Chair for Computer Science 10 (Media Computing and Human-Computer Interaction)

RWTH AACHEN UNIVERSITY

Gaze-Supported Interactions on Ultrawide Displays

Master's Thesis
submitted to the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Kerstin Kreutz

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Ulrik Schroeder

Registration date: 16.01.2020
Submission date: 28.04.2020

Eidesstattliche Versicherung

______________________________ ______________________________

Name, Vorname Matrikelnummer

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/ Masterarbeit* mit dem Titel

____

____

____

selbständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

______________________________ ______________________________

Ort, Datum Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

§ 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

______________________________ ______________________________

Ort, Datum Unterschrift

Contents

Abstract ix

Überblick xi

Acknowledgements xiii

Conventions xv

1 Introduction 1

1.1 Research Interest 3

1.2 Outline 3

2 Related work 5

2.1 Gaze and Head Interaction 5

2.2 Large and Distant Displays 9

3 Motivation and Prototype 15

3.1 Implementation 16

3.1.1 PPI Touch Detection 16

3.1.2 iOS Application 17
3.1.3 macOS Application 18
3.2 User Study 23
Apparatus and Techniques 24
Tasks and Targets 24

4 Evaluation 29
4.0.1 Completion Time 29
4.0.2 Success Rate 30
4.0.3 Questionnaire 31
4.0.4 Discussion 33

5 Summary and future work 35
5.1 Summary and contributions 35
5.2 Future work 36
Large Displays 36
In-car Entertainment Systems 37

A QUESTIONNAIRE 39

Bibliography 49

Index 53

List of Figures

2.1 Look & Touch mobile application 8

2.2 uPen 10

3.1 The calibration screen of the macOS application 19

3.2 The macOS application 20

3.3 Setup of the prototype 25

3.4 Target coordinates 26

4.1 TrackingLocking Completion Time 30

4.2 TRACKINGLOCKING Success Rate 31

4.3 User Ranking 32

Abstract

In recent years displays have increased in size everywhere, with the advancement of autonomous cars even in in-car entertainment systems. Increasing display sizes present several interaction challenges for the user. In order for the user to see the whole screen, a large display needs to be placed at a proper distance to her. Since Direct Touch is one of the most commonly used interaction techniques, even for large displays, it is difficult for a user to stay at the required distance and still be able to interact with the display.

To solve this problem with an interaction technique that is comfortable to use we propose several gaze- and head-supported touch interaction techniques in this thesis. The mentioned techniques are investigated in the context of an in-car entertainment system use case scenario. Two currently used interaction techniques, i.e, Direct Touch and Focus interaction, as well as 16 gaze- or head-supported touch interaction techniques are implemented in a prototype. These techniques are then compared regarding *Completion Time* in seconds and *Success Rate* in hit ratio for target selection tasks in a user study with twelve participants.

The results of the study show that one of the gaze-supported touch interaction techniques performs as fast and as accurate as Direct Touch. This results in a valuable alternative to Direct Touch in the context of large display interaction techniques from a seated position.

Überblick

In den letzten Jahren konnte man einen Trend in stetig ansteigenden Bildschirmgößen allerorts erkennen, durch den Fortschritt bei autonomen Autos sogar in Entertainmentsystemen von Autos. Größere Bildschirme stellen Nutzer vor neue Herausforderungen. Um den gesamten Bildschirm zu sehen, muss der Bildschirm in einer adequaten Entfernung zur Nutzerin platziert sein. Da Direct Touch eine der beliebtesten Interaktionsmöglichkeiten –auch auf großen Displays– ist, wird es für Benutzer schwieriger den benötigten Abstand einzuhalten und gleichzeitig mit dem Bildschirm zu interagieren.

Wir präsentieren in dieser Arbeit einige blick- und kopfbewegungsunterstützte Touch Interaktionen, um dieses Problem zu lösen. Diese Techniken werden in Hinsicht auf Benutzbarkeit in Fahrzeug-Entertainmentsystemen getestet. Sowohl zwei aktuell bereits verbreitete Techniken, Direct Touch und Focus Interaction, als auch 16 blick- und kopfbewegungsunterstützte Touch Interaktionen sind in einem Prototypen umgesetzt. Die Techniken werden unter den Aspekten *Durchführungszeit* in Sekunden und *Erfolgsrate* in Trefferquote durch Auswahlaufgaben in einer Nutzerstudie mit zwölf Teilnehmern getestet.

Die Ergebnisse der Studie zeigen, dass eine blickunterstützte Touch Interaktion genauso schnell und akkurat wie Direct Touch genutzt werden kann. Dadurch gibt es eine adequate Alternative zu Direct Touch bei der Interaktion mit ultrawide Displays aus einer sitzenden Position.

Acknowledgements

I want to thank Simon for giving me the opportunity to work on this thesis and being a great supervisor. Also, I'd like to thank Sebastian for answering any questions I had regarding implementation details of ARKit. Furthermore, I want to thank all participants of the study for taking the time to participate.

Thanks to Florian for keeping me company during lunch and every discussion we had about our theses. I also appreciate the advice of everyone at i10 during the construction of the study.

Last but not least I would like to thank my family, Nick, and my friends for supporting me and being patient with me.

Thanks!

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Source code and implementation symbols are written in typewriter-style text.

`myClass`

The whole thesis is written in American English. We use the plural form for the first person. Unidentified third persons are described in female form.

Chapter 1

Introduction

Problems of interacting with large screens

Large displays are placed in many different locations, for example public spaces [2012] or shop windows [2009]. They are even placed in cars as entertainment systems[1], because of the advancement in autonomous cars. When we interact with large displays we have to consider different constraints than usually, e.g., a larger distance to the screen in oder for the person interacting with it to see the whole screen. Therefore, larger displays require interaction techniques that scale to the size of the display and adjust to the large distance between the user and the screen.

Currently established techniques

A commonly used technique to interact with large screens, such as interactive tabletops, is direct touch. This interaction technique is not well suited to a large screen, since it would be difficult for the user to reach the whole display. It might even be near impossible when considering a use case such as the in-car entertainment system, because the user would be in a seated position and constrained by a seatbelt. Another possible interaction technique for large displays would be voice control. An interface with buttons, e.g., in form of a remote control, would work for all use cases, but it would be a bother to use, since large displays allow for many different interactions which would be hard to operate with a remote control. When a trackpad is used an interaction technique called focus interac-

---

[1] https://www.byton.com/m-byte/experience (Accessed: April 28, 2020)

tion is commonly used. This interaction technique is easy to use, as it consists of flicking over a touch pad, but for large distances between two targets it can become quite tedious since one has to flick quite often to cover the space between the targets. If the user is near a microphone, this technique can even work from a distance, so it would allow the user to see the whole screen. But voice control suffers in settings where multiple users might want to work together or in places where there are many background noises. Again, in the use case of in-car entertainment systems this will be error prone because of traffic noises and possible other passengers talking inside the car.

Gaze interaction as a solution

An adequate interaction technique to bridge the distance between users and the large display and that scales to larger display sizes properly would be gaze interaction. This idea is based on the fact that our gaze can cover large distances in a short period of time and even reach distant places, which would work well for both the distance between users and large displays as well as the distance between on-screen targets. In the context of interaction techniques for ultra wide displays, this would solve the reachability problem.

Problems of gaze interaction

However, gaze interaction comes with another set of problems. If gaze or head interaction is the sole interaction technique and thus used for target acquisition as well as target selection, especially the ”Midas Touch” problem makes the interaction troublesome for the user. This problem describes the difficulty of differencing between a simple look at a target, e.g, while the gaze moves, and the actual selection task that should be triggered when looking at a specific target.

Gaze-supported interaction as a solution

In this work the described problems of gaze interactions will be resolved by using a trackpad as the main input device and using gaze or head tracking as an assisting interaction technique. With gaze-supported interactions we have the ability to provide the positive aspects of gaze interaction that allow us to overcome the difficulties of large displays, but it further also allows for precise control of the interaction.

We think that the proposed gaze-assisted touch interaction technique can outperform the aforementioned problematic interaction techniques, because gaze interaction is faster than any of said techniques and indirect touch on a trackpad allows for fast and precise selection.

1.1 Research Interest

The idea for this thesis

As described above, current interaction techniques with large displays harbor many problems, especially in a context where the user interacts with the display from a seated position. With this work we want to investigate whether gaze- or head-supported touch interaction is a valuable alternative for usage on large screens from a seated position compared to currently used techniques direct touch and focus interaction. This will be done through a user study that compares the baseline conditions (*Direct Touch* and *Focus*) to 16 gaze- and head-supported interaction techniques.

