other team member, because most users gave the same answers within each pair of questions. For calculation of the scales, we averaged the results of the corresponding questions. This gives values in a range from 1 to 7, where higher numbers are better.

The calculated scales are grouped into three different categories. Figure 6.5 shows our results. The first category is co-presence and contains a single scale on mutual awareness. This scale is calculated from three question pairs and measures to which degree users were aware of the presence of the other participant. The results show a high value for both interfaces, which is expected as the users were located in the same room during the study.

The second category deals with psychological aspects, and contains three scales. Mutual attention measures whether users paid attention to each other and were not distracted or only pretended to pay attention. It is calculated from the results of four question pairs. For the mutual understanding scale, three question pairs are used. It describes how clear and comprehensible the thoughts and opinions were to the other. The last item is empathy and shows if users are influenced by the mood of their partner. Three question pairs are used as the base for this scale. The results for psychological involvement show on average high levels of mutual attention and understanding. For all three scales, there is a very slight decrease for the tablet interface. It is not clear that these small changes correspond to an actual effect, but it would be plausible, because using tablets allows the users to do partial tasks on their own.

The last category is about behavioral engagement. The first of three scales describes behavioral interaction and uses three question pairs as a source. It measures how much the actions of one user depend on the actions of the other user. Two question pairs are used to calculate the mutual assistance value. That measures how much the users helped each other to complete the task. The last scale measures dependent action and is the result of a single question pair. It shows to which degree users think they would not have been able to solve the tasks on their own. The results regarding behavioral engagement are interesting. On one hand, users rated the level of mutual assistance very high for both interfaces, meaning they worked together and helped each other. On the other hand, the low result for dependent action shows that users think they would have been able to solve the tasks on their own as well.

6.5 Users' Feedback

In this section the users' qualitative feedback on our system is presented. This includes comments on the system while the participants were using the system and from the discussion afterwards. When asked directly which kind of user interface they preferred, 11 participants chose the tablet interface and one was undecided between the tablet and desktop interfaces.

Participants described the tablet interface as being nicer, clearer, more intuitive, more fun and more usable. One user noted that the tablet application looks nicer than the desktop Java application. Another user said that he preferred navigation on the map using touch input instead of a mouse. The time selector in the tablet application was preferred over the slider in the desktop application. As a further improvement, a user suggested the possibility to move the timeframe by an exact amount of time, such as the duration of the time frame. Currently, there is only a continuous slider. Another suggestion for the tablet interface was to show the list of layers without having to swipe from the left edge of the screen.

Multiple users found that the tablet interface offers advantageous features for collaborative work. Specifically, the participants liked the ability to create and modify multiple layers at the same time using the tablet interface. Some users also related aspects that could be improved. One participants said that it wasn't obvious from the user interface which actions are synchronized just for the active layer and which ones apply to all layers. A suggestion was to allow temporal filtering only for the current layer. In our system, the temporal filtering affects all layers at once, because applying the filter to a single layer could lead to confusion. One comment by a user mentioned that there are less conflicts using the desktop interface. For example, when multiple users move the map in the tablet interface, this can lead to unexpected behavior.

There were some participants who found the content on the large screen difficult to read. Multiple users said that the window of the desktop control application and the font was very small. Some participants also had difficulties reading the names of places on the map. There are several reasons for this. First, users found that the font size is small and should be scaled. One user said that they had to lean forward in order to see the location names on the map. Second, in some cases the names of locations on the map were obstructed by the tag map. Users found that the tablet interface is an improvement in this regard, because the location names can be read on the tablet screen. As a general issue with the background map, a user noted that some city names are not shown at all on the OpenStreetMap tiles.

A common complaint was that the desktop map application responded sluggishly. Users found that the user feedback on the map application was bad. More specifically, the tile rendering of the map was too slow and that the response time while moving the map was described as annoying. One user suggested preloading the map tiles in order to improve the loading speed. However, we had previously tried to use offline map tiles, but this did not improve the performance significantly. Similar issues were reported for the automatic time filter playback. One team had 4 layers set up which caused some time frames to be skipped when selecting a fast speed.

Related to the desktop map application, users also reported that the zoom function was not ideal. Some users lost their point of interest when they zoomed quickly and the application did not respond as fast. Participants also noted that the map should zoom onto the location where the mouse is pointed instead of the center of the screen. Also, users noticed that after using the control application window on the desktop, it was not possible to zoom using the scroll wheel on the mouse. It was only working after the map window was focused by clicking in it, contrary to what some users were used to.

In general, users preferred the tablet application for panning and zooming the map. The reasons for this are that the map on the tablet allows for immediate feedback without having to wait for it to load. Participants also found the tablet application easier for zooming, because it was easier for them to zoom to an exact point. As a suggestion for improvement, some participants proposed to add a mini map to the large display in order to keep orientation while moving around. Another suggestion was to include a way to jump directly to a location on the map, for example by entering the name of a city.

One user perceived the tablet interface like an unneeded gimmick, and had no clear preference for one of the interfaces. For him, an advantage of the desktop interface was not having to switch focus between different devices. He said that staying focused on the large screen is preferential, especially for setting up filters. Another user found that it was possible to look at the large screen and move the map on the tablet at the same time. In general, users preferred entering text for filters on the physical keyboard instead of the tablet.

7 Conclusion and Future Work

In this chapter, we summarize our work of designing, implementing and evaluating our system. Then, we present the conclusion of our work. Finally, we give an outlook on future work related to multi-device interaction.

Summary and Conclusion

Today, large amounts of data are available and are still increasing. They can be used for many different applications, including emergency response. For this, methods from visual analytics are used. Large displays can be used to improve the process of visual analytics, for example high resolutions allow to show details and get an overview using the same setup. Another advantage of wall-sized displays is that they can be used by multiple users at the same time. However, existing user interfaces are not well-suited for large displays. Thus, we combined a wall-sized display with a tablet for user interaction.

We designed a number of prototypes for such a tablet interface that allows interacting with a large display. As part of that, we created a concept for navigation in context of a map application. This design includes the ability to filter a data set. We also designed two different concepts for selection of points using a tablet. In these concepts, selection works either by touch input or by using the tablet to point at the wall-display. For the actual data, we discussed two different approaches for map overlays. On one hand, the content can be shown by means of a tag map and on the other hand a heat map shows the frequency of data points. Finally, we considered features for collaborative work. The concept of layers allows multiple users to work on the system at the same time.

For the implementation we used a number of data sets from the microblog service Twitter. These data sets contain a large amount of information, because they contain textual data in addition to geolocations. We chose to implement some of our designed prototypes. In addition, we created our own communication protocol and server that allows other components of the system to exchange messages. Our map viewer for the large displays implements the previously discussed overlay types. The main part of our user interface consists of the tablet application. It is similar to the original design, but additionally supports layers for collaborative work. Additionally, we developed a desktop application for controlling the map, which is used in our study as a reference.

We evaluated our system by conducting a user study with twelve participants. In order to test the collaborative features, the participants worked in pairs of two. All of them used both of our user interfaces. For the study we used two different data sets based on a trade fair and a flood event. During the study, the participants answered questions on the data sets using our system. For the quantitative analysis, we used two different questionnaires. The NASA TLX questionnaire measured the workload and the Networked Minds measure was used to evaluate collaborative aspects. We also collected qualitative feedback from our participants.

Our system proved to be usable, as participants were generally able to solve the given tasks. The study also showed that users were not quicker when using the tablet interface. However, the perceived temporal demand was lower for the tablet interface. The results of the NASA TLX questionnaire did not show a meaningful difference between different data sets or between the first and second round. However, the overall workload was lower for the tablet interface. Subjectively, most users preferred the tablet interface over the desktop interface. Thus, the results of our tablet interface are positive overall.