1.2 Outline

In this work we investigate possible alternatives to Direct Touch and Focus interaction on ultrawide displays in the scenario of a user sitting in front of the display for the specific use case of interacting with in-car entertainment systems. The Introduction is followed by Chapter 2, where we present state-of-the-art gaze and head interaction techniques (Chapter 2.1), as well as interaction methods for large and distant displays (Chapter 2.2). Chapter 3 outlines our motivation for this work and describes the implementation in detail, as well as the setup of the user study that was conducted with twelve participants. In the evaluation (Chapter 4) we talk about the results of the study and discuss our findings. Chapter 5 concludes the thesis and focuses on our contribution and future work we want to do based on our results.

Chapter 2

Related work

In this chapter we will describe state-of-the-art techniques for gaze interaction as well as gaze and head tracking. Additionally, the combination of gaze and touch interaction will be discussed. Furthermore, we will talk about interaction techniques on large and possibly distant displays.

2.1 Gaze and Head Interaction

For more than 15 years gaze-based interaction has been researched in Mobile HCI [2018a], [2009]. This type of interaction was mostly possible with additional devices such as head-mounted gaze trackers or display-mounted eye trackers.

Motivation for gaze tracking

Zhai et al. [1999] presented three main points of motivation for gaze tracking

- hands-free interaction
- fast movement ability of eyes
- reducing fatigue

Hands-free interaction might be useful if people are already occupied with other tasks. The fast eye movement is also a motivation for gaze interaction, since the user is required to look at the target first in most cases, before interacting with it. If only gaze interaction could be used, fatigue and potential health problems like repetitive strain injury (RSI) might be reduced. Since this paper was published 20 years ago, some of the technical aspects do not match modern premises anymore, but the motivation remains unchanged.

Khamis et al. [2018a] identified three major application areas of gaze tracking on mobile devices

Main application areas for eye tracking

- gaze behavior analysis
- implicit gaze interaction
- explicit gaze interaction

Gaze behavior analysis is the tracking of the user's eyes without feedback. It is used for later analysis and not real-time analysis or interaction.
Implicit gaze interaction refers to the usage of the user's natural eye movements opposed to specific movements that need to be performed.
Explicit gaze interaction requires specific and conscious movement of the eyes. It can be used as gaze-only interaction or as part of a multimodal system.

Implicit gaze interaction

Implicit gaze interaction has many uses, Mariakakis et al. [2015] for example used it for *SwitchBack*, an application for smartphones and tablets that tracks the users reading progress through her eyes, notices when the gaze is turned away from the device and when it is turned back the app indicates the last read sentence to allow for a faster and more efficient task resumption. The authors found that SwitchBack increased the average reading speed by 7.7%.

Explicit gaze interaction

Explicit gaze interaction that uses specific eye movement or blinking gestures involves the risk that users get fatigued as well as the "Midas Touch" problem. This problem describes the circumstance where the users gaze activates every target, because explicit eye movement is used for interaction (see Kjeldsen [2001]).

Gaze-supported interaction

Explicit gaze interaction in the form of gaze-supported interaction on the other hand focuses on supporting different interaction techniques like touch input. It is for example used to fix reachability issues on smartphones that are too large for the user to reach every part of the screen (see Nagamatsu et al. [2010]). But the authors needed to use additional hardware to be able to integrate gaze-supported interaction on smartphones. Such modified devices may change the way the users hold and interact with the mobile device, since they are often inconveniently attached to the device and make it heavier and bulkier. The cameras used for gaze tracking might even be accidentally covered by the user's hands.

Commodity devices hardware can be used for gaze tracking

In recent years the front-facing cameras of everyday devices like Apple's iPhone X improved to high-quality and depth cameras. With this advancement gaze and head-tracking as well as gaze-enabled interaction may become more common, since they are already available on commodity devices. Because already integrated cameras are used, commodity devices do not suffer the same problems as modified mobile devices.

Khamis et al. [2018a] categorized gaze behavior analysis on unmodified handheld devices into two categories: *model-based eye tracking* and *appearance-based eye tracking*.
Model-based eye tracking is also referred to as geometric-based eye tracking, because the system uses the visible features of the user's face and eyes as well as head position to generate a geometric model of the users head. The gaze direction is then estimated by the intersection of the display and vectors through the users pupils and head direction.
Appearance-based eye tracking use machine learning based on datasets to map images of the user's eye to coordinates on the screen.

Problems of commodity devices

Even though the devices are now lightweight, there are still problems with gaze interaction. As shown by Khamis et al. [2018b], it is problematic to track the full face of a user with a commodity device's front-facing camera, because of multiple factors. Depending on the user's activity, the device is held in different grips, thus, the camera might be obscured by the users hand. Furthermore, the lighting in a room or

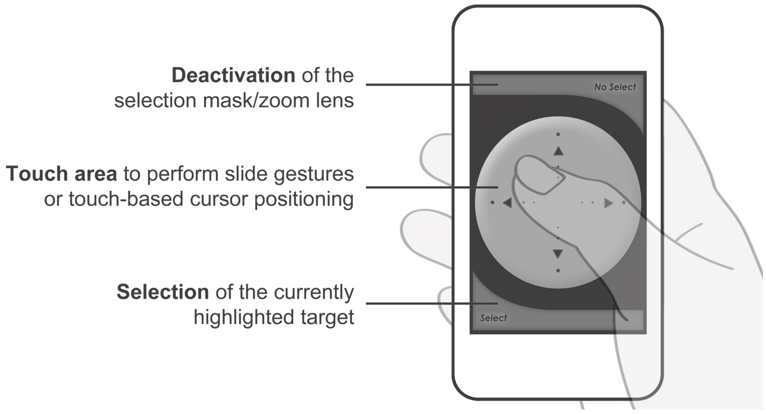

Figure 2.1: The prototype for the GUI of the mobile device application for 'Look & Touch'. Adapted from Stellmach and Dachselt [2012]

outside might make it difficult for the camera to distinguish the user's facial features. If the user is outside, parts of the face might be hidden by a scarf or hat. The authors found that the full face is visible only about 29% of the time.

Gaze-supported touch interaction enabling unimanual input on large devices

Gaze-supported interaction is often used in combination with touch interaction. Pfeuffer and Gellersen [2016] proposed gaze-supported touch interaction on tablets where touches were redirected to the gaze target. This enabled the users to reach the whole screen of the tablet without changing their grip on the device, thus, changing the selection task from requiring two hands to a task requiring unimanual input. They also proposed a cursor dragging technique in combination with the gaze-supported touch interaction for a more precise interaction type. Although the authors found their interaction techniques to be slightly slower than direct touch, it is still an improvement in terms of unimanual interaction.

Gaze suggests and touch confirms

Stellmach and Dachselt [2012] also propose gaze-supported interaction with touch input on a mobile device, because gaze interaction alone has inaccuracies based on the natural eye movement of humans, that cause measurement errors that lead to jittering and offsets. The authors counter these effects by a large-sized GUI and selecting targets through touch interaction on a mobile device (see Figure 2.1). This additional input modality also prevents the "Midas Touch" problem. Although they implemented multiple gaze-supported selection methods, all of them follow the principle *gaze suggests and touch confirms*. In these selection techniques the cursor is controlled by gaze and the fine positioning of the cursor is done by touch interaction. This method was found to be high in performance and usability according to a user study the authors conducted.

Eyes-free interaction on mobile devices as support for gaze interaction on distant screens

In another work Stellmach and Dachselt [2013] used gaze- and head-directed input again in combination with touch interaction on a mobile device to enable users to interact with distant displays. The interaction system was focused on single-users. Interaction tasks were designed to support selecting, positioning and manipulating targets on the distant display. As in the other mentioned paper by the same authors, the system follows the *gaze suggests and touch confirms* principle. The interaction on the mobile device is supposed to be eyes-free, because the main focus should stay on the distant display, and thus it should be simple like a touch and not a multitouch gesture. To overcome eye tracking inaccuracies, the authors implemented two interaction methods: a touch-enhanced gaze pointer and a gaze-directed zoom lens. The touch-enhanced gaze pointer is used in a similar way to Stellmach and Dachselt [2012] where gaze indicates the area of interest and the touch input is used for precise positioning of the cursor. The gaze-directed zoom lens increased the target sizes in three areas and the touch input can manipulate the zoom level. A crosshair is presented in the middle of the lens for target selection and selection is done by a longer touch on the mobile device. A study showed that users were fastest with with the touch-enhanced gaze pointer.