Future Work

The users gave us different suggestions for improvement during the study. A common complaint was the size of the fonts on the large screen, which should be increased in order to make it more readable. Also, users wanted improved filters, like a way to do more exact time based filtering. As some users also tended to get lost in the data set while zooming, a mini map can be added to prevent losing orientation.

In general, users liked the features of the tablet interface for collaborative work. However there is still room for further improvement. As described in our original design, it would be useful to add the possibility to work independently on the tablet screen and merge the results later. This would allow individual users to focus on their special area of knowledge without distracting others.

There are also other concepts from our original prototypes that are not yet implemented. So far, we focused on touch input using the tablet screen. An opportunity for future work is interaction involving movement of the tablet, such as using it for pointing on the screen. In general, a way to select points using the touch screen in order to view details is also a possibility.

Another example for future work that has not been discussed previously is support for real-time data and addition of a mobile application for on-field users. This would allow to improve the system by being able to verify the validity of microblog content through actual observation. In addition, further information that is not contained in the messages could be obtained.

A Appendix

Questions for the CeBIT data set

1. Which company searches for employees on Twitter, and where?
2. What is the meaning of the term “ding”?
3. Which well-known major event in IT is taking place?
4. Are there other events of this type? Name and place are sought.
5. When was the ITB in Berlin opened?
6. In which hall is the stand of Telekom located on CeBIT?
7. Where is the stand of Sipgate?
8. Which is the day when lots of messages about smart home were posted?
9. When did CeBIT open in 2012?

Questions for the flood data set

1. Which natural event is happening at the moment?
2. What is the city where a transformer station (“umspannwerk”) occurs in connection to the flood?
3. When and at which place have sandbags been filled near Magdeburg?
4. What is happening there at this time and in this context? (e.g. cause)
5. Since when is there a significant increase in messages about the flood in Dresden?
6. On which day was “Wetten, dass..?” broadcast?
7. When did a special event occur in Frankfurt? What is it about?
8. Which music event is taking place in the west of Germany?
9. On which days did “Rock am Ring” take place, or when are there many messages?

Bibliography

[AGG+13] M. T. I. Aumi, S. Gupta, M. Goel, E. Larson, S. Patel. Doplink: Using the doppler effect for multi-device interaction. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 583–586. ACM, 2013. (Cited on page 13)

[BBF12] D. Baur, S. Boring, S. Feiner. Virtual projection: exploring optical projection as a metaphor for multi-device interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1693–1702. ACM, 2012. (Cited on page 14)

[BHG01] F. Biocca, C. Harms, J. Gregg. The networked minds measure of social presence: Pilot test of the factor structure and concurrent validity. In *4th annual International Workshop on Presence, Philadelphia, PA*, pp. 1–9. 2001. (Cited on page 33)

[BHKM04] E. Braun, A. Hartl, J. Kangasharju, M. Mühlhäuser. Single Authoring for Multi-Device Interfaces. In *Adjunct Proceedings of the 8th ERCIM Workshop "User Interfaces For All".(2004) http://www.ui4all.gr/workshop2004/publications/adjunct-proceedings.html*. 2004. (Cited on page 14)

[CBF14] O. Chapuis, A. Bezerianos, S. Frantzeskakis. Smarties: An input system for wall display development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2763–2772. ACM, 2014. (Cited on pages 13 and 24)

[CLMT12] K. Cheng, J. Li, C. Müller-Tomfelde. Supporting interaction and collaboration on large displays using tablet devices. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 774–775. ACM, 2012. (Cited on page 14)

[ESRF06] K. Everitt, C. Shen, K. Ryall, C. Forlines. MultiSpace: Enabling electronic document micro-mobility in table-centric, multi-device environments. In *Horizontal Interactive Human-Computer Systems, 2006. TableTop 2006. First IEEE International Workshop on*, pp. 8–pp. IEEE, 2006. (Cited on page 15)