2.2 Large and Distant Displays

Large displays are challenging to design applications for since users might not be able to reach every part of the screen easily. Ardito et al. [2015] published a survey, where they present four categories for interaction modalities, namely *touch, external device, tangible object, and body* on

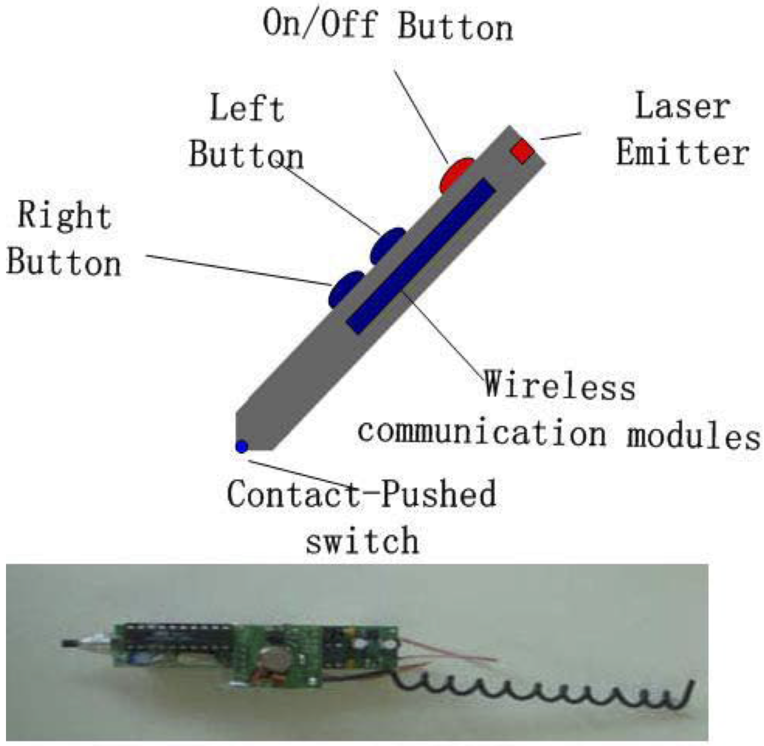

Figure 2.2: The uPen offers buttons and a laser pointer for interacting with large displays. Adapted from Bi et al. [2006]

large displays. The biggest category contains touch-based interaction with more than 50% of the considered papers' interaction systems being based on touch on the display's surface. Although not all mentioned papers consider multitouch, even displays that can recognize a hand and its orientation are mentioned.

Another 34% of the papers used for the survey described interaction with the large display over an external device that was not in direct contact with the display. The deviced used for input could be smartphones and tablets as described by Stellmach and Dachselt [2013], but furthermore might also be devices especially build for interacting with the display, like *uPen* [2006], a pen that is basically a laser pointer combined with a wireless module, three but-uPen – an external device for interaction on wall-mounted and tabletop displays

tons and a contact-pushed switch, which can be used either from a distance or directly on the screen instead of a mouse. In this case the left and right button on the uPen map to the functionalities one would expect from the left and right mouse button (see Figure 2.2). The third button emits a laser beam as well as the uPen's ID to the system. This laser beam is captured by a video camera and then interpreted by the system as the position of the cursor. In this way the uPen can be used on a wall-mounted display during a presentation, but also when the display is used as an interactive table. Additionally the uPen can be identified by the system and thus allows for multiuser and simultaneous interaction, thus multiple users might for example write on an interactive table. External devices offer even more benefits. Magerkurth and Tandler [2002] mentioned that if every user controls the large display from the distance over a private device, this device can be used to authenticate a user to grant access to for example more files on the display. In a different use case the users devices might be used to receive informative advertising or just additional information regarding the topic presented on the display.

Tangibles

21% of the papers contained in the survey describe the use of tangibles on large displays. Tangibles are physical objects that can be manipulated by users and by communicating with the tabletop system they are used on enabling manipulation of the virtual content they represent on the multitouch display, as stated by Cherek et al. [2018].
Lucchi et al. [2010] tested tangible objects made of paper in spatial layout tasks and found that they were overall faster than the direct touch alternative. With PERCs Voelker et al. [2015] presented tangible objects for capacitive multitouch tabletops that were not filtered after a specific amount of time and knew their position on the tabletop with help of a specific marker pattern, a field sensor, a light sensor and a bluetooth module. Still, these tangibles were limited to be used *on* the multitouch tabletop surface, even though they are three dimensional objects that might be lifted from the surface by a user. Cherek et al. [2019] propose that for example midair gestures might further increase the benefit of tangible interaction on large displays, because users might already be used to midair gestures from other devices such as smartphones.

Body interaction with large displays

The last category of interaction modalities described by the survey was body interaction with large surfaces. Large displays are often found in shopping malls to attract the attention of customers. Reitberger et al. [2009] used a display in a shop window to display an interactive mannequin to extend the time passersby spend in front of the shop window. Müller et al. [2012] found that displaying the mirror image of passersby, especially when somebody already interacted with the display, made potential customers notice the shop window and stop to interact with the display themselves. Therefore, large interactive displays in public places should support multi-user interaction. Although the customers otherwise need approximately 1.2 seconds to notice the possible interactions and therefore move away from the window containing the display, they often turn back, which is described as the *landing effect*.

Jota et al. [2009] presented three interaction metaphors for large displays, which offer a greater degree of physical freedom that comes with large and distant displays. The metaphors they came up with are called *grab*, *point* and *mouse*, where *grab* expects the user to walk up to the display to perform the gesture the user would use to grab a book from a bookshelf and place it somewhere else, the *point* metaphor, however does not require the user to move in front of the display, since it works over the distance with pointing to the start and end point on the screen. The mouse metaphor works like a mouse, but on the vertical plane. They conducted a user study where participants had to complete a puzzle with the three metaphors and found that *point* was the fastest and most precise technique.

Other than the whole body, systems often only track parts of a users body. That might for example be the user's hands, which can enable midair interactions above a surface with the help of depth cameras as mentioned by Wilson and Benko [2010]. Such interaction possibilities already can be found in current products like the Byton *M-Byte*[1], where cameras in the dashboard enable midair hand gestures to zoom maps for example.

Furthermore, as already mentioned in 2.1, the head and eye movement might be tracked for interaction on large displays.

---

[1] https://www.byton.com/m-byte/experience (Accessed: April 28, 2020)

In this Chapter we gave an overview of current use cases for gaze interaction and explained the problems of gaze-only interaction, which is the basis for our gaze-supported approach. Moreover, different interaction methods for large and distant displays were examined, which implied to us that gaze-supported touch interaction is a good idea for our in-car entertainment system scenario, since the other presented methods would not work based on the constraints of the use case.

Chapter 3

Motivation and Prototype

Chapter 2 of this work already showed many applications for gaze-supported interaction methods. We wanted to take a look at gaze-supported interaction on ultrawide displays, with focus on the use case of in-car entertainment systems.

Ultrawide displays in in-car entertainment systems

In recent years cars have become more and more capable, assisting the driver while driving or parking. Furthermore, autonomous cars are already in development, so step by step the driver needs to pay less attention to driving and can focus on other activities. With this advancement, car manufacturers started to build larger entertainment systems into their cars. For example, chinese car manufacturer Byton build the *M-Byte*[1], which contains a 49-inch Display that is nearly as wide as the windshield. Based on such in-car entertainment systems and considering the near future where autonomous cars can be used in everyday traffic, we wanted to investigate the possibilities of gaze-supported interaction on such displays. Considering the place for this specific use case, we had to adjust to some constraints. While sitting in a car, a person is physically constrained by a seatbelt, therefore reaching certain areas, like the other end of the windshield, is difficult from the

---

[1] https://www.byton.com/m-byte/experience (Accessed: April 28, 2020)

Constraints for in-car interactions

driver's seat. This constraint is enough to consider a different interaction technique than direct touch. Voice input was considered as a possible input technique, but there is often more than one person sitting in the car and people are talking to each other, or the radio is in use or the entertainment system might be used as a speaker for phone calls. We decided that we would not use this input technique as an alternative, because it adds new constraints like multiple users that need to be recognized. Another possibility was mouse input or cursor interaction through a trackpad, but considering the size of the screen and the space available in the center console, we also decided against this as the *only* input technique. Since there was a great advancement in recent years in gaze tracking on commodity devices, we settled on gaze-supported touch interaction. Considering that car rides are not always smooth and gaze tracking is very sensitive, we not only wanted to test gaze tracking, but also head tracking, because this alternative seems to be more stable.

3.1 Implementation

Hardware

For the implementation of the prototype an Apple iPad Pro 12,9″ was used as a device for the gaze and head tracking, a Microsoft Perceptive Pixel 55″ touch screen (PPI) was used as an ultawide display and for direct touch interaction, an Apple Magic Trackpad (Gen. 2) was used for indirect touch input, and an Apple Mac mini running macOS Mojave connected all hardware components and displayed the selection targets on the PPI. The program code was written in Swift 4.0.

3.1.1 PPI Touch Detection

Direct Touch Detection

In 2012, Microsoft bought Perceptive Pixel Inc., to create the Microsoft Surface Hub. Based on this, the device drivers for the PPI are only available on Microsoft Windows Platforms. Because the chairs' PPI has been in use for a number of years, a virtual machine running Windows 10 was already set up with a network service that announces active touches on the PPI as JSON objects.

3.1.2 iOS Application

iOS application main functionalities

We implemented an iOS application with three main functionalities:

1. Tracking the eyes and head of the user
2. Calculating the Point of Regard (PoR), i.e, the intersection of the gaze or head vector and the PPI plane
3. Sending the calculated position to the macOS application

Gaze and head tracking with ARKit

We use Apple's *ARKit Framework*, to start an ARSession that uses the iPad's front-facing camera for face tracking[2] purposes. The ARSession keeps track of multiple augmented nodes that represent a position in a three-dimensional space, which is, in our case, the real world. One node represents the device's camera, while a different node represents the center of the head, with two child nodes representing the user's eyes.
Depending on whether one wants to use gaze or head tracking, a segmented control can be adjusted in the applications' user interface (UI) that changes the calculation of the Point of Regard to either integrate the estimated gaze destination or to not integrate the estimated gaze destination but only the estimated head destination.
Sending the calculated position of the PoR is done by a framework that was previously written at the chair. Not only the PoR, but also the current setting to either use or not use gaze tracking, as well as the user's head position and angle are broadcast.

For the best possible gaze and head tracking, the iPad should be placed in portrait mode with the front-facing

---

[2] https://developer.apple.com/documentation/arkit/tracking_and_visualizing_faces (Accessed: April 28, 2020)

camera at the top of the device in between the PPI and the user with the display facing the user. This way the camera can focus directly on the face without being obscured by the iPad stand. During the interaction process with the macOS application, as long as the gaze and head tracking function is needed, the iPad needs to be active, therefore, in the 'Display & Brightness' settings we set 'Auto-Lock' to 'Never'.

3.1.3 macOS Application

macOS application main functionalities

The macOS application has six main functionalities:

1. Calibration that allows proper mapping of PoRs
2. Task selection
3. Setup of the different target selection tasks and presenting views depending on the selected task
4. Communication with the magic trackpad and correct mapping of touches on the trackpad to the presented views
5. Interpreting the received PoR and updating the views accordingly
6. Logging the received data during target selection tasks

macOS application setup

When the macOS application is started, the user sitting in front of the PPI and the iPad has to calibrate the system to enable reliable gaze interaction on the PPI. Therefore, the application is started in fullscreen and the user has to look at nine small squares in a grid of 3×3 squares and click on the magic trackpad to save the gaze coordinates for the squares as reference points throughout the whole interaction process (see Figure 3.1).
Currently the task selection is done in a separate tab or window that can be opened through the MenuBar at 'InCarGazeTracking'>Preferences or through pressing

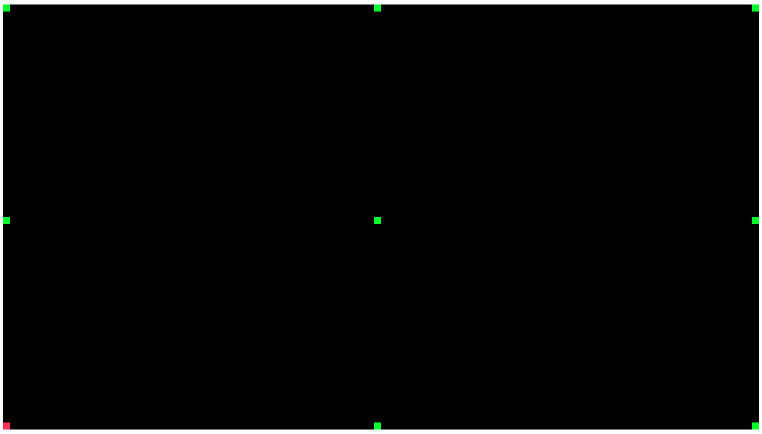

Figure 3.1: The calibration screen that is presented when the macOS application is started. Beginning from the lower left corner to the upper right corner the pink square needs to be looked at, then a touch on the trackpad needs to be performed and then the next square turns pink while the previously selected square turns green again.

command[⌘], keys. Only target selection tasks are available in this menu, because they were the focus of the user study. A total of 18 target selection tasks was implemented. The basic elements of these tasks look mostly the same. 72 green colored squares that function as targets in a grid of 12×6 squares (see Figure 3.2) are displayed. They can either be 30×30 pt or 60×60 pt in size. A color change to a light blue color indicates the currently active target that needs to be selected next. Twelve targets were chosen for the selection tasks and are active twice per trial in a random order. Due to the random ordering of the targets, a target might need to be selected two times in a row. In every condition, except for the two baseline conditions, a virtual trackpad was presented on screen in a gray color (see Figure 3.2).

Baseline conditions

Two of the selection tasks are considered as baseline conditions, as we got inspired for these conditions from interaction techniques currently used in cars.
One of them is **Direct Touch**, where the user has to select the targets directly on the touch screen of the PPI. During the touch, no matter where the user's finger is positioned on the PPI, the active target is displayed in a light blue color.

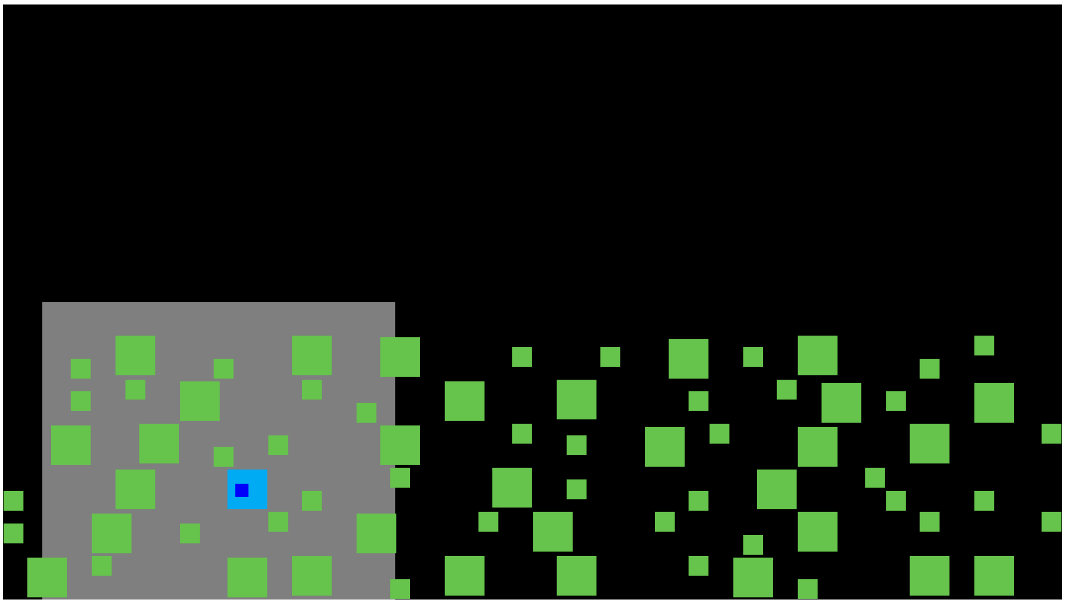

Figure 3.2: The layout of the macOS application. The gray rectangle is the virtual trackpad, the dark blue square represents the touch on the trackpad and the light blue target is the target that should be selected next.

Once the user lifts her finger, the next target will be changed to a blue color, while the last target reverts back to the green color.
The other technique is called **Focus**, which moves a pink cursor in the direction of the swipe the user performs on the trackpad. This should to be repeated until the target is reached and a click is performed, although a click will select the target that is currently selected by the cursor, independently of the question whether it is the correct target.

Condition composition

Other than the baseline techniques we decided on 16 conditions that are composited from three categories: **tracking, mapping and locking**.

Gaze and head *tracking*

We wanted to take a look at the difference between *gaze tracking* (G) and *head tracking* (H), hence, we added two **tracking methods**. Although the tracking methods are implemented in the iOS application and chosen on the iPad, we added this information to the macOS application as well, since the different parts of the tasks are combined in this place and it is important for the logging process.

Absolute and relative *mapping*

Other than the tracking methods, we wanted to see if the **mapping** of the touch points on the magic trackpad to the virtual trackpad on the screen had an effect on selecting the targets. The mappings we decided on are an *absolute mapping* (A) and *relative mapping* (R).
In *absolute mapping* conditions, the position of the user's finger on the actual trackpad is directly mapped to the same position in the virtual trackpad, e.g, if the user positions her finger in the upper left corner of the trackpad, a small dark blue square representing the touch is visualized on screen in the upper left corner of the virtual trackpad. *Relative mapping* conditions differ in the point that the user's initial finger position on the trackpad does not matter, because it is always mapped to the center point of the virtual trackpad. From that point the user can move her finger in any direction she likes and the dark blue square representing the touch in the virtual trackpad will be moved accordingly.

Locking methods: None, onTouch, onClick, onMove

In addition to these techniques we also wanted to test different methods of **locking** the virtual trackpad, so that it would stop moving with the gaze or head movement. We implemented four techniques for locking, *none* (**N**), *on touch* (**T**), *on click* (**C**) and *on move* (M). When the condition's locking type is *none*, it is not possible for the user to lock the virtual trackpad in place, therefore, it will always move with the head or gaze movement. If the selected locking type is *on touch*, the user will lock the virtual trackpad in the position it currently is in, once her finger touches the trackpad. When the finger is lifted without a click on the trackpad, the virtual trackpad will again move with the gaze or head movement. The locking type *on click* will let the user lock the virtual trackpads' position through a click on the trackpad. The second click, following the click to lock the virtual trackpad, is used to select the target, thus, the locking for one target cannot be undone. In case the selected locking type is *on move*, the locking works similar to the *on touch* locking method, but the virtual trackpad is only stopped once the user's finger moves on the trackpad. Therefore, the user can rest her finger on the trackpad without locking the virtual trackpad, and when it is positioned correctly through head or gaze movement, the finger can be moved to lock it. If the finger is lifted from the actual trackpad and no click happened, the virtual trackpad can be repositioned and locked again.

List of conditions tested in the user study

The 16 conditions were then formed by each possible combination of the different tracking, mapping and locking possibilities. Namely they are GazeAbsoluteNone (GAN), GazeAbsoluteTouch (GAT), GazeAbsoluteClick (GAC), GazeAbsoluteMove (GAM), GazeRelativeNone (GRN), GazeRelativeTouch (GRT), GazeRelativeClick (GRC), GazeRelativeMove (GRM), HeadAbsoluteNone (HAN), HeadAbsoluteTouch (HAT), HeadAbsoluteClick (HAC), HeadAbsoluteMove (HAM), HeadRelativeNone (HRN), HeadRelativeTouch (HRT), HeadRelativeClick (HRC) and HeadRelativeMove (HRM).

Additionally implemented features for future work

Although we had already implemented live feedback for the – in case of head tracking *estimated* – point of regard, which seemed promising regarding accuracy, we decided against visualizing this, because the user's eyes and attention were already caught by the virtual trackpad as well as the visualization of the touch point. Furthermore, we already implemented additional features that allow for future user studies. These features are the ability to distinguish multiple force levels on the magic trackpad, in case one wants to implement applications that use a quicklook function, and multiple gesture recognizers for dragging, zooming and rotating targets. With these additions we can take a closer look at gaze-supported interaction techniques in applications like Google Maps.

Log file structure

The last task of the macOS application is to log the user's interaction with the system. This is done through logging data in two csv files per condition. The results of each task are written to a file called ***UserID*Results*MappingLockingTracking***, where *UserID, Mapping, Locking* and *Tracking* are each exchanged for the currently selected option. For the results 26 attributes are logged, namely Task#, UserID, Tracking Method, Locking Method, Mapping Method, Total Selection Time, Target, Selected Object, Correct Target Selected, Width of Target Node, Height of Target Node, X Position of Target Node, Y Position of Target Node, X Position of Touch Down, Y Position of Touch Down, X Position of Touch Up, Y Position of Touch Up, Target Grid Area, Target Grid Row, Target Grid Column, X Head Position, Y Head Position, Z Head Position, Roll, Pitch and Yaw. This csv file is written to when a target selection was detected. The second log file, called ***UserID*Stream*MappingLockingTracking***, where *UserID, Mapping, Locking* and *Tracking* are also each exchanged for the currently selected option, but a few more attributes are logged. In addition to the aforementioned attributes, the stream files further contain a TrialCounter to map each entry to a certain trial, since the stream file is logged once per frame – thus there are multiple logs per trial and the time per frame.

Based on our work up to this point, we formulated several hypotheses:

Hypotheses

H1 There is a difference in *Completion Time* between the tracking methods, i.e., between HEAD, GAZE, DIRECT TOUCH and FOCUS.

H2 GAZE has a lower *Completion Time* than HEAD.

H3 There is a difference in *Completion Time* between the locking methods, i.e., between NONE, TOUCH, CLICK, MOVE.

H4 NONE has a higher *Completion Time* than every other locking method for both HEAD and GAZE tracking.

H5 There is a difference in *Completion Time* between the mapping methods, i.e., between ABSOLUTE and RELATIVE.

H6 There is a difference in *Success Rate* between the tracking methods, i.e., between HEAD, GAZE, DIRECT TOUCH and FOCUS.

H7 HEAD has a higher *Success Rate* than GAZE.

3.2 User Study

We wanted to take a look at gaze-supported or head-supported interaction possibilities on ultrawide displays.

Since large displays are mostly interacted with through direct touch or focus interaction techniques as described in 2.2 “Large and Distant Displays”, which are tedious or difficult to use in a seated position, we decided to test our gaze-supported interaction techniques for target selection in comparison to direct touch and focus target selection. To understand how those techniques compare, we conducted a user study with 12 participants (20-34 years, M = 26.58, SD = 3.53). We decided on two baseline conditions, Direct Touch (DT) and Focus (F) as they have been shown to be easily used and are state-of-the-art interaction techniques in cars. The other 16 conditions (GAN, GAT, GAC, GAM, GRN, GRT, GRC, GRM, HAN, HAT, HAC, HAM, HRN, HRT, HRC, HRM) we wanted to compare with the baseline conditions are composited from 2 TRACKING $\times$ 2 MAPPING $\times$ 4 LOCKING options. We asked users to select targets either with a directly with a finger on the PPI in DT or for every other condition indirectly with one finger on the trackpad while sitting in front of the left side –the driver’s side in Europe– of the PPI.

Apparatus and Techniques

As described in Chapter 3.1 “Implementation”, we used an Apple iPad Pro 12,9”, a Microsoft Perceptive Pixel 55” touch screen, an Apple Magic Trackpad (Gen. 2), and an Apple Mac mini, with the PPI presenting the targets and reacting to DT and the Mac mini capturing the interaction data (see Figure 3.3).

We implemented the baseline techniques and our techniques as described in 3.1.3 ”macOS Application”.

Tasks and Targets

Task setup

For our study participants were asked to select a target using each of the 18 techniques. At the start of each trial one of the green targets turned blue. This target was to be selected by the user either directly by hand (DT), with a pink cursor (F) or after moving the virtual trackpad over the target

Figure 3.3: The setup of the prototype during the user study.

through gaze or head movement by moving the visualized touch through indirect touch interaction on the trackpad to the target. The confirmation of a selection was either lifting the finger in the DT condition or a finished click on the trackpad. Once a target selection was confirmed a new target turned blue and the recently selected one turned back to green. If the last target was selected, a "condition finished" label appeared. The targets were arranged in a 12×6 grid over the lower half of the PPI to simulate an ultrawide display and windshield. We shifted each target in its cell to avoid a regular looking grid and to increase the level of difficulty. The decision to only place targets on the lower half of the display was made to better match our use case of a modern car's head-up display, which also matches ultrawide displays. Furthermore, we also placed the trackpad to the user's right-hand side and placed the user in front of the left-hand side for this reason.

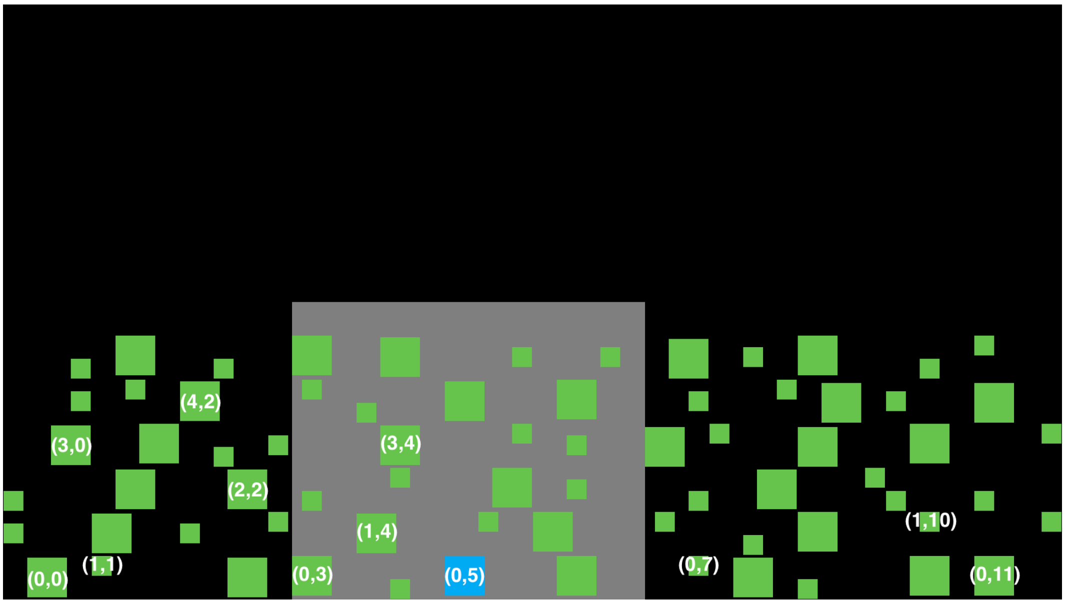

Figure 3.4: The targets were arranged in a 12×6 grid, as labeled by the coordinates, which were not presented during the user study.

Variables

Independent variables

The **Independent Variables** were TRACKING (DT, F, G, H), LOCKING (A, R), MAPPING (N, T, C, M) and TARGET. Our twelve targets were the targets (0,0), (0,3), (0,5), (1,1), (1,4), (2,2), (3,0), (4,2), (3,4), (0,7), (1,10), (0,11). For an understanding of the target order see Figure 3.4.

Order of tasks

Each participant was asked to perform 18 TECHNIQUES (4 TRACKING × 4 LOCKING × 2 MAPPING) × 12 TARGETS × 2 repetitions = 432 trials. To counter-balance TECHNIQUE we used a Latin Square for DT, F, GA, GR, HA, and HR techniques, where the order of the subtechniques for GA, GR, HA and HR were randomized for each user and target order was randomized for each user and each condition as well. Before a new TECHNIQUE was started, the user had time to do test trials to get used to the new TECHNIQUE. After these trials the user had to select each of the aforementioned twelve targets two times to complete a condition. Afterwards the test phase for the next condition started, if the participant was interested in testing it. Otherwise the next condition was started directly. Altogether the study took approximately 60 minutes to complete.

Dependent variables

The **Dependent Variables** were trial *Completion Time* [s] and user's *Success* [0,1], i.e., whether the correct target was selected or not. Once a technique was completed, the user was asked to fill out a questionnaire (see Appendix A) to evaluate it regarding her agreement on how regularly she had to change her seating position, how easy it felt to select targets, how confident she felt when selecting the targets, how much fatigue she felt during the technique, and how comfortable her head movement felt on a 7-point Likert scale (7 = totally agree). After the completion of all techniques the participant was asked to rank the techniques from highest (1) to lowest (6) by distinguishing the super-categories GA, GR, HA, and HR techniques and the baseline techniques DT and F.

Chapter 4

Evaluation

Data and tests used in the study

Our main interest in the study we conducted is the participants performance depending on the TECHNIQUE, therefore, we focus on this main effect and related interaction effects. We conducted a repeated-measures ANOVA on the *Time* data and on the *Success* data. Additionally we are interested in the participants opinion about the different TECHNIQUES, which we measured with the help of a questionnaire and Likert scales. The Likert scale data was compared using Friedman tests and the pairwise comparisons for the Likert scale data used the Bonferroni correction. The data used for the evaluation was taken solely from the logged result files as described in 3.1.3.

4.0.1 Completion Time

TRACKING had a significant effect on Completion Time

TRACKING had a significant main effect on ***Time*** ($F_{3,5113} = 134.83, p < .0001$). Tukey HSD post hoc pairwise comparisons were all significant. Participants were fastest with DT (1.76 s) followed by G (2.14 s), H (2.46 s) and F (3.3 s).

GM and GT are nearly as fast as DT

When combining TRACKING and LOCKING into one variable, we found that there is no significant main effect between DT (1.68 s) and GazeMove (GM) (1.81 s) and GazeTouch (GT) (1.84 s) on ***Completion Time*** in seconds (see Figure 4.1). All other conditions performed significantly

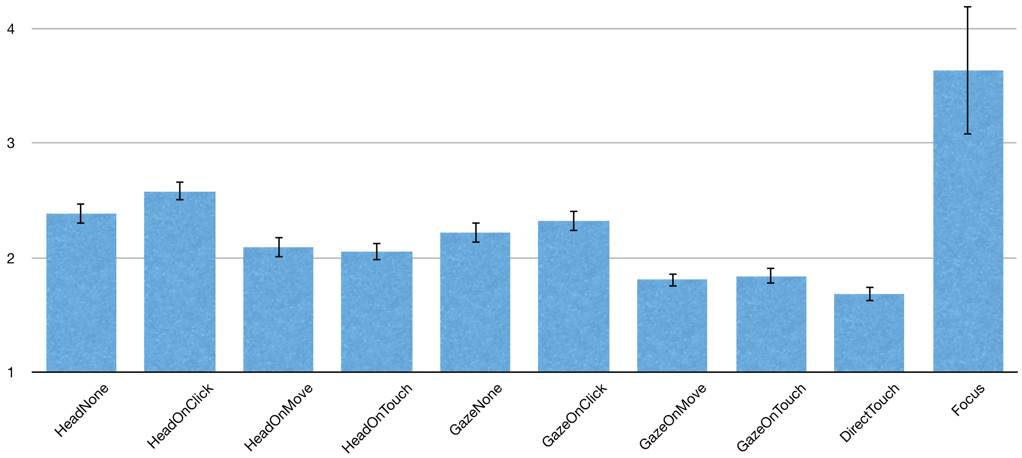

Figure 4.1: The graph shows the average Completion Times in seconds for the combined TRACKINGLOCKING methods. Whiskers denote the 95% Confidence Interval.

slower than these three conditions, with F (3.16 s) performing the slowest, as before.

LOCKING had a significant effect on Completion Time

The LOCKING method had a significant main effect on ***Time*** ($F_{3,4505} = 61.37, p < .0001$). Tukey HSD post hoc pairwise comparisons showed that onMove (2.09 s) and onTouch (2.11 s) LOCKING methods were significantly faster than no locking (2.44 s) and onClick (2.57 s).

MAPPING had a significant effect on Completion Time

MAPPING had a significant main effect on ***Time*** ($F_{1,4505} = 40.81, p < .0001$). The Student's t test post hoc pairwise comparisons revealed that an absolute mapping (2,2 s) was significantly faster than a relative mapping (2.4 s).

4.0.2 Success Rate

GM performed nearly as precise as DT

Other than ***Completion Time***, we were interested in the participants ***Success*** when selecting targets. The combined TRACKING and LOCKING variable revealed a significant main effect on ***Success*** ($F_{9,195} = 13.43, p < .0001$). Tukey HSD post hoc pairwise comparisons showed that DT (98.9%), GM (97.3%), HM (97.2%) and GT (96.3%) perform with a significantly higher success rate than HN (88.7%)

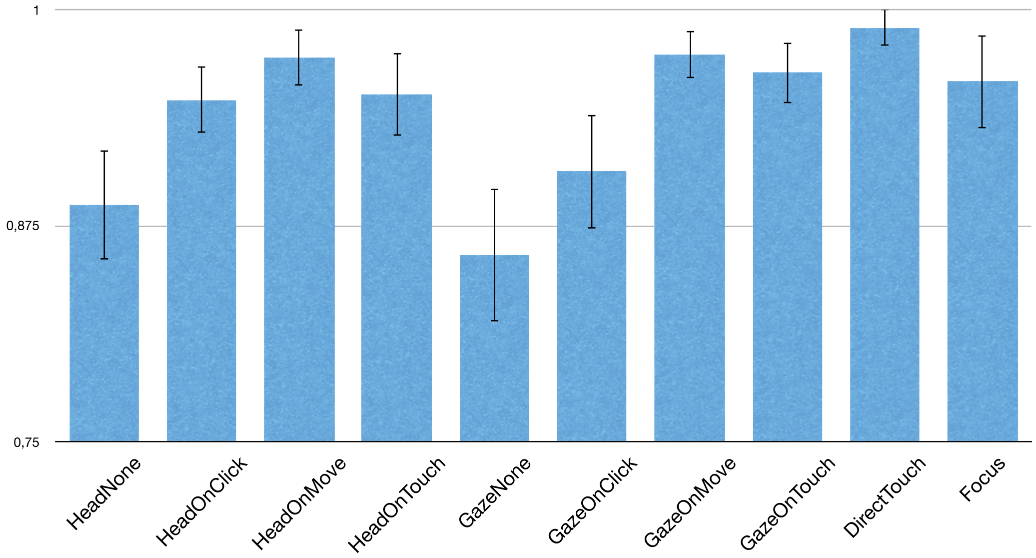

Figure 4.2: The graph shows the average ***Success Rate*** for TRACKINGLOCKING methods. Whiskers denote the 95% CI.

and GN (85.7%), with no significant difference to F(95.8%), HT (95.1%), HC (94.8%) and GC (90.6%) (see Figure 4.2).

4.0.3 Questionnaire

The questionnaire gave us insightful information about the users preference and ranking of the different conditions.

For the ranking we combined TRACKING and MAPPING, from now on referred to as TRACKINGMAPPING to reduce the number of conditions to be ranked to six different condition categories.

Ranking did not show preference of a certain category

TRACKINGMAPPING had a significant effect on the participants ***ranking*** ($\chi^2(5) = 14.190$, $p = .014$). However, due to the Bonferroni correction the post hoc pairwise comparisons did not show that participants significantly preferred one TRACKING×MAPPING over another one. The ranking can be seen in Figure 4.3.

TRACKING×LOCKING×MAPPING also had a significant ef-

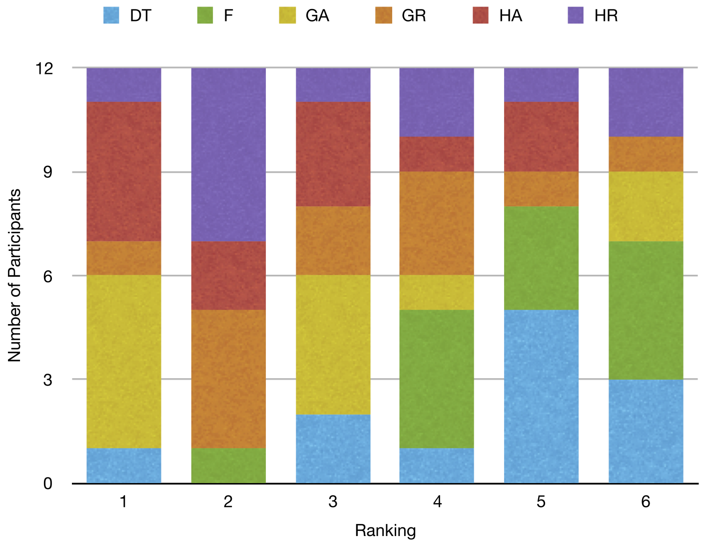

Figure 4.3: The graph shows the users ranking of the TRACKING×MAPPING methods. Most users ranked GA in the top 3 methods, while DT was mostly ranked in the lower 3 places.

DT was the worst condition regarding seating position

fect on the participants ***seating position*** ($\chi^2(17) = 77.309$, $p = .000$). The post hoc pairwise comparisons show that participants significantly preferred all other conditions over DT.

Ease of use

TRACKING×LOCKING×MAPPING had a significant effect on the ***ease of use*** ($\chi^2(17) = 78.334$, $p = .000$), too. The post hoc pairwise comparisons show that GAN, GRN and HRN significantly differ from DT, GAT, GAM, GRT, GRM, HAT, HAM and HRT. Participants did not think that conditions without a locking method were easy to use.

Confidence in selection

TRACKING×LOCKING×MAPPING had a significant effect on the ***confidence in selection*** ($\chi^2(17) = 119.259$, $p = .000$) as well. The post hoc pairwise comparisons show that GAN, GRN, HAN and HRN significantly differ from DT, F, GAT, GAM, GRT, GRM, HAT, HAM, HRT, HRC, and HRM. Participants felt less confident when using GAN, GRN, HAN and HRN, i.e., all methods without locking.

Fatigue

TRACKING×LOCKING×MAPPING did not have a significant effect on ***fatigue*** ($\chi^2(17) = 30.679$, $p = .022$).

Head movement

TRACKING×LOCKING×MAPPING also had a significant effect on participants ***head movement*** ($\chi^2(17) = 58.368$, $p = .000$). The post hoc pairwise comparisons show that HRN significantly differs from F and GRT, but nor significant difference was found regarding the other conditions. The data indicates that the head movement during HRN was perceived as less comfortable than in F and GRT.

4.0.4 Discussion

The goal of the study

We gained interesting insights from our results. Our goal was to find an alternative interaction method for large displays with the help of gaze-supported touch interaction for a use case such as in-car entertainment systems.

DT performed fast, but was disliked by participants – they preferred GM

When comparing the TRACKING methods regarding ***Completion Time***, DT performed very fast, which we expected, since it is easy to use and users already know this technique. This supports our hypotheses *H1*, but it also serves as evidence for *H2*. However, when combining TRACKING and LOCKING, we found that GM and GT did not perform significantly slower. Furthermore, participants preferred all other conditions to Direct Touch, because, as participants commented, they 'had to move a lot' and 'targets were out of reach'. During the user study we observed that participants started to overshoot the targets in the far right part of the PPI, because when they were told that the virtual trackpad would be moved by their head movement they thought that it would work faster, if they turned their head as fast and as far as they could, even though it was not necessary during the trials and the principal investigator explained it beforehand. This is an additional indicator for why there is evidence backing *H2*.

onClick performed slowest of the LOCKING methods

We found evidence for our third hypothesis *H3* when comparing LOCKING methods. The fastest methods were *onMove* and *onTouch*, but contrary to our fourth hypothesis *H4* it was not the LOCKING method *none* that had the longest completion time, but *onClick*. Even though this does not match our hypothesis, we expected this after conducting the user study, because participants mentioned that 'multiple clicks only allow a slower selection'. Some participants even started using a double click throughout the course of the study, because it felt faster using the condition this way.

Absolute mapping performed faster than relative mapping

Although we were not sure about which MAPPING would enable a faster completion time, we did think that they would significantly differ, as mentioned in *H5*. We did find a significant main effect of the MAPPING method on ***Completion Time*** and absolute mapping performed faster than relative mapping. Based on our participants comments we reckon that this is based on the users unfamiliarity with the trackpad, because they often had to reposition their fingers after placing them too close to the edges of the trackpad and thus not being able to reach the target.

GM is the most precise of our methods

When comparing the combined TRACKING and LOCKING methods regarding ***Success***, we found evidence for our sixth hypothesis (*H6*), but against the last hypothesis (*H7*). There was a significant difference in ***Success*** based on the chosen TRACKING and LOCKING, but not in the way we expected, since GM was more precise than HM, even if it was not significantly more precise.

Gaze-supported touch interaction is an alternative to DT

Our findings show that gaze-supported touch interaction is a valuable alternative in regards to ***Time*** and ***Success*** when interacting on a large display in a seated position. Even though our participants did not rank one condition significantly higher than the others, we would especially recommend using Gaze in combination with the LOCKING method *onMove*, because this allows for the virtual trackpad to be invisible and only once a finger touches the trackpad, it could become visible. This would work well for use cases like the aforementioned in-car entertainment systems, since it does not catch the user's – in this case the driver's – attention unnecessarily through constantly moving with the gaze.

Chapter 5

Summary and future work

This chapter concludes the thesis in two parts. First, the work is summarized and the contributions are pointed out. Then future work based on current results is described.

5.1 Summary and contributions

Motivation for the thesis

This thesis was based on the research interest of finding an easy-to-use interaction technique for ultrawide displays, with a focus on seated interaction based on the example of interaction techniques for in-car entertainment systems. State-of-the-art interaction techniques on ultrawide displays as well as the problems with these techniques were discussed. The most commonly found problem is the reachability issue that comes with large displays. This thesis looked into solving this problem with the help of gaze-supported interaction.

Prototype

Therefore, based on related work, this thesis focused on the implementation of a prototype for gaze- and head supported touch interaction for in-car entertainment systems. The prototype offers 16 different interaction techniques in total –two of them are baseline techniques which are currently available techniques for in-car entertainment systems.

Main contribution

Furthermore, as the main contribution of this thesis, a user study with twelve participants was conducted to determine whether one of the implemented techniques is an adequate alternative to the baseline conditions of *direct touch* and *focus* in terms of **completion time** and **success rate**.

GM is an adequate alternative to DT

The findings of this study were described in detail in Chapter 4, with the conclusion that gaze-supported touch interaction in combination with the locking method *onMove* for the virtual trackpad is recommended as an alternative to direct touch or focus based user input in the context of in-car entertainment systems.

5.2 Future work

The work of this thesis up to this point offers a good foundation for future work based on two contexts – Ultrawide or Large Displays and In-car Entertainment Systems.

Large Displays

Would the interaction technique be helpful on larger screens as well?

Considering the still increasing sizes of large displays, one task for future work is the comprehension of use cases on large displays and the mapping of the current prototype to the whole screen. This might then be used on white boards for example, when multiple people discuss something and one person uses the white board for her explanation. If the whole screen is going to be used, a slight improvement to tracking should be made as well. Although gaze tracking on the whole screen of the PPI was tested when the prototype of this work was implemented, there were still some problems with lighting. Some tests should be conducted to see what the optimal lighting would be.

In-car Entertainment Systems

When focusing on the use case for the prototype of the thesis, there is still more work to be done. Since an alternative to the currently used interaction techniques was found, it would be interesting to see if the recommended technique would work better for tasks other than target selection, e.g., maps interaction which is a frequent use case for in-car entertainment system interaction, as well. In this context, zoom, rotation and drag and drop interaction on a map would be of interest. These interaction techniques are already implemented in the current prototype, but have not yet been used. Furthermore, a closer look at the interaction areas would be interesting, since different areas of the screen can be reached from the different seats in the car. In this context the positioning issues would need to be investigated as well, since an eye-tracking camera positioned for use by the driver will not necessarily work for the front-seat passenger as well. Other than the positioning problem the lighting issues need to be considered. Since the system might also be used in the dark, if the in-car entertainment system is used at night, a constant light source in the car might irritate the driver. In future work we would need to think about an improvement of the tracking under those circumstances.

These tasks for future work show a research interest in this topic beyond the scope of this thesis.

Appendix A

QUESTIONNAIRE

Session ID: ______________________________________ **Participant:** ____________ **STUDY 1**

Incar Gaze tracking Study

Age: ________

Gender: ○ Female ○ Male

Do you own a car? What kind? ____

Which incar entertainment/ smartphone do you use? ____

Please rank all techniques regarding your preference for selecting targets.

“1” denotes highest preference, whereas “6” denotes lowest preference. Please map each rank only once.

	Rank
Direct Touch	
Focus	
Gaze Absolute	
Gaze Relative	
Head Absolute	
Head Relative	

Any comments?

Direct Touch	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Direct Touch?	***What was bad about Direct Touch?***						

Focus	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Focus?	***What was bad about Focus?***						

Gaze Absolute None	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Gaze Absolute None?	***What was bad about Gaze Absolute None?***						

Gaze Absolute Touch	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○

What was good about Gaze Absolute Touch?	***What was bad about Gaze Absolute Touch?***

Gaze Absolute Click	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○

What was good about Gaze Absolute Click?	***What was bad about Gaze Absolute Click?***

Gaze Absolute Move	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○

What was good about Gaze Absolute Move?	***What was bad about Gaze Absolute Move?***

Gaze Relative None	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○

What was good about Gaze Relative None?

What was bad about Gaze Relative None?

Gaze Relative Touch	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○

What was good about Gaze Relative Touch?

What was bad about Gaze Relative Touch?

Gaze Relative Click	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○

What was good about Gaze Relative Click?

What was bad about Gaze Relative Click?

Gaze Relative Move	**totally disagree**	**disagree**	**slightly disagree**	**neither**	**slightly agree**	**agree**	**totally agree**
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Gaze Relative Move?	***What was bad about Gaze Relative Move?***						

Head Absolute None	**totally disagree**	**disagree**	**slightly disagree**	**neither**	**slightly agree**	**agree**	**totally agree**
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Head Absolute None?	***What was bad about Head Absolute None?***						

Head Absolute Touch	**totally disagree**	**disagree**	**slightly disagree**	**neither**	**slightly agree**	**agree**	**totally agree**
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Head Absolute Touch?	***What was bad about Head Absolute Touch?***						

Head Absolute Click	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Head Absolute Click?	***What was bad about Head Absolute Click?***						

Head Absolute Move	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Head Absolute Move?	***What was bad about Head Absolute Move?***						

Head Relative None	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Head Relative None?	***What was bad about Head Relative None?***						

Head Relative Touch	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Head Relative Touch?	***What was bad about Head Relative Touch?***						

Head Relative Click	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Head Relative Click?	***What was bad about Head Relative Click?***						

Head Relative Move	totally disagree	disagree	slightly disagree	neither	slightly agree	agree	totally agree
1. I had to change my seating position regularly.	○	○	○	○	○	○	○
2. Selecting targets felt easy.	○	○	○	○	○	○	○
3. I felt confident selecting the targets.	○	○	○	○	○	○	○
4. After some time, I felt fatigue.	○	○	○	○	○	○	○
5. The head movement felt comfortable.	○	○	○	○	○	○	○
What was good about Head Relative Move?	***What was bad about Head Relative Move?***						

Bibliography

Carmelo Ardito, Paolo Buono, Maria Francesca Costabile, and Giuseppe Desolda. Interaction with large displays: A survey. *ACM Computing Surveys (CSUR)*, 47(3):46, 2015.

Xiaojun Bi, Yuanchun Shi, and Xiaojie Chen. upen: A smart pen-liked device for facilitating interaction on large displays. In *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'06)*, pages 7–pp. IEEE, 2006.

Christian Cherek, Anke Brocker, Simon Voelker, and Jan Borchers. Tangible awareness: How tangibles on tabletops influence awareness of each other's actions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 298. ACM, 2018.

Christian Cherek, David Asselborn, Simon Voelker, and Jan Borchers. Off-surface tangibles: Exploring the design space of midair tangible interaction. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, page LBW2115. ACM, 2019.

Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence*, 32 (3):478–500, 2009.

Ricardo Jota, João M Pereira, and Joaquim A Jorge. A comparative study of interaction metaphors for large-scale displays. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 4135–4140. ACM, 2009.

Mohamed Khamis, Florian Alt, and Andreas Bulling. The past, present, and future of gaze-enabled handheld mobile devices: Survey and lessons learned. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, page 38. ACM, 2018a.

Mohamed Khamis, Anita Baier, Niels Henze, Florian Alt, and Andreas Bulling. Understanding face and eye visibility in front-facing cameras of smartphones used in the wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 280. ACM, 2018b.

Rick Kjeldsen. Head gestures for computer control. In *Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 61–67. IEEE, 2001.

Aurélien Lucchi, Patrick Jermann, Guillaume Zufferey, and Pierre Dillenbourg. An empirical evaluation of touch and tangible interfaces for tabletop displays. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pages 177–184. ACM, 2010.

Carsten Magerkurth and Peter Tandler. Interactive walls and handheld devices-applications for a smart environment. In *Collaboration with interactive walls and tables, workshop at ubicomp*, volume 2, page 78. Citeseer, 2002.

Alexander Mariakakis, Mayank Goel, Md Tanvir Islam Aumi, Shwetak N Patel, and Jacob O Wobbrock. Switchback: Using focus and saccade tracking to guide users' attention for mobile task resumption. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2953–2962. ACM, 2015.

Jörg Müller, Robert Walter, Gilles Bailly, Michael Nischt, and Florian Alt. Looking glass: a field study on noticing interactivity of a shop window. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 297–306. ACM, 2012.

Takashi Nagamatsu, Michiya Yamamoto, and Hiroshi Sato. Mobigaze: Development of a gaze interface for handheld mobile devices. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 3349–3354. ACM, 2010.

Ken Pfeuffer and Hans Gellersen. Gaze and touch interaction on tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 301–311. ACM, 2016.

Wolfgang Reitberger, Alexander Meschtscherjakov, Thomas Mirlacher, Thomas Scherndl, Hermann Huber, and Manfred Tscheligi. A persuasive interactive mannequin for shop windows. In *Proceedings of the 4th international Conference on Persuasive Technology*, page 4. ACM, 2009.

Sophie Stellmach and Raimund Dachselt. Look & touch: gaze-supported target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2981–2990. ACM, 2012.

Sophie Stellmach and Raimund Dachselt. Still looking: investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 285–294. ACM, 2013.

Simon Voelker, Christian Cherek, Jan Thar, Thorsten Karrer, Christian Thoresen, Kjell Ivar Øvergård, and Jan Borchers. Percs: Persistently trackable tangibles on capacitive multi-touch displays. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 351–356. ACM, 2015.

Andrew D Wilson and Hrvoje Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 273–282. ACM, 2010.

Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (magic) pointing. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 246–253. ACM, 1999.

Index

ANOVA, 29–31
apparatus, 24

Completion Time, 29–30
conclusion, 34

discussion, 33–34

evaluation, 29–34

future work, 36–37

gaze interaction, 5–9

hypotheses, 23

Implementation, 16–23
iOS application, 17–18

large displays, 9–13

macOS application, 18–23

Questionnaire, 31–33

study design, 23–27
Success Rate, 30–31

Typeset April 28, 2020