[FES+06] C. Forlines, A. Esenther, C. Shen, D. Wigdor, K. Ryall. Multi-user, multi-display interaction with a single-user, single-display geospatial application. In *Proceedings* *of the 19th annual ACM symposium on User interface software and technology*, pp. 273–276. ACM, 2006. (Cited on page 15)

[FVWSN08] J.-D. Fekete, J. J. Van Wijk, J. T. Stasko, C. North. The value of information visualization. In *Information visualization*, pp. 1–18. Springer, 2008. (Cited on page 10)

[GPS09] G. Ghiani, F. Paternò, L. D. Spano. Cicero designer: an environment for end-user development of multi-device museum guides. In *End-User Development*, pp. 265–274. Springer, 2009. (Cited on page 13)

[GPSS09] G. Ghiani, F. Paternò, C. Santoro, L. D. Spano. UbiCicero: A location-aware, multi-device museum guide. *Interacting with Computers*, 21(4):288–303, 2009. (Cited on page 13)

[HA08] J. Heer, M. Agrawala. Design considerations for collaborative visual analytics. *Information visualization*, 7(1):49–62, 2008. (Cited on page 11)

[Har06] S. G. Hart. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pp. 904–908. Sage Publications, 2006. (Cited on page 33)

[HPN00] R. Han, V. Perret, M. Naghshineh. WebSplitter: a unified XML framework for multi-device collaborative Web browsing. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pp. 221–230. ACM, 2000.

[HS88] S. G. Hart, L. E. Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988. (Cited on page 33)

[HVHC$^+$08] J. Heer, F. Van Ham, S. Carpendale, C. Weaver, P. Isenberg. Creation and collaboration: Engaging new audiences for information visualization. In *Information Visualization*, pp. 92–133. Springer, 2008. (Cited on page 12)

[KBBC05] M. Kaltenbrunner, T. Bovermann, R. Bencina, E. Costanza. TUIO: A protocol for table-top tangible user interfaces. In *Proc. of the The 6th Int'l Workshop on Gesture in Human-Computer Interaction and Simulation*, pp. 1–5. 2005. (Cited on pages 15 and 24)

[KJM$^+$07] S. Kim, Y. Jang, A. Mellema, D. S. Ebert, T. Collins. Visual analytics on mobile devices for emergency response. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pp. 35–42. IEEE, 2007. (Cited on page 11)

[KKEM10] D. A. Keim, J. Kohlhammer, G. Ellis, F. Mansmann. *Mastering the information age-solving problems with visual analytics*. Eurographics Association, 2010. (Cited on pages 9 and 11)

[LJL+08] H. Lee, H. Jeong, J. Lee, K.-W. Yeom, H.-J. Shin, J.-H. Park. Select-and-point: a novel interface for multi-device connection and control based on simple hand gestures. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, pp. 3357–3362. ACM, 2008. (Cited on page 15)

[MWE+15] S. Mittelstädt, X. Wang, T. Eaglin, D. Thom, D. Keim, W. Tolone, W. Ribarsky. An Integrated In-Situ Approach to Impacts from Natural Disasters on Critical Infrastructures. In *System Sciences (HICSS), 2015 48th Hawaii International Conference on*, pp. 1118–1127. IEEE, 2015. (Cited on page 11)

[NSY+07] M. A. Nacenta, S. Sakurai, T. Yamaguchi, Y. Miki, Y. Itoh, Y. Kitamura, S. Subramanian, C. Gutwin. E-conic: a perspective-aware interface for multi-display environments. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pp. 279–288. ACM, 2007. (Cited on page 14)

[PC05] P. Pirolli, S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, volume 5, pp. 2–4. 2005. (Cited on page 10)

[Rek98] J. Rekimoto. A multiple device approach for supporting whiteboard-based interactions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 344–351. ACM Press/Addison-Wesley Publishing Co., 1998. (Cited on page 15)

[SMD12] M. Spindler, M. Martsch, R. Dachselt. Going beyond the surface: studying multi-layer interaction above the tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1277–1286. ACM, 2012. (Cited on page 13)

All links were last followed on December 9, 2015.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature