Using Facial Tracking for Expressive Mobile Device Interactions

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Sebastian Hueber, M.Sc.

aus Aachen

Berichter: Prof. Dr. Jan Borchers
Prof. Dr. Enrico Rukzio

Tag der mündlichen Prüfung: 30. Oktober 2024

Eidesstattliche Erklärung

Sebastian Hueber erklärt hiermit, dass diese Dissertation und die darin dargelegten Inhalte die eigenen sind und selbstständig, als Ergebnis der eigenen originären Forschung, generiert wurden.

Hiermit erkläre ich an Eides statt

1. Diese Arbeit wurde vollständig oder größtenteils in der Phase als Doktorand dieser Fakultät und Universität angefertigt;
2. Sofern irgendein Bestandteil dieser Dissertation zuvor für einen akademischen Abschluss oder eine andere Qualifikation an dieser oder einer anderen Institution verwendet wurde, wurde dies klar angezeigt;
3. Wenn immer andere eigene- oder Veröffentlichungen Dritter herangezogen wurden, wurden diese klar benannt;
4. Wenn aus anderen eigenen- oder Veröffentlichungen Dritter zitiert wurde, wurde stets die Quelle hierfür angegeben. Diese Dissertation ist vollständig meine eigene Arbeit, mit der Ausnahme solcher Zitate;
5. Alle wesentlichen Quellen von Unterstützung wurden benannt;
6. Wenn immer ein Teil dieser Dissertation auf der Zusammenarbeit mit anderen basiert, wurde von mir klar gekennzeichnet, was von anderen und was von mir selbst erarbeitet wurde;
7. Ein Teil oder Teile dieser Arbeit wurden zuvor veröffentlicht und zwar in:

   S. Hueber, J. Wilhelm, R. Schäfer, S. Voelker, J. Borchers. User-Aware Rendering: Merging the Strengths of Device- and User-Perspective Rendering in Handheld AR. In PACM MHCI ’23.

   S. Hueber, C. Cherek, P. Wacker, J. Borchers, S. Voelker. Headbang: Using Head Gestures to Trigger Discrete Actions on Mobile Devices. In ACM MHCI ’20.

   S. Voelker, **S. Hueber**, C. Holz, C. Remy, N. Marquardt. GazeConduits: Calibration-Free Cross-Device Collaboration through Gaze and Touch. In ACM CHI ’20.

   S. Voelker, **S. Hueber**, C. Corsten, C. Remy. HeadReach: Using Head Tracking to Increase Reachability on Mobile Touch Devices. In ACM CHI ’20.

   S. Hueber, E. Jang, J. Borchers: Attentive Notifications: Minimizing Distractions of Mobile Notifications through Gaze Tracking. In ACM MHCI ’23.

Aachen, 6. November 2024
Sebastian Hueber

Contents

Abstract **xv**

Überblick **xvii**

Acknowledgments **xix**

Conventions **xxi**

1 Introduction **1**

1.1 Research Questions 10

1.2 Hypotheses 10

1.3 Outline 12

2 Foundations **13**

2.1 Proxemic Interactions 14

2.2 Gaze Analysis 17

2.3 Characteristics of Facial Tracking 22

2.4 Immersive Computer Graphics 27

2.5 Adjacent Research Domains 31
2.5.1 Intelligent User Interfaces 31
2.5.2 Accessibility and Medical Applications 33
2.6 Facial Tracking on Mobile Devices 34
2.7 Evaluating Facial Tracking in ARKit for Head and Gaze Interactions 36
2.7.1 Determining the Tracked Area 37
2.7.2 Quantifying Visual Head Tracking 37
Apparatus and Task 39
Results 40
2.7.3 Quantifying Visual Gaze Tracking 41
Apparatus and Task 41
Results 42
2.7.4 Summary 42

3 Allowing Quick Menu Actions with Head Gestures 45

3.1 Motivation 46
3.2 Related Work 48
3.2.1 Touch Gestures 49
3.2.2 Tilting Interfaces 50
3.2.3 Head Tracking 51
3.3 Headbang Interaction Technique 53
3.4 Study 1: Investigating Tracking Robustness 54

3.4.1 Apparatus and Task 55
3.4.2 Variables 55
3.4.3 Results 56
3.4.4 Discussion 58
3.5 Study 2: Using Headbang to Trigger Actions in Menus 59
3.5.1 Apparatus and Task 60
3.5.2 Variables 61
3.5.3 Results 62
3.5.4 Discussion 64
3.6 Use Cases 66
3.7 Future Work 68
3.8 Conclusion 69

4 Solving Reachability Issues on Large Phones with Head Control 71

4.1 Motivation 72
4.2 Related Work 74
4.2.1 Reachability Techniques 74
Screen Transformation Techniques 75
Proxy Region Techniques 76
Cursor Techniques 77
4.2.2 Head Input on Mobile Devices 78
4.3 Head Reaching Techniques 79

4.3.1 Pure Head Selection 80
4.3.2 Head + Touch Selection 82
4.3.3 Head Area + Touch Selection 84
4.4 Study 1: Standing 85
4.4.1 Apparatus and Techniques 85
4.4.2 Task and Targets 87
4.4.3 Variables 87
4.4.4 Results 88
4.5 Study 2: Walking 92
4.5.1 Results 93
4.6 Discussion 97
4.7 Future Work 99
4.8 Conclusion 100

5 Investigating Gaze Support in Cross-Device Interactions 103

5.1 Motivation 104
5.2 Related Work 106
5.2.1 Cross-Device Systems and Tracking 106
5.2.2 Gaze Interactions 108
5.3 Gaze Tracking in an Ad-hoc Setting 109
5.4 GazeConduits 111
5.5 Study 1: Evaluating Gaze-to-Tablet Tracking 114

5.5.1 Apparatus and Task 115
5.5.2 Variables 115
5.5.3 Results 116
5.5.4 Discussion 116
5.6 Study 2: Evaluating Gaze-to-Person Tracking 118
5.6.1 Apparatus and Task 118
5.6.2 Variables 118
5.6.3 Results 119
5.6.4 Discussion 119
5.7 Interaction Scenarios 119
5.7.1 Interactions through GazeConduits' User Awareness 120
5.7.2 Interactions through Gaze-at-Device Tracking 120
5.7.3 Interactions through Gaze-at-Users Detection 122
5.8 Limitations and Future Work 122
5.9 Conclusion 124

6 Enhancing Handheld Augmented Reality with Face Tracking 125

6.1 Motivation 126
6.2 Related Work 129
6.2.1 User-perspective Rendering 130
6.2.2 Depth Perception 132
6.2.3 Impact of FOV 133

6.3 User-Aware AR 134
6.3.1 Concept 134
6.3.2 Prototype System 135
Camera Transform 137
Camera Projection 137
Video Feed 139
6.3.3 Scaling Factors 140
6.4 Study 1: Depth Perception 142
6.4.1 Apparatus and Task 142
6.4.2 Variables 144
6.4.3 Results 145
6.4.4 Discussion 147
6.5 Study 2: Searching and Selecting Objects 148
6.5.1 Apparatus and Task 148
6.5.2 Variables 150
6.5.3 Results 151
6.5.4 Discussion 153
6.6 Conclusion 155
6.7 Future Work 156

7 Optimizing Distraction on Mobile Devices with Gaze Analysis 159

7.1 Motivation 160

7.2 Related Work 162

7.2.1 Distraction Caused by Notifications 162

7.2.2 Perception of Notifications 163

7.2.3 Awareness of User's Gazing 164

7.3 Designing Attentive Notifications 164

7.3.1 Exploration of Visual Factors 165

Apparatus and Task 165

Results 167

7.3.2 Controlling Notification Placement 169

Gaze-Implicit 169

Gaze-Explicit 169

Touch-Attentive 170

7.4 Evaluation 171

7.4.1 Apparatus and Task 171

7.4.2 Variables 172

7.4.3 Results 172

7.5 Discussion 175

7.6 Conclusion 177

7.7 Future Work 178

8 Summary and Future Perspectives **181**

8.1 Summary and Conclusions 182

8.1.1 Contributions and Benefits 185

8.1.2 Reflection 186

8.2 Future Perspectives 188

Bibliography **191**

Index **223**

Own Publications **227**

List of Figures and Tables

1.1 *How the computer sees us* by O'Sullivan and Igoe [2004] 2

1.2 *Put that There* by Bolt [1980] 3

1.3 Taxonomy of interaction techniques presented in this thesis 11

2.1 *Related Work:* Proxemic interactions in today's interfaces. 14

2.2 *Related Work: Lean and Zoom* by Harrison and Dey [2008]. 16

2.3 *Related Work:* Images from the dataset of Khamis et al. [2018]. 25

2.4 *Related Work:* The *Varrier* system by Kooima [2009]. 29

2.5 *Related Work: pCubee* by Stavness et al. [2010]. 30

2.6 *Preliminary Studies:* Facial tracking in ARKit 36

2.7 *Preliminary Studies:* Obtaining a rotation from head orientation 38

2.8 *Preliminary Studies:* Mapping of head tilt to rotary angles 40

2.9 *Preliminary Studies:* Uncalibrated gaze tracking accuracy on a table . 42

3.1 *Headbang:* Image sequence of the interaction technique 47

3.2 *Headbang:* Success data from Study 1 56

3.3 *Headbang:* Time data from Study 1 57

3.4 *Headbang:* Pie menus used in Study 2 60

3.5 *Headbang:* Time data from Study 2 62

3.6 *Headbang:* Success data from Study 2 63

3.7 *Headbang:* Using Headbang for text editing 67

3.8 *Headbang:* Using Headbang as accessibility technique 68

4.1 *HeadReach:* Image sequence of the Head + Touch interaction technique 83

4.2 *HeadReach:* Visualization of techniques in the study app 86

4.3 *HeadReach:* Time data from Study 1 89

4.4 *HeadReach:* Tabular time data from Study 1 90

4.5 *HeadReach:* Success data from Study 1 91

4.6 *HeadReach:* Likert scale data from both studies 92

4.7 *HeadReach:* Participant rankings from both studies 93

4.8 *HeadReach:* The obstacle course used in Study 2 94

4.9 *HeadReach:* Time data from Study 2 95

4.10 *HeadReach:* Tabular time data from Study 2 95

4.11 *HeadReach:* Success data from Study 2 97

5.1 *GazeConduits:* An ad-hoc collaborative environment 105

5.2 *GazeConduits:* Smartphone stands 110

5.3 *GazeConduits:* Adding new devices to the ad-hoc setup 113

5.4 *GazeConduits:* Time data from Study 1 117

5.5 *GazeConduits:* An exemplary usage scenario 121

6.1 *User-Aware Rendering:* Comparison of different rendering techniques 127

6.2 *User-Aware Rendering:* Shapes and sizes of camera frustums 136

6.3 *User-Aware Rendering:* Parameters in the calculation of frustums . . . 138

6.4 *User-Aware Rendering:* Apparatus and task of Study 1 143

6.5 *User-Aware Rendering:* Depth scores from Study 1 146

6.6 *User-Aware Rendering:* Task of Study 2 149

6.7 *User-Aware Rendering:* Time data from Study 2 152

6.8 *User-Aware Rendering:* Movement data from Study 2 152

7.1 *Attentive Notifications:* Image sequence of gaze-explicit notifications . 161

7.2 *Attentive Notifications:* Visual styles tested in the preliminary study . 166

7.3 *Attentive Notifications:* Likert scale data from the preliminary study . 168

7.4 *Attentive Notifications:* Image sequence of gaze-implicit notifications . 170

7.5 *Attentive Notifications:* Image sequence of notification dismissal . . . 170

7.6 *Attentive Notifications:* Task of the main study 173

7.7 *Attentive Notifications:* Notification interaction in the main study . . . 173

7.8 *Attentive Notifications:* Likert scale data from the main study 174

7.9 *Attentive Notifications:* Participant rankings from the main study . . . 175

Abstract

For four decades, user interfaces have been mainly designed for pointing input, either with a mouse or a touchscreen. Pointing input abstracts from the human to a location on the screen. Especially visual cues that make human-to-human communication effective—like eye contact or head nodding—are ignored in this abstraction. Mobile devices, in particular, suffer from the limits of pointing input, as users cannot comfortably reach everything on the screen when using the device one-handedly. We present implicit and explicit usages of facial tracking to make mobile interactions more expressive and ergonomic.

We show the advantages of eye tracking using three interaction techniques. First, our *Attentive Notifications* remove occlusion issues and accidental activations in mobile interfaces. They determine a suitable screen edge for displaying notifications by blocking the area around the user's gaze at the moment of notification delivery. Second, we show that eye tracking can enhance the perception of content in augmented reality with our *User-Aware Rendering*. This technique provides enhanced depth perception with good performance in scene exploration. Third, interfaces can exploit that gaze input can reach anything nearby. Our *GazeConduits* concept fosters collaboration in ad-hoc multi-device environments. This enables users to interact with devices or meeting collaborators by looking at them.

However, eye tracking often comes with accuracy issues, especially when people are moving, and suffers from the Midas touch problem. To overcome these challenges, two of our interaction techniques use head tracking instead. We present a *Head + Touch* controlled cursor that increases the thumb's reach during one-handed smartphone use. This significantly reduces the overhead of touch-based reachability techniques to under 100 ms. With our *Headbang* technique, menu selections are also faster than with touch input.

Überblick

Seit über 40 Jahren fokussieren sich Benutzeroberflächen auf Zeigereingaben, bspw. mit Maus oder Touchscreen. Aus Sicht des Computers reduzieren Zeigegeräte folglich den Menschen zu einem Bildschirmpixel. Während die zwischenmenschliche Kommunikation auf die Interpretation von Augenkontakt und Gestik aufbaut, ignorieren Zeigegeräte billigend den sie bedienenden Menschen. Die Grenzen der Zeigereingabe werden bei der einhändigen Verwendung von Mobilgeräten besonders offensichtlich, da Nutzer ihren Daumen nicht frei über den ganzen Bildschirm bewegen können. In dieser Arbeit stellen wir Interaktionstechniken vor, die durch ihre explizite oder auch implizite Verwendung von Gesichtstracking dem Nutzer mehr Ausdrucksstärke verleihen und somit Effizienz und Ergonomie verbessern.

Drei dieser Interaktionstechniken bedienen sich dem Eye-Tracking. Unsere *Attentive Notifications* verringern Verdeckungsprobleme und versehentliche Aktivierungen von Mitteilungsbannern auf mobilen Betriebssystemen. Diese neue Art von Mitteilungen erscheint am jeweiligen Bildschirmrand der am weitesten von dem Punkt entfernt ist, auf den der Nutzer zum Zeitpunkt der Benachrichtigung schaut. Wir zeigen außerdem wie Eye-Tracking die Wahrnehmen von Inhalten in Augmented Reality verbessern kann. Unser *User-Aware Rendering* hilft Nutzern bei der Tiefenwahrnehmung des virtuellen Inhalts und Übersichtlichkeit bei der Exploration eben jenes. Weil Blicke eines Nutzers alles in der Umgebung erreichen können, eignet sich die Blickerfassung auch zur Stärkung der Zusammenarbeit in ad-hoc Mehrgeräteumgebungen. Beispielsweise können Nutzer in unserem Konzept *GazeConduits* durch ihren Blick Meetingteilnehmer oder anderen Geräte spezifizieren, mit denen sie interagieren möchten.

Schwächen von Eye-Tracking sind Genauigkeitsprobleme bei sich bewegenden Nutzern und das Midas-Touch-Problem. Zwei unserer Interaktionstechniken überwinden diese Schwächen durch die Verwendung von Head-Tracking. Mit unserer *Head + Touch* Zeigersteuerung erweitern wir die Reichweite des Daumens bei einhändiger Smartphone-Nutzung mit einem Overhead von unter 100ms. Mit unserer *Headbang*-Technik erfolgen auch Menüauswahlen schneller als mit Touch.

Acknowledgments

This work was only feasible with the help and support of many.

First, I want to thank my first advisor, Prof. Jan Borchers, for giving me the opportunity to do a PhD in Human-Computer Interaction. He provided a work environment in which it was possible for me to find and follow my own research interests.

Second, I want to thank Prof. Enrico Rukzio, who was kind enough to be the second advisor. Thank you for spending the time and effort to co-supervise this thesis.

Special thanks to Simon Voelker, who provided me with important guidance in writing my first publications and always gave me immensely valuable feedback on my research ideas.

Many thanks go to my colleagues at i10, who created a joyful working environment where we had many good conversations beyond work topics. I especially want to thank Adrian Wagner for always having good advice when I got stuck coding computer graphics stuff. Having you as an office neighbor and implementing so many software projects with you was always fun.

Thanks to all my thesis students and student assistants who realized my partially crazy ideas and requirements. I enjoyed working with all of you.

Finally, I want to thank my friends and family for their constant support, enthusiasm, or whatever was needed at the moment.

—Sebastian

Conventions

Throughout this thesis we use the following conventions:

- The thesis is written in American English.
- The first person is written in plural form.
- Unidentified third persons are described in female form.

Summaries and short excursuses are set off in colored boxes.

EXCURSUS:
Excursuses are set off in orange boxes.

SUMMARY:
Summaries of own publications are set off in blue boxes.

Where appropriate, paragraphs are summarized by one or two sentences that are positioned at the margin of the page.

This is a summary of a paragraph.

We use the term *facial tracking* to summarize the visual tracking of eyes and head.

Chapter 1

Introduction

"When you walk up to your computer, does the screen saver stop and the working windows reveal themselves? Does it even know if you are there? How hard would it be to change this?"

—Bill Buxton [1997]

Current technology especially receives inputs using our fingers. Our eyes and ears perceive the output.

The features of today's user interfaces are derived from the inputs and outputs the current computing devices offer: Typically, this is a pointing device, like a mouse or touch digitizer, supplemented by a keyboard. O'Sullivan and Igoe [2004] summarized these common modalities of HCI in their illustration titled *''how the computer sees us"*. In Figure 1.1, you see a single finger with an attached eye and ears. The finger creates inputs by pointing, touching, or clicking the mouse and keyboard. The eye and ears receive the outputs made by the computer. Despite voice input, the illustration does not contain a mouth. The omission of a mouth probably emphasizes that speech interfaces are often perceived as awkward for users and passers-by, primarily when used in public spaces [Baier and Burmester, 2019].

Reducing the human to a 2D pointing action discards information.

Even though this illustration might look bizarre initially, one must admit it caricatures the typical communication between today's computing interfaces and their users well. Like a machine, the human is reduced to a closed system that receives inputs and performs outputs. Reductions like

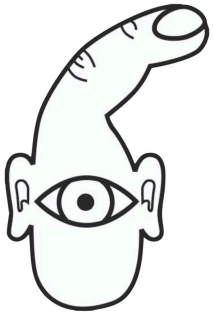


Figure 1.1: *How the computer sees us*. This illustration characterizes the human user by the input and output capabilities detected by a computer. Image taken from O'Sullivan and Igoe [2004].

these are common in many models across the field of HCI. For instance, in the famous *"human information processor"* by Card [1983], the task of the human is to use its visual inputs to make physical pointing actions. Reducing a human to a pointing location, however, discards multiple pieces of information: The actual pointing does not take place in 2D; it includes the bodily movement of different joints in our 3D world. The computer does not capture the parts of the interaction that do not occur on an interactive surface. Other aspects of communication are also left behind during this reduction: For instance, the emotional state indicated through facial expressions or other ongoing tasks, e.g., whether the person is on the go or resting at home.

Bodily movements, emotional states, and context are ignored by current systems.

In contrast, human-to-human communication functions differently. When humans communicate with each other, they benefit from the increased communication bandwidth of multiple modalities. For instance, interpreting audio-visual signals is essential for communication [Sebe, 2009]. Consider sitting in a room with three windows, and your friend asks you to open *the* window. At first, this request might sound ambiguous. Yet, the message's intention becomes clear if this friend is concurrently pointing at one specific window. Similarly, when many people stand together, we can reliably address a question to a specific person simply by looking at them.

The combination of audio-visual signals makes communication between humans effective.

Figure 1.2: *Put that There* was the first system to support multimodal input. It allowed users to control a cursor by pointing at a location on the screen. Voice commands were used to select elements and activate other functionalities. Image taken from Bolt [1980].

Excursus: *Multimodality*

MULTIMODALITY:
Each of our senses—vision, audio, haptics, smell, and taste—provides a unique mode of communication. Multimodal systems receive input and provide output via more than one of these modes. Bateman et al. [2017] explains that "multimodality is a way of characterizing communication situations [...] which rely upon combinations of different 'forms' of communication to be effective." Multimodal systems supposedly provide richer and more natural interactions. Moreover, the overall bandwidth increases as each mode has its own communication bandwidth.

In HCI, the first multimodal interactions were presented as early as 1980.

The effectiveness of multimodality in human-to-human communication inspired seminal work in HCI as early as the middle of the 20th century to add more senses to the computer. The first working system that presented multimodal inputs was *"Put that There"* by Bolt [1980]. It allowed users to control a cursor on a wall-sized screen with their

finger and initiate actions by speaking voice commands. At this time, however, voice commands only supported a fixed set of phrases. Also, while point-and-speak is technically multimodal, it makes only limited use of the new input modes and instead borrows from the mouse metaphor [Oviatt, 1999]. Therefore, it would be exaggerated to call the interaction of *Put that There* natural.

For interaction designers, it is important to consider possible input and output modalities, as well as their communication bandwidths.

Since *Put that There*, HCI researchers strived to enrich interactions with user interfaces. This requires the computer to understand its user's multimodal cues better, just as a human can read another human. Thus, Wendy Ju [2015] stated the importance of an interactive system to understand its user across all modalities it can perceive and deduce inputs. She, therefore, recommends interaction designers ask themselves "How do you do? How do you feel? How do you know?" [Verplank, 2009] both from the user's and system's point of view. The quality of answers to these questions is ultimately limited to the overall communication bandwidth, i.e., the amount of data that can be communicated via the supported modalities.

Excursus: *Communication Bandwidth*

COMMUNICATION BANDWIDTH:
Information theorists quantify data-transfer rates not only for digital media but also for information processed inside humans. The amount of data humans can perceive via the communication channels of the different modalities varies notably. Zimmermann [1989] calculated communication bandwidths of 10^7 bit/s for vision, 10^6 bit/s for haptics, 10^5 bit/s for for audio and smell, and 10^3 bit/s for taste. In other words, our eyes communicate with our brain at Ethernet speed.
Further proof that humans are optimized for visual processing can be found in our nervous system: Out of all modalities, visual information activates the most areas in our cortex.

Remote human-to-human communication became richer through higher bandwidth.

The importance of high communication bandwidths becomes apparent in remote human-to-human communication: Consider the differences between a letter, a phone call, and a video call. The simple exclamation "Good for you!" will switch between a nice or sarcastic and rude meaning

based on intonation. In letters, the intonation of the message is entirely in the reader's disclosure. Yet, this ambiguity is removed in phone calls, and hearing the other person simultaneously helps interpret their emotional state. Lastly, communication is also possible nonverbally in video calls: People can read each other's moods by analyzing facial expressions, body language, mimics, and gestures.

In history, novel interfaces of higher dimensionality were enabled through new technologies.

Jakob Nielsen [1994] also knew the impact of communication bandwidth on a computing system. The *dimensionality* of user interfaces he coined as a term in his book "Usability Engineering" explains how, in history, the usability of computers was enhanced by using new input methods. For reference, early computers provided a *one-dimensional* interaction: Line-oriented interfaces process one command at a time, display its outcome, and wait for the following user input. The succeeding full-screen textual interfaces were *two-dimensional*: Users could enter text in multiple lines instead of just one, which, e.g., made form-filling dialogues easier. However, the focus of textual interfaces remained on the commands. Graphical user interfaces (GUIs) with overlapping windows, denoted as *two-and-a-half-dimensional*, shifted from this function-oriented to an object-oriented paradigm, in which one window corresponds to one file. This jump from text-based to pointing-based interfaces led to the widespread adoption of computers, as 2.5D interfaces have better usability characteristics. This was enabled by increasing the communication bandwidth of both input and output. On the input side, the mouse made it easier for humans to communicate their objects of interest to the computer. On the output side, the raster graphics display allowed for a flexible presentation of the new data-centered windows.

The mouse allowed 2.5D interfaces with better usability characteristics than textual interfaces.

In recent years, we are approaching the area of Nielsen's 3D interfaces.

At the time of writing this book, Nielsen envisioned the future *three-dimensional* interfaces to include more media types, be highly portable and personal, and achieve tight connectivity through new technologies. While this definition remained vague, we must acknowledge that smartphones and wearable devices provide a variety of sensors and cameras that drive new personal communication methods and experiences, such as augmented reality. However, like the mouse was vital for 2.5D interfaces, new input

methods will also be required to achieve 3D interfaces. Yet, what could this new type of input be?

Adding a visual input sense to the computer could be the next step toward 3D interfaces.

Considering the importance of seeing and reading our conversation partner in human-to-human communication, adding the visual sense to a computer seems to be a promising candidate for three-dimensional interfaces: Making the computer see its user adds a lot of communication bandwidth for new interaction techniques. Visual tracking offers many capabilities: It can identify what a person is doing, detect gestures, and even determine what the user is looking at. Since visual information processing is so deeply wired into our brains, gaze tracking can be a reliable indicator of what somebody focuses on. This makes tracking the user's gaze, in particular, an important area of research.

Gaze is an indicator of attention.

In psychology, gaze was studied as early as 1879, beginning with analyzing eye movements and saccades [Duchowski, 2002]. It quickly became apparent that visual information is crucial to human information processing. This makes gaze an essential indicator of what the user is paying attention to in her environment [Kahneman, 1973]. In a broader sense, the head posture of a user provides a piece of similar information, as our heads follow our gaze to achieve comfortable eye positions [Stiefelhagen et al., 1999]. At the same time, head gestures also add to communication as their meaning follows explicit social conventions. According to Kettner and Carpendale [2013], infants learn to nod and shake their heads before they are 18 months old.

Our head orientation follows our gaze.

Head gestures also convey cultural meanings.

Gazing can be used to infer areas of interest on screens.

HCI researchers also explored the use of gaze tracking for novel interaction techniques. The *World of Windows* by Bolt [1981] used gaze as an indicator of the user's attention: If users gazed at a window over a longer time, it would increase in size. In contrast, not looking at a window for some time would make it disappear.

Controlling UI elements with gaze

One early work that explored gazing in WIMP interfaces comes from Jacob [1990]. He discovered that the naive approach of activating on-screen elements simply by looking at them does not work out. The problem is that neither the system nor the users themselves can tell which gazes

should be used as input and which not. This is what Jacob called the *Midas touch problem*: If every gaze is used as direct input, one cannot look anywhere without continuously triggering commands. Jacob also considered using blinking to make selection more explicit and ran into a different issue, as blinking is a subconscious process to keep the eyes moist. If a user wants to control her blinking, this results in a conscious intervention, making the interaction less natural. Instead, he proposed resting the gaze on a specific item or pressing an explicit button as an explicit selection mechanism. He also presented appropriate gaze-based interaction techniques for UI elements of WIMP interfaces: selecting and moving objects, activating and selecting elements in popup menus, and scrolling lists by gazing at arrows beneath the list.

The Midas touch problem arises when using gaze as direct input: One cannot look anywhere without triggering commands.

One strength of gaze tracking in interfaces is that it allows for significantly faster interaction times. For instance, in a study by Sibert and Jacob [2000], participants selected both synthetic targets and specific letters in texts faster with their gaze than with a mouse. Gaze interactions also provide advantages beyond the confined space of the display: In a VR user study by Tanriverdi and Jacob [2000], participants could select items significantly faster by gaze than by pointing. This effect was even more prominent with distant virtual content.

Advantages of gaze interactions are speed and reachability of distant objects.

Since this early work, various interaction techniques using gaze have been explored, more recently in combination with touch. From an evolutionary point of view, using hands and fingers is the most natural way for a human to interact with an object [Kivell, 2015]. Since smartphones, everybody knows how widespread and intuitive touch input is. In combination with gaze, touching a handheld device allows easy control to make gaze selections explicit. For instance, Stellmach and Dachselt [2012] projected the local touch inputs onto a distant screen to perform actions. Their evaluation shows that touch is an appropriate input for explicitly confirming intended actions that apply to the gazed-on object. Pfeuffer et al. [2014] presented gaze-touch interactions on the touchscreen itself. By leveraging the gaze target, they mapped touch inputs either directly or indirectly, depending on whether the user looked toward

Touch input is natural and widespread on mobile devices.

Touch input can reliably be used to make gaze inputs explicit, solving the Midas Touch problem.

her finger or away. Possible advantages of this interaction include eliminating the fat finger problem, resulting in no occlusion, and making content accessible across the whole screen.

Gaze is a noisy input due to eye movements and tracking errors.

One challenge of working with gaze input is that it is inherently jumpy. Interaction techniques using gaze as input must be robust against short fixation durations as low as 180 ms [Rayner, 2009] before the gaze jumps to a new location. Yet, determining the three-dimensional orientation is also challenging from the hardware perspective. Niehorster et al. [2020] found that even with dedicated eye tracking hardware the gaze estimation will be off by up to 3.1° if the user is speaking or making facial expressions. In the context of mobile devices, where both the user's face and the devices move a lot, these errors amplify, and gaze tracking can be unreliable [Lei et al., 2023]. Therefore, head tracking could be a promising substitution depending on the purpose of gaze tracking. As our heads tend to follow our gaze, the early work of Stiefelhagen et al. [1999] already showed that head orientation alone can be used to approximate a user's visual target of attention. The head also bears the advantage of making less subconscious movements, and it is visually easier to identify in the camera feed. As head tracking provides additional unique pieces of information on the user, we summarize eye and head tracking under the term *facial tracking* in this thesis.

Depending on the usage context, head tracking is a promising substitution for eye tracking.

The smartphone has become the most used computing device, making research for these devices especially relevant.

Today's increasingly powerful mobile devices contain advanced camera systems and extensive calculation capabilities, making on-device processing of facial tracking and novel interaction techniques using this data possible. The technical possibilities and the continuous increase in smartphone usage make them an exciting area of research. Since 2019, mobile devices have a higher market share than desktop computers[1]. This makes it no surprise that in a survey by Statista conducted in September 2023[2] 96% of 5,990 consumers ranked the smartphone as their most used consumer electronics device. Globally, the numbers look simi-

[1] `https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/2019`. Accessed April 2024.

[2] `https://statista.com/forecasts/998677/most-used-consumer-electronics-in-germany`. Accessed April 2024.

lar: Datareportal presented usage statistics with data from GWI and We are social in October 2023[3]: Their data shows that 93.4% of consumers worldwide use their smartphone to access the internet, a slight year-on-year change of +2.6%. On the other hand, they measured that only 54.5% of the global population use a personal desktop or laptop computer. Computer usage declined by –8.6% when compared year-on-year. While these usage trends make research on mobile interaction techniques relevant, the plethora of sensors built into smartphones also makes them an exciting platform for exploring multimodal capabilities.

Smartphone usage continues to grow globally, while desktop computer usage declines.

Designing interaction techniques for mobile devices, however, comes with additional challenges irrelevant to desktop interactions. Firstly, the input of a single touch is even less expressive than mouse input, as it lacks the hovering state [Buxton et al., 1985]. Secondly, the devices are used in a different context, often on the go. Therefore, users can only pay reduced attention to the device as they, e.g., might have to look out for traffic while walking. Thirdly, their hands might be occupied because they are carrying a bag. This makes hands-free interactions relevant.

Smartphones are used across different mobile contexts.

As the portability of smartphones affords using the device in different environments, it also becomes exciting to apply facial tracking to different types of content: Of course, like in the seminal work of gaze interactions, one can identify what a user is looking at on the screen or allow her to control elements in the UI and perform actions. Those types of *screen-space* interactions all revolve around actions in the UI itself, e.g., input in addition to or as a replacement of touch. In the broader sense, however, facial tracking can also be applied to contents around the user and her device. For instance, Nagai et al. [2022] combined the camera feeds of the front and back-facing cameras in a smartphone to identify objects of interest that a user is looking at inside a room. We denote remote content—physical or virtual—like other devices or augmentations in mixed reality as *world-space* content.

Possible interactions realized through a smartphone can target both screen-space and world-space content.

[3] `https://datareportal.com/reports/digital-2023-october-global-statshot`. Accessed April 2024.

1.1 Research Questions

The work in this thesis revolves around the effects of the computer no longer perceiving its user as in Figure 1.1, focussing on mobile handheld experiences due to their omnipresence and continuously growing usage in our everyday lives. The presented interaction techniques in this thesis were driven by the following questions:

- Which aspects of facial tracking, i.e., eyes or head, are appropriate as input under which conditions?
- Should facial tracking be used as an explicit or implicit input?
- Is facial tracking differently suitable to world-space and screen-space content?

1.2 Hypotheses

H1 *Facial tracking enhances ergonomic one-handed use of mobile devices*. Users cannot reach everything on the screen with their thumb when holding their phone in one hand. Gazing is suitable for reachability techniques as it is not physically limited in the same way.

H2 *Facial tracking enhances perception of virtual content*. In the real world, the composition of our visual field changes with every slight movement of our eyes and head. However, virtual content in augmented reality remains static without displacing the handheld device. Thus, facial tracking could enhance how virtual content is rendered on screen.

H3 *Facial tracking leverages multi-device environments*. As we can visually target any object in an entire room, facial tracking has the potential to effortlessly reach and control contents that are out of the arm's reach of the user. This could be especially beneficial in meetings with multiple users and devices.

H4 *Facial tracking allows designing for distraction*. Notification banners often result in undesired content overlap. When a device knows where the user is looking on the screen, it could place alerts at a specific distance from the locus of attention.

To verify these hypotheses, we designed a variety of interaction techniques presented in this thesis. They can be characterized based on their mapping and controlled content.

	screen-space	**world space**
explicit	Headbang ▷ *Chapter 3*	GazeConduits ▷ *Chapter 5*
	HeadReach ▷ *Chapter 4*	
implicit	Attentive Notifications ▷ *Chapter 7*	User-Aware Rendering ▷ *Chapter 6*

Table 1.3: The interaction techniques presented in this thesis explore interaction techniques utilizing face tracking with both implicit and explicit mappings across content that is either part of the GUI on-screen or contents surrounding the user (world-space).

Chapters 3–7 contain artifact contributions evaluated using controlled experiments.

These interaction techniques are presented in artifact contributions. As established in the community, we conducted empirical studies for evaluation. For novel interaction techniques, in particular, controlled experiments are required to precisely measure the benefit to human performance. For an overview of the research types in HCI and their evaluation methods, we recommend the reader to refer to the work of Wobbrock and Kientz [2016].

1.3 Outline

The remainder of this thesis is structured as follows. Chapter 2 presents the foundations of the different research domains associated with this thesis. We present two explicit screen-space interactions in Chapters 3 and 4 that support ergonomic smartphone use. Chapter 5 revolves around explicit world-space interactions and presents how to utilize gaze tracking to leverage ad-hoc cross-device interactions. Chapters 6 and 7 present implicit interactions for screen- and world-space content, combining facial tracking with augmented reality rendering and notification placement. Chapter 8 summarizes the thesis and presents future perspectives.

Chapter 2

Foundations

This chapter presents different research domains in HCI that added a visual sense to an interactive system and closes with three preliminary studies that evaluate the facial tracking software used in the own work.

Visual tracking allows a computer to see its user and the world around it. Concretely, the user's gaze target and head orientation were commonly tracked features in the related work. This chapter contains the background knowledge required for the work presented later in this thesis. It also provides a quick and shallow overview of the different research domains that use visual tracking. We first examine early research prototypes using the user as input in the field of proxemics and then continue with the field of gaze analysis. The third section derives prevailing conditions from the intrinsics of visual facial tracking. The fourth and fifth sections provide an outlook on other research domains that use visual tracking of facial features, such as intelligent user interfaces or immersive 3D visualizations. The chapter closes with techniques to track facial data on mobile devices and three preliminary studies in which we evaluate the accuracy of the facial tracking software we used for our research prototypes in the following chapters.

This chapter only provides foundational related work that commonly fits all own artifacts presented in this thesis. Additional related work regarding specific research questions is presented in each of the following five chapters.

Figure 2.1: An example of spatially aware proximity-based interaction: When an iPhone is held close to a HomePod, it displays an interface to transfer music playback and control the speaker. On the other hand, the iPhone itself is spatially aware of its user and prevents the screen from turning off while somebody is looking at it.

2.1 Proxemic Interactions

Ubiquitous computing envisions a future in which technology disappears and is interwoven with everyday life. One requirement for enabling simple data transfers between devices is some awareness of their surroundings.

In 1991, Mark Weiser [1991] formulated his vision of *ubiquitous computing*, a future in which technology disappears and weaves itself into our everyday lives. Today, three decades after this vision, we use many computing devices of different sizes every day: Smartwatches, smartphones, tablets, and computers complement a variety of smart home and IoT devices. Yet, only in recent years have our devices become more interwoven, with seamless data transfers between different devices. While the integration of cloud services can partially explain this trend, the more critical technical improvement was that our devices became aware of other devices around them. For example, the devices depicted in Figure 2.1 use ultra-wideband connectivity and other sensors to enable seamless interactions.

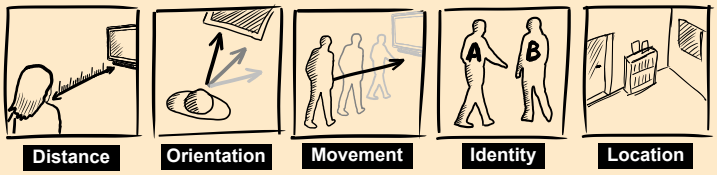
Proxemic interactions are enabled by device awareness of the environment.

Nonetheless, most of today's devices still are agnostic of what the users are doing in front of them. The research field of *proxemics* explores different interaction techniques that rely on devices being aware of their user through various sensors.

While proxemic interactions can be based on information of various dimensions, already measuring the very rudimentary distance between the user and the screen can greatly enhance the interaction. For example, based on the idea that users naturally move their head closer to the screen to inspect some content in detail, Harrison and Dey [2008] created *Lean and Zoom*. This technique uses the camera feed of a webcam to estimate the distance between user and screen. As seen in Figure 2.2, this measurement is then used to magnify on-screen content like websites when the user leans forward. In a user study, participants perceived the interaction technique as highly intuitive and that it increased comfort while working with visual media.

Lean and Zoom increases comfort while working with digital media by tracking the distance between user and screen to adapt the size of on-screen contents.

PROXEMIC INTERACTIONS:
Systems that react to the spatial presence of their users allow for proxemic interactions.



Graphic taken from Greenberg et al. [2011].

Greenberg et al. [2011] characterized the five proxemics dimensions depicted above.
In a nutshell, the spatial **distance** between two entities (human or technology) is the most fundamental category, and it is often classified into discrete zones. Edward Hall [1966] initially established the term *proxemics* as an area of study on the human use of space. He defined four distinct zones of interpersonal distances: intimate ($< 0.5\ m$), personal ($< 1.2\ m$), social ($< 3.7\ m$), and public (up to $7.6\ m$). Software could use these categories to enable different interaction modes.
Orientation adds more meaning to the distance measure by determining with which angle an entity faces another. By capturing distance and orientation over time, one yields **movement**. **Identity** identifies a specific entity. **Location** denotes the physical context in which the interactions happen.

Excursus:
Proxemic Interactions

Figure 2.2: In *Lean and Zoom*, the computer reacted to the proximity of its user to change the content scale. Images taken from [Harrison and Dey, 2008].

Tracking the orientation of a handheld device relative to its user allows for around-body interactions. For instance, to show different keyboards or adapt notification behavior.

An essential characteristic of handheld devices is that they afford to be displaced frequently. Thus, distance, orientation, and identity become promising dimensions for new interactions. Chen et al. [2014] presented different interaction techniques for *around-body interaction*. The small footprint of mobile devices limits the room for displaying buttons in the UI. Tracking a device's posture can substantially enlarge the interaction space of mobile devices. They propose to make the software keyboard aware of the posture and show the most common keys while the user holds the device in front of the body, and less frequently keys like numbers when the device is held sideways. Around-body interaction also allows for increased context awareness. For instance, notifications are only visible while the device is held close to its owner.

Large shared screens could switch between different modes depending on what the people in front of them are doing.

Proxemic systems also enable novel interactions when multiple users try to interact with one system simultaneously. For instance, Ballendat et al. [2010] explored how large screens can benefit from the awareness of their surroundings. Their system provides split-screen views if multiple users are present, pauses movie playback when people are talking with each other or making a phone call, and presents touch controls only while somebody is standing in front of the screen.

Proxemic interactions aware of the spatial device arrangement allow for new collaboration techniques.

When people get together in meetings, they bring many devices with them. Rädle et al. [2014] presented interaction techniques that foster collaboration by enabling ad-hoc data exchange between devices. One possible use case is peephole navigation, in which the whole table contains a large content, and the respective area covered by a de-

vice is displayed on its screen, resulting in a large shared screen. On the other hand, separating a device from the others could provide a unique UI to provide annotations. However, their system requires an external tracking device known as *HuddleLamp*. It is a camera integrated into a desk lamp in the middle of the shared space and provides tracking for 0.6 m^2.

The design of the available cross-device interactions directly impacts the collaboration. Homaeian et al. [2018] evaluated two different interaction techniques to connect independent tablets to shared content. In their study, participants had to select an area of interest via touch on a shared or independent personal device. Their results show that either approach has advantages and disadvantages. Touching on a shared surface communicates to others what someone is working with, which can enhance collaboration. On the other hand, selecting content from the personal device was more comfortable and less distracting for others. In conclusion, in collaborative settings, interaction designers should not only optimize the interaction of the individual user but also need to assert transparent communication of their actions to collaborators.

Collaborative settings pose a tradeoff between individual comfort and transparency of actions to collaborators.

All of the presented proxemic techniques provide richer interactions by obtaining knowledge of the spatial constellation between devices and users. However, proxemics are only one step toward capturing the user in detail. For instance, the orientation of a user's body already provides a rough direction of what she is facing. However, to understand what a user focuses on exactly, we also need to analyze her eyes and head.

2.2 Gaze Analysis

Vision is our primary input modality. The first interaction with an object is usually looking at it [Zhai, 2003]. With around 10 million bit/s, our gaze has a bandwidth roughly corresponding to an ethernet connection [Koch et al., 2006]. The information gained from looking is so deeply wired in our brains that people will look at an object of thought even

Humans point their gaze toward any element in the vicinity that relates to what is going on in their mind.

if it does not provide valuable visual information. Kahneman [1973] placed different line drawings in front of his study participants while conducting an interview. When participants were, for instance, asked to list different car makes, they pointed their gaze toward a vehicle drawing despite the absence of a brand logo or type label.

In HCI, gaze was studied as early as the 1980s in the hope of providing a fast selection method.

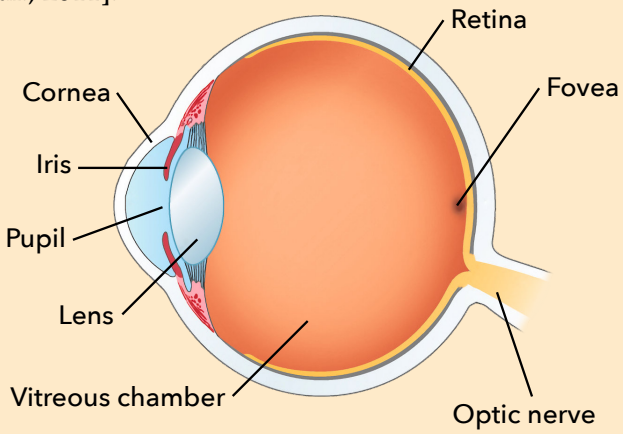
Researchers frequently used gaze to specify targets in pointing tasks.

The fast and nearly effortless movement of our gaze and the fact that we can quickly look at objects out of our arm's reach made it attractive for HCI researchers to evaluate gaze as an input modality as early as in the 1980s [Ware and Mikaelian, 1987; Starker and Bolt, 1990]. Since then, gaze has proven itself to offer promising input for *pointing tasks* or *attention analysis*. For instance, with MAGIC pointing by Zhai et al. [1999], one can quickly warp the cursor to a different screen area by gazing. To overcome accuracy issues, subsequent refinement of cursor location and clicking is performed using familiar mouse interaction. Alternatively, like in the work of Strapper et al. [2017], one could magnify the gazed screen area to make targets larger and thus easier to hit. Even multitouch gestures can be applied to gazing targets. For instance, Stellmach and Dachselt [2012] presented methods to pan and zoom content on a remote screen by tilting or touching a handheld device.

Challenge 1: Gaze input is jittery.

While gaze input seems promising as it is fast, requires no effort, and can reach any object in the vicinity, different challenges occur. The first challenge is that gaze tracking is inherently jittery. The simple fact that people cannot recall specifically where they looked [Clarke et al., 2017] already shows that subconscious processes influence eye movements. Even during fixations, our eyes do not remain still, exposing microsaccades and ocular drift [Krauzlis et al., 2017]. One reason for frequent eye movements is the composition of our retina, which influences the acuity of different areas in our visual field. Humans do not notice blurriness in their vision despite their paracentral vision covering less than 5% of the horizontal visual field only due to frequent fast eye movements.

HUMAN EYE:
Our eyes perceive visual inputs. They translate light in our environment into electrical impulses that create an image inside the brain. On its way to the optical nerve, the light passes through multiple layers of the eye: It enters our eyes through the *cornea*, a protective outer layer, and is then bundled by the *lens* whose focal point is controlled by the *iris*. After passing through the vitreous chamber, the light reaches the *retina*.
The retina contains two types of photoreceptors, *rods* and *cones*. Both react to light differently. Cones allow us to see in color as they react to red, green, or blue wavelengths. Conversely, rods measure light intensity, i.e., whether it is bright or dark.
The *fovea*—the center of our visual field—contains large amounts of cones. This results in a high-resolution image in the central area. In comparison, in the slightly pigmented *macula* around the fovea, the density of cones is already lower. The farther away a location on the retina is from the fovea, the more it consists of cones rather than rods. Thus, the visual acuity deteriorates toward the edges of our visual field [Campbell et al., 2011].



Graphic modified from Ling et al. [2016].

Excursus:
Human Eye

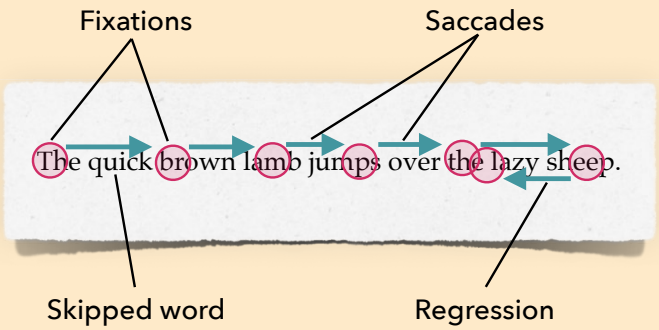
Gaze trackers have to estimate the three-dimensional position of the eye based on a camera feed. This makes slight errors in the orientation angles likely. As the gazing location can only be identified based on the eye orientation, the intercept theorem makes it evident that errors will grow with

Challenge 2:
Tracking and calibration errors

increasing distance between eyes and what they are looking at. Especially in the context of cursors on screens, even slight errors in gaze tracking will result in an infinite loop of the user correcting her gaze to match the cursor only for the cursor to change its position again [Jacob, 1995]. Thus, algorithmic stabilization of gaze quickly became an area of research. For instance, Zhang et al. [2008] evaluated different approaches to stabilize gaze inputs by progressively snapping the gaze input to possible targets.

Excursus: *Eye Saccades and Fixations*

EYE SACCADES AND FIXATIONS:
Eye movements are split into two alternating phases: *saccades* and *fixations*. During a *saccade*, the eye shifts the center of the visual field in a fast ballistic movement. During a saccade, the eye can move with speeds up to 500°/s when the change in focus exceeds 10–30°. For smaller movements, the minimum duration of a saccade is 20–30 ms [Binder et al., 2009]. While reading, a single saccade takes about 30 ms [Abrams et al., 1989] during which it jumps about eight character spaces or 2°of visual angle [Rayner, 1978].
The eye rests in between saccades to perceive the world around us. Depending on the current task, these *fixations* can take between 180 and 330 ms. For instance, the eyes make the fastest movements during visual search and perform larger and slower movements during scene perception. When reading silently, fixations typically take 225 and 250 ms [Rayner, 2009].



Challenge 3: The Midas touch problem

Gaze interaction is also prone to what Jacob [1990] called the *Midas Touch* problem: If one uses gaze as both input and output modality, one cannot look at anything without

activating it. Therefore, gaze interactions use dwelling or an additional mode for explicit confirmation.

One can use gaze tracking to derive what a user is paying attention to.

While concurrent usage of gaze as input and output modality is problematic, and accuracy issues exist, using gaze only as the user's output channel provides a helpful indicator of areas of attention. Humans are trained to infer what someone is paying attention to just from their eye gaze. Frischen et al. [2007] found that children already learn at ages between three and five to interpret other people's gazing as directional information for objects of interest. As this information is obtained from head and pupil orientation, it becomes more salient if multiple people look at the same object. The urge to look at something that crowds of other people also look at is denoted as *collective gaze* by psychologists [Sweeny and Whitney, 2014].

Highlighting the gazing of others can positively impact teaching.

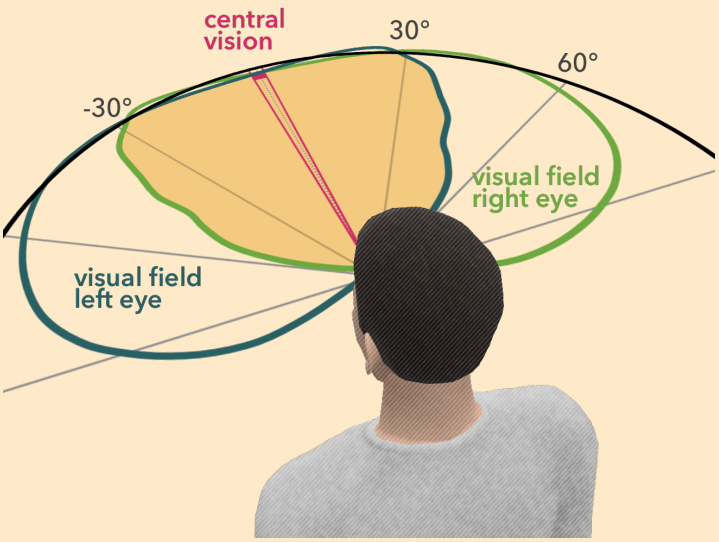
HCI researchers integrated gaze tracking to infer the attention of an audience. For example, Sauter et al. [2023] evaluated different approaches to highlight classroom attention in online learning formats. Their results indicate that instructors prefer a simplified ellipse over a heat map as it combines suitable precision with an unobtrusive visualization. Vice versa, students value a visualization of their instructor's gaze and pointing on the slides [Wagner et al., 2023].

Head tracking alone is sufficient to determine what somebody is paying attention to.

Head orientation is not only an essential aspect of determining an object of interest for another human. As it is uncomfortable to look at objects at an angle, our head follows our gaze. In fact, head orientation can reliably be tracked as a substitute to identify the focus of attention, as presented by Stiefelhagen et al. [1999]. In their study, a Hidden Markov Model identified the focus of attention with 98% accuracy based on head orientation alone. Similarly, Esteves et al. [2017] found that object selection with smooth pursuit tracking feels just as natural as with gaze. Multiple of their study participants could not tell the difference between gaze and head tracking as their movements were tightly coupled. Even so, head tracking has been used less commonly than gaze tracking in research so far.

Excursus:
Visual Field

VISUAL FIELD:
The visual field describes the visually perceivable area in front of our face. For adults, the visual field extends up to 214° horizontally. Vertically, it is a little smaller [Strasburger, 2020]. The inner 90° of the human visual field are perceived by both eyes and allow stereoscopic depth vision.
The visual system has multiple levels of acuity. The fovea only perceives 5° of the inner visual field at our line of sight. The irregular distribution of rods and cones on the retina results in high-resolution perception of only the tiny foveal part of the visual field. The visual acuity at 2° is already halved compared to foveal vision. Contents at most 30° away from the line of sight still belong to *near peripheral* vision. After that, color perception and acuity deteriorate even faster [Anstis, 1998; Abramov et al., 1991].



2.3 Characteristics of Facial Tracking

Multiple physical restrictions affect the visual tracking of the eyes and head equally. While some are intrinsic to the human eye or gaze as a modality, others are external, like usage posture or camera occlusion. We identified six com-

mon characteristics of visual tracking that we needed to consider when designing our own artifacts.

People hold their smartphone approximately 26.8 cm away from their face

Understanding usage posture. Boccardo [2021] evaluated how 233 people hold their smartphones while reading content on the display. She found that, on average, people held the display 36.8 cm away from the display (SD = 6.6 cm), with the visual distance tending to be slightly closer when sitting and slightly farther away while standing. According to Lei et al. [2023], the distance to the display is typically smaller when lying in bed. Considering that text is everywhere in UIs, we assume these findings adapt well to general smartphone usage.

Therefore, the smartphone fills a visual angle of 23.8° inside its users visual field

Visibility of on-screen content. Knowing the typical visual distance between the smartphone and the user's face, we can calculate the acuity with which it fits in the user's visual field. As acuity decreases toward the borders of our visual field, the *visual angle* of the smartphone in front of the user becomes relevant. The approximate visual angle of a modern phone of around 23.8° means that most of the phone fits is perceived with relatively sufficient acuity. The edges of the phone, however, bleed into near-peripheral vision, and reduced color perception needs to be considered in these areas.

In a study by Khamis et al., the face of smartphone users was only visible half of the time while standing or walking.

Visibility of the user. While we can assume the user has good device visibility, this does not apply to the inverse direction. Of course, for visual tracking to work, the device needs to see its user, not only in lab settings but also in the wild. Khamis et al. [2018] conducted a two-week study in which 11 participants were frequently photographed with the selfie camera of their phone in everyday situations while using their phone. The authors analyzed the 25,726 captured images regarding their face visibility. They found that the faces of their participants were fully visible only 29% of the time. However, this number increases to about 50% while people are standing or walking. A full face was seldom visible in the camera when lying or sitting. How-

ever, the eyes were often still visible in these cases. The authors could also find differences in face visibility across app types, as different apps often have varying usage postures.

Excursus:
Visual Angle

VISUAL ANGLE:
The visual angle V describes the angular size of an object inside the human visual field. For an object of a diameter S and a distance D, the visual angle is calculated as follows:

$$V = 2 \times arctan(\frac{S}{2D}) \tag{2.1}$$

For example, nowadays, a standard smartphone display size is 6.1" (= 15.5 cm). This results in a visual angle of 23.8° at a usage distance of 36.8 cm. For reference, the edges of macular vision that span the inner 17° of our visual field are considered the boundary to peripheral vision. The graphic below tries to visualize the reduced acuity and color perception toward the screen edges when looking precisely at the phone's center at the typical usage distance. For reference, the screen contents of the visual angles of foveal, paracentral, and macular vision are highlighted.

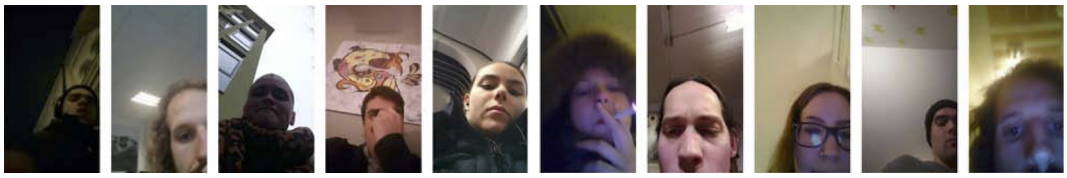


Figure 2.3: The dataset collected by Khamis et al. shows how smartphones see their user in the wild. In their study, the full face was only visible 29% of the time. Image taken from Khamis et al. [2018].

This seems less of an issue with explicit usages of facial tracking in the interface.

While the overall time the user's face is visible seems relatively low in this work, it is important to remember that facial tracking was not used as an input modality in this study. In particular, implicit usages of face tracking must expect a frequent lack of facial data. This problem likely does not transfer to explicit interactions. Face ID on iPhones does not have visibility issues, as users understand that their faces need to be visible to unlock their devices.

Gaze requires minimal effort, but our eyes also make subconscious movements.

Eye tracking errors are likely amplified with increasing distance to the gazed-on surface.

Intrinsics of eye tracking data. As pointed out above, the advantage of gaze is that we can move our eyes fast with little conscious effort. This quickly becomes a disadvantage when it comes to tracking our eyes. Noninvasive visual tracking requires deriving the inherently jittery eye postures from a camera image. As the pupil is only a small part of this image, even a small error of 1 px can lead to errors. Yet more importantly, as the eye orientation is not a meaningful input alone, ray casts against interactive surfaces are required. The further away this surface is from the eye, the larger the error becomes. For interfaces that are supposed to be operated primarily via gaze, the target size becomes an essential factor to consider [Ware and Mikaelian, 1987].

Head tracking provides a more stable input feed than gaze tracking.

Intrinsics of head tracking data. While head movements require more effort than gazing, head orientation as input has the advantage of being more stable than gazing. First, intrinsic data noise is lower as the head makes no subconscious ballistic movements like the eye. Second, from an external standpoint, the head is more straightforward to track as it is a larger object in the camera feed with multiple unique features. The information of tracking head and

gaze movements is interconnected. For instance, Lanman et al. [1978] observed monkeys performing object tracking. While having more variability, both head and eye movements could be used to identify the tracked object. Also, in the already mentioned study by Stiefelhagen et al. [1999], head orientation alone was sufficient to identify the focus of attention.

People might prefer head-based selections over gaze-based selections if this omits additional steps for the interaction.

The unique strengths of gaze and head tracking become evident in the study by Kytö et al. [2018], who compared selection times and accuracy across these two modalities for HMD AR use cases. In their study, gaze was faster than head-based input.However, the lower precision of gaze required an additional refinement technique using head- or hand-based control. Notably, their participants found head-based inputs easier to control, as they have less calibration error and data noise.

Both head and gaze tracking allow to select elements anywhere in the vicinity of the user.

Reaching and selecting contents beyond the screen. Both gaze and head tracking are great for selecting targets that are out of reach, as they can target any object in the 3D space in front of the user. With gaze, however, where the eye serves as both input and output modality to the user, a new problem related to the Midas touch appears. Namely, it is impossible to use interface elements like a slider to control continuous input parameters, as one cannot look at the slider to perceive its boundaries without specifying a value. Stellmach and Dachselt [2012] used gaze in combination with touch to overcome this issue. They presented different interaction techniques using a handheld device to interact with distant content. For instance, users would look at an item on the distant screen and tap on the touchscreen to select it. Similar to the refinement in MAGIC pointing, every action that cannot be performed via gaze is then performed via touch. For instance, dragging the finger to refine selections or applying familiar multitouch gestures to gaze-selected distant objects [Stellmach and Dachselt, 2013]. Please note that using head tracking to control a slider would also solve the double role of gaze: One can gaze-shift the slider to see its boundaries without side effects in controlling its value with head orientation.

Introducing an additional modality like touch for confirmation helps with Midas touch issues.

2.4 Immersive Computer Graphics

Not only interaction with GUIs but also with virtual 3D content can benefit from tracking the user's face and gaze. Rendering techniques can increase the realism of the virtual content by synchronizing the user's head with camera frustums that converge into the user's eyes.

Excursus: *Camera Frustum*

CAMERA FRUSTUM:
In computer graphics, a virtual camera moves through the 3D scene and renders it into a planar image we can see on the screen. The so-called *camera frustum*, a cut-off pyramid in front of the virtual camera, specifies the visible area of the camera and its perspective distortion. Mathematically, the two properties of a virtual camera that influence its frustum are the transformation matrix and the projection matrix: The *transformation matrix* specifies the location and orientation of the camera in the scene, i.e., what the user sees on the screen. The *projection matrix*, on the other hand, defines how a point inside the frustum is mapped to the planar image plane.

User-perspective rendering is used in different setups to increase the realism of the virtual content.

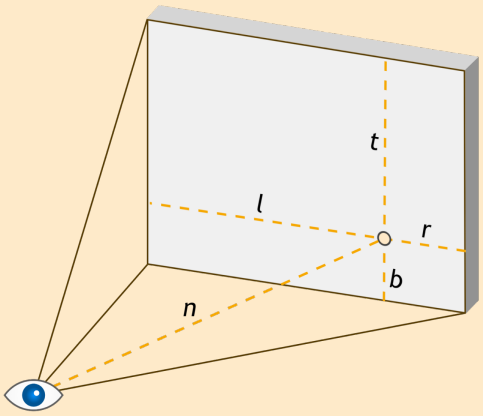
No matter whether one looks at CAVE systems with wall-sized displays or head-mounted displays, allowing users to change what they see based on their physical movement is achieved by synchronizing the transformation matrix between the user's head and the virtual camera. When the projection matrix is additionally determined so that the rendered image on the output surface fits naturally into the user's natural visual field, this is called *user-perspective rendering* (UPR).

UPR provides correct alignment and size of virtual content and motion parallax effects.

UPR promises more natural experiences of virtual content. It makes understanding what is visible on screen easier, as the content correctly aligns in space with the user's natural visual field: a window into a parallel world. UPR also helps observers estimate the size of virtual content and adds motion parallax effects that help with depth perception.

Excursus: *User-Perspective Rendering*

USER-PERSPECTIVE RENDERING:
A generic camera frustum is symmetric and imitates the idealized vision cone, as the human visual field is also symmetric. This is suitable when interpreting the virtual scene as a self-contained world. For example, following the player's character with a symmetric frustum makes sense in a game observed from a third-person perspective. However, when the visual output of the virtual world interacts with its user's natural visual field, the frustum becomes only a part of the user's overall frustum. Hence, a symmetric frustum is only correct if the screen is centered in front of the user's face. User-perspective rendering, therefore, calculates the camera frustum so that it expands across the fraction of the user's natural visual field that contains the output screen.



UPR defines the transformation matrix so that the camera is moved into the location of the eye. The projection matrix P is calculated by inserting the distances of the intersection point of the normal to the camera to the screen edges:

$$P = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{n+f}{n-f} & \frac{2fn}{n-f} \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{2.2}$$

For a detailed formula explanation, see Kooima [2009].

Figure 2.4: The Varrier system is a tiled semicircular display array. Each display aligns the content to match the user's natural visual field for its respective area with a unique UPR frustum. Note how the content of the outer displays seems stretched from the photo's perspective. That is because the visualization only makes sense from the user's perspective. Image taken from Kooima [2009].

CAVE systems use head tracking information and user-perspective rendering to render virtual content in the natural visual field of the users.

Cave automatic virtual environment (CAVE) systems are prominent examples of head tracking in combination with rendering virtual worlds. A CAVE is a VR environment in which the walls, floor, and ceiling display the rendered virtual world through projectors or displays. While HMDs artificially restrict the visual field of their users as they cannot address a full 214° FOV the human eye can perceive, this limitation does not apply to CAVEs, where users perceive the virtual world through their natural vision. To display an image that correctly immerses the users into the virtual world, what is depicted on each side of the room needs to be calculated dynamically based on the location of the user's head and, thus, eyes. This means that the virtual content is rendered in user-perspective (previous page). The system calculates offset renderings for the user's eyes and uses shutter glasses to create the 3D perception of the virtual world. Cruz-Neira et al. [1992] presented the first CAVE system. At this time, the system still had notable processing delays and only worked for a single user. For

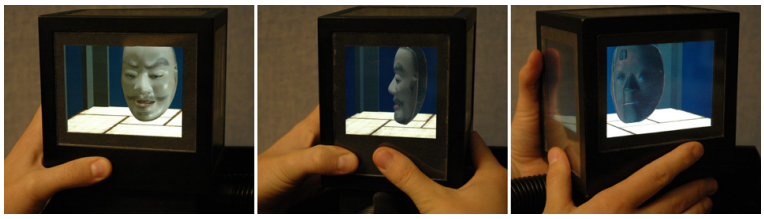


Figure 2.5: pCubee was a handheld cube with five small LCDs. The use of UPR created the illusion of a 3D object residing inside of the cube. Image taken from Stavness et al. [2010].

instance, Lebiedź and Mazikowski [2021] show how active shutter glasses can deliver different images to two users inside one CAVE.

Fish tank VR and VR experiences on WSDs also rely on user-perspective rendering.

While their spatial arrangement of screens differs, fish tank VR and immersive visualizations on wall-sized displays (WSDs) present similar tracking and rendering concepts. The original *Varrier* by Sandin et al. [2005], for instance, consisted of 35 LCD panels arranged semicircular around the user, enabling a FOV over 120°. An extended version of the Varrier is depicted in Figure 2.4.

pCubee was one of the first handheld prototypes using user-perspective rendering.

Different research tried to bring the advantages of user-perspective rendering to handheld devices. For instance, the *pCubee* by Stavness et al. [2010] consisted of five small LCDs that together formed a cube. Combined with additional head tracking hardware, the system calculated dynamic camera frustums for each visible screen to create an immersive depth effect. As the head tracking and continuous calculations of camera frustums have high computing costs, early systems were still wired to a PC. UPR rendering on mobile hardware was only possible multiple years later, e.g., in the work by Yang et al. [2018].

In combination with handheld AR, UPR allows for true magic lenses.

When combining UPR with handheld augmented reality, one obtains "true" magic lenses in which the device becomes virtually transparent. However, integrating the camera feed required for AR to UPR adds more computational complexity. Therefore, early systems imposed multiple restrictions on usage. For instance, the system by Hill

et al. [2011] consisted of two cameras attached to a graphics tablet connected to a PC. However, the achievable frame rate was relatively low (up to 30 fps), and the technique only worked at a specific fixed distance between the user and the display. Likely due to the prevailing hardware limitations in processing power and camera resolution, we could not find stable AR prototype systems using UPR running on smartphones. For example, Andersen et al. [2016] presented three alternative implementations of a handheld UPR display, all having unique shortcomings.

Hardware limitations made stable UPR experiences on handheld devices impossible during the last years.

2.5 Adjacent Research Domains

Facial features are also tracked in other research domains. Two interesting research areas related to this work are intelligent user interfaces, as well as accessibility and medical applications.

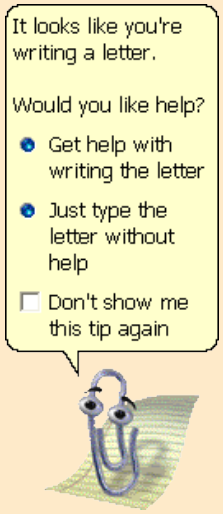
2.5.1 Intelligent User Interfaces

Beyond static GUIs, a variety of research domains in HCI use additional user inputs to create interfaces that adapt to users' goals and needs. One prominent example is intelligent user interfaces (IUIs). IUIs leverage a variety of inputs, from the conventional pointing and keying inputs, the user's task performance, their location or physiological state, e.g., heart rate, their gazing and facial expressions, and more. By monitoring these inputs over time, IUIs can identify commonly used actions and adapt their functionality to provide easier access to them. This can help users to perform routine tasks [Lavie and Meyer, 2010].

IUIs can use additional inputs like eye tracking to adapt their functionality or provide interventions.

Excursus: *Intelligent User Interfaces*

INTELLIGENT USER INTERFACES:
An *adaptive* or *intelligent user interface* (IUI) monitors the users's inputs over time and uses a recommendation system to identify how it could help its user. Rothrock et al. [2002] summarized the variety of IUI definitions in literature into the following: "An adaptive interface autonomously adapts its displays and available actions to current goals and abilities of the user by monitoring user status, the system task, and the current situation." IUIs can also proactively deliver recommendations at opportune moments in time.
One early and infamous example of an IUI is *Clippy*, part of Microsoft Office applications between 1996 and 2006. Microsoft conducted usability studies over 25,000 hours [Horvitz, 1998] and carefully developed an algorithm to identify opportune moments to notify the users about Clippy's recommendations [Horvitz et al., 1999]. According to Whitworth [2005], many users disliked Clippy because it did not respect the rules of polite computing.



Screenshot of Clippy in Office 2000, used with permission from Microsoft.

Proactive intervention of IUIs is especially promising to increase user safety.

In recent times, the recommendations in IUIs rely on machine learning and artificial intelligence, for instance, to display frequently used apps on the home screen of a smartphone. Since Clippy, developers have implemented proactive interventions of IUIs only with deep considerations. One area where the proactive intervention of an IUI actually makes sense is when it comes to safety, for instance, in the context of cars.

The survey of Wells-Parker et al. [2002] showed that a driver's emotional state influences driving style. Angry drivers perform dangerous maneuvers more frequently, which increases their likelihood of being involved in an accident. Thus, lowering the aggression of car drivers in-

creases the safety of all road users. Braun et al. [2019] explored different communication strategies with angry or sad drivers. While their main finding is that proactively approaching drivers works better with a voice assistant than with standard UI or ambient lighting, it is notable that the gazing trajectories of their participants differed between emotional states. This shows that gaze tracking is also a promising modality for deriving emotional states.

Braun et al. were able to infer the emotional state of a car driver by analyzing gaze trajectories.

2.5.2 Accessibility and Medical Applications

Finally, facial tracking also has promising potential in the areas of accessibility techniques and medical applications.

Silent speech input (SSI) is one of these novel input techniques from which both disabled and able-bodied users can benefit from tracking facial features. Acoustic speech input provides a comfortable hands-free input technique in private settings. Yet, in public, it can be too exposing for users, annoying for bystanders, or unreliable in loud environments. Instead, SSI visually tracks lip movements with the front-facing camera. A study by Pandey et al. [2021] suggests that SSI is perceived as better socially acceptable than loud speech input.

Silent speech input uses visual tracking of lip movements to allow hands-free text input, which is less disturbing in public environments.

Paraplegic patients also rely on special input techniques to control their motorized wheelchairs. Facial tracking provides a promising alternative input mode in comparison to a chin-controlled joystick. For example, Lu et al. [2007] used head tilt as input, and Araujo et al. [2020] used gaze tracking to specify the driving target.

Head tilting or gazing is commonly used to control motorized wheelchairs of paraplegic users.

Gaze tracking could also have potential in future medical applications, e.g., for detecting functional psychoses. Evidence for that can be found in the work of Bestelmeyer et al. [2006], who observed that gazing patterns of schizophrenic people differ from those of healthy patients when looking at images. Among others, their gaze fixations are prolonged and at different locations than the ones of the control group. What is more, Phillipou et al. [2015]

Gazing trajectories of looking at images could be used to detect functional psychoses in the future.

found out that people with Anorexia Nervosa have shorter fixation times when viewing faces than healthy people.

2.6 Facial Tracking on Mobile Devices

Increasing the accuracy of visual facial tracking is an ongoing research topic.

Accurately estimating the user's head orientation and gaze target from the front-facing smartphone camera is an ongoing research topic. While earlier models tried to map the camera image directly to eye geometry, newer models use machine learning techniques to provide accurate results under more circumstances.

The gaze-interaction of EyePhone could differentiate between nine different screen areas.

The *EyePhone* system by Miluzzo et al. [2010] was one of the first systems that allowed gaze-based interaction on unmodified mobile devices. With EyePhone, users moved the phone relative to their face so that their left eye was in one of nine possible positions in a 3×3 element grid and then blinked to trigger input.

Model-based approaches map the camera image to an eye model to infer their position and orientation.

Model-based approaches try to map the camera image to a 3D eye model by identifying specific features of the eye. These features can use the outline of the iris or pupil. For example, Wood and Bulling [2014] used the RGB camera in an unmodified commercial tablet and Alberto Funes Mora and Odobez [2014] used an RGB-D camera. The depth data of these cameras made it easier to identify the location of facial landmarks, which increased the accuracy of head posture tracking. Goswami et al. [2014] improved gaze tracking accuracy by using this data to geometrically model the user's face and eyes. However, model-based approaches require clear images in order to work reliably. Thus, outside of lab settings, their accuracy is limited.

The calculation of appearance-based eye tracking approaches omits the detour of mapping the image to a model.

On the other hand, appearance-based approaches try to map the camera image directly to gaze direction and location vectors. The required trackable features of the eye to identify the gaze direction can vary heavily between different people and even fluctuate depending on their head postures. Supervised machine learning on large gaze datasets helps to make gaze estimation more reliable independent of user appearance. One early example is

Krafka et al. [2016], who estimated gaze with an end-to-end appearance-based approach using deep learning. In a survey by Lei et al. [2023], 25 of 27 appearance-based gaze estimation models published between 2015 and 2023 use convolutional neural networks (CNN). In fact, CNN-based gaze tracking is quite reliable even with low-resolution images and can, therefore, also be used outside the lab [Bao et al., 2021]. The recent methods by Cheng et al. [2022] and Bao et al. [2021] show that fusing facial and eye features instead of concatenating them can further increase the accuracy of gaze tracking.

Most recent gaze tracking research uses an appearance-based approach in combination with a CNN.

In the past, gaze tracking systems needed extensive calibration, used specific hardware, or had a high price tag [Khamis et al., 2018]. Yet modern smartphones contain a variety of sensors, including RGB-D cameras, and allow developers comfortable access to their data. This makes it possible for off-the-shelf phones to perform facial tracking and use this data for novel interactions.

Gaze tracking required specific and expensive hardware in the past.

For iOS devices, Apple introduced facial tracking as part of ARKit[1] in 2019. ARKit visually tracks the user in the feed of the front-facing camera and creates a three-dimensional mesh of the face topology located relative to the device. This provides information on head location and orientation relative to the device, as well as facial expressions (Figure 2.6). Moreover, based on the pupils in the camera image, the system also individually infers a location and orientation specifying the optical axis for each eyeball. These APIs are optimized for the mobile chipsets they run on and offer appropriate performance without needing to outsource calculations to an external computer.

Today, modern smartphones allow developers to track the user's face as part of their standard API.

[1] `https://developer.apple.com/documentation/arkit`

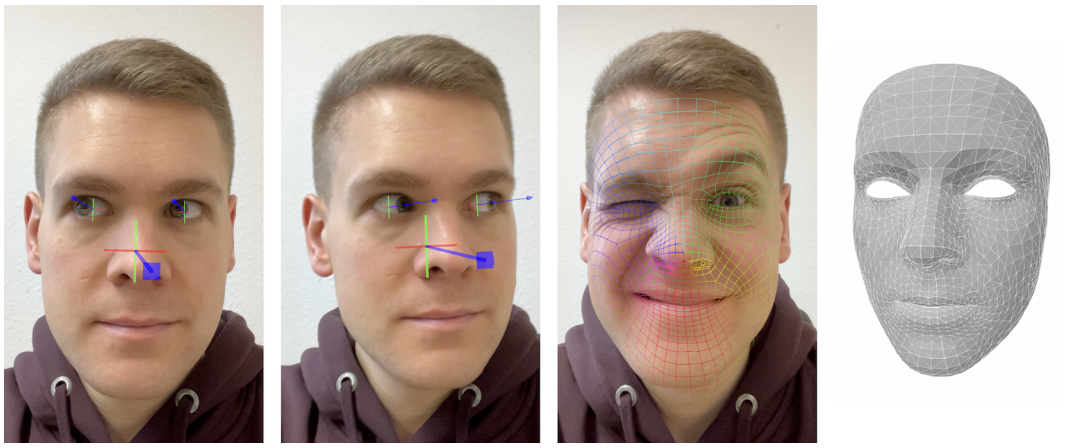


Figure 2.6: ARKit tracks both the facial features as well as eye and head positions. The right side of the figure shows a face mesh with the accuracy of 1220 vectors and how it is mapped to a grimace. The left side shows the origin and orientation of each eye and the head. The tracked coordinates are either the center of the eyeballs or the center of the head. The blue rays show how the gazing lines cross the pupil.

2.7 Evaluating Facial Tracking in ARKit for Head and Gaze Interactions

We evaluated head and eye tracking accuracy in three preliminary studies.

As even with specialized hardware gaze and head tracking contains errors, we first had to evaluate the accuracy achievable with the native eye and head tracking capabilities of iPhones. We conducted three preliminary studies to evaluate whether the spread and error of data are accurate enough to leverage novel interaction designs. The first study determined in which area in front of the phone the user can be tracked. The second study evaluated the precision of users performing head rotations of arbitrary angles using head tracking. As we knew from related work that gaze estimations could be influenced by the usage posture, we evaluated gaze tracking within a radius of 40 cm around the device in the third study.

Publications: The three studies in this section were published as parts of papers published at MobileHCI 2020 [Hueber et al., 2020] and CHI 2020 [Voelker et al., 2020]. The author of this thesis implemented the software artifacts used in the studies, which he also conducted and evaluated.

2.7.1 Determining the Tracked Area

We measured the part of the frustum that supports head and eye tracking.

The tracking range and accuracy depend on the camera frustum and resolution. Therefore, we conducted a preliminary study to determine the boundaries of reliable tracking. Five people of both genders and with different haircuts participated in this quick test.

Along the z-axis, facial tracking works when the user's head and phone are between 10 and 88 cm apart.

We placed an iPhone X on a camera stand and asked participants to move their heads away from the phone slowly. We measured the distance from the participant's nose to the phone using a laser distance meter at the farthest point the phone could still detect the user's head. The results show that the phone could track the user's head at a distance between 10 and 88 cm (SD = 3.2 cm).

Facial tracking was functional within a frustum of 30°.

Using the same approach, we measured the frustum angle that supports face tracking by asking the participants to move their heads left, right, up, and down while maintaining their distance to the phone. We found that face tracking is functional within a frustum of 30° (SD = 2.1°) in both the horizontal and vertical directions.

The facial tracking was robust against head rotations of up to 30° vertically and 35° horizontally.

Lastly, we asked participants to rotate their heads in front of the device to find out how far users could turn their heads away from the smartphone while still being tracked by the phone. We found that users could turn their heads by about 35° (SD = 3.5°) horizontally and about 30° (SD = 2.3°) vertically.

2.7.2 Quantifying Visual Head Tracking

We can map both head and eye tracking data to a two-dimensional location by ray casting.

As our previous findings supported that facial tracking on the iPhone X was appropriate for the typical usage postures (Section 2.3) we aimed to measure the accuracy of using the head as actual input next. Both head and eye tracking result in three-dimensional locations paired with a three-dimensional orientation. ARKit provides these vectors in a coordinate system that originates at the front camera of the device. However, for the graphical UI, we need to convert this into a two-dimensional location on the display. The dis-

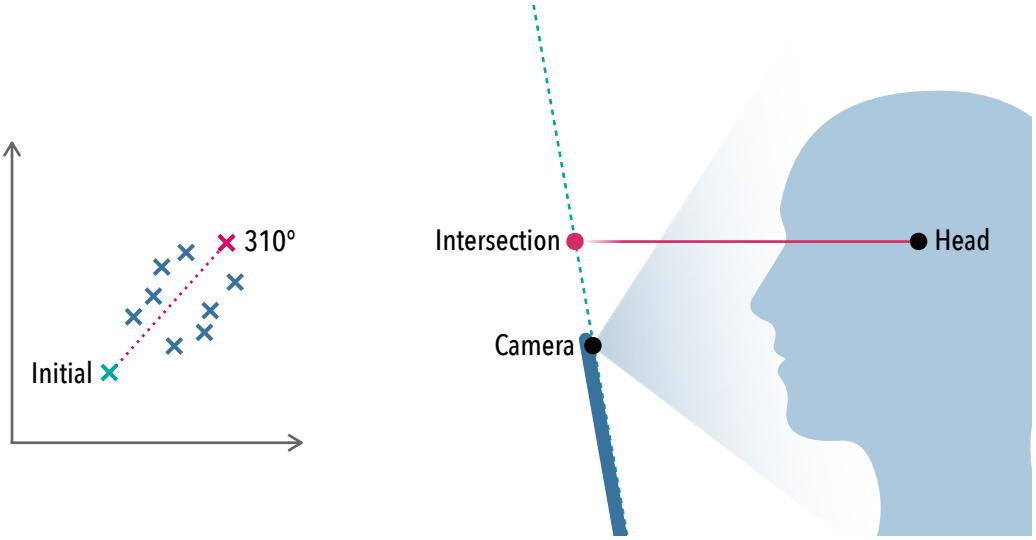


Figure 2.7: We convert head orientation to a location on the device by intersecting a ray from the head with a plane defined by the camera's orientation vector as normal (right). The initial intersection point and a buffer of up to 60 recent intersection points are stored to determine the head tilt. The point with the farthest distance from the initial point (left, magenta colored) is used for the angle calculation.

play can be modeled as a plane at the coordinate origin that uses the camera's orientation vector as normal. By shooting a ray cast from the head onto this plane, one obtains an intersection point that is controlled by the head's rotary yaw and pitch.

Head tilting allows to make controls with relative head movements.

For our first evaluation of head tracking as input we decided to focus on head tilting. Head tilting suits the modality of head position well: As the virtual ray cast from the head is invisible to the user, blindly specifying an absolute location on the intersection plane will be hard. However, humans are trained to make relative head movements, e.g., to adjust their visual field. This makes relative head input easy to control and accurate.

Buffering up to one second of head orientations provides stable tracking and allows users to correct themselves.

To determine the angle of head tilting, we analyze the trajectory of the interaction point on the device plane (see Figure 2.7). We store the point measured when the interaction started and a buffer of the 60 most recent points. As ARKit uses a 60 Hz sampling rate, our buffer can store up to one second of head movement. The point in the buffer with

the farthest distance to the starting point is then used to calculate the angle on the device, converting the 3D rotation into a 2D one. This approach allows users to change the angle they want to specify while it is being calculated. We could track continuous clockwise or counterclockwise rotations without a noticeable delay in our preliminary tests. We found the measured data stable enough not to need further filtering, which would slow down tracking.

Intersecting the ray shot from the head with the device plane has usability advantages over a plane that is orthogonal to the head.

For two reasons, we calculated the intersection point on a plane along the device instead of a plane orthogonal to the head. First, when holding their phone comfortably in one hand, users do not align the phone perfectly straight with their head, and the phone is tilted around the z-axis. Using the device plane allows us to keep the mapping of angles intact and avoid misconceptions about angles. Second, in a typical posture holding a smartphone, users slightly bow their heads toward the phone. This limits head rotations toward the chin more than any other direction. The intersection point on the device plane, however, counters this by requiring less movement to specify a point toward the bottom of the screen.

We conducted a study with 8 participants (22–28 years, M = 25.25, SD = 1.75, three female) to determine the deviation of measured angles.

Apparatus and Task

In this study, we evaluated the precision of head tilt for the smooth pursuit of a dot moving on the screen.

The participants were asked to hold an iPhone X at a typical location where they usually hold their phone. The study software displayed a white line ranging from the center of the screen to its border, specifying an arbitrary angle. A small dot at the center of the screen would begin to move linearly along the line with a speed of 2.4 $^{\text{cm}}/_{\text{s}}$ once the participant touched the screen. While the mapping of the dot to a head movement remained subjective, it assisted participants in smoothly pursuing a visual target. We tested all multiples of 10° in a random order with two repetitions, i.e., 72 trials per participant.

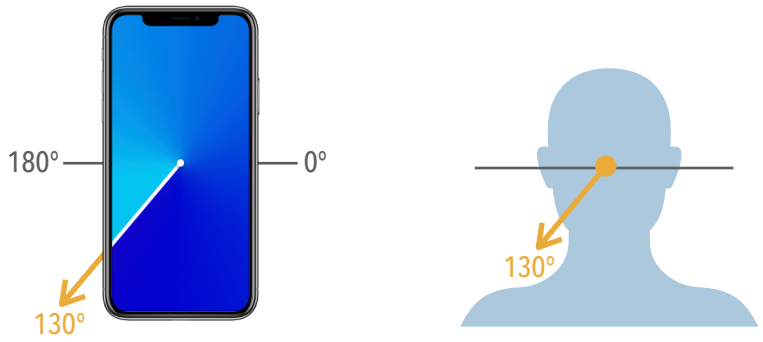


Figure 2.8: To evaluate the precision of head-based rotary input, participants had to tilt their heads along the indicated line in the interface (left). The rotary angle specified by the line in the interface clearly defines a change in the yaw and pitch of the head (right). Rotation is specified clockwise starting at 3 o'clock.

Participants were asked to target the dot with their heads, touch and hold anywhere on the screen, and follow the dot along the path with their heads. Once the measured intersection point moved farther than 4.8 cm, the system gave haptic feedback. Participants then lifted their fingers and continued to the next angle. The interface is depicted in Figure 2.8.

Results

Without feedback and including the human interpretation as error source, the error of the specified angle was 11.67°.

Since we wanted to evaluate the feasibility of head rotation as input, we measured the achievable accuracy with this system without feedback, including the human as error factor. The movement speed required to follow the dot was left for participant interpretation. However, trials always took less than two seconds to complete. The average offset between the targeted angle and the angle measured from our system was 11.67°. The spread of the data had a large standard deviation of 8.13°. While a sixth of the measured samples had a very high accuracy of less than 3°error, the maximum error we measured was 35.4°.

2.7.3 Quantifying Visual Gaze Tracking

In this study, we measured how reliable the gaze prediction is for targets in an area of $0.5m^2$ around the device without additional calibration.

One strength of gaze is that objects do not need to be within our arm's reach to see them. Therefore, for our investigation of gaze accuracy, we decided to evaluate not only the screen but also locations around the device within a radius of 40 cm. Similarly, as in the previous study, we calculated where a ray cast intersected a plane to specify an input location. However, this time, we used the positions and orientations of the tracked eyes instead of the head, and the plane was specified by the table instead of the screen. The gyroscope inside the device determines the relation between the table and the phone. Again, we used an iPhone X in this study. To determine the uncalibrated accuracy of the phone's gaze prediction toward targets on the table, we conducted a study with 10 participants (23–42 years, M = 32.73, SD = 2.31, four female).

Apparatus and Task

Participants had to look at nine targets on the table for five seconds while rotating their heads.

One iPhone X in a stand was placed on the table at a distance of 60 cm from the table edge. We defined the bottom of the phone as the origin of the coordinate system $(0, 0)$, and highlighted nine target locations around the phone on the table as depicted in Figure 2.9. The locations distances (in cm) of the gaze targets from the coordinate origin were $(0, 0)$, $(-40, 0)$, $(40, 0)$, $(0, -40)$, $(0, 40)$, $(-20, -20)$, $(20, 20)$, $(-20, 20)$, and $(20, -20)$. Participants were asked to look at each of the targets for five seconds. During this time, they were asked to move their head around while keeping their eyes on the target position. Throughout each trial, the phone recorded the intersection points of both gaze and head vectors with the table plane 30 times per second. This allowed us to analyze the difference between head and gaze tracking.

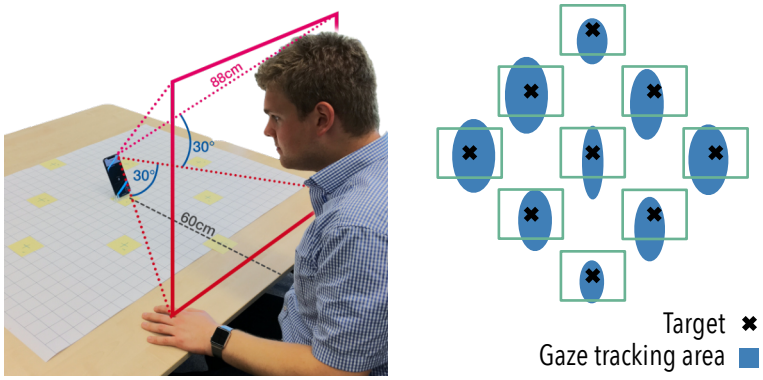


Figure 2.9: To analyze tracking accuracy, participants were asked to fixate each target for five seconds. The blue ellipses represent the gaze targets on the table calculated by the phone. Their spread can be explained by participants rotating their heads during their fixation times. For comparison, the green boxes represent the size of a 9.7″ tablet.

Results

Even without calibration and under difficult tracking conditions, gaze predictions were precise enough to identify objects with the size of a tablet.

To evaluate the accuracy of the smartphone's gaze estimation, we measured the distance between the estimated gaze location on the table and the marked location at which participants were looking. Our results show that the average distance over all participants and all targets was 2.5 cm on the x-axis and 9 cm on the y-axis. However, the data spread had a large standard deviation of 8.1 cm on the x-axis and 11.5 cm on the y-axis. Using this approach, the smartphone can detect if a user is looking at a specific table area with a size of 20×25 cm, which is roughly the size of a tablet.

2.7.4 Summary

Facial tracking on smartphones enables new interaction techniques.

Head and eye tracking enables novel multimodal interaction techniques. However, this required complicated setups and expensive specialized hardware in the past. Therefore, researchers could not explore the impact of facial tracking in mobile contexts. Modern smartphones provide facial tracking using their front-facing RGB-D camera. In

three preliminary studies, we showed that this technology is precise enough for HCI research.

Using ARKit on an iPhone X, we found that the user's face is trackable within a 30° FOV of the camera. Tracking was functional when holding the device close to the face or with an extended arm. Using head orientation for smooth pursuit tracking, we measured an average error of 11.67° even without feedback. Gaze tracking of targets on a table spread across an area of 0.5 m^2 had an average error of 2.5 cm horizontally and 9 cm vertically. Please note that all of these measurements were made without a previous calibration. The plot in Figure 2.9 clearly shows that the error systematically enlarges the farther a target is away from the phone. Thus, both calibrating the system with a few known locations in advance or limiting gaze tracking to the screen bounds will increase accuracy.

Head tracking was reliable without calibration.

User calibration of gaze tracking applications should increase accuracy.

After examining existing interaction techniques and the foundational data of our preliminary studies, we will present new techniques that use facial tracking as input. We will start with a discretization of head gestures that adds semantics to concurrent touch input.

Chapter 3

Allowing Quick Menu Actions with Head Gestures

SUMMARY:
The increasingly powerful mobile devices allow users to apply a variety of tools to on-screen content. However, they lack screen space to display many menu items at once. Inspired by social facial expressions like nodding and shaking the head, we present *Headbang*, an interaction technique that enriches touch input on handheld devices through slight head movement gestures. This way, users can easily execute shortcuts, like Copy, Paste, or Share, to on-screen targets while touching them.
We compared Headbang in two studies against device tilting interaction and touch interaction. The interaction technique can be reliably used while sitting and walking and offers a similar accuracy as touch interaction. Depending on the number of elements in a menu, Headbang could be operated even faster than touch input.

Publications: The work presented in this chapter was done in collaboration with Christian Cherek, Philipp Wacker, Jan Borchers, and Simon Voelker. The author of this thesis developed the research idea and relevant research questions, including the motivation of the work. Furthermore, he designed and implemented all experiments and the presented use cases. Most of this work has been published in the Proceedings of ACM MobileHCI 2020 [Hueber et al., 2020]. The author of this thesis is the main author of the paper. Most sections in this chapter are taken from the paper publication.

3.1 Motivation

The limited screen space and touch input expressiveness result in time-based inputs for context menus.

Touch input is prevalently used in mobile devices. However, its expressiveness is limited. Simple inputs such as tapping and swiping are usually already occupied with the semantics of selection and scrolling. Options that are familiar from desktop interaction, such as right-click or keyboard shortcuts, do not exist, and many everyday tasks require longer sequences of selections that hardly benefit from multitouch [Li, 2010]. The limited screen space on mobile devices amplifies this issue, as there is insufficient space to present large toolbars. Therefore, modern smartphone operating systems make use of long press gestures to show context menus. This comes at the cost of slowing down the interaction.

Related attempts to increase the expressiveness of touch input include stroke and multitouch gestures, force, and tilting input.

A variety of touch techniques have emerged to alleviate this issue. They aim to provide an additional semantic dimension to the touch input, which can be temporal or physical. Temporally, both stroke gestures [Appert and Zhai, 2009] or multitouch gesture sequences [Hinrichs and Carpendale, 2011] can be performed faster than long presses. Physically, one can use the force applied to the finger while touching the screen [Corsten et al., 2018] or utilize further sensors built into smartphones and tablets, such as motion sensors for tilt input [Baglioni et al., 2011].

Discretized head tilt is a promising and reliably trackable input to increase the expressiveness of touch.

With facial tracking, one can augment the touch input without requiring time-based input or additional screen space. As the first step in our research, we wanted to use a rudimentary, clearly defined, and discrete input. The head tilting accuracy we measured in our preliminary study (Section 2.7.2) suggests that one can reliably differentiate up to 16 states. These different head states while touching could then be used, for example, to perform shortcuts on specific items or enhance one-handed use by removing the need for specific menu buttons on the screen edges.

Head movement is a standard social communication method [Kettner and Carpendale, 2013; McClave, 2000] that has also been used to interact with interactive systems. For example, to move a cursor on a desktop computer [Gorod-

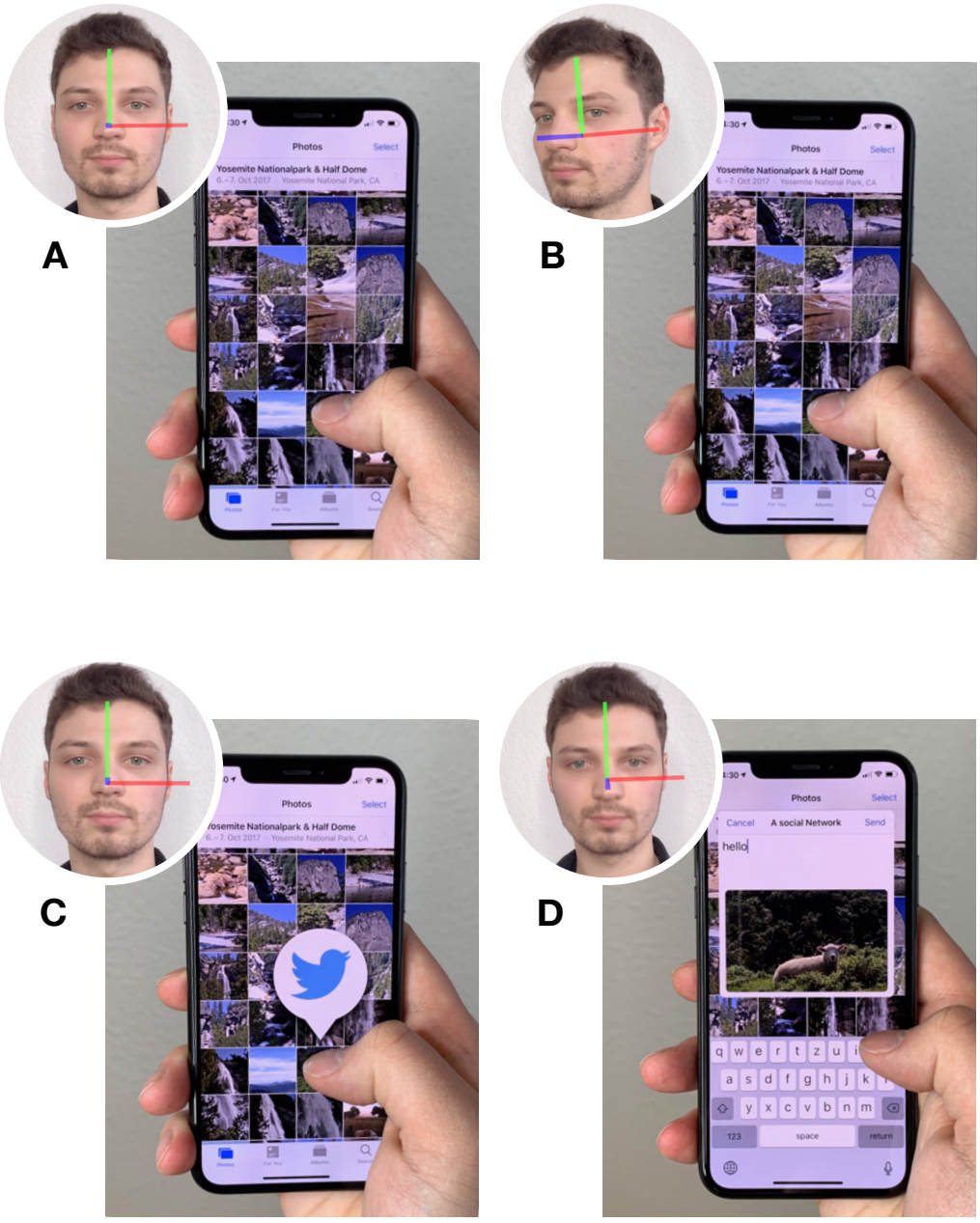


Figure 3.1: (a) To share an image to a social media application with a Headbang gesture, a user touches an image with his finger and (b) immediately moves his head slightly away from the device and back again. (c) When the gesture is detected, an indicator above the image displays the selected action. (d) To confirm this action, the user lifts the finger from the image.

HCI researchers used head movement as an input across desktops, mobile platforms, and head-mounted devices.

nichy and Roth, 2004], as command gestures while using head-mounted displays [Yan et al., 2018], or to interact with mobile devices [Crossan et al., 2009]. However, earlier research required additional external hardware to realize the head tracking. Visual head tracking omits the need for additional hardware on a smartphone. It can reliably track its user as she already focuses the screen and points her head toward the device while using it.

Our Headbang technique allows users to trigger actions with head rotations. It requires no additional hardware and saves screen space.

In this chapter, we present *Headbang*, an interaction technique that allows users to trigger actions by slightly rotating their heads in different directions. Headbang does not require additional hardware as it works with the visual input from the built-in front-facing camera. While users make their head input, the possible actions are visible in a compact menu. This saves valuable screen space that would otherwise be occupied by toolbars. After discussing related work, we present our Headbang interaction technique and its implementation in more detail. In our studies, Headbang gestures were detected reliably while sitting and walking, offering a promising alternative to context menus. We close with a discussion and a collection of use cases for the Headbang interaction technique.

We evaluated Headbang in two user studies.

The key contributions of this chapter are as follows: First, we present the Headbang interaction technique to increase the expressiveness of touch input and propose different use cases. Second, we provide a quantification of head tracking precision on a commodity smartphone and its robustness against walking. Third, we provide a comparison of Headbang with alternatives such as device tilting.

3.2 Related Work

Common app designs require mode switches to access all their features.

The fat finger problem and limited screen space of mobile devices make it unfeasible to present all app features in toolbars like desktop applications. On the other hand, the limited expressiveness of touch input also does not allow for a secondary click. Therefore, mobile apps commonly implement context menus accessed by mode switches, e.g., a long press.

3.2.1 Touch Gestures

Touch gestures add new semantics to touch input. For instance, stroke gestures that can be performed with single-touch input could be shortcuts to certain functionalities. They are even easier to recall than keyboard shortcuts.

A variety of different touch techniques have been introduced to mitigate the issue of the limited expressiveness of touch input. Several interaction techniques use single-point stroke gestures to access menus or shortcuts. Appert and Zhai [2009] presented the use of stroke gestures instead of keyboard shortcuts to access menu entries in a desktop interface. One advantage of stroke gestures was that their participants could recall the stroke gestures better than keyboard shortcuts. For handheld devices, Li [2010] envisioned *Gesture Search*: It provided a full-text search functionality of apps, contacts, or media stored on the user's phone by stroking the characters of the search term anywhere on the screen. The feature was well received by over 100 study participants. A similar concept by Zhang et al. [2016] even allowed users to define custom stroke patterns to launch applications. However, swiping and flicking have become standard system gestures for navigation and interaction by now. Therefore, the gestures mentioned above cannot be used to access additional shortcuts anymore easily. On another note, the menus themselves require screen space that is large enough for every item to allow reliable touch selection.

While multitouch gestures are intuitive, they are often not feasible in the mobile context.

Another common way to customize interaction is to use multi-finger touch gestures [Hinrichs and Carpendale, 2011]. These gestures are intuitive but not feasible in many situations: Especially users on the go often interact with their smartphones using only the thumb for input [Karlson et al., 2008].

The shape of the finger on the screen or the force it applies adds more properties to touch input.

Touch interfaces usually only use the location of the finger as input. Boring et al. [2012] also used the extent of the finger tracked on the touch digitizer and interpreted it for input mode switches. While the shape of a touch on the digitizer already enlarges when applying more force, one can also use the applied force for mobile interactions. For example, Corsten et al. [2018] used force input and thumb rolling to select values in picker menus. However, force input requires significant learning [Corsten et al., 2019], is difficult to control while moving [Wilson et al., 2011], and

However, force input requires significant learning.

is unavailable on most smartphones. Chen et al. [2014] explored the combination of touch and in-air input. However, additional hardware makes these concepts challenging to use in a real-world scenario.

3.2.2 Tilting Interfaces

In contrast, the hardware required to detect tilting input is built into most smartphones.

As most smartphones include gyroscopes to switch between portrait and landscape views when turning the device, tilting inputs are possible without additional hardware. One of the first systems using tilt input on a handheld device was developed by Rekimoto [1996]. Tilting does not need a display for input, so it can be used on small devices and is especially promising for smartphone UIs with small screen space.

Tilting input is especially suitable for devices with small screen space. In previous studies, pie menus could be controlled well using device tilt.

As device tilt inherently makes the most sense along two axes, Rekimoto [1996] used it to control pie menus. They provide a straightforward mapping between the intended input direction and UI. Similarly, Tian et al. [2008] used pie menus to provide a UI that could be controlled by tilting the stylus on a touch surface. Tilting also allows for commonly used actions even without an explicit UI. Baglioni et al. [2011] successfully used quick back-and-forth tilting inputs on mobile devices in up to nine directions. One use case is eyes-free music playback control, where tilting the device sideways skips songs.

While tilt input is effective in triggering discrete actions, it is outperformed by touch for continuous actions.

Device tilt can be mapped to more than just triggering discrete actions. Oakley and O'Modhrain [2005] and Sad and Poirier [2009] used tilting to scroll through lists. Researchers also created tilt-based text entry systems for small devices like *TiltType* by Partridge et al. [2002] and *TiltText* by Wigdor and Balakrishnan [2003]. However, the use cases of scrolling and text entry are outperformed by touch input on smartphones.

3.2.3 Head Tracking

Head tracking has advantages over gaze tracking when users are moving.

Modern mobile devices support facial tracking without additional hardware. However, especially the tracking of the eyes can become inaccurate when people are moving [Khamis et al., 2018]. Different research projects used head input instead, as the head tilt can be controlled reliably while walking [Crossan et al., 2009].

While head gestures are commonly used in human-to-human communication, they were primarily used for accessibility use cases in HCI.

Head gestures are a common communication method when people interact with each other. For example, nodding and shaking are used to express *yes* or *no* [Kettner and Carpendale, 2013], or more complex messages such as acknowledgment or disinterest [McClave, 2000]. In HCI, head gestures were also explored to be used as input technique [LoPresti et al., 2000], especially for users with limited arm mobility. For example, Craig and Nguyen [2005] attached tilt sensors to the heads of motor-impaired people. The tilting data was then processed on a PDA to control their motorized wheelchairs. Lu et al. [2007] used a less intrusive hardware setup by registering the head tilting via a webcam.

For able-bodied users, head input could allow completely hands-free input.

Still, head movement can also be a useful additional input method for able-bodied users. Mardanbegi et al. [2012] used a head-mounted eye tracker to detect head gestures that allowed users to interact with screens around them. One of their proposed interactions is *iRecipe*, a digital cookbook in which users can switch the steps hands-free while cooking.

Quick back-and-forth movements are the preferred type of head gestures.

Yan et al. [2018] conducted an elicitation study exploring what kind of head gestures could be used to create hands-free input while wearing an HMD device. They found that users preferred head gestures that involved turning the head in one direction and back again, which can be reliably distinguished from normal head movement.

Previous applications of head input include cursor control and changing viewports.

Head tracking has also been used as continuous input to move the cursor on desktop computers. For example, Gorodnichy and Roth [2004] coupled the mouse cursor to the user's nose in the camera feed of the webcam. Jacob

et al. [2016] explored using head control to change the viewport in a 3D application. In their study, the participants solved a docking faster with head controls than with mouse and keyboard.

Head tilt should be mapped directly, not time- or rate-based.

Discrete operations are also promising application areas for head gestures. One example is *HeadTurn* by Nukarinen et al. [2016], which allows users to change numeric values by rotating their heads left or right. In follow-up work, Špakov et al. [2016] compared different mappings of the head tilt to input. Their results indicate that time-based and rate-based controls perform worse than directly mapped inputs in this context.

Head tracking was often used for short inputs, like turning pages, closing a dialogue window, or entering short texts.

HeadPager by Tang et al. [2017] enabled users to turn pages by leaning their heads to the left or the right area, and *HeadNod* by Morency and Darrell [2006] allowed users to quickly answer *yes* or *no* in a dialogue by nodding or shaking their head. Even text entry is feasible to a certain degree via head tracking, as explored by Gizatdinova et al. [2018].

Previous work shows that head tracking can surrogate eye tracking while providing higher accuracy.

Head-mounted devices can use the tracked head position in many ways. For instance, Yi et al. [2016] envision authenticating users based on their head gestures. For object selection, Esteves et al. [2017] show that head tracking can reliably surrogate eye tracking for smooth pursuit selection of moving targets by following their trajectories. According to Kytö et al. [2018], head-based selection is easy to control and more accurate but slower than eye-based selection while using an AR headset.

Research on head input on handheld devices showed that absolute controls perform better than velocity-based ones. Moreover, head tilt is similarly accurate to wrist or device tilt.

Head movements have also been explored on handheld devices. Crossan et al. [2009] explored how accurately users can select a target on a smartphone while walking when using head tilting to control the cursor. They found that absolute cursor control was faster and more accurate than velocity cursor control in a static context but significantly worse while moving. Williamson et al. [2013] used shake sensors and compared head gestures with wrist and device tilting gestures and showed that head gestures have similar accuracy to wrist or device motion gestures. However, they also showed that users felt uncomfortable making head gestures while in a conversation with other people. The aforemen-

tioned mobile device approaches, however, require trackers that are attached to the user to detect head movement. This additional hardware makes these approaches difficult to use in the real world. In our preliminary study in Section 2.7.2, we measured an error of 11.67° in head input without feedback. Therefore, we conclude that the mobile head tracking is accurate enough to discriminate up to 16 discrete directions of head gestures.

In contrast, visual tracking requires no additional hardware.

3.3 Headbang Interaction Technique

Headbang allows you to trigger an action on a specific object, such as sharing a photo on social media, by rotating the head slightly. To do so, the user touches the photo she wants to share (Figure 3.1.a) and then immediately rotates her head slightly away from the screen and back again (Figure 3.1.b). The connected action is then displayed on the screen (Figure 3.1.c), and the user can lift the finger to perform the action (Figure 3.1.d).

Headbang allows to trigger actions on objects in the interface by making a back-and-forth gesture with the head.

Since touch interfaces typically trigger actions upon lifting a finger from an object rather than touching it, the Headbang interaction sequence does not overload the existing interaction concept of handheld touch devices. Thus, it can co-exist with common touch-based interaction techniques such as *tap*, *long press*, *swipe*, or *drag*. As most swipe gestures have become standard system commands so far and multitouch is not feasible in one-handed situations, Headbang adds further actions without relying on any of them. As users only have to move their heads slightly, they can still keep their eyes on the screen and maintain their visual context. Furthermore, the back-and-forth movement can be easily performed and distinguished from normal head movement [Yan et al., 2018].

Headbang can co-exist with common touch-based interactions since it specifies the object of interest on touch-down events.

Slight head rotations can be performed easily while keeping the eyes on the screen.

The head gesture takes place between tapping and releasing the object. This makes it possible to give information about the currently determined action before the user confirms it by releasing the finger. In Figure 3.1, for example, tapping and holding an image and performing the head gesture already shows a popout indicating that *'Share to X'*

Both feedback and feedforward can be integrated into Headbang interactions.

Changing the selection by dragging the finger allows to cancel the Headbang interaction.

has been detected. If this is the action the user wants to perform, she can lift her finger. Alternatively, it is possible to cancel the action by sliding the finger outside the selected image before lifting it, similar to canceling a button press on current mobile operating systems. This provides an easy way to cancel unwanted actions and also enables new users to use this system without the risk of immediately performing unwanted actions. Expert users do not have to wait for the visual indication. Instead, they can touch a target, perform the head gesture, and release the finger from the screen before the head gesture is detected. However, using this faster mode of Headbang prevents users from canceling their actions.

Both spatial and cultural mappings allow for a meaningful arrangement of Headbang actions.

Various natural mappings are promising for Headbang inputs. Cultural mappings can imitate common head gestures: A down movement can be used to confirm a message as it imitates a nod; a sideways gesture could be used for rejections. Spatial mappings can be inspired by the direction of the action or their typical icon representation in the UI. For example, forwarding, replying, sharing, or printing an email with head gestures to the right, left, top, or bottom.

We implemented Headbang with different numbers of directions and evaluated the performance in the two following user studies. Furthermore, we implemented several example applications using this technique and present them in Section 3.6 “Use Cases” (p. 66).

3.4 Study 1: Investigating Tracking Robustness

In Study 1, we investigated the robustness of Headbang in the mobile usage context.

To understand how well our envisioned interaction technique works, we conducted a study in which participants triggered Headbang actions with 4 and 8 directions both while sitting and walking. We recruited 12 people between 19 and 65 years old (M = 36.3, SD = 14.9, 5 female) who participated in this study.

3.4.1 Apparatus and Task

Participants performed Headbang gestures using an iPhone X while sitting and walking an obstacle course.

We used an iPhone X with our tracking software described in Section 2.7.2 "Quantifying Visual Head Tracking" (p. 37). Participants were asked to perform back-and-forth Headbang gestures without feedback while sitting and walking. Similar to the walking condition by Crossan et al. [2009], our participants were instructed to walk a figure-of-eight across a 3×4 m rectangle. We placed obstacles for them to walk around to ensure they still had to pay attention to where they were walking. Each participant performed each gesture twelve times. As we investigated both 4 and 8 directions, this resulted in 288 gestures across all four conditions for each participant. Conditions were assigned in a Latin square design to counterbalance possible learning effects.

We used a counterbalanced within-subjects study design.

Participants had to apply Headbang gestures to buttons at pseudorandom locations in the UI. An arrow specified the direction of the gesture. We did not provide feedback.

Our participants were asked to perform Headbang gestures in the direction of an arrow in the UI. Feedback was deactivated for the study. At the start of each trial, users were shown a button in the bottom third of the screen with an arrow pointing to the direction of the gesture they should perform. The button position varied among four different positions. The different gestures were pseudorandomly distributed over the buttons to mimic a more natural interaction. We made sure that all buttons could be easily reached in one-handed portrait mode. To start a trial, users had to press and hold the button and then perform the Headbang gesture. After performing the gesture, users had to release the button to start the next trial.

3.4.2 Variables

The independent variables were the activity CONTEXT, the number of AREAS and their Headbang gesture DIRECTION.

Since we were mostly interested in how reliable the system could distinguish in which direction the users turned their heads, we used CONTEXT [sitting, walking] and AREAS [four, eight] as **independent variables**. Additionally, we analyzed whether the detection rate differed between the different areas. We used DIRECTION as an additional independent variable for that. In the four area condition the directions were defined by multiples of 90°, i.e. right,

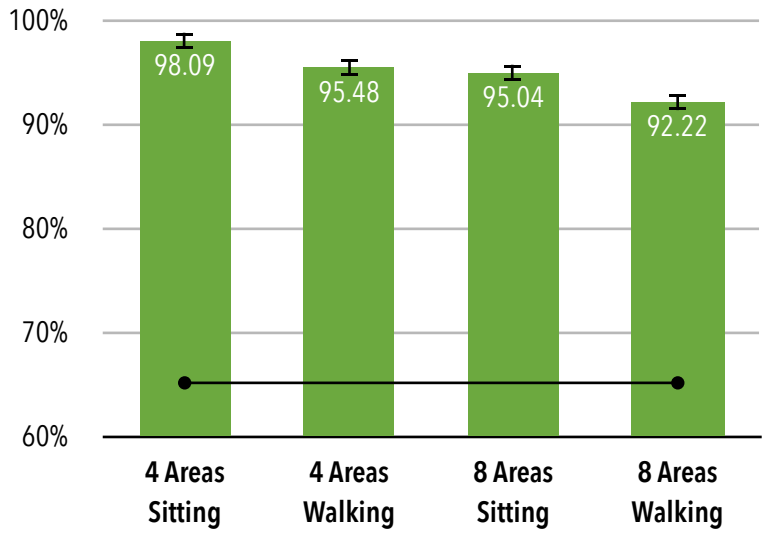


Figure 3.2: *Success* [%] by AREAS × CONTEXT. Whiskers denote 95% CI.

We measured *Success* and *Time* as dependent variables.

bottom, left, and top. We used multiples of 45° in the eight area condition respectively. As **dependent variables**, we measured ***Success*** [0,1] if the system was able to identify the correct area and the task completion ***Time*** [s] for each trial.

3.4.3 Results

On average, Headbang gestures took 1.39 s and were detected with a 95.22% success rate.

The overall success rate was 95.22% (SD = 8.38%) with an average task completion time of 1.39 s (SD = 0.73 s) across all trials and users. For a more detailed analysis, we used McNemar and Cochran's Q tests for the dichotomous *Success* data. We conducted a repeated-measures ANOVA on the log-transformed *Time* data.

Success was higher while sitting than walking.

Success was also better when detecting only four AREAS.

CONTEXT had a significant main effect on ***Success*** ($Q(1) = 5.19$, $p < .023$). The success rate in the sitting condition (96%) was significantly higher than in the walking condition (94%). Also, the AREAS had a significant main effect on ***Success*** ($Q(1) = 6.47$, $p < .011$). The success rate in the four-area condition (96%) was significantly higher than in the eight-area condition (94%). There was also an AREAS

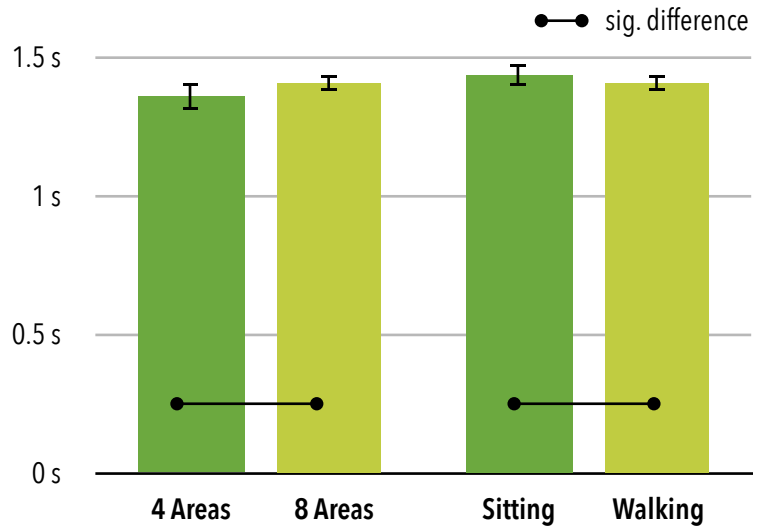


Figure 3.3: *Time* by number of AREAS (left) and by CONTEXT (right). Whiskers denote 95% CI.

× CONTEXT interaction effect ($Q(3) = 13.71$, $p < .003$). Post hoc tests revealed that the four-area sitting condition (97%) had a significantly higher success rate than the eight-area walking condition (94%). Figure 3.2 shows the results for this interaction.

Participants had significantly less *Success* when performing downwards gestures.

In the four-area condition, DIRECTION had a significant main effect on ***Success*** ($Q(3) = 48.303$, $p < .001$). Post hoc tests revealed that the bottom direction had a significantly lower success rate (91%) than the other direction (top: 99%, left: 98%, right: 100%). Also in the the eight-area condition DIRECTION had a significant main effect on ***Success*** ($Q(7) = 74.220$, $p < .001$). Post hoc tests revealed that the bottom-right (45°) direction (86% success rate) had a significantly lower success rate than the other directions.

Both CONTEXT and AREAS had a significant effect on *Time*. The effect sizes were, however, smaller than 70 ms.

CONTEXT had a significant main effect on ***Time*** ($F_{1,3441} = 30.518$, $p < .001$). Users were significantly faster in the four-area condition (1.29 s) than in the eight-area condition (1.35 s). Also AREAS had a significant main effect on ***Time*** ($F_{1,3441} = 31.602$, $p < .001$). Users were significantly faster in the walking condition (1.28 s) than in the sitting condition (1.35 s). Figure 3.3 shows the results of both main effects.

3.4.4 Discussion

Headbang gestures were registered with high accuracy across conditions.

With an overall success rate above 95%, the evaluation confirms that Headbang gestures can be used reliably for input. However, we found significant differences in the success rate between the conditions. The higher success rates in the four area conditions had to be expected, as the cone of a single section is twice as large as in the eight-area condition. Likewise, the difference in success between the sitting and walking conditions is no surprise. In the sitting condition, both the device and the user are static, which allows the system to track the user's head more accurately. Nonetheless, the effect sizes of both are negligible on their own. Only when combining multiple elements with walking did the accuracy drop significantly to 92%.

As our participants already pointed their heads down to the phone while using it, it was hard for them to trigger downward Headbang gestures.

The bottom direction's slightly lower success rate in the four-area condition can be explained by how the users hold and look at the device. We observe that users hold their smartphones not directly in front of their heads but much lower to maintain a comfortable arm position. This means that the users already rotate their heads downward to look at the content displayed on the smartphone. An even further downward rotation of the head to select the bottom area could be limited due to the neck muscles or discomfort for the users. Therefore, participants overshoot to the top when performing a back-and-forth gesture. A similar effect, however not significant, also appears in the eight-area condition. Here, all three bottom directions have a lower success rate than the other areas. However, only the lower right direction differed significantly from the other areas. This could be because most of our participants (10 out of 12) were right-handed and, therefore, held the device on the right side. In this case, participants had to move their heads further to the right to select the lower bottom direction.

The study also showed significant differences in task completion times between the conditions. While walking, participants were around 30 ms faster than while sitting. This could be because they shifted their attention away from the phone earlier to avoid tripping over an obstacle while walk-

ing. However, this difference is minimal and probably not noticeable in a real-world use case.

3.5 Study 2: Using Headbang to Trigger Actions in Menus

While an on-screen context menu adds feedforward to the Headbang interaction, the head gestures' spatial intrinsics help to recall menu entry locations. Thus, Headbang is a good match for radial context menus.

As we have seen, Headbang can reliably be used to trigger actions on items of interest. However, we did not include any visual feedforward that presents the user with all available actions in Study 1. Due to the limited screen space on mobile devices, actions are often selected from context menus. The size of these menus can become quite large. On iOS 17, for instance, we can find context menus with 13 items in the *Files* app, nine items in the *Mail* app, or 11 items in the *Music* app. In touch interaction, tapping and swiping are already occupied, so holding an item of interest for a specific duration is required to bring up the menu. As users can quickly recall spatial positions [Scarr et al., 2013], Headbang seems to be a promising modality for menu items. Thus, we wanted to compare Headbang with touch and tilting input for menus.

The headbang menu is a 3 cm wide circular menu that saves valuable screen space compared to conventional list-style menus.

Headbang menu. Our Headbang menu is a pie menu whose selected item is controlled by tilting the head. Upon putting a finger on an item on the screen, the camera system activates. When tilting the head slightly, i.e., by approximately 10°, a pie menu appears and the item corresponding to the current angle is highlighted. The selection is changed by rotating the head and confirmed by lifting the finger. With a diameter of 3 cm the pie menu offers a compact menu visualization that takes less screen space than the list menu at the cost of omitting labels. However, labels will fit into the menu when there are at most seven menu items. For more extensive menus, labels can be placed inside the pie by increasing its diameter. However, they are still smaller than a complete list and occupy around 25% of the phone screen.

Figure 3.4: Pie menu segments were equally sized and oriented so that four segments were exactly aligned with the horizontal and vertical axis, i.e., the 90° steps.

In Study 2, we compared Headbang for context menu selections with device tilt and touch input.

For a fair comparison in our evaluation, we also implemented a pie menu for device tilt and touch input. The device tilt condition behaves exactly like the Headbang menu and shows up when holding an item and tilting the device. As recommended by Teather and MacKenzie [2014], we used the absolute device rotation for input. In the touch condition, the menu appeared after a long press of 400 ms, which is 20% faster than the iOS default. The selection in the menu was then selected by swiping in the direction of the item and confirmed by lifting the finger inside or outside of the menu.

As the default context menu style on mobile OS is a touch-controlled list menu, we added this style as a baseline condition.

As a baseline, we implemented a list-style menu with touch targets that are 28×7 mm large, adopting the same size as the system menus in iOS. While users often have to scroll through the menu in real-world applications, we made sure that all items were always visible on screen, limiting the maximum number of items in this condition to 16. We used a new set of twelve participants in this study aged between 20 and 31 years (M = 25.5, SD = 3.34, 3 female).

3.5.1 Apparatus and Task

We used the same hardware as in the previous study.

For the study, we continued to use the iPhone X from the previous study but extended our implementation with the different menu techniques as described above.

The interface presented the participant with an emoji and a red box with the size of an app icon (1 cm^2) in whose context menu the depicted emoji had to be selected. Moreover, an arrow next to the emoji pointed to where it would appear in the menu once it became visible. This hint was provided to mitigate the search time during the interaction and to consider that users create muscle memory for actions they perform frequently. We asked our participants to select the menu items as quickly as possible without compromising accuracy while using only one hand to operate the phone.

The study interface showed participants an emoji and directional hint. They had to select this emoji in a context menu of a target as large as an app icon.

While all presented techniques work with an arbitrary number of items, we picked three different menu sizes with 8, 12, and 16 elements for evaluation. We found 12 items to be a reasonable number from our observation of system menus and due to its benefit of a mapping known from the hour marks of a clock. For each menu size, we selected 8 different items at the representative angles 35°, 80°, 120°, 167°, 210°, 260°, 305°, and 350° (see Figure 2.8). With three repetitions we measured a total of 288 selections from each participant (8 items × 3 menu sizes × 3 repetitions × 4 menu types). We used Latin squares for both the menu size and input conditions to counterbalance possible learning effects. Participants tested all four input conditions with the same menu size before switching to another one. The new menu size then had a new order of input conditions.

We used a counterbalanced within-subjects study design.

Participants ranked their preference between the four techniques for each menu size before switching to the next study phase. Further questions were filled out at the end. Participants were allowed to test the input techniques before each trial. There was no monetary compensation for participation.

Participants filled out post hoc questionnaires during the study.

3.5.2 Variables

As we conducted the study to find out whether Headbang can be used as a reliable modality for selecting menu actions, we used the INPUT TECHNIQUE [Headbang, Device Tilt, Touch Pie, Touch List] and the MENU SIZE [8, 12, 16] as

The independent variables were four INPUT TECHNIQUES and three MENU SIZES.

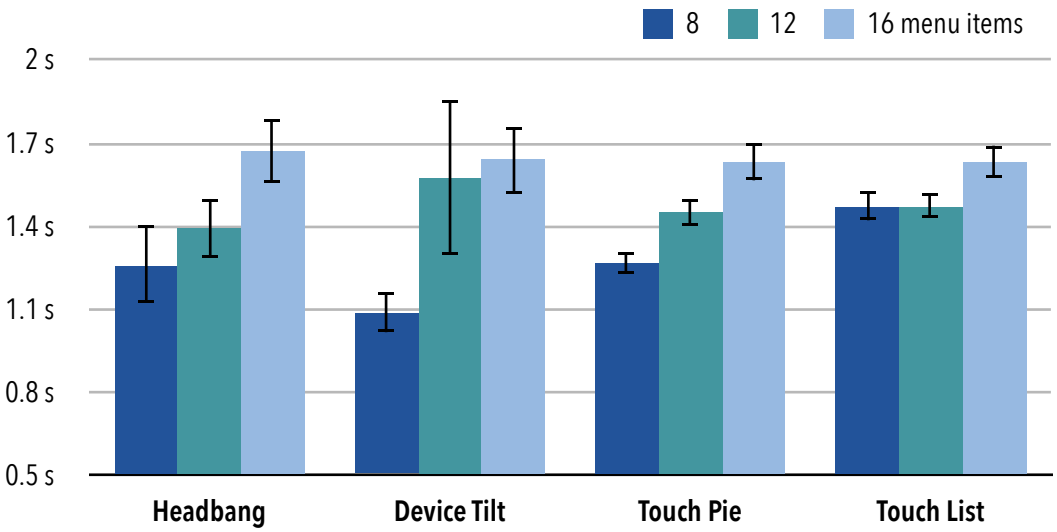


Figure 3.5: *Time* [s] by INPUT TECHNIQUE × MENU SIZE. On average, it took our participants 1.5 s to open the menu and select an item. Conditions using pie menus were operated faster. Whiskers denote 95% CI.

independent variables. As **dependent variables**, we measured ***Success*** [0,1] if the correct item was selected, and the task completion ***Time*** [s] for each trial.

We measured the *Time* and *Success*, as well as information on the rotary *Offsets* when initiating and committing the menu selection.

We also measured the angles obtained from the head or device tilting when initiating the gesture, i.e., before feedback, and on selection, i.e., when feedback was visible. When calculating the offset to the angle representing the center of the target menu segment, this results in ***A Priori Offset*** [°] and ***Post Hoc Offset*** [°] respectively. From the questionnaires, we obtained a forced ***Preference*** ranking [1–4].

3.5.3 Results

The INPUT TECHNIQUE had no significant effect on the speed of the interaction.

We used repeated-measures ANOVAs to evaluate our measurements. In this study, we were most interested in the participants' performance depending on the INPUT TECHNIQUE used. However, we were not able to find a significant effect of INPUT TECHNIQUE on ***Time*** ($F_{3,2833} = 1.755$, $p = .154$).

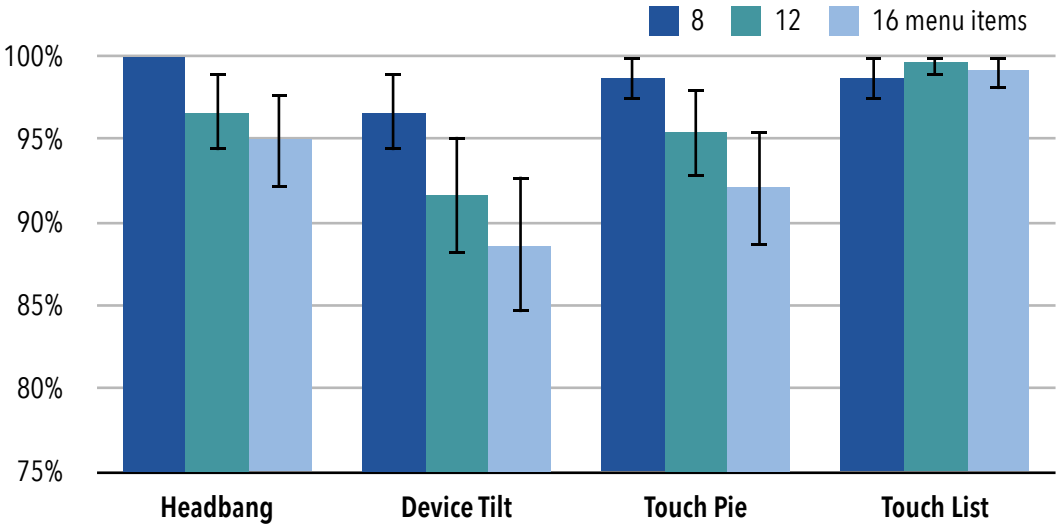


Figure 3.6: *Success Rate* [%] by INPUT TECHNIQUE × MENU SIZE. When using pie menus, the success rate decreased with larger menu sizes. We did not observe this effect in the list menu. Whiskers denote 95% CI.

MENU SIZE, on the other hand, had a significant main effect on the log-transformed ***Time*** ($F_{2,2833} = 43.856$, $p < .001$). Tukey HSD post hoc pairwise comparisons were all significant. Menus containing 8 items were the fastest (1.27 s). On average, menus with 12 items were 15% slower (1.47 s), and 16 items were 29% slower (1.65 s) than menus with 8 items.

On average, doubling the MENU SIZE resulted in 29% slower input times.

There was also a MENU SIZE × INPUT TECHNIQUE interaction effect on ***Time*** ($F_{6,2833} = 3.901$, $p < .001$). Again, we used Tukey HSD post hoc pairwise comparisons for further analysis. On average, Device Tilt with 8 menu items was the fastest condition (1.09 s), and it was significantly faster than all other conditions except for Headbang with 8 menu items and Touch Pie with 8 menu items. Larger menu sizes took longer to operate, ranging from 1.64 s (Touch Pie) to 1.67 s (Headbang). However, Headbang with 16 menu items was only significantly slower than Device Tilt with 8 menu items and Touch Pie with 8 items. When comparing Headbang with the Touch List there were no significant differences independently of the menu size. Figure 3.5 shows the measured times for all conditions.

Looking at the interaction effect, the Headbang menu with 16 elements was only significantly slower than Device Tilt and Touch Pie with half as many menu elements.

For the analysis of the ***Success*** values, we calculated the ***Success rate*** as the share of successful trials per condition and user. We then conducted a repeated-measures ANOVA on the calculated success rates.

Across conditions, doubling the MENU SIZE significantly lowered the success rate.

MENU SIZE had a significant main effect on the ***Success Rate*** ($F_{2,99} = 9.004$, $p < .001$). Tukey HSD post hoc pairwise comparisons revealed that the success rate significantly drops from 98.5% to 94.0% when the number of menu items is doubled from 8 to 16.

Our participants were significantly more accurate using Headbang than Device Tilt.

INPUT TECHNIQUE also had a significant main effect on ***Success Rate*** ($F_{3,99} = 7.997$, $p < .001$). Tukey HSD post hoc pairwise comparisons show that Device Tilt (92.2%) was significantly worse than Headbang (97.5%) and Touch List (99.2%). Furthermore, the Touch List also had a significantly higher success rate than Touch Pie (95.4%). We did not find a significant MENU SIZE × INPUT TECHNIQUE interaction effect on ***Success Rate*** ($F_{6,99} = 1.227$, $p = .299$).

3.5.4 Discussion

Headbang provided both a quick and reliable input modality for context menus.

Overall, we measured similar times to activate (with an initial tilt or a long press) and select an item across all techniques. The slowdown in interaction with growing menu size in any condition is no surprise, as the target segments become smaller with more elements in the pie menu. Likewise, a larger distance must be traveled with the thumb in a list menu. Notably, apart from Device Tilt, all input techniques delivered high success rates, making them feasible to use on mobile devices.

Our participants found Headbang similarly unconventional to Device Tilt, but not awkward.

In addition to the similar performance, the rankings of our participants were unsettled, too: The Likert scale data from the questionnaires, including preference, comfort, and easiness, yielded similar ratings across all conditions, with no significant effects found by using a Friedman test. In contrast to the findings of Williamson et al. [2013], our participants did not perceive the head-controlled input as awkward but rather similarly unconventional to device tilt.

This might result from Headbang requiring only subtle head rotations for input.

Headbang triggers vertical targets faster than horizontal ones.

When using Headbang, items at the bottom and top of the pie menu, which are selected with nodding gestures, were selected around 400 ms faster than items at the left and right, i.e., shaking gesture items. We presume this originates in users looking at the phone downwards and thus already performing a rotary pitch in their resting position.

Our participants found that Headbang requires less homing than Device Tilt as the head rests in a more neutral position.

The qualitative feedback we received from our participants on what they liked and disliked about the techniques was similar, too. Four participants enjoyed that Headbang requires less homing than Device Tilt, as the head is typically in a more neutral position when initiating the gesture than the wrist. However, three participants also stated that they found reaching the upper targets with Headbang less comfortable than with Device Tilt. Five participants perceived 16 items as too many with all input conditions.

Interestingly, pie menus decreased performance when controlled with touch only.

The low success rate measured in the Touch Pie was surprising, as it was possible to swipe the finger out of the menu, achieving large target sizes that should have been used reliably. In conclusion, we recommend using the Touch List over the Touch Pie for touch-only systems.

The use of feedback noticeably increased the precision of the Headbang interaction.

While the effect of feedback was noticeable, the study further supports that Headbang can be used without feedback. The average *Post Hoc Offset* we measured across all trials was 6.59° (SD = 5.08°). The average *A Priori Offset* was four times as large (20.49°). This matches our findings from Study 1, as this corroborates with eight possible actions in the menu without feedback.

Even without training, our participants made selections faster with Headbang than with Touch List in small and medium-sized menus.

In conclusion, the cost of implementing Headbang as menu technique is quite low with its high accuracy and while not being slower than touch. One advantage of Headbang over the Touch List is the reduced screen space needed, thus decreasing occluded content. Moreover, all participants were familiar with touchscreen menus but not with head input. Even though our participants were untrained and unfamiliar with the interaction, Headbang was already slightly

faster in small and medium-sized menus than Touch List. We expect users to become even faster with training.

Headbang seems promising when screen space is scarce, or users cannot freely move their hands.

Therefore, Headbang is a helpful menu selection technique in scenarios where users cannot reach the whole screen, e.g., one-handed smartphone and tablet use or in-car controls operated while driving. It allows for compact menus with many options that preserve the context, which is favorable for apps such as drawing and image editing.

3.6 Use Cases

Headbang is an interaction technique that can be used in various application domains and use cases. To underscore its utility, we envisioned and developed several fascinating use cases and applications.

Headbang could speed up content sharing by leveraging spatial recall when specifying the target application.

Content Sharing. Sharing digital content between different applications is a common task on a smartphone. An example of this is to share an image from the image library with a social media application such as Twitter or Instagram. To do so, users typically have to first select the image, click the share button, and then select the app to which the image should be shared. With Headbang, users select the image and perform a Headbang gesture to directly share the image with the designated application (Figure 3.1).

In the context of text editing, Headbang could be used to format the marked text quickly.

Text Editing. We also developed a simple text editor, shown in Figure 3.7, that enables the user to use Headbang to trigger actions on selected text or the current text cursor location. For example, to copy selected text, she performs the gesture to the top; to replace the text with another text, she performs the gesture to the bottom. She can also make the text bold, underlined, or italics via Headbang gestures in different directions.

Figure 3.7: In this example, the user selected a part of the text that he wants to cut. After selecting the text with his finger (left), he performs a Handbang gesture to the left (middle) to cut the text (right).

Pasteboard Management. Headbang gestures can also be used for a multi-level pasteboard where users can store digital objects by selecting them and performing a Headbang gesture in the direction in which the object should be stored. This approach allows users to use their spatial memory to retrieve in which direction they stored which object. To retrieve the item, users repeat the same Headbang gesture while placing a finger on an add button or an area where the object should be placed.

Headbang could make multi-level pasteboards more accessible.

Objects could be copied into and pasted from different memory slots around the head.

Accessibility Features. Another use case for the Headbang interaction technique is to use it as an accessibility feature for users with tremors who have difficulties typing on an on-screen keyboard [Wacharamanotham et al., 2011]. For this use case, we developed an on-screen keyboard with three large buttons that are much easier to select than the typical keyboard buttons (Figure 3.8). Each button encodes nine characters: eight for the Headbang gesture and one (the middle) for touching and releasing the button without performing a gesture. This can be used to type letters by selecting one of the buttons and then performing the Headbang gesture in the direction of the designated letter.

Headbang could support tremor patients by increasing touch target sizes and making selections of actions using the head.

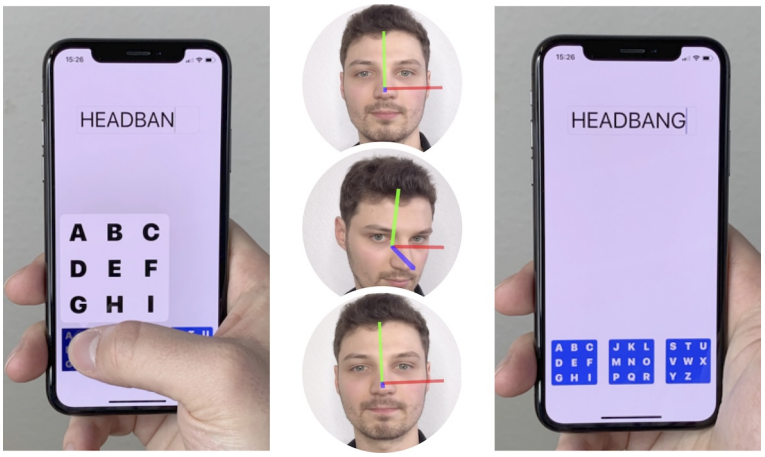


Figure 3.8: In this example, the user uses an accessibility keyboard for users with tremors who have difficulties selecting small buttons. To type the letter G, the user touches the left button (left) and then performs a Headbang gesture to the lower left (middle). After lifting the finger from the button, the selection is confirmed (right).

Headbang could also be used to trigger global actions without specifying an object of interest.

Hands-free Controls. We also envision hands-free use cases in which the users perform Headbang gestures to trigger actions that are not applied to a currently selected object but rather global actions. A simple example of that is using Headbang gestures to turn a page in a digital cookbook while preparing a meal. To execute this gesture, the user has to look directly at the device and then execute the Headbang gesture. The same approach can be used for a variety of different use cases in which the users would like to interact with the device hands-free.

3.7 Future Work

An evaluation with tremor patients could further refine Headbang and how it handles accidental activation.

In future work, we want to explore the Headbang interaction technique in more detail and investigate its use as an accessibility feature based on the already presented use cases. An evaluation with users who have tremors will help us understand how the Headbang interaction technique can be further improved. Like most user interfaces,

Headbang can also suffer from accidental activation. While it is already possible to discard the menu by swiping to prevent accidental selections, mechanisms to prevent accidental activation remain a relevant area of further work.

Always-on hands-free control using Headbang requires investigating the ideal selection method.

Moreover, our implementation of Headbang is currently activated by touch input, which also limits the camera system's power consumption. The opportunities Headbang offers to accomplish completely hands-free system navigation still need to be uncovered. We want to investigate different selection methods for a hands-free interaction, including dwell-time-based head resting, blinking, other facial gestures, and voice input. These approaches have their own caveats, including battery consumption, user acceptance, and accuracy issues. We also want to explore the social acceptance of interacting with a smartphone using head gestures in public spaces further, although our participants had no bias against this input modality.

Uncertainties with the social acceptance of head-based inputs remain.

3.8 Conclusion

We designed and evaluated a new interaction technique, called Headbang, that lets users perform actions typically found in context menus by performing slight back-and-forth head movements.

With Headbang, we presented an interaction technique that increases the expressiveness of touch input by using head gestures as an additional input channel. Our studies showed that a commodity smartphone can reliably detect the Headbang technique while the users are sitting or walking with a success rate of over 95%. We have seen that the number of menu items influences the execution time of a head gesture, whether the user is walking or standing, and the target location. In our studies, Headbang was not slower than touch input, so its use is low-cost while offering the advantage of needing less screen space and enhancing one-handed use. This enables Headbang gestures to be used in various everyday tasks to select elements from menus, perform shortcuts without visual feedforward, or as an accessibility feature for users with difficulties selecting small targets on a touchscreen.

Headbang used head tracking to make discrete selections in a one-dimensional list of choices. Next, we will map the head input into the two-dimensional space.

We made a foray into visual facial tracking as an input source with Headbang. The input domain of Headbang, however, remained relatively small: We discretized head tracking data into a selection from a one-dimensional list of actions. As our evaluations of Headbang proved suitable precision of the head input, we were motivated to increase the size of the input domain with two-dimensional controls. In the next chapter, we will directly link a cursor on the screen to the user's head as a reachability technique for any object.

Chapter 4

Solving Reachability Issues on Large Phones with Head Control

SUMMARY:
People often operate their smartphones with only one hand, using just their thumb for touch input. With today's larger smartphones, this leads to a reachability issue: Users can no longer comfortably touch everywhere on the screen without changing their grip. We investigate using head tracking in modern smartphones to address this reachability issue. We developed three interaction techniques, *Pure Head (PH)*, *Head + Touch (HT)*, and *Head Area + Touch (HA)*, to select targets beyond the reach of one's thumb. In two user studies, we found that selecting targets using HT and HA had higher success rates than the default direct touch (DT) while standing (by about 9%) and walking (by about 12%) while being moderately slower. HT and HA were also faster than one of the best touch-based techniques, BezelCursor (BC) (by about 20% while standing and 6% while walking), with the same success rate.

Publications: The work presented in this chapter was done in collaboration with Simon Voelker, Christian Corsten, and Christian Remy. The author of this thesis developed the research idea and relevant research questions with his co-authors. Furthermore, he designed and implemented the presented head-based interaction techniques and study software. Most of this work has been published as a paper in the Proceedings of ACM CHI 2020 [Voelker et al., 2020]. The author of this thesis is one of the principal authors of the paper. Most sections in this chapter are taken from the paper publication.

4.1 Motivation

The increasing display size of smartphones amplifies reachability issues during one-handed usage.

Ever since their introduction, our smartphones have become bigger and bigger. For reference, the first generation iPhone from 2007 had a 3.5″ screen, and the 2024 Samsung Galaxy lineup contains screens ranging from 6.2″ to 6.8″. Consequently, today's smartphone screens are nearly four times as big as those of 2007. While these larger screens allow for displaying more content, users often interact with their smartphones using just one hand [Boring et al., 2012; Kim et al., 2012], using only the thumb for input [Karlson and Bederson, 2008]. This introduces reachability issues, as users cannot reach all parts of the screen comfortably anymore without having to re-grasp the device [Corsten et al., 2019].

Existing reachability techniques introduce cursors or apply screen transformations. The latter results in a reduction of screen real estate.

Several techniques have been proposed to address this problem: BezelCursor by Li and Fu [2013], ForceRay by Corsten et al. [2019], and MagStick by Roudaut et al. [2008] create a cursor that is activated via touch to select targets beyond the reach of the user's thumb. Other techniques introduce mode changes to transform the on-screen content to make it reachable. For instance, Kim et al. [2012] shifted the interface to the lower half of the display, and Chang et al. [2014] completely resized it closer to the thumb. While these approaches address the reachability problem, they require explicit mode switching or reducing the screen real estate.

Facial tracking adds expressiveness to the touch input without occluding on-screen content.

The inputs of facial tracking could address the issues mentioned above by increasing the communication bandwidth from user to smartphone. Early explorations of head and gaze tracking features built into smartphones have begun to uncover this potential, creating interactions such as browsing through photo albums using gaze tracking [Zhang et al., 2013] or unlocking the phone using eye gestures [Khamis et al., 2017]. Using the head as input has the benefits that users can reach the entire screen just by rotating their head and that the head control does not occlude content on the screen.

In this chapter, we explore how head tracking can address the reachability problem on smartphones. We designed interaction techniques that use head tracking to select objects on a smartphone touchscreen in three different ways, which are also the conditions of our evaluation: *Pure Head*, *Head + Touch*, and *Head Area + Touch*.

We designed and evaluated three interaction techniques that use head tracking to select objects on the screen.

We use only the head for target selection in the *Pure Head* condition. *Head + Touch* combines head tracking and touch by letting the user adjust the head selection with a brief touch gesture. *Head Area + Touch* selects a quadrant of the screen via head tracking, after which the target is selected via indirect touch. We compare these and two baseline conditions, *Direct Touch* and *BezelCursor*.

Pure Head selects the target only using head input. Our other two techniques allow for the selection to be refined via touch.

Since our head tracking techniques are designed especially for one-handed smartphone use, we evaluated the five conditions with participants (n=15) standing rather than sitting. That is because single-handed smartphone use is more likely while standing than sitting in practice. Our results show that *Head + Touch* and *Head Area + Touch* selections while standing were only 5% and 7% slower than *Direct Touch*, but have a 9% respectively 8% higher success rate. For added realism and ecological validity of our findings, we also investigated the viability of all five techniques while walking [Wilson et al., 2011] with ten additional participants. Here, both *Head + Touch* and *Head Area + Touch* selection were 24% and 25% slower than *Direct Touch*, but had a significant higher success rate (97% and 94% vs. 82%). We discuss our findings and the usefulness of the various techniques in different application contexts and provide recommendations for developing mobile input techniques on larger mobile devices that rely on one-handed interaction.

We found that these head-based reachability techniques resulted in higher success rates than *Direct Touch* both when our study participants were standing and walking.

The main contributions in this chapter are the design and the evaluation of *Head + Touch* and *Head Area + Touch* selection. Both new input techniques address the reachability issue in one-handed smartphone use by combining head tracking and touch input with promising performance.

Our *Head + Touch* and *Head Area + Touch* techniques enable accurate one-handed target selections.

4.2 Related Work

Our goal of this work was to explore the use of facial tracking to extend thumb reach on smartphone touchscreens. Thus, this section discusses related work in reachability techniques. It also highlights previous use cases of head tracking to provide some background for our implementation.

4.2.1 Reachability Techniques

The model of Bergstrom-Lehtovirta and Oulasvirta calculates the small parabolic area of the smartphone that the thumb can reach.

Bergstrom-Lehtovirta and Oulasvirta [2014] developed a model that predicts which areas of the phone screen a user can comfortably reach with her thumb depending on the hand size and orientation. When the thumb is opposed to the palm, the possible thumb movements limit the reachable area to a parabolic shape. Hence, the simplest way to solve the reachability problem would be to constrain input GUI elements of a smartphone to the region within comfortable reach of the user's thumb. However, this approach restricts the space in which interactive objects can be placed to the lower area of the screen, ignoring the remaining space. Instead, reachability techniques provide manipulations that transiently allow specifying touch targets outside the thumb's reach or moving them to the thumb tip.

The design space of reachability techniques differentiates between screen transform, proxy region, and cursor techniques.

A design space to classify reachability techniques was developed by Chang et al. [2015]. This design space classifies techniques based on two dimensions: Their trigger and selection mechanisms. Some techniques use explicit mode switching gestures, while others are always active or only activate when dragging the finger over the screen. To help select a target, some techniques apply a *screen transform*, while others provide a *proxy region* or a *cursor*. For our discussion of related work, we follow this taxonomy and present techniques using each targeting mechanism.

Screen Transformation Techniques

UI transformations result in contents that are either hidden or hard to read because of their reduced size.

Screen transformation techniques can be found in the stock user interfaces of Apple's and Samsung's recent smartphones. On an iPhone, swiping down across the bottom edge of the screen slides the screen downwards so that the user's thumb can reach the upper targets. However, this still leaves targets on the far side opposite the thumb unreachable, and context information is lost. On many Android devices, such as Samsung and Asus smartphones, triple-tapping the home button scales down the entire screen to the lower corner near the thumb. This brings all targets into reach but impedes readability and targeting because of reduced content size.

Screen transformations of the whole UI were triggered in different HCI prototype systems by tilting and sliding gestures on the front and back of the device.

In HCI research, several projects have proposed alternative screen transformation techniques to address the reachability problem: Similarly to Android devices, *TiltReduction* by Chang et al. [2015] scales down the interface when the user tilts the device. *Sliding Screen* by Kim et al. [2012] moves the screen diagonally closer to the thumb by a swiping gesture. Alternatively, it is also feasible to use device tilt to trigger this reachability method, as explored by Chang et al. [2015]. Tsai et al. [2016] used a swiping gesture along the screen edge to activate the reachability technique, thus giving the user more control over how far the screen should move toward the thumb. Le et al. [2016] allowed users to shift the screen contents downwards by sliding their index finger across a touchpad at the back of the device to trigger this transformation.

Less frequently, researchers applied transformations to only specific UI parts, e.g., just the keyboard.

Instead of manipulating the whole screen, some techniques are applied only to parts of the user interface. For instance, Eardley et al. [2017] observed that users tilt the phone toward their thumb while aiming at targets that cannot be comfortably reached with the thumb. This motivated them to use device tilt as an activation method to shift, among others, the keyboard to one side of the screen. Future interfaces could also adapt to the user's handedness and shift the UI toward that hand. In this context, Löchtefeld et al. [2015] showed that one can detect usage handed-

ness from the finger movements made while unlocking the phone.

Disadvantages of these techniques include concealing information and, in the case of tilting, also looking at the screen at an angle.

However, these interaction techniques hide parts of the digital content, conceal context information, or scale down the interface, making objects challenging to read and select. Some of them need additional hardware, and others use tilting. Tilting, however, comes with the caveat that reading content on screen at an angle is difficult. Moreover, according to Spelmezan et al. [2013], tilting is also prone to overshooting.

Proxy Region Techniques

Using a proxy of the entire display has the drawback of small touch targets, which are difficult to hit.

ThumbSpace is a reachability technique for PDAs by Karlson and Bederson [2007]. This technique creates a pop-up view around the thumb's touch location that represents the entire screen. While this offers a straightforward mapping where each corner of the pop-up also corresponds to one corner of the PDA's screen, it also means that the already small touch targets in the UI become even smaller and more difficult to hit. As the technique highlights the selected UI element on touchdown, users can drag to adjust their selection and confirm it by releasing the finger.

Using the *TapTap* technique the whole screen serves as a proxy. Thus, selections now require the two inputs of coarse and fine selection.

TapTap by Roudaut et al. [2008] allows disambiguating previously performed touch input in a pop-up view. With this technique, each input requires two taps. The first tap only coarsely specifies the intended touch area. A pop-up view then shows the magnified screen area around this tap. To make the selection, a user taps again inside the pop-up. A clear drawback of this technique is that it doubles the amount of inputs required to control the interface.

Back-of-device proxies exploit that the index finger rests behind the screen corner that the thumb cannot reach.

Yoo et al. [2015] used back-of-device touch input with the index finger in addition to the thumb's direct touch. When using the phone in the right hand, the index finger typically rests behind the upper left screen corner. Exactly this area is hard to reach with the thumb. While such techniques require additional sensing hardware, they can increase the thumb's reach by 15%.

Hasan et al. [2016] developed an approach that uses the mid-air space above the touchscreen as a proxy region or to control a virtual joystick. However, this method requires an external tracking system, and mapping an imaginary curved proxy surface to the planar screen under it is perspectively unintuitive.

Mid-air proxy regions suffer from an unintuitive mapping.

Cursor Techniques

Cursor techniques provide a digital cursor that reaches targets outside of their thumb's reach. Li and Fu [2013] presented *BezelCursor*, an accelerated cursor that is activated by swiping from the bezel of the device and controlled by continuous dragging. In recent versions of mobile operating systems, however, flicks starting at the screen edge trigger many system-wide actions. There exist further similar interaction techniques with different activation gestures: For example, *ExtendedThumb* by Lai and Zhang [2015] is activated by double tapping; *TiltCursor* by Chang et al. [2015] activates from tilting the device.

Cursors can reach any element in the interface, but system gestures already occupy many of the activation gestures used in research prototypes.

Corsten et al. [2019] explored using force touch to aim at out-of-reach targets. With *ForceRay*, a cursor moves along a ray that crosses the touch location and the closest screen corner. Applying more pressure moves the cursor further away from the touch location. As small touch movements can alter the ray, the thumb can remain at a comfortable location while using this technique. However, as presented by Wilson et al. [2011], force input with an absolute mapping is imprecise and difficult to use while walking.

The inclusion of force input can solve this activation problem but lacks precision while walking.

As the cursors are controlled by thumb movement, this can also lead to occlusion problems. Both *MagStick* by Roudaut et al. [2008] and *Extendible Cursor* by Kim et al. [2012] address this issue by steering the cursor in the opposite direction of the thumb movement. Instead of using one larger dragging gesture, which could also require thumb stretching, *2D-Dragger* by Su et al. [2016] lets users step through objects with small dragging operations. Yet, a technique like this becomes tedious with many potential targets on screen.

Some cursor techniques steer the cursor in the opposing direction of the thumb or automatically snap to possible targets.

Techniques that simplify the reachability of targets at the screen edge come at the cost of less accurate selections at the center of the screen.

Yu et al. [2013] presented two reachability techniques that focus on the outermost targets on screen: *BezelSpace* lets users reach targets at the screen's edge using a cursor controlled by small thumb movements, and *CornerSpace* places a remote cursor at the corners of the screen to access them quickly. However, both these techniques make the selection of objects outside the corners or edges of the screen less accurate.

Cursors that are controller by back-of-device touch input require additional hardware.

Another way of avoiding occlusion while controlling a cursor could be back-of-device input, which was explored by Yang et al. [2009] and Löchtefeld et al. [2013]. With these techniques, users move their index finger over a touch surface on the back of the device to control the cursor location. However, these techniques require additional hardware.

Head input could overcome the drawbacks of the different existing reachability techniques.

While many of these techniques successfully address some of the problems such as occlusion and offer viable alternatives for cursor selection, they introduce drawbacks. Such drawbacks include a decreased success rate, discomfort for the thumb, or fatigue. We seek to explore whether head input can address the reachability problem while avoiding some of these issues, also in combination with other techniques.

4.2.2 Head Input on Mobile Devices

Previous use cases of head input cover accessibility, games, and gestures-based actions.

The related work of *Headbang* presented in Section 3.2.3 already covered different uses of head tilting. In a nutshell, previous work used head tilt as an accessibility technique, in games, and to operate gesture-based controls. For accessibility purposes, users with limited arm mobility can make inputs via head movement, e.g., to steer the power wheelchairs presented by Craig and Nguyen [2005] and Lu et al. [2007]. In addition, the work of Gorodnichy and Roth [2004] evaluated using head movement to control the cursor on a desktop computer. Head movement as input in games was explored with continuous control and detection of facial expressions by Ilves et al. [2014]. In contrast, Williamson et al. [2013] used discrete actions in their game. Noteworthy, however, is that their study participants felt

uncomfortable performing head gestures during conversations with other people. Examples of UI controls that react to head movements are steppers for numeric values like in the work of Nukarinen et al. [2016] and turning pages like in the work of Tang et al. [2017].

Absolute vs. velocity cursor control using head input perform differently when moving and when stationary.

The already in Section 3.2.3 presented work by Crossan et al. [2009] showed that head tilting can be used to control a 1D-cursor on a smartphone while walking. They found that absolute cursor control, compared to velocity cursor control, was faster and more accurate when stationary but significantly poorer when users moved.

Using the tracking capabilities of modern smartphones allows the evaluation of input techniques in more natural settings.

The presented previous work required the use of additional hardware. We, however, wanted to use the tracking functionalities of recent smartphones to address the reachability problem. One advantage of this reduced hardware complexity is that the system can be better evaluated outside the seated lab context, as people also tend to use their smartphones while walking.

4.3 Head Reaching Techniques

We designed three reachability techniques that use head tracking. This means they do not face the accuracy of issues of gaze tracking in mobile scenarios.

We designed three head tracking techniques for reaching elements on the smartphone screen outside of thumb reach in three different setups: *Pure Head* tracking, *Head + Touch*, and *Head Area + Touch*. While mobile gaze tracking technology has improved significantly and a plethora of contributions has highlighted its potential for use in mobile settings [Dalton et al., 2015; Franchak et al., 2011; Höller et al., 2009; Khamis et al., 2017], one shall not overrate gaze accuracy, especially in the mobile context. Even when using dedicated hardware, gaze tracking accuracy significantly deteriorates when the user's face moves [Niehorster et al., 2020]. Therefore, we intentionally decided to employ head tracking exclusively instead of gaze tracking.

Several caveats and disadvantages significantly constrain the real-world applicability of gaze tracking in mobile scenarios. Most of the applications of gaze above utilize head-mounted eye-tracking hardware, limiting the feasi-

Especially walking causes reliability issues with eye tracking.

bility of using them on the go. While eye tracking has become available on mobile devices, applications of eye tracking for target selection experience have severe limitations outside controlled lab settings [Lappi, 2015]. Even small body movements such as head re-positioning interfere with eye tracking, rendering its results unstable and unreliable as soon as the head is not in a fixed position anymore [Lappi, 2015]. More active movement, such as walking, further complicates eye tracking as gaze and foot are connected [Matthis et al., 2018]. The problem of the impact of walking on eye and head tracking has been the subject of investigation for several decades, and early studies proved that eye tracking becomes increasingly unreliable, especially during walking [Mcdonald et al., 1983]. Additionally, Kytö et al. [2018] showed that head-based selections are easy to control and have a higher success rate than gaze-based selections while being slower, and Gizatdinova et al. [2018] highlighted that this is especially true for small targets. For these reasons, as our goal was to identify a reliable and stable interaction technique that can be used when users are on the go, we chose to rely on head tracking rather than eye tracking.

Our cursor implementations build on our previously presented head tracking pipeline.

To track the user's head and face, we used an iPhone XS Max and our tracking software described in Section 2.7.2 "Quantifying Visual Head Tracking" (p. 37). The intersection point determined by this software is then processed to control the cursor. Below, we detail the specific implementations of our three different interaction techniques.

4.3.1 Pure Head Selection

We use force input to activate our reachability techniques as a quasi-mode.

Our **Pure Head (PH)** technique uses head tracking only for target selection. To activate this technique, the user touches somewhere on the screen and applies a light amount of force to enter the head tracking mode. This temporary *quasi-mode* [Raskin, 2000] via force allows the system to differentiate between regular touch events and our interaction technique, making it more applicable in real-world scenarios. Force as a quasi-mode is an established tech-

nique on mobile devices [Corsten et al., 2019, 2018; Roudaut et al., 2008].

When initiating head control, the cursor begins at the center of the screen.

While users interact with a smartphone, their heads typically face their phones. However, in a preliminary study we found that the head is typically directed at a point 20–30 cm above the screen, with users looking downwards with their eyes. For this reason, we assume that when a user enters the head tracking mode, she looks at the screen, and we display a virtual cursor at its center. Then, the user's *relative* head movements move the cursor. For example, to select a target in the top right-hand corner, she slightly rotates her head in this direction. As soon as the cursor is above a target, it is highlighted to indicate the current selection. Once she releases her thumb from the screen, the currently selected target is confirmed, completing the interaction.

Users move the cursor with relative head movement and confirm their selection by lifting the finger.

A sigmoid transfer function increases precision on the center area of the screen.

To improve the success rate of this selection, we used a similar transfer function approach as described by Kjeldsen [2001]. He used a sigmoid transfer function for the user's head movement in a multi-screen desktop environment that allowed users to perform fast yet accurate cursor movements. Since a smartphone screen provides much less screen real-estate than a multi-screen desktop environment, it is even easier to reach items at the edge of the screen. However, more precise control for the middle area of the screen is needed.

Both head position relative to the device and its rotation need to be tracked for our implementations.

The position the user faces on the device is inferred from the head's Euler angles α and distance to the device, which is obtained from the magnitude of the positional vector v. The vector v goes from the center of the screen to the center between the user's eyes, right at the root of the nose bone. We transfer head rotations to positions in a resolution-independent coordinate system ranging from -0.5 to $+0.5$, i.e., the origin is located at the screen's center. Prior to scaling, the measured point u on a screen with the physical size s is calculated as follows:

$$u = \begin{pmatrix} \frac{|v| \times tan(\alpha_y)}{s_x} \\ \frac{|v| \times tan(\alpha_x)}{s_y} \end{pmatrix} \tag{4.1}$$

We then scale the point u with a sigmoid function, where o is the point that was measured at the time when the user initiated the head control:

$$r = \begin{pmatrix} -1.4 \times (\frac{1}{1+e^{7(u_x - o_x)}} - 0.5) \\ -1.4 \times (\frac{1}{1+e^{7(u_y - o_y)}} - 0.5) \end{pmatrix} \tag{4.2}$$

We explored multiple transfer functions and scale factors in preliminary studies and discovered that this function was the most effective.

Lastly, we convert the relative point r into a pixel coordinate p by multiplying it with the screen resolution z:

$$p = \begin{pmatrix} \frac{z_x}{2} \times (1 + r_x) \\ \frac{z_y}{2} \times (1 + r_y) \end{pmatrix} \tag{4.3}$$

Only small head rotations are required to control the cursor.

At a head-to-phone distance of 20 cm (a typical value we measured for the given task), users have to move their head by 9.6° horizontally and 19.9° vertically to select targets in the corners of the smartphone. As the system depends on the perspective, these angles decrease with an increased distance between the user and the device. With a distance of 40 cm, for instance, head rotations of 4.8° and 10.5° are sufficient.

4.3.2 Head + Touch Selection

Head + Touch allows to refine the cursor location via touch.

The **Head + Touch (HT)** selection technique combines head tracking and touch input by allowing the adjustment of the head-selected target with a brief touch gesture. This interaction technique extends the pure head technique, as the user activates and selects targets in the same way. To improve the selection accuracy, the user can momentarily increase the force of the thumb press to lock the head cursor, switching into an adjustment mode. In this mode, a

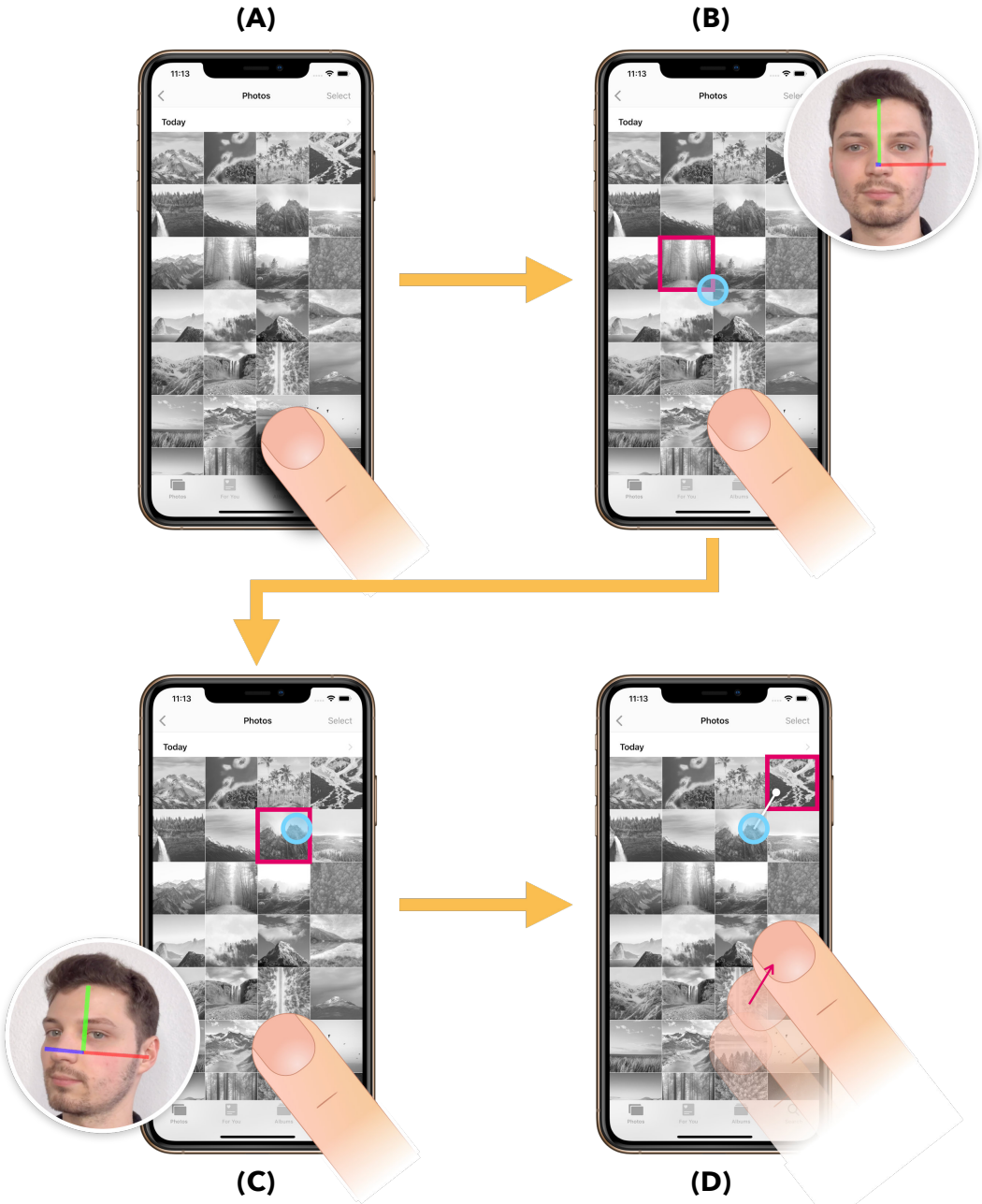


Figure 4.1: A user wants to select the image in the upper right corner, which is inconvenient to access without changing his device grip. The *Head + Touch* interaction technique enables selection by combining head and touch input: **(A)** The technique is activated by applying a small amount of force anywhere on the screen. **(B)** Above a certain force threshold, a virtual cursor is displayed at the center of the screen. **(C)** The user can now control the cursor by rotating his head. **(D)** For a more fine-grained selection, he increases the force momentarily to lock the cursor in place and then drags his finger to adjust the cursor position. Releasing his finger confirms the selection.

small indicator at the center of the head cursor appears, and dragging the thumb will draw a line in the direction of the movement (see Figure 4.1). By applying less force, the user can switch back to the normal head tracking mode anytime. Releasing the touch entirely confirms the selection, just as in the pure head tracking interaction.

4.3.3 Head Area + Touch Selection

Head Area + Touch divides the screen into four quadrants.

In the **Head Area + Touch (HA)** selection technique, the screen is divided into four quadrants. We call the lower right quadrant the *touch input area*, as it is within reach of the user's thumb. Therefore, she can use direct touch to select targets in this quadrant. If the user wants to select a target further away, i.e., in another quadrant, she can select the quadrant by head movement and make the input in the touch input area. The division of the screen into four areas has two advantages. First, even less head rotation is required to select an area, which should result in faster selection times. Second, the touch input area is small enough to be reached comfortably by the thumb.

The touch input area allows absolute indirect touch inputs in the quadrant specified by the head.

To do so, the user activates the head tracking selection mode by applying a small amount of force, as in the HT technique. Instead of a cursor as in the HT technique, the system shows a frame around a selected quadrant (see Figure 4.2), and the user can choose one of the three other areas by rotating her head in the desired direction. Like the previous technique, the area can be locked by momentarily applying a stronger force with the thumb. Now, the touch point from the touch input area is mapped to the selected area using absolute mapping. This mapping is indicated by a virtual cursor representing the thumb's touch location inside the selected area. By moving the thumb, the user can control the cursor position inside the selected area and finalize their selection by lifting it.

4.4 Study 1: Standing

We evaluated our techniques in a user study with 15 participants.

To understand how our different reaching techniques compare to each other and established methods, we conducted a user study with 15 right-handed participants (23–69 years, M = 37.66, SD = 13.94, 6 female). Their average thumb length was 73.56 mm (SD = 7.3 mm) and all of them were smartphone users (screen size: M = 5.3'', SD = .68''). We compared our three techniques (PH, HT, HA) to BezelCursor (BC) [Li and Fu, 2013] and Direct Touch (DT) input as baselines.

4.4.1 Apparatus and Techniques

Participants had to select targets with their thumb using an iPhone XS Max

Participants were asked to hold and operate a smartphone in portrait orientation only using their primary hand. Their task was to select targets with their thumb using each technique while standing and holding the device in their right hand in portrait orientation. We used an iPhone XS Max to present the task to our users and capture data. The iPhone screen measured 896×414 pt (149×69 mm).

Force thresholds for our head-based techniques were adapted from the related work.

To activate all three head-based reaching techniques as described above, we set the force activation threshold to 1.33 units (about 0.7 Newton), which is significantly higher than a typical touch on iOS devices [Corsten et al., 2019]. To lock the cursor in HT and the area in HA, we used the maximum force value the iPhone can detect (about 4 Newton).

We added Direct Touch and BezelCursor as baseline conditions.

We also compared our techniques to **Direct Touch (DT)** and **BezelCursor (BC)**. We chose BC as a baseline condition as Corsten et al. [2019] showed in a similar study setup that BC is faster and has a higher success rate than most other reaching techniques such as MagStick [Roudaut et al., 2008] or Samsung's edge-triggered ThumbSpace. In a pre-study, we also tested the ForceRay technique [Corsten et al., 2019] but found that selecting a specific force value while walking is difficult due to hand and arm movement. BezelCursor was implemented as described by Li and Fu [2013] and included the additional details described by Corsten

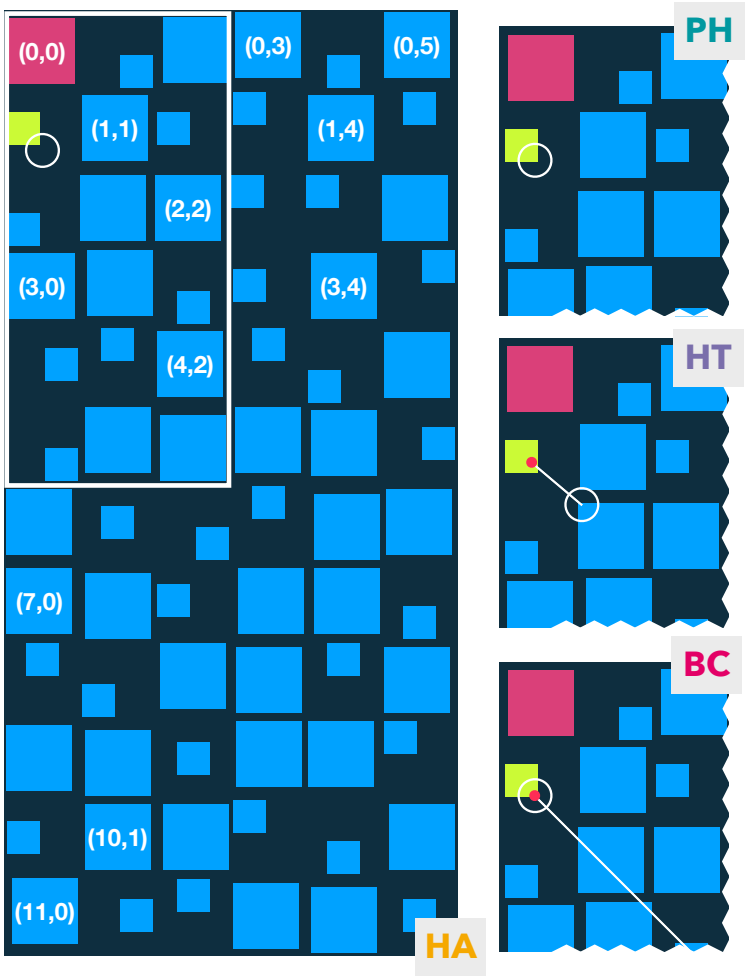


Figure 4.2: The targets were arranged in a 6×12 grid, as labeled by the coordinates (not in the actual trial). The participants were asked to select the red target in each trial. This example shows the large target SIZE. The four screens show the details of our visualization in different conditions. HA: The user moved her head toward the upper left area and moved her thumb to select the green target. HT: The user increased the force of the thumb to lock the virtual cursor and is currently dragging the thumb upwards to the left. PH: The user activated the head tracking technique and rotated her head so that the green target is currently selected. BC: The user is selecting the green target with BezelCursor.

et al. [2019]. It is triggered by a swiping gesture from the screen edge. After detecting the gesture, a line that expands linearly by a factor of three in the direction of the thumb is displayed. Similar to *DynaSpot* by Chapuis et al. [2009], the end of the line has a circular area cursor that expands exponentially up to 7.3 mm depending on the speed of the swipe movement. If the speed drops below 2 $^{mm}/_{s}$ the area shrinks co-exponentially. When a target is below the cursor, it is highlighted. When multiple targets intersect with the area cursor, the target with the smallest distance from its center to the cursor location is chosen. Lifting the thumb selects the target.

4.4.2 Task and Targets

Participants were asked to select targets in a 6×12 grid as quickly as possible.

Participants were asked to select the targets as quickly as possible using the five techniques. At the beginning of each trial, one target was highlighted in red, and the currently selected target was marked in green, as shown in Figure 4.2. The participants had to release the thumb from the touchscreen to confirm the target selection. After selecting a target, the next trial was automatically shown after a delay of 500 ms. The targets were arranged in a 6×12 grid (see Figure 4.2) across an area of 414×864 pt; each cell measured 69×72 pt. We excluded the top 32 pt due to the camera notch of the iPhone. As recommended by Karlson and Bederson [2007], we shifted each target within its cell to avoid a regular-looking grid.

4.4.3 Variables

The independent variables were the TECHNIQUE used to select two different types of TARGETs of different SIZE.

The **independent variables** were TECHNIQUE [PH, HT, HA, BC, and DT], TARGET, and SIZE. Our twelve targets (Figure 4.2) were split into two groups: targets (0,0), (0,3), (0,5), (3,0), (8,0), and (11,0) located at the border of the screen, while the remaining six targets were more toward the center of the screen. The SIZE represented typical iOS widget sizes, i.e., the height of a 30 pt button (4.8×4.8 mm) and a 60 pt app icon (9.6×9.6 mm).

We used the same counterbalancing approaches as similar studies in the related work.

Each participant was asked to perform 5 TECHNIQUE × 12 targets × 2 SIZE × 2 repetitions = 240 trials. TECHNIQUE was counter-balanced using a Latin square, and the order of the targets was randomized. We also randomized the SIZE but, similar to the ForceRay study [Corsten et al., 2019], we ensured that each participant started half of each TECHNIQUE with small targets and the other half with large targets. Before the participants started with a new TECHNIQUE, they were given two minutes to perform trials to familiarize themselves with the new technique. After these test trials, they selected twelve targets two times, followed by the remaining SIZE for the current TECHNIQUE, again starting with the test trials. After both sizes for a TECHNIQUE were completed, a new TECHNIQUE was presented. Overall, the participants took approximately 35 minutes to complete the study.

We measured the *Time* and *Success* of each selection.

Dependent variables were trial completion ***Time*** [s], and user's ***Success*** [0,1], i.e., whether they selected the correct target or not. We measured the *Time* from when a new target was displayed until the user released the finger from the touchscreen to confirm the selection. After the participants finished a technique, they were asked how much they agreed that the technique was easy to use, how fatiguing the technique was, how stable they could hold the device, and how comfortable the head movement was for the three head techniques on a 7-point Likert scale (7 = totally agree). At the of the study, the participants were asked to rank all techniques by ***Preference*** from highest (1) to lowest (5).

4.4.4 Results

In this study, we were most interested in the participants' performance depending on the TECHNIQUE used. Therefore, we focused our analysis on this main effect and related interaction effects. We conducted a repeated-measures ANOVA on the log-transformed *Time* data and calculated the effect size using the partial eta squared measurement. For the dichotomous *Success* data, we ran McNemar and Cochran's Q tests and used the approach from Berry et al. [2007] to determine the effect size. We compared

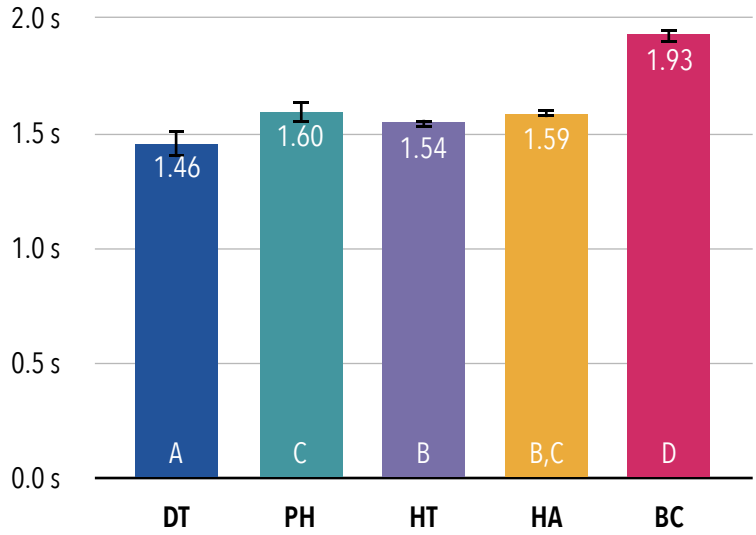


Figure 4.3: *Time* [s] by TECHNIQUE while standing. For each variable, pairs of levels that do not share a letter are significantly different ($p < .001$). Whiskers denote 95% CI.

Likert scale data using Friedman tests and used Kendall's Concordance Coefficient W to calculate the effect size. The pairwise comparisons for the Likert scale data used the Bonferroni correction.

TECHNIQUE had a significant effect. Participants were fastest using DT, slightly slower using HT and HA, and slowest using BC.

TECHNIQUE had a significant main effect on ***Time*** ($F_{4,3560} = 217.74$, $p < .001$, $\eta_p^2 = .196$). Tukey HSD post hoc pairwise comparisons were all significant ($p < .001$) except between PH and HA. Figure 4.3 shows the mean selection times per condition. Not surprisingly, participants were the fastest with DT, followed by HT. HA and PH were the third fastest techniques, followed by BC. These results are also visible in Figure 4.3.

On average, targets with a large SIZE were selected 100 ms faster than the smaller ones.

The target SIZE had a significant main effect on ***Time*** ($F_{1,3560} = 67{,}14$, $p < .001$. $\eta_p^2 = .018$). The Student's t post hoc pairwise comparisons revealed that the large targets (1.55 s) were selected significantly faster than the small targets (1.65 s). The TARGET position had a significant main effect on ***Time*** ($F_{1,3560} = 15.26$, $p = < .001$, $\eta_p^2 = .004$). The Student's t post hoc pairwise comparisons revealed that the targets at the border (1.62 s) were selected slower than the

other targets (1.58 s); however, with an almost unnoticeable effect size. There was also a TECHNIQUE × TARGET interaction effect on ***Time*** ($F_{4,3560} = 13.67$, $p < .001$, $\eta_p^2 = .015$). The HSD post hoc pairwise comparisons are shown in Table 4.4.

		DT	PH	HT	HA	BC
Border	*M*	1.536	1.554	1.581	1.584	1.937
	CI	±.050	±.044	±.030	±.025	±.043
	*	B,C	B,C	C,D	C,D	E
Center	*M*	1.384	1.640	1.507	1.586	1.913
	CI	±.041	±.044	±.030	±.033	±.044
	*	A	D	B	C,D	E

Table 4.4: *Time* [s] by TECHNIQUE × TARGET. Pairs of levels that do not share a letter are significantly different (*Time*: all $p < .001$). CI denotes 95% CI.

DT and PH were significantly less accurate than the other techniques.

TECHNIQUE had a significant main effect on ***Success*** ($Q(4) = 95.56$, $p < .001$, $\mathcal{R} = .025$). Post hoc tests revealed that *Success* for BC, HA, and HT were significantly higher compared to DT and PH. Figure 4.5 shows each technique's mean *Success* rates.

Large targets were easier to select than small ones.

Again, the SIZE of targets had a significant main effect on ***Success*** ($Q(1) = 13.68$, $p = < .001$, $\mathcal{R} = .862$). Post hoc tests revealed that the *Success* rate for larger targets was 3.51% higher than for smaller targets.

Our participants had the lowest *Success* selecting small targets using DT.

There was also a TECHNIQUE × SIZE interaction effect on ***Success*** ($Q(9) = 163.02$, $p = < .001$, $\mathcal{R} = .063$). The post hoc pairwise comparisons showed that the ***Success*** rate for small targets using DT was significantly lower (75.4%) than all other conditions. The ***Success*** rates for PH for small (89.5%) and large targets (87.3%) were also significantly lower than the remaining other conditions. All other conditions, except DT for large targets (94.0%), were not significantly different.

Furthermore, there was a TECHNIQUE × TARGET interaction effect on ***Success*** ($Q(9) = 131.62$, $p = < .001$, $\mathcal{R} = .028$).

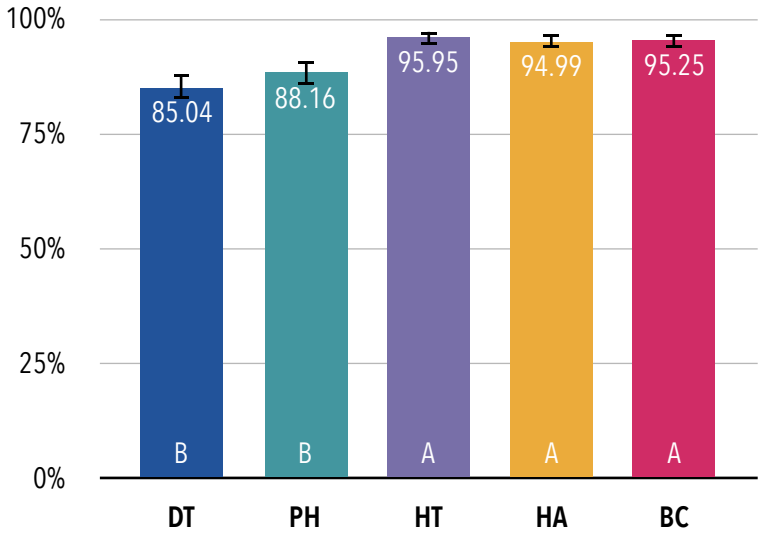


Figure 4.5: *Success* by TECHNIQUE while standing. For each variable, pairs of levels that do not share a letter are significantly different ($p < .05$). Whiskers denote 95% CI.

The post hoc pairwise comparisons revealed that the SUCCESS rate for DT on targets at the border of the screen (80.11%) was significantly lower than all the other conditions. The SUCCESS rate for PH on non-border targets (85.3%) was significantly higher than DT on border TARGETS, but significantly lower than all the other conditions.

We also found significant differences in the Likert scale data regarding grip stability and comfort of head movements.

TECHNIQUE had also a significant effect on the ***ease of use*** ($\chi^2(4) = 40.16$, $p < .001$, W = .586). Users found BC similarly easy to use to DT and HT but significantly easier to use than HA. PH was significantly more difficult to use than the other techniques. The TECHNIQUE had also a significant effect on the ***grip stability*** ($\chi^2(4) = 45.22$, $p < .001$, W = .754). The participants found that they did have a more unstable grip using DT than the other techniques. The TECHNIQUE had no significant effect on the participants perceived ***fatigue*** but on how ***comfortable*** the head movement was rated ($\chi^2(4) = 45.22$, $p < .001$, W = .866).

Participants preferred HT, HA, and BC over the other techniques.

Finally, TECHNIQUE had a significant effect on participants' ***Preference*** ranking ($\chi^2(4) = 41.21$, $p < .001$, W = .804), as depicted in Figure 4.7. The post hoc pairwise comparisons

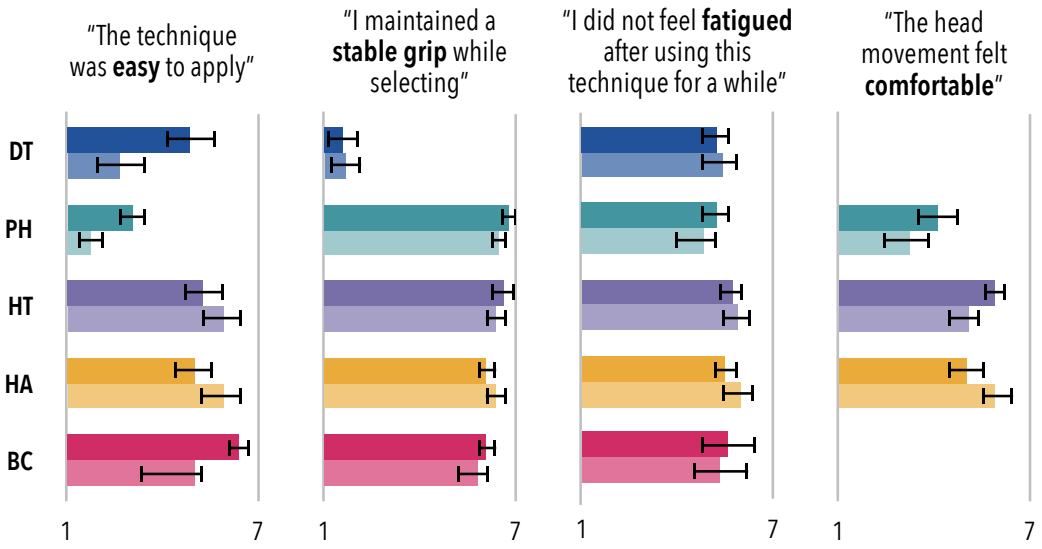


Figure 4.6: The graphs show the results of questionnaires for the standing (upper bars, saturated colors) and walking (lower bars, desaturated colors) conditions. The bar charts depict participant agreement with each statement above on a 7-point Likert scale (1: totally disagree, 7: totally agree). Participants experienced PH as too difficult and DT as unstable. Whiskers denote 95% CIs.

show that participants significantly preferred HT, BC, and HA over PH. DT was rated significantly worse than all other techniques.

4.5 Study 2: Walking

To evaluate our techniques while moving, we conducted a second study with a fresh set of participants.

We evaluated our reaching interaction technique in the first study while participants stood. However, users often interact with handheld devices while walking. Our next study, therefore, explored how these techniques performed in that situation. We conducted this user study with 10 right-handed participants (19–69 years, M = 36.50, SD = 13.54, 4 female). None of these participants participated in the first study. All of them own a smartphone with a mean screen size of 5.1" (SD = .45"). Their mean thumb length was 70.21 mm (SD = 7.5 mm).

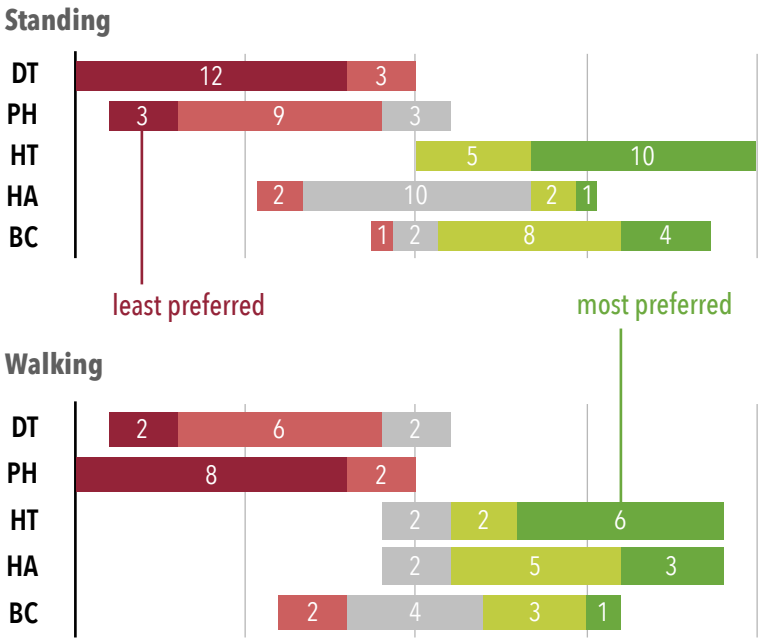


Figure 4.7: These graphs show our participants' forced ranking of techniques while standing (top) and walking (bottom). Options were labeled in the questionnaire from "least preferred" to "most preferred". Stacks that are aligned further to the right were preferred by participants.

Apparatus and task remained the same as in Study 1, except that participants walked around an obstacle course.

The study was conducted in the same way as the first study, using the same study setup, device, and dependent and independent variables. The only difference was that the user had to walk. Similar to the setup from Crossan et al. [2009], the users were asked to walk around a set of obstacles (in our case small tables) in a 4×4 m rectangle, as shown in Figure 4.8.

4.5.1 Results

We used the same statistical methods as in the previous section to evaluate the data measured in this study.

TECHNIQUE had a significant main effect on ***Time*** ($F_{4,2336}$ = 152.03, $p < .001$, $\eta_p^2 = .206$). Tukey HSD post hoc pairwise

Figure 4.8: To evaluate the technique in the walking condition, participants were asked to walk around tables on an eight-shaped path.

While participants were walking, we measured the longest interaction *Time*s with PH.

comparisons were all significant ($p < .001$) except between PH and HA. Also, while walking, users were fastest with DT followed by HT, HA, and BC; PH was the slowest. The average completion times are also shown in Figure 4.9.

Across conditions, large targets were selected faster than small ones.

SIZE of the targets had a significant main effect on ***Time*** ($F_{1,2336} = 11{,}82$, $p = .006$, $\eta_p^2 = .335$). Like the first study, the Student's t post hoc pairwise comparisons revealed that the large targets (1.65 s) were selected faster than the small targets (1.72 s).

Other than while standing, border targets were now selected faster than center targets.

The TARGET position had a significant main effect on ***Time*** ($F_{1,2336} = 11.48$, $p < .001$, $\eta_p^2 = .329$). In contrast to the first study, the Student's t post hoc pairwise comparisons revealed that the targets at the border (1,72 s) were selected faster than the other targets (1,65 s). There was also a TECHNIQUE × SIZE interaction effect ($F_{4,2336} = 5.08$, $p = .008$, $\eta_p^2 = .335$) and a TECHNIQUE × TARGET interaction effect on ***Time*** ($F_{4,2336} = 54.51$, $p < .001$, $\eta_p^2 = .085$). The Tukey HSD post hoc pairwise comparisons for both effects are shown in Table 4.10.

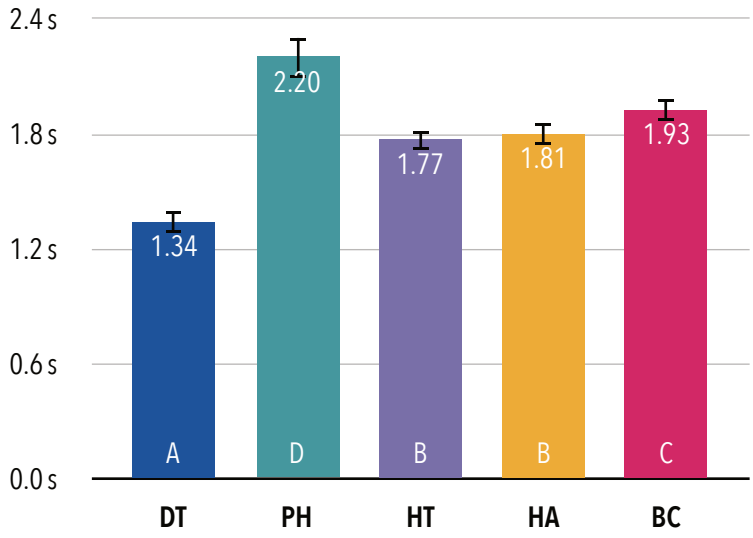


Figure 4.9: *Time* [s] by TECHNIQUE while standing. For each variable, pairs of levels that do not share a letter are significantly different ($p < .001$). Whiskers denote 95% CI.

			DT	PH	HT	HA	BC
TARGET	**Border**	*M*	1.453	1.804	1.774	1.802	1.976
		CI	±.097	±.145	±.065	±.075	±.086
		*	B	C	C,D	C,D,E	E
	Center	*M*	1.233	2.604	1.769	1.810	1.885
		CI	±.047	±.136	±.075	±.084	±.062
		*	A	F	C,D	C,D	D,E
SIZE	**Small**	*M*	1.419	2.170	1.885	1.871	1.927
		CI	±.098	±.159	±.084	±.080	±.071
		*	A	E	C,D	C,D	D,E
	Large	*M*	1.269	2.237	1.741	1.741	1.935
		CI	±.048	±.139	±.048	±.079	±.078
		*	A	D,E	B	B,C	D,E

Table 4.10: *Time* [s] by TECHNIQUE × TARGET and TECHNIQUE × SIZE. Pairs of levels that do not share a letter are significantly different (all $p < .001$). CI denotes 95% CI.

HT, HA and BC provided significantly higher *Success* rates than the other two TECHNIQUES.

For the dependent variable ***Success*** we found the following effects: TECHNIQUE had a significant main effect on ***Success*** ($Q(4) = 380.019$, $p < .001$). We only measured ***Success*** rates above 90% with HT, HA, and BC. This difference to DT and PH was significant. The results of the post hoc pairwise comparisons are shown in Figure 4.11.

Success was higher when targets were larger and/or at the screen border.

The target SIZE also had a significant main effect on ***Success*** ($Q(1) = 14.290$, $p < .001$, $\mathcal{R} = .162$). Larger targets (87%) were selected significantly more reliable than smaller targets (82%). The TARGET position had a significant main effect on ***Success*** ($Q(1) = 7.197$, $p < .001$, $\mathcal{R} = .003$). Targets at the border (86%) had a significantly higher success rate than targets in the middle of the screen (83%).

There was also a TECHNIQUE × SIZE interaction effect on ***Success*** ($Q(4) = 439.440$, $p < .001$, $\mathcal{R} = .003$). The post hoc pairwise comparisons show that selecting both small and large targets with PH had the lowest success rate. Selecting small targets with DT has a significantly higher success rate than PH but a significantly lower success rate than all other conditions.

The TECHNIQUE × TARGET position interaction effect had also a significant effect on the ***Success*** ($Q(4) = 497.896$, $p < .001$, $\mathcal{R} = .034$). Here, the post hoc comparisons show again that selecting targets with PH has the lowest success rate and that selecting not-border targets with DT has a significantly higher success rate than border targets.

The preference ranking was similar to the previous study.

Similar to study 1, TECHNIQUE had a significant effect on the ***Preference*** ranking ($\chi^2(4) = 30.001$, $p < .001$, W = .750). Users preferred HT, BC, and HA over DT, followed by the least preferred technique, PH.

Significant effects in the Likert scale support this preference ranking.

The TECHNIQUE also had significant effect on the ***ease of use*** ($\chi^2(4) = 36.237$, $p < .001$, W = .906). Users found HT and HA significantly easier to use than DT and PH. There was no difference between BC and the other techniques. The effect of TECHNIQUE on ***grip stability*** was significant, too ($\chi^2(4) = 27.739$, $p < .001$, W = .693). Similar to the first study, our participants found that their grip was much more unstable using DT than any other technique. The effect on

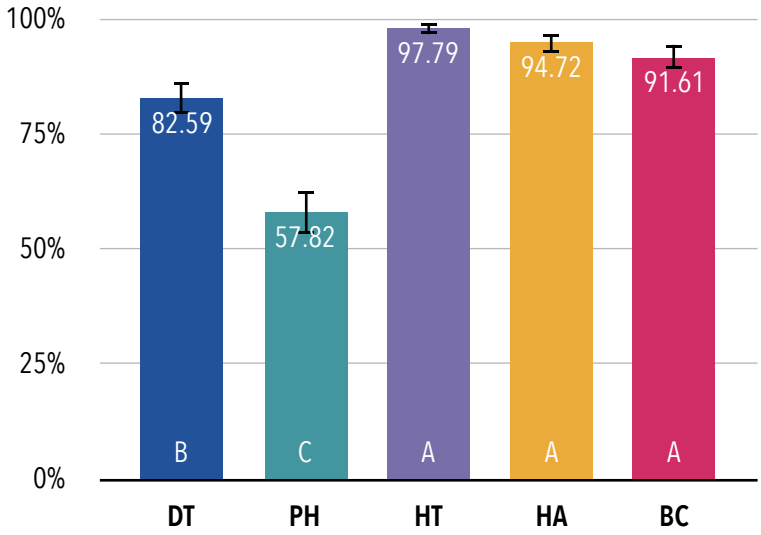


Figure 4.11: *Success* by TECHNIQUE while standing. For each variable, pairs of levels that do not share a letter are significantly different ($p < .05$). While walking, neither DT nor PH provided sufficient accuracy. Whiskers denote 95% CI.

the head movement ***comfort*** was significant as well ($\chi^2(2) = 12.684$, $p < .001$, W = .357). Participants experienced the head movement required by HA as more comfortable than using PH. Other than in the first study, TECHNIQUE had no significant effect on the users' perceived ***fatigue*** ($\chi^2(4) = 9.560$, $p = .049$, W = .239). Figure 4.6 summarizes the means and CIs of the questionnaire data.

4.6 Discussion

The results of our evaluation led us to important insights into how head tracking can be used in reachability techniques. While our goal was to increase the thumb reach on smartphones specifically, our findings can more generally inform the design of interaction techniques that aim to leverage head tracking.

Especially in the walking condition, HT and HA outperform BC with a higher success rate and faster completion times.

Supplementing head tracking with touch input. The results of both studies show that our HT and HA reaching techniques have a higher success rate than DT and a similar success rate as BC. Especially in the walking condition, HT and HA have a higher success rate than BC (see Figure 4.5) even though there is no statistical significance. This makes HT and HA particularly useful for scenarios in which an accurate target selection is important.

HT and HA are slower than DT because they combine two input modalities. Also, DT provides a significantly lower accuracy.

Both techniques are only slightly slower than DT while standing and about 25% slower while walking, a trade-off that was partially expected as our selection technique combines two different input modalities. However, HT and HA were significantly faster than BC, and we believe that once users become more familiar with those new input techniques, the gap to DT will narrow further. These findings show that head tracking input in combination with touch offers a good trade-off between speed and accuracy.

Head-based techniques likely scale better than BC on larger screen sizes.

As the users can target any point in their vicinity using their heads, both HT and HA can also be used on larger devices such as tablets with similarly small finger movements. On the other hand, when using BC on large devices, the movements of the fingers to reach a target increase. However, the head-based techniques require the camera of the devices to track the user's head, which could lead to increased battery consumption.

While PH was comparable to other techniques while standing, it failed while walking.

Importance of realistic testing conditions. Our study revealed that while PH performance in the standing condition is similar to other head tracking techniques, it becomes almost unusable while walking. We believe this to be rooted in the problem that while a user is walking, not only the head is moving but also the arm and, thus, the smartphone. Those results align with previous works that have identified issues with head tracking in real-world scenarios [Crossan et al., 2009].

Realistic testing scenarios are essential for research with mobile devices.

Our study contributes an important data point due to the comparative study of different techniques in two different conditions. It presents solutions for this problem by supplementing head tracking with touch input. We hope this

not only inspires future research to develop new interaction techniques but also encourages to consider evaluations in more realistic scenarios, as the results of controlled lab studies might not be replicable in the real world, rendering promising interaction techniques unusable.

User preference and mitigating frustration. While our participants preferred HT the most, the results also see HA ranked quite high in the walking condition. We assume this is because it requires slightly fewer movements, and the control of each movement requires effort while walking. The rankings highlight that participants accepted the general concept of combining coarse head selection and touch fine-tuning. Although in a different context, a similar insight was reported for gaze interaction by Stellmach and Dachselt [2012] where users made a coarse selection on a wall-sized display with the eye and fine-tuning via touch input.

The combination of a coarse head selection tweaked by touch input was well perceived by participants.

The participants' preference plays an important role when comparing novel approaches with established techniques: While direct touch was generally the fastest selection method, it led to a high level of frustration. Not only did participants rank DT extremely low in both conditions in terms of grip stability, but three participants even dropped the phone while they tried to reach a target in the upper corner of the display. Our post-study questionnaire also highlighted another issue of head tracking for input: Fatigue that is well-known from previous research [LoPresti et al., 2000]. However, when pairing head tracking with another technique, this effect can be mitigated as the head movement is less enunciated due to it only being used for coarse pre-selection, as highlighted in our HT and HA conditions in both studies (see Figure 4.6).

Our novel interaction techniques were also ranked better because they offered a less frustrating experience for the participants.

4.7 Future Work

Across all conditions, we saw no significant difference between HT and HA in terms of performance, both in terms of

success rate and time. Most questionnaire responses show a similar result as we saw no substantial differences between the two techniques or relatively small preferences for either technique in one of the conditions (head movement comfort in Figure 4.6). The most significant difference between the two techniques can be found in the ranking, as most participants ranked HT as their favorite selection technique in both conditions and HA as second (walking) or third (standing, behind BC).

For future work, one could investigate larger screen sizes and different aspect ratios.

However, further work is required to investigate the differences between those and potential other techniques in more detail. For example, which concrete real-world use cases can best be supported by which technique? We only considered portrait view—but are there scenarios using landscape orientation, and how does head + touch perform in those situations? How do those techniques scale, e.g., on larger screens such as tablets, where reachability even becomes an issue in multitouch environments when using two hands to hold a tablet while talking? Can head tracking in combination with touch input also be helpful in such a scenario, and if so, how is it best implemented?

4.8 Conclusion

We designed and evaluated different interaction techniques that solve the reachability problem on smartphones.

In this chapter, we investigated the use of head tracking to address the reachability issue on handheld touchscreens. In addition to pure head tracking as a target selection input technique, we developed *Head + Touch*, an approach that complements head tracking with touch input for refining the target selection, and *Head Area + Touch*. This additional technique allows users to first select a target area via head tracking and then refine the selection within that screen area. We compared those three techniques to traditional direct touch input and a well-known technique that aims to address reachability, BezelCursor by Li and Fu [2013]. To ensure that our evaluation reflects a realistic use case, we conducted two user studies in different conditions: in the first study, participants selected targets using all five different techniques while standing, and in the second study while walking.

The results of our evaluation show that our combination of head tracking and touch input not only addresses the reachability problem but also performs well compared to existing techniques. While we identified that pure head tracking as a selection technique encounters issues, particularly in the walking condition to the point of not being viable as an alternative, our refined techniques are viable interaction techniques. Both approaches that combine head tracking with touch input were more accurate but slightly slower than direct touch and faster than BezelCursor, with almost similar success rates. Our *Head + Touch* and *Head Area + Touch* techniques offer a useful tradeoff between success rate and speed of input for target selection tasks. Especially in real-world scenarios with one-handed smartphone operations, they provide an improvement over traditional touch input. We believe that this work can also inform future research into identifying ways how to leverage head tracking as a complementary input technique on touch devices, as head tracking becomes more ubiquitous in today's technology.

We found a useful tradeoff between accuracy and speed by combining head tracking for a coarse selection with touch input for more precise control.

In this chapter, we used head tracking as a two-dimensional input to specify elements on the screen. The two studies prove that head tracking was reliable both while standing and walking. As our participants' experience with the interaction techniques and their preferences were quite similar across standing and walking conditions we will explore the seated setting only in the next chapter. The increased tracking stability during the seated interaction will allow us to integrate gaze tracking, which can target any object in the vicinity. This will allows us to make the jump from screen-space to world-space content.

Head + Touch links the user's head to a screen-space cursor. Next, we will select world-space content using facial tracking instead.

Chapter 5

Investigating Gaze Support in Cross-Device Interactions

SUMMARY:
We present *GazeConduits*, a calibration-free ad-hoc mobile interaction concept that enables users to collaboratively interact with tablets, other users, and content in a cross-device setting using gaze and touch input. GazeConduits leverages recently introduced smartphone capabilities to detect facial features and estimate users' gaze directions. To join a collaborative setting, users place one or more tablets onto a shared table and position their phones in the center. The system then tracks their presence and gaze direction to determine which tablets they look at. We present a series of techniques using GazeConduits for collaborative interaction across mobile devices for content selection and manipulation. Our evaluation with 20 simultaneous tablets on a table shows that GazeConduits can reliably identify which tablet or collaborator a user is looking at, enabling a rich set of interaction techniques.

Publications: The work presented in this chapter was done in collaboration with Simon Voelker, Christian Holz, Christian Remy, and Nicolai Marquardt. The author of this thesis developed the research idea and relevant research questions with his co-authors. Furthermore, he created the software artifact and planned and conducted the study. Most of this work has been published as a paper in the Proceedings of ACM CHI 2020 [Voelker et al., 2020]. The author of this thesis is one of the principal authors of the paper. Most sections in this chapter are taken from the paper publication.

5.1 Motivation

Cross-device interactions rely on seamless interactions to shift data between devices connected in an ad-hoc network.

Cross-device interaction between multiple co-located mobile devices has become an emerging field of research in human-computer interaction [Hamilton and Wigdor, 2014; Rädle et al., 2014]. These setups of co-located mobile devices can be used to create *ad-hoc device communities* [Jetter and Reiterer, 2013] almost anywhere to allow users to share or collaborate on content across devices. Especially using the spatial relationship between co-located devices is beneficial for applications such as brainstorming [Rädle et al., 2015], collaborative photo sharing [Jin et al., 2015], and productivity tasks [Marquardt et al., 2018]. For seamlessly interacting across devices, researchers have explored a variety of input techniques, such as touch [Rekimoto, 1997] and gestures [Hinckley, 2003; Rekimoto et al., 2003].

In multi-user cross-device interactions, gaze tracking can be used to identify objects of interest. As this requires no ongoing control using the eyes, the double role of gaze is minimized.

One input modality receiving increased attention lately is gaze as a communication channel for recognizing the user's interaction with a device [Pfeuffer et al., 2014]. The previous chapters prove that users can reliably use their heads to specify objects out of reach for touch. As our eyes move to perceive visual input, head tracking allows us to use different input and output modalities that can transfer information concurrently, preventing a double role of gaze. However, targeting a distant object using the head requires more physical movement than just looking at it. In specific scenarios, the double role of gaze is less of an issue, e.g., when the interactive system relies only on identifying the user's object of interest. Especially in multi-user scenarios, gaze sensing enables new ways to interact with objects out of reach for touch. Particularly, mobile cross-device interactions can benefit from gaze as input, as studies have shown its benefits for interacting with multiple devices [Turner et al., 2014; Voelker et al., 2015].

This chapter explores calibration-free gaze interaction techniques.

In this chapter, we integrate mobile gaze sensing into cross-device interaction to explore the design space of collaborative interaction on everyday mobile devices. Our approach is guided by the following research question: How

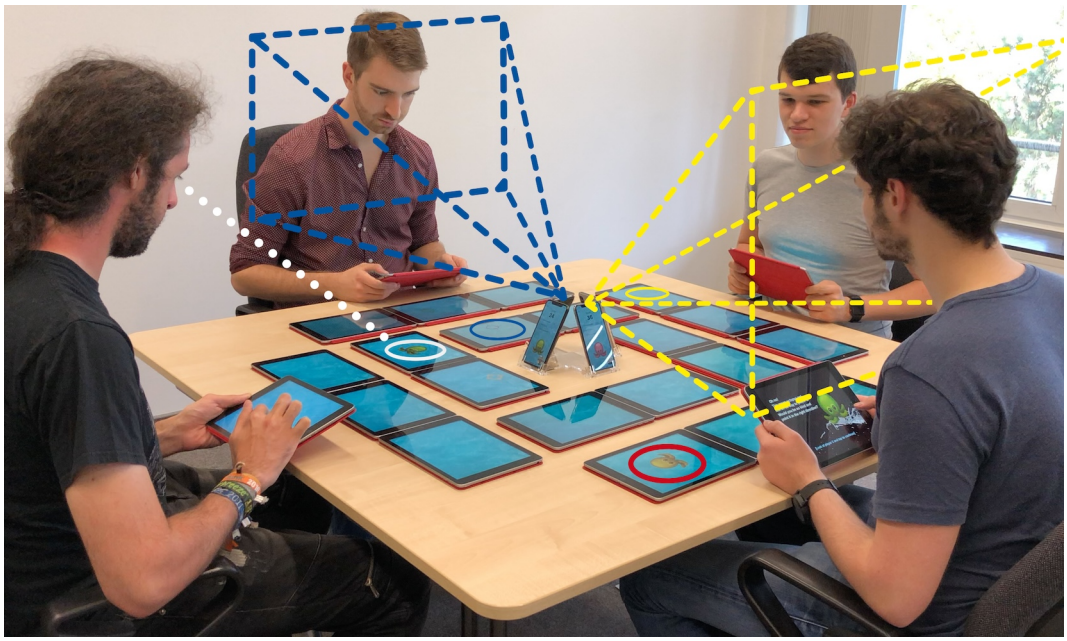


Figure 5.1: GazeConduits is a mobile and ad-hoc gaze tracking system that allows one or more users to interact across the devices placed in front of them. GazeConduits leverages the eye-tracking capabilities of recent commodity devices, allowing them to participate in cross-device interaction scenarios.

can we implement calibration-free collaborative interaction techniques that support cross-device interaction?

GazeConduits showcases interaction techniques that can be used on off-the-shelf ad-hoc device setups.

We present our gaze tracking system *GazeConduits*, which leverages the gaze tracking capabilities built into recent smartphones to detect and track users around a set of devices as well as predict their gaze direction toward specific devices for selection and interaction. GazeConduits requires no calibration, which makes it simple to set up in ad-hoc group scenarios. We briefly demonstrate multiple scenarios that build on GazeConduits in conjunction with touch input to enable dynamic groups of users to select and manipulate content across their devices collaboratively. We evaluated GazeConduits and found that it can detect which tablet or collaborator a user is looking at with 95.6% accuracy. GazeConduits contributes to the field of interactive technologies and showcases how off-the-shelf devices can be used to achieve calibration-free, collaborative cross-device interaction by combining gaze and touch input.

5.2 Related Work

We position the GazeConduits techniques in the context of three areas of related work: the design of cross-device systems and their requirements for tracking, interaction techniques leveraging a person's gaze as input, and the recent approaches toward mobile gaze tracking. As the latter was already presented in Section 2.6, we will focus on the former two here.

5.2.1 Cross-Device Systems and Tracking

Cross-device interactions focus on sharing data and controls across multiple co-located devices. For these interactions, the different dimensions in the framework by Greenberg et al. [2011] (Section 2.1, p. 15), e.g., distance and orientation, provide semantics to which actions are appropriate in different device and user constellations [Grønbæk et al., 2020].

Mobile cross-device interactions were explored in different contexts, including sensemaking, curation, media sharing, and content editing.

The survey on cross-device systems by Brudy et al. [2019] identified a recent trend in ad-hoc mobile cross-device uses. Researchers envisioned novel interaction techniques for such cross-device setups across different application domains. For example, *VisTiles* by Langner et al. [2018] couples and coordinates data visualizations across multiple co-located mobile devices to simplify sensemaking. The work of Wozniak et al. [2016] shows how cross-device sensemaking on mobile devices benefits from spatial awareness of devices. Brudy et al. [2016] explored a cross-device document presentation system to help curate digital content. Lucero et al. [2011] presented interaction techniques that help to share and consume digital media like photos across multiple phones. Klokmose et al. [2015] investigated collaborative content editing across multiple devices.

Usually, these setups distribute an interface across a number of devices, such as tablets, smartphones, interactive walls, or tabletops, and provide techniques for effectively using the input/output modalities of these multi-device se-

tups. The research has led to the exploration of supporting both individual and collaborative tasks. Early work on supporting individual usages includes Hinckley et al. [2009], who presented interaction techniques that allow for fluent interactions with dual-screen tablet computers. Related to that, *SurfaceConstellations* by Marquardt et al. [2018] is a system of 3D-printed brackets to arrange multiple mobile devices optimized for specific tasks stably. Regarding collaborative activities, multiple interaction techniques to interact with media on a shared large screen were developed, e.g., by Izadi et al. [2003] and Wigdor et al. [2009]. However, collaborative tasks often also contain individual subtasks. In this context, Homaeian et al. [2018] showed the trade-off between accessing data display on a shared display and mirroring it on a personal device: When users can interact with shared content directly from their personal device, this is more comfortable for them and less distracting to collaborators. However, it comes at the cost of reduced awareness of other people's actions.

Cross-device setups can be used for collaborative or individual tasks. The latter also occurs as part of collaboration.

A central research area of cross-device work is investigating effective techniques for a person to interact with this ecology of devices [Brudy et al., 2019]. Inspired by early seminal work such as *Pick-and-Drop* by Rekimoto [1997], different techniques have been designed and evaluated for linking devices and transferring digital content. One technique for pen-based tablets similar to Pick-and-Drop is *Stitching* by Hinckley et al. [2004]. With this technique, dragging content with the digitizer pen from one device surface to another will initiate data transfers. Simeone et al. [2013] envisioned drag and drop gestures with a bimanual interaction between desktop and phone: One hand aligns the devices next to each other; the other hand is used to swipe contents with touch. Hinckley [2003] used IMU data to trigger device connections. One example of their *synchronous gestures* is extending the desktop of one device to the other display by bumping the sides of the two nearby devices against each other.

Researchers explored various interaction techniques that make data transfers within the devices of a cross-device system seamless and tangible.

All these techniques rely on some physical contact between the two devices to initiate data transfers. Contrarily, Hamilton and Wigdor [2014] envisioned a technique where the different devices could remain spaced apart. A broadcast is

Data transfers with spatially agnostic broadcasts require no physical contact.

sent to all other devices after initiating a possible transfer from one device. Therefore, without specifying a target device in advance, this transmission can be accepted from any device nearby. Not requiring physical contact might have advantages, as the authors found out during their sensemaking tasks study. People place devices so that they can remember which content is where.

Spatially aware techniques integrate external tracking mechanisms of devices. They can reduce mental workload during operation.

The above techniques are spatially agnostic: They do not know about the actual arrangement of the devices in the room or on the table. Therefore, overhead-positioned depth-sensing cameras have been used to add context to different cross-device systems. For instance, Hu et al. [2014] and Wu et al. [2017] used them to track the position and orientation of the users in front of a shared digital surface, Rädle et al. [2014] tracked locations of devices on a table, and Marquardt et al. [2012] combined tracking of both to offer different functionalities and spatial gestures based on different proxemic distances between participants. Later, related approaches leveraged screen polarization, like the *PolarTrack* by Rädle et al. [2018] or tracking a person's face with front-facing RGB cameras like Grubert and Kranz [2017] to detect device location. The work of Rädle et al. [2015] suggests that spatially aware techniques can reduce mental workload and are preferred by people. Because most of these approaches require complex technical setups, finding new ways toward ad-hoc and flexible cross-device collaboration remains an ongoing research challenge.

5.2.2 Gaze Interactions

The double role of gaze and Midas touch problem make direct manipulation of contents using our eyes infeasible.

Gaze interactions promise a fast and effortless input, as gaze is our primary sensory channel (Section 2.2, p. 17): The first interaction with an object is to look at it [Zhai, 2003]. However, several obstacles complicate the use of gaze in interfaces, such as its inaccuracy, the double role of gaze, and the Midas touch problem defined by Jacob [1990]. These circumstances render it ineffective to directly manipulate digital content or control cursors using gaze [Stellmach and Dachselt, 2013].

Instead, gaze has proven to be more useful to indicate areas of interest or in applications as a method for selecting content. Vertegaal et al. [2005] used eye contact in combination with a universal control to determine which home theater appliance is being controlled. Zhai et al. [1999] used gaze to coarsely position the mouse cursor while supporting refinement and click actions using the familiar mouse interaction. Stellmach and Dachselt [2012] presented methods to pan and zoom gazed content on a distant display by tilting or touching a handheld device or the mouse. Multiple studies by Pfeuffer et al. [2014, 2015]; Pfeuffer and Gellersen [2016] and Voelker et al. [2015] showed that gaze can be utilized to switch between direct and indirect touch input seamlessly. This is achieved by using direct touch while users are looking at their hands and warping the touch point to the gaze location when they look at a different part of the display or even at a different display.

Successful applications of gaze in interfaces are mode switches and pre-selections of cursor locations.

An advantage of gaze over touch input is that it does not share the reachability problem [Remy et al., 2010], as any objects in the visible range can be interacted with. Studies by Turner et al. [2011, 2014, 2015] have shown that a combination of gaze and touch can be used to extend the touch input on a large surface, to transfer objects between multiple devices, or to modify objects on a distant screen. Gaze input can also be a handy addition to collaborative settings. Studies by Zhang et al. [2017], van Rheden et al. [2017], and Pfeuffer et al. [2016] used gaze tracking to show that all collaborators recognize at which location a user is looking, thus increasing the awareness of collaborators. All these interaction concepts can be applied in an ad hoc mobile cross-device setting; however, so far, they have required either extensive calibration, specific hardware, or a controlled lab environment to track gaze.

In collaborative settings, gaze tracking was often combined with touch, e.g., to extend the reach on large touch surfaces.

5.3 Gaze Tracking in an Ad-hoc Setting

In ad-hoc cross-device setups, multiple users can create device communities on the fly by arranging multiple mobile devices on a surface. With GazeConduits, we followed this ad-hoc approach and designed it for easy setup without cal-

The ubiquitous smartphone serves as a central hub within GazeConduits.

Figure 5.2: The smartphone case (left) has an included stand that can be expanded and collapsed. It ensures that the phone can easily be placed at the correct angle on the table. The connector widget (right) allows the users to arrange up to four smartphones in a fixed arrangement easily.

ibration. Our choice of using the smartphone to track gaze instead of a separate eye tracker allows the system to be set up at any time without specific hardware requirements beyond the ubiquitous smartphone that people carry in their pockets.

For GazeConduits, we placed the smartphone in an upright position on a table using a stand integrated into a smartphone case.

GazeConduits builds on iOS running on Apple's iPhone X or later being able to extract facial features from its front-facing RGB-D camera. In our preliminary study in Section 2.7.3 "Quantifying Visual Gaze Tracking" (p. 41) we already saw that uncalibrated gaze tracking accuracy suffices to identify regions with the size of a tablet. Yet, for the phone to see the environment and its user, we need a stand that holds the phone upright at the center of the table where the users are sitting. For this purpose, we built a prototype smartphone case with an expandable stand, shown in Figure 5.2

Determining the optimum angle. We wanted to provide users with comfortable seating around devices and providing enough space to move around without unintentionally leaving the tracking area. Assuming a user's typical sitting

height of 30–40 cm above the table and close to the table edge, the phone should be placed 60 cm away from that table edge, following basic trigonometry. At this position, we varied the tilting angle of the phone and found that an angle between 25° and 35° allowed users to move their heads without the phone losing tracking.

We found that a tilt angle of 30° (±5°) was most suitable to allow users to sit comfortably at the table.

The GazeConduits case. Our smartphone case allows the smartphone to be placed as a central hub at the correct position and angle. Using an additional connector widget, we can ensure that four phones can be placed in a fixed position, similar to the idea of SurfaceConstellations of Marquardt et al. [2018]. To obtain each phone's direction in this constellation, we can use the compass to determine their relative positions. GazeConduits then creates a shared coordinate system across devices that registers the location of each tracker phone.

Multiple smartphones can be locked together using magnets.

Using a smartphone case as a stand directly results in no need for users to carry additional custom hardware to set up the cross-device system. This makes it more practical. We built this prototype using *LEGO Technic*[1]. However, such a case could also be easily built in a thinner form factor using a 3D printer or laser cutter.

Using a stand integrated into the phone case omits the need for users carrying additional hardware.

5.4 GazeConduits

With the GazeConduits case, we created a stable and scalable solution to support gaze tracking in a cross-device setup. In this section, we present interaction techniques that GazeConduits uses to create a calibration-free collaborative environment that supports a combination of gaze and touch interactions on commodity devices and minimal setup requirements.

To set up the GazeConduits system, the first user places an iPhone on a table and starts our GazeConduits app, which acts as the central hub for gaze input. The integrated stand

[1] `www.lego.com/en-us/themes/technic`

To start a session, the user launches the app and uses the phone case to position the phone at the center of the table.

of the case enforces a suitable tilt angle and provides the flexibility for users to expand and collapse the setup during operation. Using the measurements of the IMU in the smartphone, we can calculate the exact position on a plain surface that represents the table in the smartphone's coordinate system.

Each tablet added to the session requires users to provide its coarse position relative to the hub phone(s).

After placing her smartphone on the table, the user can add up to 20 tablets around the smartphone following a rough grid layout (Figure 5.1). To participate in the shared device environment, the GazeConduits app is launched on each tablet, which automatically connects it to the smartphone, in our case via Wi-Fi. Each new tablet then displays a message asking the user to select its rough position on the grid (Figure 5.3).

GazeConduits supports up to 20 tables with a size of up to 25×25 cm.

Several design decisions have influenced the layout and size of our grid and the limitation to 20 tablets. Each grid element has a size of 25×25 cm; based on our accuracy tests, this size ensures that the gaze tracking mechanism can always reliably detect whether a user is looking at a target within that grid element. It is also slightly larger than common tablets like the iPad (23.8×16.7 cm), Microsoft Surface GO (24.5×17.5 cm), or Samsung Galaxy Tab (24.9×16.4 cm), making sure that common tablets fit into a grid element without overlap issues, regardless of their orientation.

Tablets can be placed in a 5×5 grid, as at most two tablets fit in the tracking frustum between a user and the phone.

Assuming a setup with up to four collaborators around the table, we can support a maximum of a 5×5 grid, with the central grid element being reserved for the smartphones. Two rows of tablets are placed in front of each user; with more rows, the distance from a user to a phone would exceed the maximum 88 cm derived in previous tests, especially if the user is leaning back or rocking back and forth in their chair. Furthermore, the corners of the grid do not allow for reliable gaze tracking, as users looking at tablets in the corners of the table have to rotate their heads more than 30 degrees. Thus, we limited the number of grid elements in the first row in front of each user to three, resulting in the grid used in Figure 5.1.

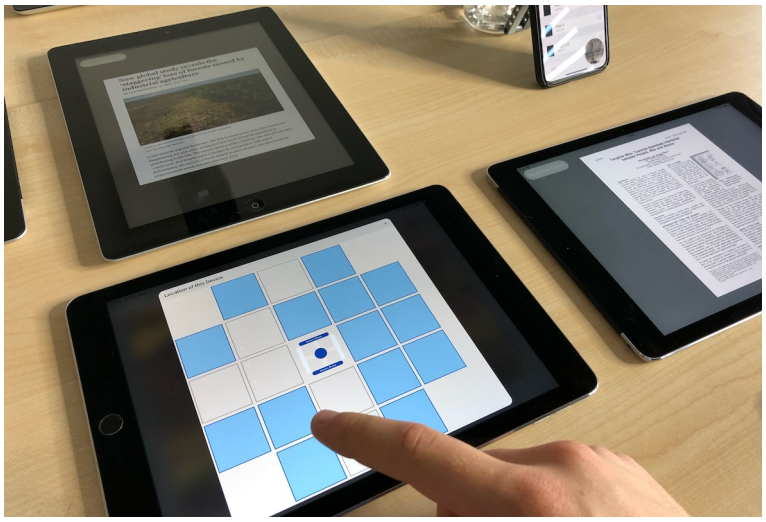


Figure 5.3: When the user places a new tablet on the table, GazeConduits connects to existing devices and asks the user to select its grid position.

The number of users, tablets, and arrangement of tablets is fully flexible.

GazeConduits works on layouts other than the 20-tablet arrangement, too, e.g., in setups with only one or two users or scenarios where not all tablets should be selectable via gaze. As soon as one or more tablets are placed on the table and the user selects at which position these tablets are placed, GazeConduits can directly detect and track which tablet or other collaborator the user is looking at.

GazeConduits uses a Voronoi grid to map the interaction space on the table to tablets. This makes the system robust against sparsely filled grids.

While GazeConduits displays a static, predefined grid on its tablet UI during this positioning phase, the system does not require tablets to align with the grid elements perfectly. It displays a grid to suggest good placement positions, but after tablets have been placed and the users have tapped on the closest grid element for each, the system splits up the entire shared space between the existing tablets, mapping every area to the nearest existing device in a simplified Voronoi grid approach. This makes our system robust against sparsely filled tablet grids and misaligned devices. The grid becomes more important if two tablets are placed directly adjacent to each other. In this case, aligning them to the grid helps disambiguate which tablet is being looked at.

Orientation awareness rotates contents on tablets so that they face their viewer.

By comparing the compasses of each tablet and smartphone, GazeConduits can roughly determine each tablet's orientation. This enables GazeConduits to, e.g., change the display orientation of a tablet dynamically based on which user is looking at it.

User awareness allows identifying and locating users around the table.

Moreover, GazeConduits can also detect if one or more users are present, their locations around the table, and via face identification who is located where. We implemented this detection by comparing the geometry (nose, chin, and mouth) of the detected faces, achieving a simple user identification to simulate the use of a more advanced face recognition system in collaborative scenarios.

Users can have a private tablet that other collaborators cannot access.

In addition to the shared tablets, GazeConduits also supports one private tablet per user. These tablets can be freely moved around and used for private content that the other users should not see. Other users cannot select it using gaze. However, they can use the respective person as a proxy for this tablet. For example, if one user wants to share an object from her private tablet with another user such that the others cannot see the object, he could just look at the user and perform a touch gesture on his private tablet to send it to the other user's private tablet. We explore this and several other interaction concepts in the scenarios section below.

5.5 Study 1: Evaluating Gaze-to-Tablet Tracking

To evaluate tracking accuracy, we conducted a study with 10 participants.

To understand how well the gaze tracking of the phone and our correction method can identify which tablet a user is looking at, we created a game for a user study with 10 participants (23–35 years, M = 28.73, SD = 3.31, two female). In this study, users had to perform a gaze-and-drop interaction similar to the content transfer techniques introduced by Turner et al. [2014].

5.5.1 Apparatus and Task

The game task was inspired by typical cross-device movement operations.

We set up GazeConduits with one iPhone X that tracked the participant's gaze and 20 shared tablets placed on the grid around it as shown in Figure 5.1. Participants were asked to sit in front of the smartphone so that it could track their faces. The system provided feedback at any time by displaying a crosshair on the shared tablet they were looking at. In addition, they held one private tablet, which they were instructed to use for touch input. The goal of the game was to feed octopuses on the shared tablets with a shrimp displayed on the private tablet. This task mimics a typical cross-device object movement operation, in which a user wants to transfer an object from one tablet to another.

Integrating a private tablet in this task provided a homing location for the gaze and touch selections.

At the beginning of each trial, all shared tablets only showed a blue water background, and the private tablet displayed a 3×3 cm large shrimp at a random location. As soon as participants touched and held the shrimp, an octopus appeared on one of the shared tablets. Participants then had to find the tablet with the octopus and, while looking at it, release the touch from the shrimp on their private tablet. The octopus and shrimp disappeared as soon as they released their touch, and a new trial was started. With this study design, we ensured that participants looked at their private screens at the start of each trial. Each octopus appeared four times on each of the 20 tablets in random order, such that each participant had to conduct 80 trials for a total of 800 recorded gaze selections across 10 participants. Before starting the study, the participants had the opportunity to familiarize themselves with the system by conducting one test trial for each tablet.

5.5.2 Variables

We used POSITION as independent variable and measured the *Time* and *Success* of each selection.

Since we were primarily interested in how reliable the system could detect which tablet the participants were looking at and if this differed between tablet positions, we used the tablet POSITION as **independent variable**. As **dependent variables**, we measured ***Success*** [0,1] if the system was able

to identify that the user looked at the correct target, and the task completion ***Time*** [s] for each trial.

5.5.3 Results

We measured high *Success Rates* independent of POSITION.

The overall success rate was 95.58%, To analyze the relation between POSITION and ***Success*** events, we calculated the ***Success Rate*** [%] for each user at each position. We used a repeated-measures ANOVA to analyze the data. We could not find a significant effect of POSITION on the ***Success Rate*** ($F_{8,1336} = 779$, $p = 0.0758$).

POSITION had a significant effect on the *Time* needed to select a tablet.

We measured an average task completion time of 2.13 s (SD = 1.33 s). For analysis, we used a repeated-measures ANOVA on the log-transformed data. POSITION had a significant main effect on ***Time*** ($F_{19,779} = 8.13$, $p < .001$). Tukey HSD post hoc pairwise comparisons showed that the tablet positions are divided into four groups that are significantly different from each other in terms of task completion time. Figure 5.4 shows these groups.

5.5.4 Discussion

The study confirms the reliability of gaze-to-tablet tracking under feedback.

Our study results highlight that GazeConduits can reliably identify which tablet a user is looking at. The success rates show that the system could identify each tablet position similarly. However, since the system displayed a cursor that indicated the tablet at which participants were looking, participants could correct the selection by moving their eyes and head until the system selected the correct tablet.

The majority of tablets (groups 1 and 2) were directly identified correctly by the tracking.

Task completion times suggest that for most tablet positions (groups 1 and 2), GazeConduits was able to identify the tablet the user was looking at directly. We also observed that it was sufficient for participants to look at most of these tablet positions naturally.

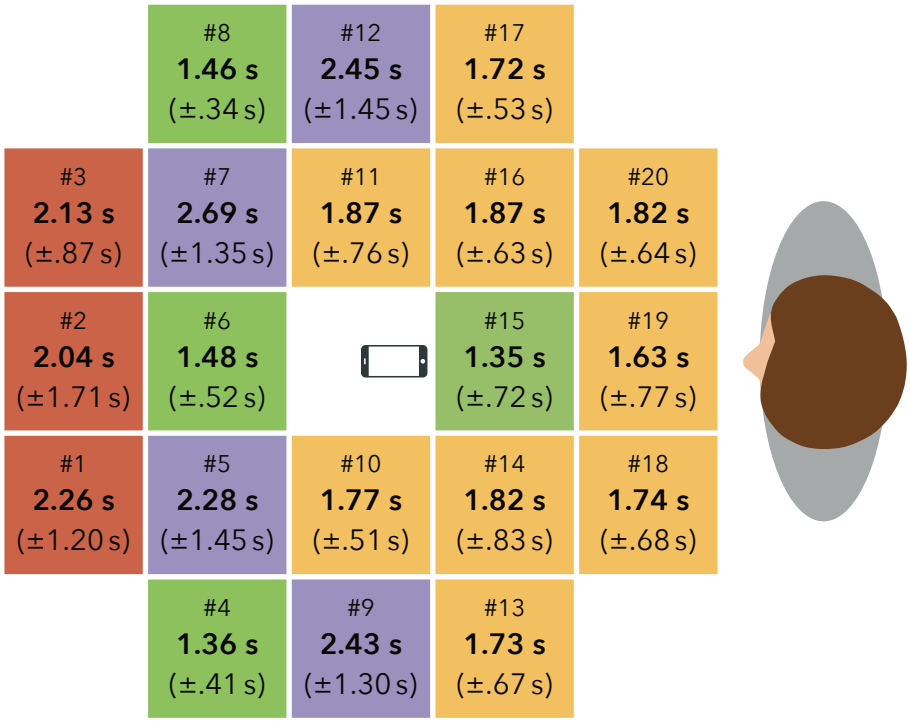


Figure 5.4: Task completion times in the Gaze-to-tablet study. Colors differentiate the groups whose measurements were significantly different. Participants were able to select the green tablets fastest, followed by yellow, red, and purple.

The difference in times suggests that tracking was subpar for tablet positions in groups 3 and 4.

However, for the tablet positions in groups 3 and 4, the system could not always identify which tablet users were looking at, and the cursor jumped between multiple tablets. Most users tried to stabilize the cursor by actively moving their heads and eyes toward the tablet position, resulting in longer task completion times.

Especially for tablet positions in group 4, we could not find a final explanation for why it took participants more time to select them. We hypothesize that because several other tablets surrounded them, and due to inaccuracy, the system was not always sure which tablet participants were looking at, thus jumping frequently between different tablets. However, this is also true for tablet positions 10 and 11, which were selected significantly faster.

5.6 Study 2: Evaluating Gaze-to-Person Tracking

In Study 2, we evaluated how reliably GazeConduits tracks gazing at collaborators.

We conducted another study to evaluate how GazeConduits tracks the gaze of four users simultaneously and how reliably it can detect if a user is looking at another user. 12 people aged between 22 and 33 years (M = 26.17, SD = 3.02, two female) participated in this study.

5.6.1 Apparatus and Task

We used a similar setup and task as in the previous study.

Setup and task were similar to the first study, except now with four participants around the table instead of one, all playing the game simultaneously (Figure 5.1). In this setup, the game acted merely as a distractor to keep participants busy and engaged. The actual task we focused on in our analysis was that participants had to look at other participants from time to time. This was triggered by a notification on the participant's private tablet displaying the other participant's ID. The participant then had to look at the person indicated and confirm this selection with a touch on their private tablet. During the study, each participant had to look at every other person eight times, for a total of 24 selections per participant.

5.6.2 Variables

We used COLLABORATOR as independent variable and measured *Time* and *Success*.

As **independent variable**, we used the COLLABORATOR location (left, front, right) that described at which other person a participant had to look. As **dependent variables**, we used the same as in the first study: ***Success*** [0,1] and task completion ***Time*** [s] for each trial. We measured the time from when the notification appeared on the private tablet until the participants confirmed their selection.

5.6.3 Results

The overall success rate was 95.14% across all trials and participants. As in the first study, we calculated the ***Success Rate*** as a percentage for each user and each COLLABORATOR and used a repeated-measures ANOVA to analyze this data. COLLABORATOR had no significant main effect on ***Success Rate*** ($F_{2,284} = 1.2$, $p = .086$).

We measured high *Success Rates* independent on which side of a participant the COLLABORATOR sat.

We measured an average task completion ***Time*** of 2.26 s (SD = .88 s) across all trials and participants. We used ANOVA on the log-transformed data but did not find a significant difference between COLLABORATORs ($F_{2,284} = 0.55$, $p = .574$).

User selection took 2.26 s on average with no significant differences between conditioins.

5.6.4 Discussion

The results of this study show that GazeConduits can reliably detect when a user is looking at a particular other user. It also shows that the shared virtual 3D space between the four smartphones is stable enough to support such interactions across users. However, task completion time was relatively large. This was likely due to the game being too much of a distractor, as we often observed participants searching for the next octopus instead of noticing the notification on their private tablet.

We attribute this slowness to participants not registering the task quickly enough.

5.7 Interaction Scenarios

GazeConduits takes a step toward gaze as a real-world input modality by removing the requirement for calibration altogether while expanding its functionality to include multiple users and devices simultaneously, all with off-the-shelf devices. We believe that the increased availability of gaze tracking has the potential to enable a variety of new cross-device, collaborative interaction scenarios. To illustrate this potential, we present applications and benefits that GazeConduits enables through its awareness of users

We illustrate the potential of GazeConduits in the scenarios below.

around a table, gaze-at-device interactions, and gaze-at-user interactions.

5.7.1 Interactions through GazeConduits' User Awareness

User awareness can benefit competitive usage scenarios.

Where competition exists, e.g., in multiplayer games, GazeConduits provides benefits by maintaining a map of users' presence around the table, detecting who enters and leaves the tracking space where and when.

User authentication prevents collaborators from seeing private content.

For example, in *GazePoker* (Figure 5.5), each user can only see their own set of cards. When a player temporarily gets up and leaves the table, her cards flip to blanks so that a neighboring player cannot peek. Only when the player returns does GazeConduits authenticate her and restores her cards. Similarly, when a second player attempts to peek while the first person is present, GazePoker detects this and temporarily hides the cards until the second player has left the frame again. Due to the player detection, users could even switch seats or change devices while maintaining privacy, while each player will always see her cards only.

Others are blocked from actions on private devices as each touch is associated with a user.

In *GazeScrabble*, the system can ensure that only the owner of a set of letters can place them. This way, neighboring players cannot advance the game when a player temporarily leaves the table. GazeConduits enables this app's behavior by enabling it to associate a touch-input event with a particular user in front of the device.

5.7.2 Interactions through Gaze-at-Device Tracking

Different gaze interactions can foster collaboration.

The continuous tracking of people's gaze across the tablet devices allows for different gaze & touch interaction techniques for cross-device group collaboration.

With *GazeMirror*, we suggest a new technique for users to rapidly mirror the content of any other device onto the screen of their tablet directly in front of them. The mir-

Figure 5.5: GazePoker prevents cheating by continuously authenticating users in front of their cards based on their faces. Those cards are hidden when a player leaves in GazePoker, or somebody else peeks in.

Ad-hoc screen sharing from a shared tablet can be initiated with gaze & touch interaction.

roring is triggered by a four-finger multitouch gesture on a private tablet while the user gazes at another tablet on the table. As long as the user holds the four-finger touch gesture on her screen, she can view and, using her other hand, interact with the mirrored remote content on her local device. The technique is designed as a lightweight, ad-hoc technique for collaborative settings. In traditional setups, each user maintains their own tablet, and collaboration frequently involves invading each others' private space. Using gaze in this scenario improves the accessibility of shared content.

Combining gaze with multitouch allows for data transfers.

This technique can also be combined with content transfers between devices, called *GazeDrop*. To transfer content from the remote device to the local device, a user touches the object that should be moved while releasing the four-finger mirroring gesture. When the mirroring ends, the object stays at the user's finger and is moved to the local device. GazeMirror can also be used to mirror the content of the gaze-tracking smartphone to a tablet. Examples of applications for this gesture are content transfers from the user's smartphone to a private or public tablet, such as a private message, or accepting a call that a user received during the cross-device session. Since only the user who is tracked by the smartphone can select it using gaze, she is the only one who can mirror content from her phone to a different device, accounting for privacy concerns.

Easily accessible analysis of gazing patterns can help to reflect materials in design sessions.

At the end of a meeting session, *GazeHeatMap* visualizes how much time people in the meeting looked at any device by changing the background color of each tablet. This can, for example, support design critique sessions: The heat map provides feedback about which designs drew the most attention during the session, which in turn can trigger discussions about the reasons.

5.7.3 Interactions through Gaze-at-Users Detection

Handshake interactions could integrate gazes at other users, which occur naturally during conversations.

GazeConduits maintains the location of each user and can also detect if a user is looking at another person. We use this feature to detect if two users look at each other. This virtual handshake can trigger actions that both users have agreed upon. If a user wants to edit the personal content of another user, she first asks for permission by looking at the content and then at the content owner. If the owner looks back at the user and performs a particular touch gesture, the owner provides the user permission to edit the content. This feature can also be used to synchronize object transfer between two users.

5.8 Limitations and Future Work

Increased gaze tracking accuracy from technological advancements could enable further interactions.

We primarily designed GazeConduits to allow users to interact with a set of mobile devices using gaze input in an ad-hoc cross-device setting without the need for gaze calibration. However, GazeConduits was only evaluated with tablets of a certain size, and it can only detect that a user is looking at a tablet but not at which location on the tablet. Gaze tracking accuracy would need to be improved to support smaller devices such as smartphones or to detect the exact gaze location on a tablet. This could be achieved by using more sophisticated tracking and calibration methods. However, we also anticipate an increase in gaze tracking accuracy, as it only recently became a feature in commodity devices, and a more widespread adoption may lead to increased demand. While gaze tracking is only available in the most recent smartphone models, a more

widespread availability will allow for more reliable use of GazeConduits due to expected improvements in tracking accuracy, with the basic interaction technique remaining the same. For example, the face detection on more recent iPhone models covers a wider FOV, which allows GazeConduits to track a user's face and gaze in a larger area in front of the device.

The dynamics of direct touch interaction with nearby tablets should be investigated further.

Touch can be more useful than gaze interaction when devices are within easy reach. However, in general, using gaze can resolve situations in which a device is out of reach (the distance to the furthest tablet could be over 1 m), and even when within reach, gaze can provide a better solution when reaching into another user's personal space to pick up a tablet would be awkward—and if the other user is actually holding a tablet, then, rather than grabbing that tablet out of their hands, GazeConduits will help to select that tablet using the person as a proxy (see our Study 2).

With a refined case and stand system, one could add even more users to a GazeConduits session.

The maximum table depth is limited by the phone's tracking area and resolution to ensure that users' faces remain recognizable. GazeConduits is currently limited to four users who are sitting in fixed positions. Depending on the available technology, this can be increased to more users in specific scenarios. In some scenarios, these constraints match the actual use case well. For example, GazePoker entails social constraints (staying close to your cards), and user positions tend to be stationary in this scenario.

Future work should also explore interactions around the displacement of shared tablets.

GazeConduits currently does not support moving tablets on the table, as it cannot directly track their position. While the system can detect when devices are moved, it does not detect their location, and very slow movements are currently not detected. Therefore, users have to update positions manually on the device when prompted. In future work, we want to explore methods in which users can use their gaze to specify the location of a tablet, which requires a more accurate gaze tracking algorithm.

5.9 Conclusion

We created a calibration-free ad-hoc gaze tracking system for collaborative use cases called GazeConduits. We empirically evaluated the precision of the system and presented the benefits of gaze and touch interactions in different interaction scenarios.

This chapter presented GazeConduits, a system that uses a combination of gaze and touch input in a collaborative ad-hoc setting. We exploited that in such a setting, users remain seated around a table and placed the phone in a fixed position to obtain more accurate gaze estimations. Both measures reduced the data noise in our previous facial tracking applications, where our study participants moved. We could, therefore, omit any calibration or time-consuming setup in our implementation of the gaze tracking system: Users can simply sit down at a table, connect their off-the-shelf devices to the GazeConduits system, and start using a combination of gaze and touch to interact seamlessly with multiple devices and users. Our evaluation shows error rates of less than 5%, even with up to four users, four phones, four private tablets, and 20 shared tablets combined in one scenario. Not only can users interact with all these devices, but GazeConduits can distinguish between different users, which enables additional new interaction techniques. We also highlighted scenarios that arise from those new opportunities to provide an outlook of how gaze input can shape the future design of interactive technologies.

In GazeConduits, users made explicit selections of world-space content. Next, we will explore the advantages of making eye tracking implicit and activated steadily.

This was our first research project that tracked the users' eyes and not only their heads. In GazeConduits, eye tracking is used to specify tablets or collaborators across the room. However, apart from GazeConduits' user authentication features, all of the presented interactions require combining gazing with explicit touch input. In the next chapter, we will explore a use case of implicit eye tracking, which allows the on-screen content to continuously update with the user.

Chapter 6

Enhancing Handheld Augmented Reality with Face Tracking

SUMMARY:
In handheld AR, users only have a small screen to see the augmented scene, making decisions about scene layout and rendering techniques crucial. Traditional device-perspective rendering (DPR) uses the device camera's full field of view, enabling fast scene exploration but ignoring what the user sees around the device screen. In contrast, user-perspective rendering (UPR) emulates the feeling of looking through the device like a glass pane, which enhances depth perception but severely limits the field of view in which virtual objects are displayed, impeding scene exploration and search.
We introduce the notion of *User-Aware Rendering*. By following the principles of UPR but pretending the device is larger than it actually is, it combines the strengths of UPR and DPR. We present two studies showing that User-Aware AR imitating a 50% larger device successfully achieves both enhanced depth perception and fast scene exploration in typical search and selection tasks.

Publications: The work presented in this chapter was done in collaboration with Johannes Wilhelm, René Schäfer, Simon Voelker, and Jan Borchers. The author of this thesis developed the research idea and relevant research questions. Furthermore, he designed, implemented, and evaluated the study. Most of this work has been published as a paper in the Proceedings of ACM MobileHCI 2023 [Hueber et al., 2023]. The author of this thesis is the main author of the paper. Most sections in this chapter are taken from the paper publication.

6.1 Motivation

AR using DPR results in severely limited depth perception.

Augmented Reality (AR) renders 3D objects into a view of the real world. The popularity of smartphones and their technical advancements have established handheld AR as the type of AR most commonly used by the masses [Dey et al., 2018]. However, depth perception in handheld AR is severely limited [Kruijff et al., 2010; Liu et al., 2020; Swan et al., 2017], leading to interaction problems when navigating a scene or selecting targets. One limiting factor is that smartphones lack a stereoscopic image. Furthermore, depth perception is also limited because the image you see on screen uses *device-perspective rendering (DPR)*, i.e., only the position and orientation of the phone camera determine what is visible, as known from taking photos. This means that the device ignores your own field of vision, and moving your head around in front of your smartphone, as we naturally do to look at a scene from different angles, will not change what is displayed on the screen.

UPR provides better depth perception than DPR at the cost of a smaller FOV at a typical usage distance.

User-perspective rendering (UPR) tries to overcome these issues by matching the *frustum*, a cut-off pyramid representing the field of view displayed on-screen, to the area that the screen itself covers within the user's natural field of vision [Baričević et al., 2014; Mohr et al., 2017]. This offers a simple metaphor to the user: The device becomes a transparent glass window into the world that adds augmentation inside that window. This means that the frustum of UPR is dynamic. When holding the device closer to the face, it covers a larger part of the user's natural visual field, and thus, the frustum extents increase. However, with typical distances between face and device of 40 cm [Boccardo, 2021], this results in a very narrow field of view (FOV), so that users can only see augmentations in a small part of the world around them: through a window the size of their device at their arm's length.

Without a stereoscopic screen, UPR will always produce a small horizontal offset.

Other limitations of user-perspective rendering include the fact that the camera frustum must originate in the user's eyes for a correct rendering. However, only one eye can be addressed precisely without a stereoscopic screen. This already results in a trade-off between using the device with

(A) User-perspective

(B) User-aware 1.5x

(C) User-aware 3.0x

(D) Device-perspective

Figure 6.1: Different rendering techniques when holding the device at an angle. User-Perspective Rendering (UPR, A) and Device-Perspective Rendering (DPR, D) differ in both the orientation from which the camera looks at the scene and their field of view (FOV). In UPR (A), the device aims for virtual transparency: the cupboard in the background is aligned between the device viewport and peripheral vision. However, the FOV is limited, and the sheep is slightly too large to fit on the screen. In contrast, DPR (D) creates a noticeable offset between screen and real world. For instance, the cupboard's real-world and on-screen locations are disjoint in DPR. Our User-Aware Rendering (UAR) techniques (B, C) serve as a middle ground between the two, combining a large FOV with approximate alignment.

one eye shut or accepting that the horizontal alignment of the content is slightly off.

User studies suggested that the FOV of UPR is too small to be usable.

Yet even when these limitations of UPR are overcome, the increased realism obtained from it may not offer enough benefit for users to prefer it over the traditional, much wider-angle FOV of typical smartphone camera lenses: In a study by Baričević et al. [2012] participants preferred DPR over UPR for smartphones simply because notably less content fits into the small FOV of UPR while operating at a normal posture. If the FOV is too small, it becomes impossible to fit large scenes on one screen, which may also negatively affect depth perception [Kline and Witmer, 1996] and search times [Ren et al., 2016; Kruijff et al., 2010].

Our goal was to combine the strengths of DPR and UPR in a hybrid rendering technique.

User-perspective rendering has been analyzed in many environments [Baričević et al., 2014; Mohr et al., 2017; Čopič Pucihar et al., 2013; Yang et al., 2018], and the default device-perspective AR has reached the daily lives of many. However, *hybrid* rendering techniques that combine aspects of both techniques above have received no attention, opening up an exciting research opportunity: Is there a hybrid rendering approach that combines the strengths of DPR, like fast scene exploration, with the strengths of UPR, like enhanced depth perception? How do these different perspectives affect the AR experience and interaction?

The UAR frustum is calculated similarly to UPR, but with a virtually increased screen size.

In this chapter, we introduce *User-Aware Rendering* as a novel approach: By taking a UPR implementation and virtually increasing the device size, we created a technique with a larger FOV that still reacts to the user's natural gaze direction. Moreover, the larger FOV also makes horizontal misalignments of the screen content less obvious, thus mitigating typical instabilities occurring in UPR systems.

In summary, the key contributions of this chapter are:

- We introduce User-Aware Rendering as a new rendering and interaction technique for handheld AR.
- We present results from a study on how this technique affects depth perception, a measure in which UPR is known to perform better than DPR.

- We report how User-Aware Rendering affects search + selection tasks, in which DPR outperforms UPR due to its larger FOV.

We evaluated UAR in two user studies.

In two user studies on depth perception and object selection, we demonstrate that handheld AR leveraging User-Aware Rendering combines the strengths of UPR and DPR in these two areas, making it a promising candidate for many typical AR use cases. In the remainder of this chapter, we first review related work and then introduce User-Aware Rendering from a technical point of view before discussing the two user studies. We close with the limitations of our approach and resulting opportunities for further research.

6.2 Related Work

Handheld AR creates the illusion of depth through kinetic depth cues.

In handheld AR, users see the virtual world through a single small screen. While using the system, people move their devices around to change what is on screen. Doing so, they perceive 3D contents on a 2D screen by generating kinetic depth cues, such as the motion parallax induced through camera movement.

Motion parallax greatly benefits depth perception.

Motion parallax has long been known to substantially impact depth perception independent of other visual characteristics of the screen. For instance, in a fundamental study conducted by Rogers and Graham [1979], participants were able to identify the depth geometry of different planes that were only visualized through random dot patterns through the dot displacement achieved through motion parallax. In the AR context, Furmanski et al. [2002] found that people often falsely perceive virtual objects as being in front of the real world when additional motion parallax and occlusion effects were missing.

UPR also generates motion parallax from head movement.

With UPR, the on-screen content is dependent not only on the device posture but also on the user's head. Thus, even without physical arm movements, users constantly trigger new inputs to the system and create motion parallax. In this section, we first present related work using UPR and continue with the influence of visual characteristics on depth perception and the impact of the FOV on 3D scene understanding.

6.2.1 User-perspective Rendering

UPR makes the device virtually transparent. This enhances the mapping between the virtual and the real world.

User-perspective rendering allows for AR "magic lenses" in their original sense as envisioned by Bier et al. [1993]: Instead of looking at the virtual scene through the perspective of the device, the device itself becomes transparent, like a sheet of glass. With UPR, objects appear in the same size and location in the user's vicinity as they would if they were real. In a user study by Čopič Pucihar et al. [2013], participants had to use handheld AR with either DPR or UPR rendering to identify locations they needed to tap on a touchscreen. Their participants completed the task faster using UPR and also had a higher preference for it despite their UPR implementation requiring a fixed distance between the handheld device and the head. This shows how the correctly aligned camera feed of UPR enhances the mapping between the augmented and real worlds.

UPR is technically challenging and requires precise eye tracking as small errors can diminish the whole effect.

However, user-perspective rendering is a technically and computationally challenging problem. It requires tracking of the user's eyes and knowledge of the device location in the real world to calculate an off-axis projection of the virtual scene. Therefore, small errors in head tracking can diminish the effect. Prototype systems are not entirely stable yet, require specialized hardware, and/or force the user into a specific, fixed position in front of the AR system. For example, Andersen et al. [2016] presented three alternative UPR display implementations, all with unique shortcomings.

One of the first handheld systems exploring head-coupled perspective rendering was *pCubee* by Stavness et al. [2010],

Researchers created various handheld systems that integrate UPR. All of them have different caveats.

a small fish tank VR with displays on each side of a cube. Shortly after, handheld AR prototypes leveraging UPR were created with a variety of technical caveats. For instance, a system by Hill et al. [2011] required a fisheye camera and a fixed point of view from which the user had to observe the scene. One can also compensate the required fisheye camera with a homography transformation of the planar camera image to approximate UPR with fidelity, like in the works of Tomioka et al. [2013] and Samini and Palmerius [2014]. Baričević et al. [2014] were able to create a UPR simulation with acceptable stability by integrating a stereo-matching algorithm. Mohr et al. [2017] increased the stability (though not correctness) of their UPR simulation by lowering the sampling rate of head input. Yet, the work of Andersen et al. [2016] shows one main challenge across all prototypes remains that robust pixel-perfect alignment of the virtual scene has been impossible so far.

While user-perspective AR could also allow for different interactions than device-perspective AR, this has received little attention so far. One example of such interaction techniques is using the user-perspective occlusion of the real world as a target selection mechanism Qin et al. [2023].

In user studies, UPR performed better on large devices, like tablets, as these compensate for the small FOV.

Since making the device transparent massively narrows the FOV when holding it at arm's length, several studies favor using large screens for user-perspective rendering. For example, already the work of Oh and Hua [2006] indicated how larger magic lenses containing more information help to solve information-gathering tasks. EhT study of Baričević et al. [2012] also showed advantages for tablet-sized magic lenses in comparison to smartphones. To circumvent the technical issues of UPR on handheld devices, they prototyped user-perspective AR in VR. The results of their search and select task show that UPR could slightly enhance selection times when using a tablet. But more importantly, their study participants strongly preferred device-perspective rendering when using a smartphone, as it allowed them to see much more of the scene at once by providing a significantly larger FOV.

6.2.2 Depth Perception

Pictorial depth cues supplement the kinetic depth cues, e.g., shadows and reflections.

As handheld AR uses a single screen, depth perception cannot rely on physiological cues like stereoscopy to convey depth. Therefore, depth is inferred by our brain from pictorial and kinetic depth cues instead. Early work in the field of AR, e.g., by Drascic and Milgram [1996], already pointed out the importance and challenge of creating suitable depth cues. While UPR can enhance motion parallax and thus kinetic depth cues, *pictorial depth cues* can be generated by the AR system independent of the chosen perspective. Pictorial depth cues include visual characteristics of the virtual objects, e.g., shading, shadows, relative size and shape of an object, or its texture [Cutting and Vishton, 1995]. While each of these various features enhances the visual realism of an object, neither increased depth perception significantly on its own in a study by Diaz et al. [2017]. Instead, they found an interaction effect between multiple pictorial cues that enhances depth perception.

Only combining different pictorial depth cues provides significantly enhanced depth perception.

The Shape and complexity of the virtual content intrinsically influence depth perception.

The virtual content itself also has an impact on perception. For example, participants in a study by Diaz et al. [2017] were faster at identifying the depth of virtual content with a planar shape than with a torus shape. In the study of Roo et al. [2018], participants could reliably transfer target locations between physical and virtual models by matching them to unique visual landmarks of known positions in the model. Do et al. [2020] also found the complexity of the virtual content's shape, color, and texture luminance to impact depth perception.

Harsh lighting and shadows can amplify depth perception.

Likewise, light and shadow can help users perceive depth. Spotlights that are manipulated with the viewport can enhance the depth effect. For instance, the UPR implementation of Yang et al. [2018] placed a light-emitting node at the user's head position. Much depth information is inferred from a 2D representation of the scene, as can be seen in shadow projections. Even though they look less realistic, drop shadows performed better than ray-traced shadows in the study of Diaz et al. [2017].

Still, even with a realistic AR rendering pipeline, the limited depth cues current systems provide are insufficient to precisely judge the distance from user to virtual content, independent of the device size used as the viewport. In the case of Liu et al. [2020], participants underestimated the distances to virtual objects when observing them through a smartphone, which matches the results of Swan et al. [2017] who used a tablet. The issue of limited depth perception partially seems to be intrinsic to the content being virtual. For instance, Witmer and Sadowski [1998] found that distance judgments when walking through virtual environments on a treadmill are less accurate than walking in the real world. What is more, the data by Thompson et al. [2004] showed that the rendering quality does not influence the lack of accuracy of depth estimations in virtual environments. While requiring additional input hardware, Wacker et al. [2020] showed that rendering synthetic depth cues by shaders that react to a real-world physical object as a reference point also enhances depth perception in virtual scenes.

Yet, even with realistic AR rendering, distances between user and objects are often underestimated.

6.2.3 Impact of FOV

The size of the FOV determines how much of the virtual scene fits on the screen. Therefore, the difficulty of visual tasks increases with a limited FOV [Kruijff et al., 2010]. The FOV is also known to have a strong influence on distance perception. For example, according to the studies of Kline and Witmer [1996], when observing content through small FOVs, humans tend to overestimate distance as the content appears zoomed in. Large FOVs have the contrary effect. A user study by Ren et al. [2016] showed that when searching for annotations in large virtual models, a larger FOV leads to faster completion times. While handheld AR uses central vision, wide-FOV AR systems are also possible. However, the findings of Sun and Varshney [2018] show that people will take longer to notice changes in their peripheral vision area.

The FOV determines how much content fits on the screen. As a result, it impacts the size of content and distance judgments.

Common mobile AR using DPR suffers from the *dual-view problem*, a term defined by Čopič Pucihar et al. [2013] to

The dual-view problem arises from three visual errors when using DPR: A mismatched FOV and misaligned camera image in terms of position and orientation.

summarize the three mismatches of the on-screen content with the real world: Different FOV, non-centered screen capture, and an angular offset of views. All of them are present in Figure 6.1, where a comparison between UPR (A) and DPR (D) shows that the virtual content has a different on-screen size and a different alignment with the real world. Especially this viewing angle offset can bias inter-object relations of the virtual content [Kruijff et al., 2010]. Čopič Pucihar et al. [2013] showed that users of UPR have a higher spatial perception in comparison to DPR.

6.3 User-Aware AR

Despite the advantages of UPR, its small FOV and tracking requirements make it unfeasible for mobile devices.

By mitigating the dual-view problem and offering increased motion parallax, UPR seems to be a promising technique for enhancing AR experiences. However, even small errors or jitters in the head tracking can diminish the entire effect. Especially the need to "zoom in" the camera image in order to align the content makes tracking errors easily noticeable.

6.3.1 Concept

Our idea was that UPR with a larger FOV would mitigate the existing problems while adapting the benefits of DPR.

The motivation for our user-aware rendering (UAR) was to combine the advantages of UPR and DPR without increased hardware requirements. We knew from the related work that a small FOV can lead to perceptual issues and make any tracking errors more noticeable. Therefore, UAR was designed to mitigate only two out of three aspects of the dual-view problem, focusing on the alignment of the overall content while remaining flexible in content size.

In Figure 6.1(A–C), the center of the on-screen content spatially correctly overlaps its real-world counterpart using both UPR and UAR. The increased content scales of UAR, however, render content smaller so that more fits on screen. This results in a slight misalignment toward the edge of the screen, which increases with higher content scale factors. Due to the device borders separating the screen from

the real world, especially with UAR 1.5x, this offset is very minor. Using DPR, however, only the device position defines the on-screen content. Figure 6.1(D) shows how the center location of the on-screen content does not match its real-world counterpart and how content at the edge of the screen is noticeably apart from its real-world counterpart.

UPR and DPR differ in their transformation and projection matrices.

Computationally, what we see through a virtual camera is defined by two matrices. The transformation matrix describes the camera location and orientation in space. The projection matrix defines the visible frustum before the clipping plane (i.e., screen). Figure 6.2 provides a visualization of different frustums.

In DPR, the transformation matrix is defined by the device location, and the projection matrix is static.

When using a device-perspective frustum, the "eye" that observes the scene is the device camera. Thus, the transformation matrix encodes the location and orientation of the device, and the projection matrix is a constant matrix that fits the device camera's characteristics. This stands in contrast to the user having her own field of vision and the device only covering a small part of this area. For instance, in Figure 6.1(D), there is only a limited spatial relationship between the background image on the device vs. around the device: Although the device only covers the lower part of the cupboard, the entire cupboard is visible on the screen.

In UPR, the transformation matrix is defined by the user's eyes, and the projection matrix is calculated from the spatial distance between device and user.

User-perspective rendering overcomes this issue by calculating a frustum that converges at the user, depicting the parts of the scene that are covered by the device. Thus, in Figure 6.1(A), only the lower part of the cupboard can be seen, and the camera image is approximately aligned with the real world. Since smartphone displays are not stereoscopic, one has to define one location for the camera inside the user's head: This can be the right or the left eye or the center between both eyes as an approximation.

6.3.2 Prototype System

UAR borrows from the calculation of a dynamic viewing frustum from UPR, yet it increases its FOV by pretending that the device used is actually larger (Figure 6.2), i.e.,

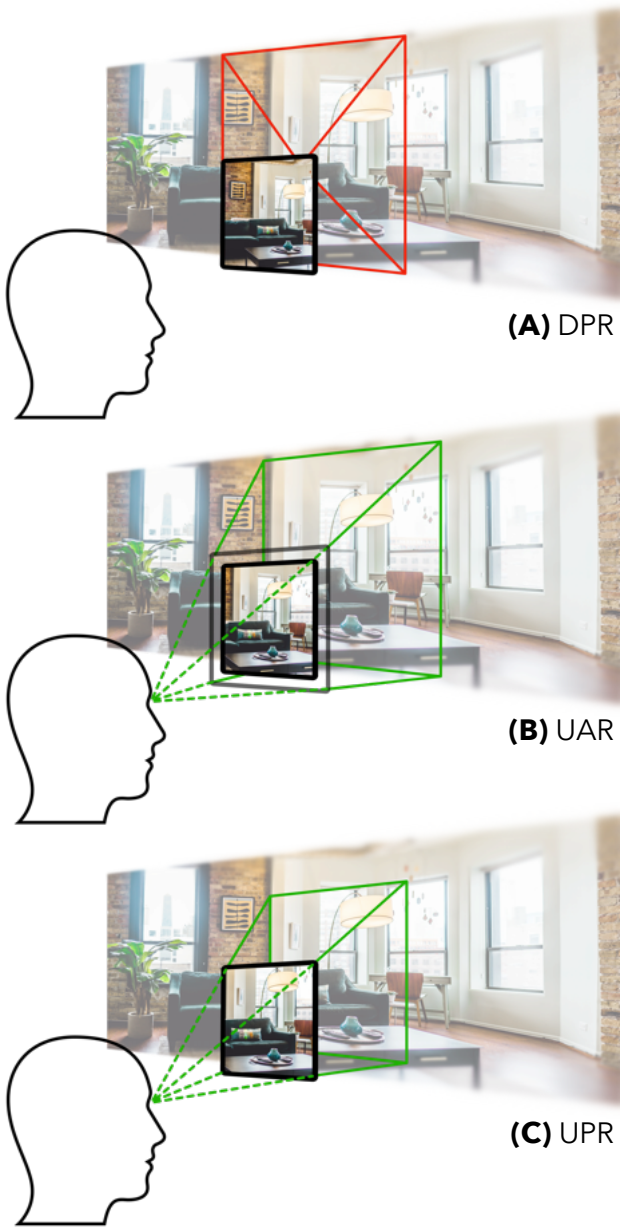


Figure 6.2: Comparison of the camera frustum when the user looks at the device at an angle. In DPR (A), the frustum of the virtual camera is completely defined by the hardware camera. In UPR (C), the virtual camera sits in the user's eye and is defined by the corners of the screen. Therefore, only a fraction of the actual camera image is visible. In UAR (B), we increase the size of the virtual screen (semitransparent frame around the device), which lets the user see a larger part of the camera image.

its screen corners are farther away from the center of the screen. Therefore, we created an AR implementation that supports UPR with an adjustable device size parameter, as well as DPR. We used the Unity graphics engine and tracking functionalities in Apple's ARKit to implement our prototype. We created a custom UPR rendering pipeline using techniques found in previous work. We adapted these implementations to support a variable device size. Tracking and rendering were performed at 60 Hz.

We created an AR prototype system that supports both DPR and UPR with an adjustable device size parameter to explore different UAR versions.

Camera Transform

The transformation matrix of the virtual camera can be derived directly from the head-tracking capabilities of ARKit. When users have both eyes open, we place the camera at the center location between both eyes, creating an image that addresses our two-eyed vision. If the user closes one eye, the camera is placed into the open eye.

With both eyes open, in UAR, the camera originates at the center between both eyes, right under the nose bone.

Camera Projection

The projection matrix P can be built from six parameters: the z distances of the near and far clipping planes (n and f) and the frustum extents on the near plane (t, r, b, l). We already saw the generalized projection matrix of UPR in Section 2.4:

$$P = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{n+f}{n-f} & \frac{2fn}{n-f} \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{6.1}$$

For UPR, the corners of the camera frustum need to be cast from the eye location through the corners of the screen. Geometrically, one can obtain these parameters as follows: n can be defined by the Euclidean distance of the vector along the normal of the screen between the device and camera

The parameters for the UPR projection matrix can be calculated from the estimations of ARKit.

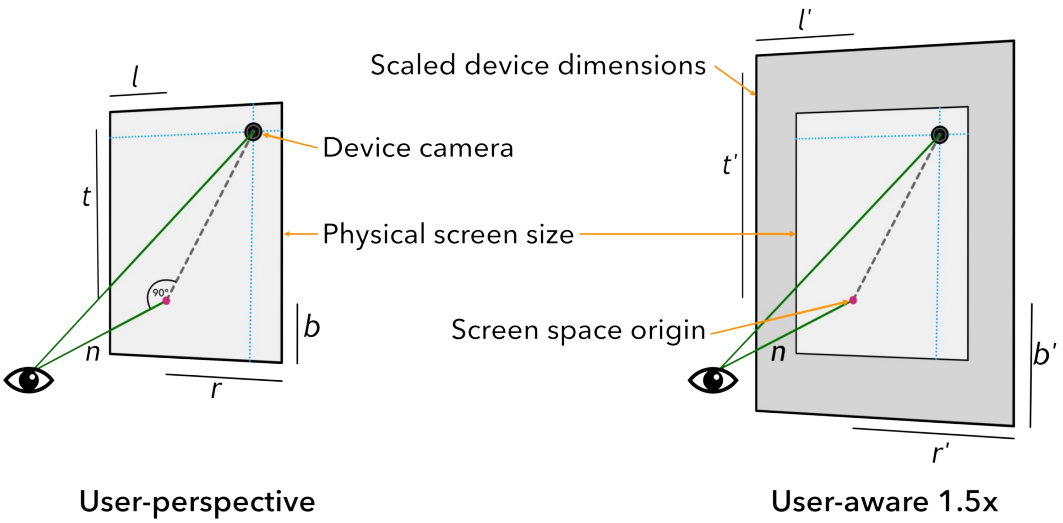


Figure 6.3: The screen space origin (pink dot on the light gray plane) is the closest point on the screen plane (light gray) to the eye. It is easy to calculate, as both the eye and hardware camera locations are given, and the normal of the plane is known (green). The distances between the camera sensor and screen edges (blue dotted lines) are constant, so one can easily infer the values for t, r, b, l. With UAR, the distances between the camera sensor and screen edges are scaled up, and by doing so, result in larger values for t, r, b, l and ultimately a larger FOV.

(eye) locations. t, r, b, l need to be the horizontal/vertical distances from the screen space origin (under the tip of the frustum) to the edges of the screen (see Figure 6.3). Thus, in UPR, $r - l$ is the physical width of the screen / clipping plane. ARKit provides the location of the hardware camera. One can infer the four coordinates of the screen corners by measuring the physical distances from the sensor to the individual edges of the screen in advance. This way, the output of the rendering algorithm shows exactly what is covered by the device in the vicinity of the user.

For UAR, a scale factor is applied to the parameters t, r, b, and l.

The goal of UAR is to provide a larger FOV. Thus, the coordinates of the screen edges used to calculate t, r, b, l are spaced further apart (see Figure 6.3). This is achieved by pretending we measured larger actual distances in the previous step.

Video Feed

For UPR and UAR, the planar camera image needs to be transformed to match the virtual content.

Lastly, the camera image needs to be placed in the background of the virtual content. This is easy for DPR, as the projection is already suitable for the camera. For UPR, this is harder, as the planar camera image has to be mapped behind the 3D content so that the anchoring required for AR holds. The back-facing camera of mobile devices offers developers planar images only, which, e.g., cannot be mapped to a skybox to be used as background for the virtual scene. We used the approach of Samini and Palmerius [2014] and placed the camera image on a plane behind the virtual objects.

We use the camera feed as the texture of a plane whose position adapts to the distance between device and virtual scene, respectively floor.

For our device, using their method resulted in an image plane that is 10×13 m large and roughly 10 m away from the screen. This approach works as long as the distance between the user's eyes and the device is negligible compared to the distance between the device and the observed object. For virtual scenes closer than 2 m in front of the device, the image plane needs to be positioned logarithmically further away from the device camera based on the camera intrinsics. To identify a suitable mapping, we manually adjusted the distance of the image plane d based on the scene depth s for 60 samples with a varying scene depth between 0.2 and 7.5 m and interpolated these data points.

$$d = 10 + 23.5e^{-1.85d} \tag{6.2}$$

Looking at an angle is enabled by rotating the image plane if needed.

In addition, users could look through the device at an angle. Therefore, the image plane must be rotated dynamically around the camera to counterbalance possible misalignments. Similarly to the work of Samini and Palmerius [2014], the horizontal rotation r_h of the image plane is calculated by using the following formula:

$$r_h = \alpha - tan^{-1}\left(\frac{0.5 \times w}{d} \times \frac{tan(\alpha)}{tan(0.5 \times HFOV)}\right) \tag{6.3}$$

where w is the width of the image plane, and α is the horizontal angle between the screen normal and the user's line of sight. The vertical rotation follows analogously.

Devices with LIDAR scanners could refine image plane placement further.

The approach of positioning the image can be refined further; e.g., Kyriazakos and Moustakas [2015] proposed a technique to segment the camera image into multiple layers based on the depth data that is available from devices equipped with LIDAR scanners. But even without LIDAR, we can still assume that the user visually focuses on the virtual content. Thus, we can optimize the alignment of the camera image to the virtual content by using the distance to the object in the virtual scene as a value for our scene depth s.

We fill out screen borders to which no camera feed is available in black color.

It is also possible that a user faces a part of the scene for which no camera image exists as a background. In that case, we decided to show black color as it blended in with the frame of the device we used. This is a hardware limitation that could be mitigated with wider-angle cameras. Yet, as the typical FOV during operation is still smaller than in DPR, there is usually enough leeway to operate a UPR system without seeing these unspecified areas.

We observed that this approach works sufficiently well under most usage postures.

Overall, with this technique, we observed good alignment of real world, camera image, and virtual content for scenes that are at least 25 cm away from the device. When the next physical surface is closer than 25 cm, visual quality deteriorates for different reasons. First, the image lacks resolution as we digitally zoom into a small part of the camera feed. Second, the camera feed can also become grainy or blurry due to cropping the camera image into a peripheral and slightly unfocused area of the camera sensor. Third, toward the corners of the planar camera image, the projection of the real and virtual world might disperse.

6.3.3 Scaling Factors

The scale factor of UAR provides a continuous gauge to increase the FOV.

Our idea of UAR was to extend the dynamic frustum of UPR to provide motion parallax with a larger FOV even when holding the device at arm's length. Therefore, our system applied a custom scale factor for the measured real-world device size when calculating the frustums. Thus, the scale factor provides a continuous gauge to increase the FOV of the system. As one can see in Figure 6.2, a UPR

frustum is calculated by casting rays to the four corners of the screen. By virtually scaling up the device size and thus relatively moving these points farther away from the center of the screen, the resulting FOV becomes bigger.

We used an iPad mini as its ultra-wide front-facing camera enlarges the space of facial tracking.

With mobile devices becoming larger and the expected uptake in foldables in the future, we picked an iPad mini with an 8.3″ screen running iOS 15 for our user studies. This device was specially chosen for its ultra-wide front-facing camera that tracks the user. In Section 2.7.3, we already saw that the iPhone camera system can track the user's head reliably within a cone of 30° for distances under 90 cm. The additional trackable area from the ultra-wide camera is beneficial for fast, physically interactive tasks like AR experiences.

We tested two different UAR scaling factors.

As we were not sure which scaling factor would turn out most beneficial for AR interaction, we tested two different ones, which are also visible in Figure 6.1. The rationale for the scaling factors is as follows:

At a typical viewing distance the FOV is 23° with UPR, 35° with UAR1.5x, 63° with UAR3.0x, and 70° with DPR.

The standard device-perspective AR is based on a wide-angle camera and offers roughly a 70° (vertical) FOV. UPR and UAR, however, have a dynamic FOV that changes based on the distance between face and device (smaller distance = wider FOV). At a typical viewing distance of 37.4 cm while standing and holding a smartphone [Boccardo, 2021] which we confirmed in our own preliminary observations, this results in a 23° FOV in *UPR* when using an iPad mini. To examine the impact of the FOV in UAR, we selected two scaling factors: *UAR3.0x* denotes a 3.0x device magnification, resulting in an FOV of around 63°, and thus comparable to DPR. On the other hand, we also wanted to test a magnification level between *UPR* and *UAR3.0x*. *UAR1.5x* halves this magnification to a 1.5x magnification of the device. It results in a noticeably larger FOV than UPR, which is still only about half as large (35°) as the DPR FOV.

UAR also hides alignment issues better than UPR.

One beneficial side effect is that with UAR techniques, slight issues in alignment are no longer as noticeable as with UPR, mitigating the horizontal alignment problem in two-eyed UPR usage.

We compared our UAR techniques against DPR and UPR in two user studies.

We conducted two user studies to find out how UAR performs compared to DPR and UPR and whether it can combine their individual strengths, not their weaknesses. Moreover, we analyzed the device and selected body movements to identify whether people change their behavior while operating UAR compared to DPR and UPR AR. Our research questions were:

RQ1 Can UAR convey cues that enhance depth perception similarly to UPR?

RQ2 Does the increased FOV of UAR simplify search and selection tasks compared to UPR?

RQ3 Does user-aware rendering impact the amount of body movement while using AR?

6.4 Study 1: Depth Perception

Study 1 investigated the effect of the kinetic depth cues provided by the different techniques on depth perception.

We conducted a first user study to better understand how the user-aware rendering techniques impact depth perception in handheld AR and the physical effort required. With our task, we aimed to evaluate to which extent the kinetic depth cues achieved from altering the camera frustum help to identify the order of virtual mid-air objects. We removed pictorial depth cues like shadows and reflections to trace the measured effects back to the visualization used. In this first user study, we had four conditions: We compared *UAR1.5x* and *UAR3.0x* with the two baselines (*UPR* and *DPR*).

12 participants aged 23 to 29 took part in the study (M = 25.9, SD = 3.13), eight male and four female. Four reported no previous usage of AR or VR. The others reported that they use AR or VR systems only occasionally.

6.4.1 Apparatus and Task

We created a set of AR scenes that contained a constellation of three virtual cubes each. The cubes could appear

Figure 6.4: In Study 1, participants observed a scene with three cubes and had to judge their apparent depth order.

in a space of 3.0×2.5×5.0 m^3 positioned 2 m in front of the participant. The diameter of the objects in a scene differed between .2 m and .8 m, so the relative size could not be used as a depth cue. All objects had at least .3 m distance to the floor, i.e., they were floating in mid-air so that the floor could not be used as a reference point. Due to the rendering and study setup, the depth cues our participants could use were the perspective projection, occlusions, and motion parallax from changing the mobile phone camera position (Figure 6.4). Object colors were randomized while ensuring a similar contrast and color saturation ratio. We asked participants if they suffered from color blindness to select a suitable color scheme.

The tasks consisted of participants judging the depth order in constellations of three cubes each. All cubes had a random size and were floating in mid-air.

Participants were asked to stand at a certain predefined location, observe the AR scene through the device, and determine the distance of the three virtual cubes from front to back. Objects were identified by their horizontal position in the scene and their colors. We asked our participants to be as precise as possible without becoming unnecessarily slow. People were asked to hold the tablet comfortably with either hand or bimanually, and were encouraged to rest their upper arm on their body to prevent fatigue. We fixed the lighting conditions with activated ceiling lamps over the virtual scene. The iPad display was fixed at 80%

Participants were allowed to move the device while standing at a predefined location freely.

brightness to ensure legibility without becoming too warm over the course of the study.

We used a within-subjects design and counterbalanced the order of tested techniques using Latin squares.

We used a within-subjects study design, i.e., all participants tested all four conditions. We counterbalanced their order using Latin squares, in which each condition will precede another condition exactly once. AR scene order was randomized in each condition. All conditions started with an opportunity for participants to familiarize themselves with the rendering technique and explore how the device reacted to their input. To allow for familiarization with ordinary 3D models without giving away information about the task, we created a scene with multiple barnyard animals placed on the floor across the room.

Participants filled out a post hoc questionnaire to explain which visual features they used to solve the task.

After each condition, we asked participants to rate their experience with this technique on 5-point Likert scales in a questionnaire. They had to rate which of the following four visual characteristics they found most helpful while solving the task: occlusion, motion parallax, anchoring of items in the real world, and grouping of virtual items. In addition, we asked them whether they found the task physically and mentally demanding, as well as whether the on-screen content was the one they intuitively expected, and whether they found the viewport to be restrictive, prohibiting them from seeing the whole scene comfortably.

To prevent learning effects, only a subset of constellations was the same in all conditions. Only these were used to assess depth perception.

Each condition consisted of a set of 18 cube constellations that were tested in random order. Due to learning effects, we could not use the same set of cube constellations across all conditions despite randomization. Therefore, we used seven measured constellations that were the same across all conditions. The other 11 filler constellations were specified randomly. Only the data of the measured constellations was used for evaluation.

6.4.2 Variables

The four techniques served as IV.

We used TECHNIQUE [*UPR*, *DPR*, *UAR1.5x*, *UAR3.0x*] as the main **independent variable**.

As **dependent variable**, we measured ***Depth Score*** as the number of correct relative orderings in the participant's answer, with three being the highest and zero the lowest score possible. We calculated ***Device Movement*** as the traveled distance the device was moved while solving the task. We calculated ***Head Movement*** as the traveled distance the head was moved while solving the task relative to the tablet. We also measured the ***Time*** it took a participant to solve the task after a scene became visible.

We measured a *Score* to compare depth perception as well as *Movement* and the *Time* required by participants to solve the task.

6.4.3 Results

In this study, we were interested in the effect that user-aware rendering TECHNIQUEs had on depth perception. To analyze the effect of TECHNIQUE on our dependent variables, we used one-way ANOVAs for evaluation and Student's t-tests for post-hoc pairwise comparisons on the aggregated data for each participant and technique.

UAR1.5x achieved the highest depth score (2.93), which corresponds to a correctness of 98%. Scores were similar in *UPR* (97%) and *UAR3.0x* (96%). The average success with *DPR* was only 90%. There was a significant main effect of TECHNIQUE on the ***Depth Score*** ($F_{3,33} = 3.430$, $p = .028$). *DPR* performed significantly worse than *UAR1.5x* ($p = .007$), *UPR* ($p = .015$) and *UAR3.0x* ($p < .029$). Other comparisons were not significant. The individual scores per condition are visualized in Figure 6.5.

Depth perception was significantly worse using DPR than any other condition.

We could not find a significant effect of the rendering TECHNIQUE on ***Device Movement*** ($F_{3,33} = 1.818$, $p = .163$). On average, our participants moved the device in similar amounts in all tested conditions. We measured mean distances of 2.5 m with *DPR*, 2.2 m with *UPR*, 2.1 m with *UAR1.5x*, and 2.0 m with *UAR3.0x*.

We measured similar device movement in each condition.

It also took our participants a similar ***Time*** to solve the task across all conditions. We measured an average of 13.4 s with *UPR*, 11.7 s with *UAR1.5x*, 11.1 s with *DPR*, and 10.4 s with *UAR3.0x*. Thus, we noticed that the task was solved

Also the *Time*s to solve the task were similar across conditions.

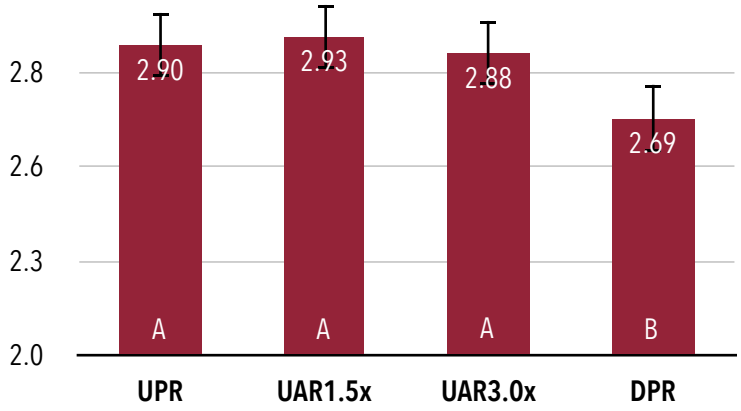


Figure 6.5: Mean *Depth Scores* measured in the four conditions. DPR is the only technique that does not react to head inputs. It resulted in significantly worse average depth perception than any other condition. Levels that do not share a letter (A, B) are significantly different (all $p < .05$). Whiskers denote 95% CI.

slightly slower in conditions with small FOVs. However, this effect was not significant ($F_{3,33} = 2.120$, $p = .117$).

Participants used the same depth cues in all conditions.

In the questionnaires, we asked participants to rate the visualization features they used to solve the task on a 5-point Likert scale (1 = completely agree). They reported a similar high use of occlusion effects (M = 1.3, SD = .7) and motion parallax (M = 2.0, SD = 1.0) across all conditions. Reference points in the real world close to virtual objects (M = 3.4, SD = 1.3) were used less frequently, also because there was no furniture in the area in which objects could appear. We could not identify a significant effect between techniques using a Wilcoxon signed rank test.

Our participants perceived the small FOV of UPR as restrictive.

When it comes to comments on the overall AR experience, our participants found that the size of the AR content prohibited them from seeing the whole scene in *UPR* (M = 1.9), a significantly worse experience than any of the other three conditions ($p < .02$). The intuitiveness of the visible camera frustum was not rated significantly different across conditions. Yet, *DPR* and *UAR1.05* were rated slightly better (M = 1.9) than the other two (M = 2.8). This is partially be-

cause people know the behavior of *DPR* "from taking photos". The virtual transparency of *UPR* irritated one participant, who asked why the camera zoom changes when moving her head.

6.4.4 Discussion

From related work we knew that UPR sets the standard for *Depth Score* in this task.

The task in this study tested how the different rendering techniques affected depth perception. In this study, *UPR* performed significantly better than *DPR*, increasing the success in depth perception from 90% to 97%. The good performance of *UPR* was expected from the related work by Kline and Witmer [1996].

While UAR3.0x has a similar FOV to DPR, we measured a *Depth Score* similar to UPR.

As *DPR* suffers from the dual-view problem, it varies from *UPR* in its FOV, the angular offset, and the center of the screen capture. Our *UAR* techniques all have different FOVs, yet they mitigate the other two aspects. Overall, we anticipated a decline in depth perception with a larger FOV, as this results in objects becoming smaller on screen, making depth cues harder to see. However, with *UAR3.0x*, we measured similar depth perception scores as with *UPR*.

We might have obtained larger differences in the results with a more difficult task.

Overall, the results were quite similar across all techniques. A possible reason for this is that our scenes often made it possible to achieve perspective overlapping between objects and thus being too simple. While using smaller objects could have increased the task difficulty, we still found *DPR* to perform significantly worse than the other techniques.

Depth cues from motion parallax were usable independent of the FOV.

We used this setup to answer whether user-aware rendering is able to preserve this strength of UPR (RQ1). In our study, both *UAR1.5x* and *UAR3.0x* performed significantly better than *DPR* on average. Both UAR techniques were able to match the score of *UPR* (see Figure 6.5). This shows that the depth cues obtained from motion parallax were still usable independent of the increased FOV from our techniques.

DPR performed significantly worse than any other tested technique. Yet all other techniques leveraged continuous

Motion parallax seems less effective when initiated by hand instead of the head movements.

head tracking to change what is visible on screen, which resulted in a constant input stream of subtle motion parallax effects. While these effects are the likeliest explanation for this difference, it is important to note that motion parallax is also obtained in *DPR* from moving around the device, and our participants all moved around a lot during operation; e.g., the tablet was moved 2.5 m on average in *DPR*.

No technique harmed usability.

Considering the similar time and movement across conditions and the questionnaire data and comments, this suggests that all rendering techniques were usable with ease and that participants did not have to adapt their usage patterns for our user-aware techniques. Overall, this study answered RQ1 positively.

6.5 Study 2: Searching and Selecting Objects

Study 2 investigated search times in large virtual scenes. Thus, DPR benefits from its large FOV in this task.

In many AR application domains, the virtual scenes can be too large to fit on one screen, e.g., when working with large data sets, virtual desktops, or reconstructions of buildings. Ren et al. [2016] showed that larger models can complicate searching elements inside of them. To find out to which degree our user-aware techniques retain the fast scene exploration known from DPR, we conducted another study. This study was conducted after a 10 min break with the same group of participants from Study 1. Again, we used a within-subjects design in which our participants tested the four conditions in a counterbalanced order, using a different Latin square than in Study 1.

6.5.1 Apparatus and Task

We conducted a search and selection task similar to Baričević et al.

We adapted the design of the search and selection task that Baričević et al. [2012] used in VR for use in handheld AR. Participants were asked to stand in front of a 1.6×0.7 m table and select a virtual object by tapping it on the screen. Targets could only appear in the mid-air space up to 0.5 m above the table.

Figure 6.6: The three phases of the task in Study 2. Left: Participants had to tap on the red ball to spawn a target on the table. Middle: During the search phase, the frame was yellow as long as the target was not visible on the screen. Right: When the target was in the viewport, the frame turned green, and participants tapped on the target to select it. The images depict DPR.

To make a new target appear, participants needed to tap a homing object next to the table.

To spawn a new target, participants had to tap a virtual ball to their right. As this ball was located below the table, it was intended to function as a homing target that shifted the AR frustum away from the table so that targets needed to be searched for and were not immediately visible in front of the user.

The state of the task was indicated through the color of the border around the display.

A border around the display indicated what to do next (see Figure 6.6): A red border implied that no object was visible and the homing target needed to be tapped. A yellow border was visible during the search phase when the object appeared over the table but was not visible in the current frustum. A green border denoted the selection phase, which started once the object was visible on the screen for the first time. Our participants had to stand at most 1.2 m away from the table. Thus, even with DPR, only parts of the table fit onto the screen at once, as seen in Figure 6.6.

A visible score was intended to engage participants to be as fast as possible.

The task was designed to minimize possible interaction effects of the task design by reducing the interaction to panning over a virtual scene and tapping an object. Participants were instructed to select the targets as fast as possi-

ble. To keep them engaged, we displayed their score at the top right corner of the screen.

We conducted a trial and two measured runs for each condition.

For each condition, we conducted three runs. The first one allowed participants to explore the technique and learn how to operate the system best. As suggested by Baričević et al. [2012], this was limited to at most 8 min. Afterward, two runs taking 2 min each were made with a short break between them. We sampled 12 arrays containing 100 random target locations each once before conducting our studies. These 12 arrays were hard-coded in the software and used as the locations in the study for all participants in their individual runs. Thus, all participants saw the same target locations in their n-th run. In a questionnaire after each condition, we asked them to rank on 5-point Likert scales which parts of their bodies they moved the most while solving the task. Options included tilting of head, hand, and movement of forearm, neck, and torso. We also asked them to describe their search strategy in their own words briefly.

6.5.2 Variables

TECHNIQUE and PHASE served as independent variables.

We used TECHNIQUE [*UPR*, *DPR*, *UAR1.5x*, *UAR3.0x*] as the main **independent variable**. The two different PHASEs [*Search*, *Selection*] were logged independently for additional analysis.

We measured *Time*, *Device Movement* and *Head Movement* as dependent variables.

Time was measured for both phases. For the search phase, this is the time it took from tapping the homing ball till the target cube was rendered in AR on at least 1 px of the display. In the selection phase, it is the duration between the end of the search phase until the participant tapped on the cube. We calculated ***Device Movement*** as the traveled distance the device was moved [m] while solving the task. Moreover, ***Head Movement*** denotes the traveled distance the head was moved [m] while solving the task relative to the tablet.

6.5.3 Results

To analyze the effect of TECHNIQUE on our dependent variables, we used one-way ANOVAs for evaluation and Student's t-tests for post hoc pairwise comparisons. Overall, we obtained over 11,000 measures for analysis. To make the statistical test more reliable, we calculated the mean values for each participant, technique, phase, and run. We performed the analysis on the data averaged over both runs.

Participants were the slowest using UPR, similarly fast with DPR and UAR1.5x, and fastest with UAR3.0x.

There was a significant effect of TECHNIQUE on ***Time*** ($F_{3,33}$ = 82.268, p < .0001). *UPR* was slowest. Using this technique, it took our participants 2.05 s on average to select a cube after it appeared in the scene. With similar average times of 1.50 s and 1.41 s, *UAR1.5x* and *DPR* performed significantly better than *UPR* (p < .0001), and not significantly different from each other. *UAR3.0x* was faster than any other technique (p < .0001). However, this measurement has to be interpreted carefully, as this is due to the annihilation of the search phase with *UAR3.0x*: With all other techniques, the average search duration was 0.54 s (green parts in Figure 6.7). Using *UAR3.0x* this measurement dipped to 0.11 s. This is because, over the course of the study, all participants found a way to hold the device so that both the table and the homing target were visible at once, thanks to the dynamic frustum of UAR.

Device Movement exposed the same differences as *Time*. UPR required significantly more movement than any other condition. UAR3.0x required significantly less movement than any other condition.

There was also a significant effect of TECHNIQUE on ***Device Movement*** ($F_{3,33}$ = 21.790, p < .0001). Device movement helps us understand how bodily the search interaction was using each technique. *UPR*, which already took the longest, also required the most device motion: 46 cm on average. *UAR1.5x* (33 cm) and *DPR* (27 cm) required significantly less device movement (p < .001). As mentioned above, *UAR3.0x* was used rather statically (9 cm on average) and cannot be appropriately compared here. Fig. 6.8 shows mean device movements during both phases. TECHNIQUE only had a significant effect on ***Head Movement*** when comparing the static usage of *UAR3.0x* with the other conditions.

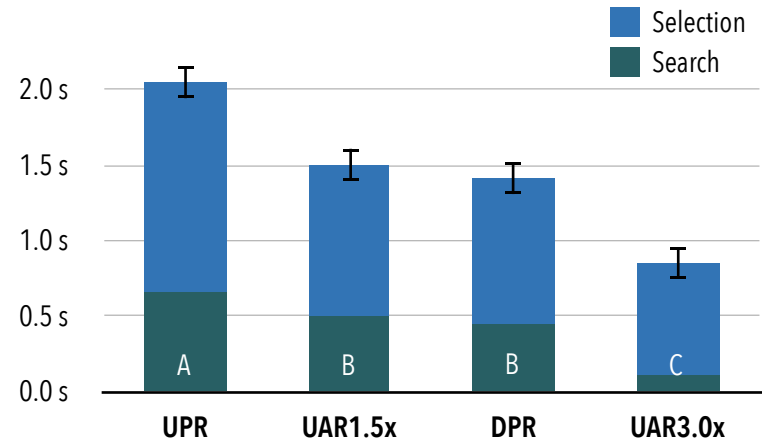


Figure 6.7: Average time [s] to search and select a target by condition. The two phases are color-coded into the graph. Using UPR, which offers the smallest FOV, our participants were the slowest. UAR3.0x required close to no search time because our participants held the device in such a way that the whole table fit on the screen. Levels that do not share a letter (A, B, C) are significantly different (all $p < .001$). Whiskers denote 95% CI.

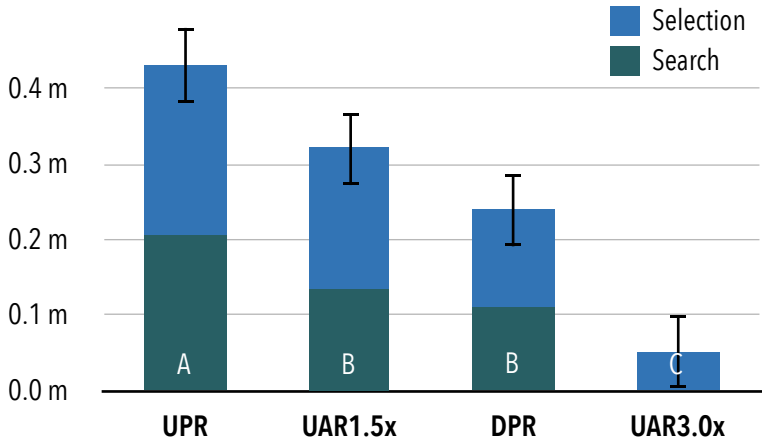


Figure 6.8: Average device movement [m] (right) to search and select a target by condition. The two phases are color-coded into the graph. Using UPR, which offers the smallest FOV, our participants had to move the device the most. Levels that do not share a letter (A, B, C) are significantly different (all $p < .001$). Whiskers denote 95% CI.

As part of the task, participants went over the scene from right to left while searching for the target. During their first run with this technique, all participants found a way to hold the device with *UAR3.0x* in such a way that the majority of the scene and a part of the homing target were visible on screen. To do so, they decreased the distance between their eyes and the device, enforcing ultra-wide FOVs up to 100°. Thus, "no search was required": Upon tapping the homing target, a cube appeared on the screen. Its appearance was a strong visual cue to directly spot its location. Again, our participants described *UPR* as too restrictive in its narrow viewport.

The reason for UAR3.0x requiring close to no *Device Movement* during search and also the shortest *Time* was that participants abused the dynamic FOV by holding the device close to their face.

As the interaction with *UAR3.0x* was not representative of the intended usage, we left it out in the analysis of ranked body parts involved in controlling the AR system. Across the other three techniques, people preferred to rotate with torso movement (M = 1.6, SD = 0.2) and use their forearm (M = 2.5, SD = 0.2) to further adjust the frustum. Device tilt, head, and neck (M = 3.6, SD = 0.2) were not controlled actively but "rather subconsciously".

Participants responded they especially relied on torso and forearm movement to solve the task.

6.5.4 Discussion

Based on the visual characteristics of the different techniques, we already expected *DPR* to outperform *UPR* in this task. When combining search and selection time, *UPR* (2.0 s) took 43% longer than *DPR* (1.4 s) on average. This difference is due to the different characteristics of these techniques. First, the FOV of *UPR* is the smallest across all tested techniques. Second, the part of the virtual scene visible on screen can easily be influenced by rotating the device in *DPR*, but it only changes slightly in *UPR*. Instead, in *UPR*, the angle from which one is facing the device makes a difference. This difference in usage pattern is also visible when looking at the traveled distance while solving the task: *UPR* required more movement as the search was complicated with less FOV, and (in-place) rotations having no effect on what is visible through the screen.

From related work we knew that DPR sets the standard for fast search and selection times in this task. To no surprise, UPR was significantly slower than DPR.

UAR1.5x was able to match the performance of DPR. UAR3.0x even outperformed DPR, yet with our participants cheating.

When adding *UAR1.5x* and *UAR3.0x* into the mix, we see that both search and selection times, as well as device movements, decreased significantly the larger the FOV of the AR system was. Especially notable in the data is the non-existing search phase with *UAR3.0x* as a result of the large FOV. However, these measures are incomparable to the other techniques, as all participants found a way to abuse *UAR3.0x*. In order to solve the task faster by reducing the movement required, they moved the device in such a position close to their face that most of the table and a part of the homing target were visible on screen. By doing so, upon tapping the homing icon, the target directly appeared inside the visible part of the scene. The device movement during the selection phase was then likely performed to better reach the target with the thumb for selection.

Our data suggests that good performance in this task arises from a large FOV. It is not relevant whether the frustum originates in the device camera or the user's head.

Thus, regarding RQ2, it makes more sense to compare *UAR1.5x* with the two baselines. With a combined search and selection time of 1.5 s, this hybrid technique was similarly fast as *DPR*. Significance tests prove that both were significantly faster than *UPR*. This is especially interesting as the foundation of user-aware rendering is in *UPR*: While one could expect that losing the ability of *DPR* to quickly pan over the scene by rotating the device in place might have a negative impact on performance, this was not the case. Looking at the graphs, one can rather see a relationship between FOV and search and selection times. This makes it likely that user-aware AR with a device scale factor between our two versions could yield an even better result than *DPR*.

The difference between *UAR1.5x* and *UPR* was already 0.5 s for a very simplistic task of tapping on a box.

Just like in study 1, the effect size between the best and worst performing techniques is rather subtle. Search and selection using *UAR1.5x* took only 0.5 s less than *UPR*, which still is a speed increase of 25%. However, one must also consider that our task design was very simplistic. As AR tracking and rendering are usually provided by a system library and not required to be implemented by app developers, modern versions of AR toolkits could leverage head input for enhanced AR perception without an effort required by app developers.

6.6 Conclusion

We combined UPR with a larger frustum similar to DPR and evaluated this novel rendering technique in two user studies.

This chapter introduced *User-Aware Rendering*, a new approach to handheld augmented reality. UAR combines the better visual context and depth perception of user-perspective AR with the larger FOV of traditional device-perspective rendering that enables fast scene exploration. The technique works by calculating UPR frustums with virtually increased device sizes. We tested two scaling factors: 1.5x and 3.0x.

In each study, UAR1.5x was able to match the performance of the respective known gold standard for this task.

Our two studies provide valuable insights into how people perceive content in handheld AR. Each study focused on an individual known strength of *DPR* and *UPR*. Study 1 suggests that the additional motion parallax effect obtained from the head tracking positively affected the depth perception (RQ1). Study 2 shows us that a large FOV is also important to quickly search for an object in the virtual scene (RQ2). On the other hand, our participants could not use the angular offset of *DPR* to an advantage in decreasing search times. We have seen no change in head or device movement in Study 1 while operating the techniques, showing that they allowed for natural usage (RQ3). In Study 2, larger FOVs resulted in reduced movement to solve the task, proving that the idea to scale up the virtual device size made sense. Our user-aware techniques, especially *UAR1.5x*, were able to combine the strengths of *UPR* and *DPR* in these studies by mitigating the dual-view problem without lowering the FOV.

UAR1.5x combined the enhanced depth perception of UPR with fast search and selection times of DPR.

In our two studies, UAR was able to match the performance of the respective existing favorable rendering technique [Kruijff et al., 2010; Liu et al., 2020; Swan et al., 2017; Ren et al., 2016; Baričević et al., 2012]. The design of both studies was reduced to very concrete aspects of perception. We saw in Study 1 that a constant input stream of subtle motion parallax effects enhances depth perception. In Study 2, we measured the positive impact of a larger FOV on scene exploration. Real-world AR experiences require both of these aspects: Estimating the distance and order of multiple objects and getting an overview of the overall constellation of objects is relevant for any AR experience. Thus overall,

UAR provides an interesting technique that combines motion parallax with a large FOV.

A FOV of around 35° seems to be a sweet spot for handheld AR.

Both the comments our participants made and the quantitative measurements suggest that the FOV of around 35° one obtains with *UAR1.5x* at a typical usage distance between device and head offers a sweet spot at which users can see enough content on screen (Study 2) while still benefiting from the added motion parallax for depth perception (Study 1). Thus, we attribute the enhanced performance measured with UAR to the more useful FOV it provides at a comfortable viewing distance.

6.7 Future Work

For the future, we envision further hybrid rendering techniques that are less computationally expensive than UAR.

Our studies show that hybrid rendering techniques like UAR can combine large FOVs with motion parallax from head input to enhance the overall AR experience. UAR, however, is not the only imaginable implementation that can achieve this combination. We can also envision a rendering technique based on DPR instead of UPR that pans into specific areas of the captured camera. This approach would result in a similar amount, yet a different type of motion parallax effect. Thus, it could lead to a performance similar to *UAR1.5x* while being less computationally complex.

UAR with a dynamic scale factor based on the content could also enable new interactions with AR.

Our UAR approach also tested only two possible scaling factors for user-aware rendering. In our studies, we often found a relationship between the FOV and the performance. Alternating this scale factor offers further interesting research trajectories. We could also envision a dynamic scale factor based on the visible virtual content, where the device would zoom in to show smaller scenes and increase the FOV for large scenes.

UAR should be evaluated with other screen sizes, too.

One should also analyze user-aware rendering on other screen sizes. In our preliminary tests, the prototype also worked well on a phone. However, the impact of device size is likely small, as according to Paillé [2015], operating distance decreases with a smaller device. Consequently, the

relative size of the area blocked in the user's field of vision remains steady.

Finally, the transformation of the background image based on the floor and model distance worked well for us. Still, there were minor issues in this perspective transformation, which could be refined by using a different (fish-eye) camera. None of our participants mentioned any alignment issues of the virtual content in the real world. However, the black borders around the camera image that were especially present when *UAR3.0x* ran out of camera feed should be tackled with different camera technology.

The transformation of the camera feed to match the virtual content could be refined further.

Overall, this chapter serves as a first exploration of our concept of User-Aware Rendering. The two studies showed that it was able to combine the advantages, but not the disadvantages, of UPR and DPR. At the heart of UAR is a continuous and implicit processing of the facial tracking data. This data is used to update both transformation and projection matrices, enhancing the perception of the virtual world-space contents. In the next chapter, we will shift our focus to an implicit usage of gaze tracking for screen-space content to complete our taxonomy of interaction techniques.

UAR shows that the perception of virtual world-space content is enhanced through implicit facial tracking. Next, we shift our focus back to screen-space content.

Chapter 7

Optimizing Distraction on Mobile Devices with Gaze Analysis

SUMMARY:
Notifications on smartphones typically appear at the top of the screen, resulting in interruptions caused by content overlaps of toolbars and potential accidental activation of a notification. As returning to a workflow that got interrupted proves difficult for the general user, interface designers should thoughtfully design the visual disruption caused by notifications. We explore possible designs of *gaze-attentive notifications* to overcome this issue. By placing the notification banner as far from the user's current gazing point as possible they result in less visual overlap and our study participants experienced them as less distracting.

Publications: The work presented in this chapter was done in collaboration with Eunae Jang and Jan Borchers. The author of this thesis developed the research idea and relevant research questions. Furthermore, he designed and evaluated the study. Most of this work has been published as late-breaking work at ACM MobileHCI 2023 [Hueber et al., 2023]. The author of this thesis is the main author of the paper. Most sections in this chapter are taken from the paper publication.

7.1 Motivation

Implicitly controlling screen-space content with gaze will result in the Midas touch problem.

Any direct usage of gaze as input modality in 2.5D user interfaces always results in the Midas touch problem: It is simply not possible to look at something without activating it. In our two explicit interaction techniques for screen-space content, we overcame this issue by leveraging a combination of gaze and touch. For instance, the touch-based selection in Headbang in Chapter 3 even added to the explicit nature of the interaction. For implicit interactions, however, that is not possible.

Therefore, we explore the use of gaze as anti-location of the interaction instead.

With the work in this chapter we set out to explore the effects of inverting the meaning of the gazing location on the interaction: What if we only place new windows in screen areas the user is not looking at? Mobile devices provide an especially promising testbed to investigate the potential of face tracking in determining an on-screen area of user interest for two reasons. First, they have a small screen that fits into the visual field well. Second, they frequently have alert windows (notification banners) appear while using them.

In multiple studies, notification banners were perceived as distractive.

Pielot et al. [2014] found that smartphone users receive over 60 mobile notifications daily. Still, the number of notifications people actually react to is way lower, with some surveys reporting reaction rates under five percent. Reasons for these low reaction rates can be found in studies by Mehrotra et al. [2016] and Sigitov et al. [2016]. Their findings indicate that users experience notifications as an interruption from or a distraction to their ongoing tasks.

Different projects tried to optimize the distraction and acceptance of notifications by delaying their delivery.

Designing mobile notifications to incorporate a suitable level of distraction has been an ongoing research challenge. In the study of Avraham Bahir et al. [2019], the click-through rates increased significantly the later they were received over the day. Fischer et al. [2011] proposed to delay the delivery of notifications based on usage patterns. For instance, a notification could be displayed when the user just finished a task. However, delaying the delivery is not suitable for many types of information. A field study by Iqbal and Horvitz [2010] found out that people actually

Figure 7.1: *Gaze-explicit* notifications reduce distraction and occlusion of the primary on-screen content. When a notification is about to be presented (left), the system checks the user's current gazing location to determine an area of her interest (orange) where no notifications are supposed to appear. The notification then appears on the most distant screen edge. By shifting her gaze toward the notification, a user can enlarge it, revealing more content or additional options.

value the awareness of information supplied by notifications of different importance levels.

To optimize for distraction without delaying information delivery, we used gaze tracking.

In this chapter, we explore gaze tracking to enhance the presentation of notifications so that information is delivered as timely as possible but with less distraction from a primary task. We explore both the presentation characteristics of notification banners, like contrast levels and size, as well as how explicit gaze interaction can enhance notification interaction. We also present qualitative feedback that helps to design further iterations of gaze-attentive notifications.

Overall, the two research questions for the work in this chapter are as follows:

RQ1 Can gaze tracking reduce undesired content overlaps when presenting notification banners?

RQ2 Can gaze tracking be used as an effective input modality in the context of notification UIs?

7.2 Related Work

Notifications interrupt a primary task through visual overlaps and auditory interference

While using a smartphone, notifications deliver additional information that typically is unrelated to a currently ongoing primary task. Hence, notifications require the user's secondary attention and distract from the ongoing task through visual overlaps [Bahr and Ford, 2011] or auditory interference [Stothart et al., 2015].

Even though they are despised for being disruptive, users value the information awareness from notifications.

Despite users finding notifications disruptive, they opt into them. The studies by Iqbal and Horvitz [2010] and Chang et al. [2023] show that notifications provide an increased information awareness that smartphone users value. Thus, it is also no surprise that completely disabling notifications, like Fitz et al. [2019] did in a user study, results in increased anxiety and fear of missing out.

7.2.1 Distraction Caused by Notifications

Resuming to an interrupted task is challenging.

The diary study of Czerwinski et al. [2004] showed that the task-switching required to return to a previously interrupted task is difficult for humans. Mobile notifications interrupt us frequently each day. Therefore, researchers explored different approaches to minimize the perceived disruption and improve the use of mobile notifications.

Balance the tradeoff between disruption and noticeability of notifications based on subjective importance could be promising, but no heuristic exists yet.

As proposed by Sahami Shirazi et al. [2014], one intuitive approach could be to filter notifications based on their subjective importance. Their assessment identified communication apps and calendar apps as especially important. This is in line with the earlier finding of Fischer et al. [2010], in whose study people were more receptive to notifications whose content provided them with a good gut reaction. However, no general heuristic exists to determine the importance of notifications from arbitrary other apps.

Another approach to reducing disruptions—and thus annoyance and frustration—caused by notifications is to optimize the timing of their arrival. For desktop interfaces, Adamczyk and Bailey [2004] propose interrupting users af-

ter finishing a task but before the save process. Fischer et al. [2011] found out that users deal with mobile notifications faster if they are delivered after an episode of mobile interaction, i.e., after finishing a task or at least subtask in a single screen of the mobile UI. Chen et al. [2022] analyzed the intensity of activity and engagement during VR sessions to predict when users can best be interrupted by notifications. Ogawa et al. [2021] used IoT devices to identify breakpoints in daily routines to time the delivery of notifications on a smart speaker.

Different approaches delayed notification delivery to opportune moments, e.g., after the user finished a task.

By identifying physical or mental context, a phone can obtain hints about when to deliver notifications. The Android library *InterruptMe* by Pejovic and Musolesi [2014] used user activity, location, and current time, among other indicators, to delay notification delivery. In their study, these delayed notifications yielded faster response times and higher satisfaction. Mehrotra et al. [2015] used the notification's content and social relationship between the user and its sender to classify its relevance to the user using machine learning. But also context-agnostic systems, like Fitz et al. [2019] did with batching notification delivery to a few selected points in time, can result in a slight productivity gain and less distraction.

Notifications delivered at such opportune moments yield faster response times.

7.2.2 Perception of Notifications

The visual appearance of notifications is an important aspect that allows them to fulfill their purpose. To engage a sufficient level of perception, especially when notifications are displayed in the peripheral vision, previous literature suggested visual enhancements of notification placement [Rzayev et al., 2019], extents [Janaka et al., 2023], and color [Mairena et al., 2019] to reduce users' reaction times or preference.

Visual characteristics of notifications also influence reaction times.

Avraham Bahir et al. [2019] examined the effect of visual manipulations of mobile notifications on users' reaction times. While disadvantages of adding graphics or images to notifications include covering more screen space and

Users are more likely to respond to notifications with images or emojis.

adding more clutter on smartphones, they also increase response rates.

Pulsing glows around notifications effectively capture the user's attention.

Tasse et al. [2016] conducted a desktop-based user study to identify the effects of 15 different types of visual attention grabbers using different combinations of visual factors such as color, position, size, and animation. They measured their participants' reaction times for each visual design while they played a memory game as a primary task. The more noticeable—and thus obtrusive—a notification was, the faster the measured reaction time. Overall, they recommend pulsing glowing shadows as the most likable and effective way to capture the user's attention.

7.2.3 Awareness of User's Gazing

Due to the differences in the visual field's acuity, results from desktop studies do not transfer to mobile.

The study by Klauck et al. [2017] already presented that the distance between the gazing point and a notification affects the user's attention and subjective distraction. Much previous research focused on the visual perception of notifications on large screen setups and even VR. Due to the intrinsics of human vision, these results might not be easily transferable to phones and their smaller screens. Nonetheless, as smartphones have become bigger in recent years, their screen edges move into mid-peripheral vision at a typical usage distance (see Section 2.3). Thus, notifications are displayed in the area of the user's visual field where color perception and acuity already diminish. However, changes in screen brightness caused by a notification appearing highly trigger the perception in mid-peripheral vision. Therefore, notification designs with a smaller footprint should also reliably capture the user's attention. What is more, reading a notification is impossible without moving the visual focus closer to it.

7.3 Designing Attentive Notifications

It is important to state that external events trigger notification delivery—thus, their delivery is not under the

user's control. When notifications appear right under the user's fingertips, they are prone to trigger accidental inputs [Guleria and Kaur, 2021]. This issue worsens as on modern smartphone platforms, notification banners cover the whole toolbar at the top of the screen, overlapping important navigation and functionality buttons.

Notifications overlap a commonly touched screen area. This increases the chance of accidental input.

Gaze tracking offers a promising way to mitigate this issue: As a user's gazing provides indicative information on her intention or likely next action within a second, Çığ and Sezgin [2014] were able to automate mode-switches in pen-based interactions simply from gaze analysis. As we have already shown in Section 2.7.3, the high-resolution front-facing cameras in recent smartphones allow for gaze tracking with sufficient accuracy, especially as we are only interested in the region of the screen the user is gazing at.

Gaze tracking with sufficient accuracy can predict the user's next touch target within a second.

The interaction design of mobile notifications is a combination of factors influencing their visual design and delivery process [Pejovic and Musolesi, 2014], and enhancing current notification design requires looking into all of them. First, visual aspects (size, contrast, ...). For instance, the work of Klauck et al. [2017] showed that reducing the visual footprint of notifications tends to be less disruptive and reduces content occlusion. Second, introducing and mapping gaze data (notification position, size, ...). Interaction effects between these factors are also likely. For example, increasing the spatial distance between the location the user is interacting on screen and the notification banners should have a similar effect to size reduction. Third, the possibility of using gaze explicitly as input arises. To find out more about how size and contrast influence perception and disruption, we conducted a preliminary study.

Different factors influence the interaction design of notification banners, including their visual intrinsics, placement, size, and input modality.

7.3.1 Exploration of Visual Factors

Apparatus and Task

In this study, 10 participants aged between 20 and 30 (6 male) were given an iPhone XS on which notification banners of different styles arrived silently during usage.

We conducted a study on visual factors with 10 participants.

Figure 7.2: The nine different notification styles explored in the preliminary study differed in their size (from left to right: small, medium, large) and their contrast level (from top to bottom: low, medium, high).

Participants watched a video of their interest and had to dismiss distractive notifications.

They watched a video of their choice in landscape orientation, which had to be at least 20 min long. The personal selection of a video was intended to capture the individual interest and guide the focus on the video content. Participants were asked to dismiss notifications by swiping whenever they noticed them on the screen.

The independent variables were SIZE and CONTRAST.

We used two **independent variables** in this study. Notifications were presented in three different SIZEs [small (23×10 mm), medium (60×10 mm), large (60×20 mm)] and three different CONTRAST levels [low (3:2 contrast, 70% opacity), medium (3:1 contrast, 85% opacity), high (7:1 contrast, 100% opacity)], as depicted in Figure 7.2. Each combination was tested three times with each participant in a random order in which the same condition was not presented two times in sequence. For reference, the design of the standard iOS notifications matches our look of a medium-size high-contrast notification.

We measured the *Perception Time* as the time participants needed to touch a notification after it became visible.

We measured the ***Perception Time*** as the time between the notification beginning to animate on screen and the moment the participant began to dismiss the notification. Moreover, participants rated their perceived distraction level and preference in a post hoc questionnaire on a 5-point Likert scale.

Results

To analyze the effects of SIZE and CONTRAST on the mean response times, we used two-way ANOVA for evaluation and Student's t-tests for post hoc pairwise comparisons on the aggregated data.

SIZE had a significant effect on the ***Perception Time*** ($F_{2,18} = 10.379$, $p < .001$). Pairwise comparisons revealed that all three size classes performed significantly different ($p < .03$). On average, we measured 2.0 s with large notifications, 2.4 s with medium-sized notifications, and 2.8 s with our smallest notifications.

With each increment of SIZE notifications were perceived significantly faster.

CONTRAST, on the other hand, had no significant impact on the ***Perception Time*** ($F_{2,18} = 2.843$), $p = .085$). We measured average response times of 2.3 s with medium and high contrast and 2.5 s with low contrast.

CONTRAST had no significant effect on the *Perception Time*.

There was also an interaction effect of SIZE×CONTRAST ($F_{4,36} = 3.512$, $p = .016$). Small-size low-contrast notifications were perceived significantly slower than any other notifications ($p < .01$). However, small-size medium-contrast notifications were already not perceived significantly slower than large high-contrast notifications ($p = .066$).

Small-size low-contrast notifications were perceived significantly slower than any other condition.

The ratings of our participants suggest that their self-reported distraction correlated to their response times (see Figure 7.3). However, our participants also mentioned that not much content will fit into the small notifications, limiting their use: *"I do not think many messages will fit into that."* Regarding the SIZE, six participants responded that they preferred the small-sized notifications most, especially as they cover less screen real estate: *"The small ones blended in nicely with the video, so that both contents can coexist."*

While our participants liked small notifications, they wondered how much content would fit into them.

The data of the preliminary study shows promising optimization potential for the visuals of mobile notifications: Small notifications with sufficient contrast seem to provide a good trade-off between perception and screen occlusion. Even without the additional sound cue, they were perceived less than a second slower than the standard iOS no-

Small-sized medium-contrast notifications were perceived less than 1 s slower than the default iOS style.

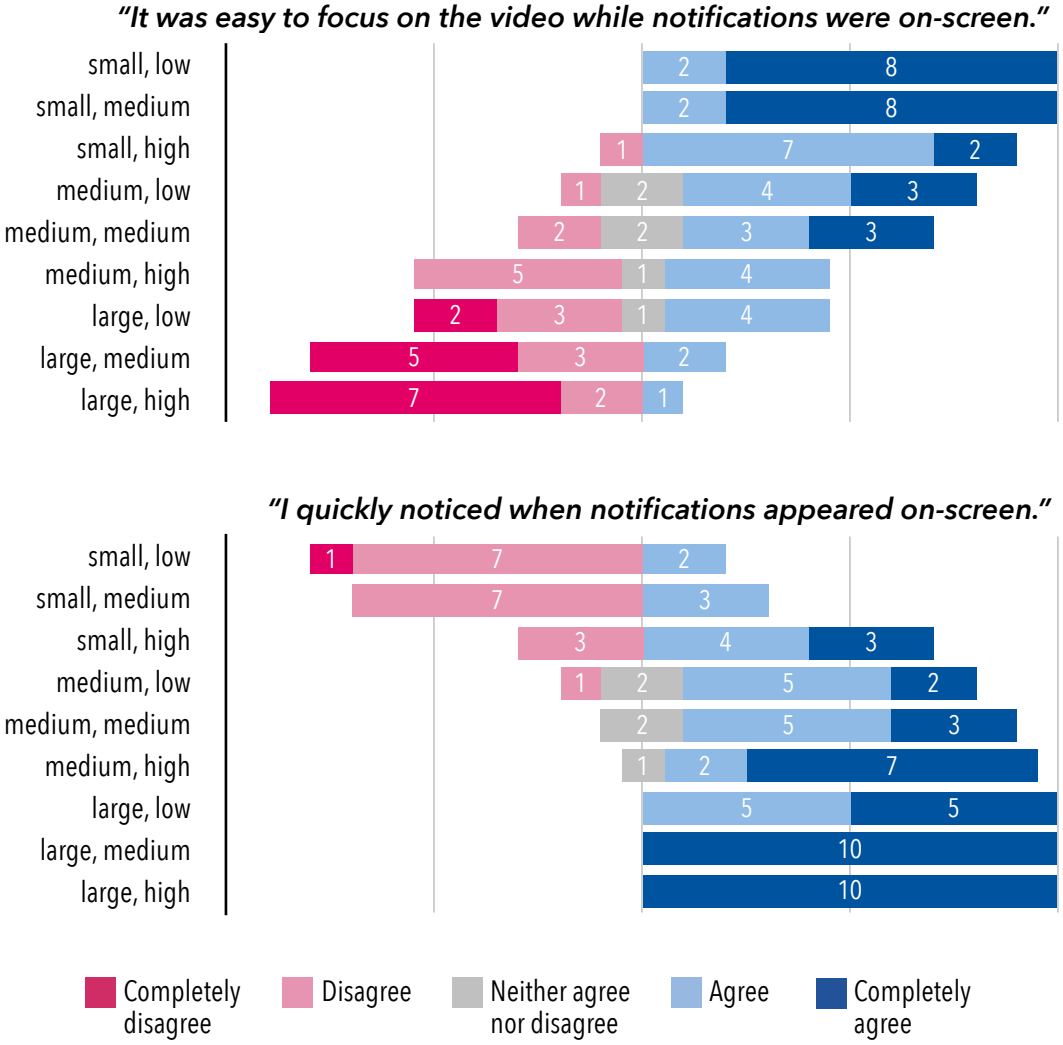


Figure 7.3: participants were asked to specify their agreement to statements on 5-point Likert scales after the preliminary study. The responses regarding the distraction (top) and perception (bottom) of different notification styles are presented in stacked charts. Stack alignment toward the right suggests a higher agreement. The impact of notification size is stronger than that of contrast. Large notifications were rated only slightly better noticeable than medium-sized notifications, yet more distracting. Small high-contrast notifications provide a good compromise of self-reported perception and distraction.

tifications. However, by reducing the footprint of notifications, the problem of fitting sufficient content into the notification arises. This could be compensated by enlarging the notification once the user actively gazes at it.

7.3.2 Controlling Notification Placement

We designed two gaze-aware interaction techniques.

To explore further aspects of the previously mentioned design factors, we designed two interaction techniques utilizing gaze tracking and the previously tested visual designs to answer our research questions.

Gaze-Implicit

Gaze-implicit notifications display notification banners of regular size on the horizontal screen edge that is furthest away from the user's gazing location.

Our first interaction technique activates the front-facing camera to estimate the user's gaze location shortly before a notification is presented. Depending on whether the user looks at the upper or lower half of the screen, the notification will be displayed on the screen edge which is vertically farthest away. Thus, notifications are moved from the user's central vision into the peripheral vision. They use the medium-size high-contrast design that is default on iOS. These notifications can be pressed to expand them (see Figure 7.4). They are dismissed by swiping or looking away from the notification for 1.2 s, a duration that fits into the range of typical dwell times with gaze interactions [Esteves et al., 2020].

Gaze-Explicit

Gaze-explicit notifications are initially smaller and enlarge once the user shifts her gaze toward them.

Our second interaction technique additionally allows further gaze interaction with the notification. Notifications are placed using the same rule as gaze-implicit notifications but use the small-size medium-contrast design to further reduce occlusion while being sufficiently perceivable based on the results of the preliminary study. When the user moves her gaze toward the notification, it enlarges as if it was pressed, and all options are revealed (see Figure 7.1).

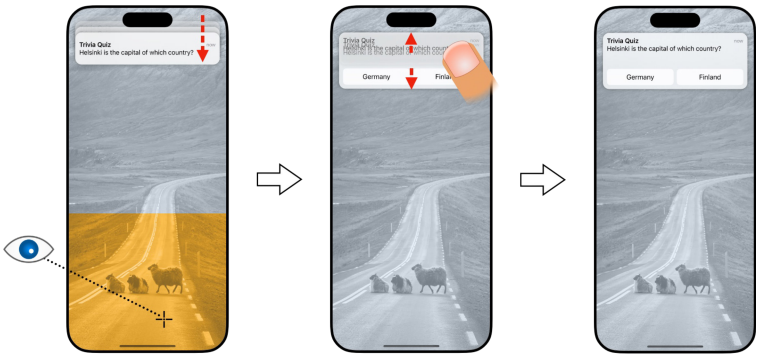


Figure 7.4: Just like our *gaze-explicit* notifications, *gaze-implicit* notifications appear on-screen as far as possible from the user's gazing location. However, they always appear in full width. To reveal additional options, users then touch the notification with their fingers.

Figure 7.5: Both *gaze-explicit* as well as *gaze-implicit* notifications are automatically dismissed when the user moves her gaze away from the notification back to the original position.

Touch-Attentive

For comparison, we also created a similar condition that uses the recent touch data instead of gaze.

For comparison, we also created a notification presentation style that does not rely on gaze tracking as an input modality. Touch-attentive notifications work like gaze-implicit notifications, but they use the last location of the user's finger instead of her gaze to determine on which side of the screen the notification is supposed to appear.

7.4 Evaluation

We evaluated our techniques against the system default of current mobile platforms in a user study with 10 participants.

To answer our research questions, we conducted a user study comparing these three techniques with a *baseline* condition that always slides in notifications from the top of the screen. Thus, this condition mimics the default system style on common mobile platforms. All conditions used a slide-in animation from the screen edge that matched the default style in iOS. Ten people participated in this study aged from 20 to 30, three male.

7.4.1 Apparatus and Task

A drawing app served as a primary task that kept users engaged during the study.

While we were interested in the usage of notifications, we needed to keep participants engaged in a primary task. Therefore, we created a simple drawing application in which users could pick different drawing tools and colors. A selection of different template outlines that participants could paint in was provided to assure their interest and make it unnecessary to think about an own design first. The drawing task was chosen because it requires focusing on moving the finger precisely so that one does not cross the outlines while painting. Typically for mobile UIs, toolbars for drawing tools and color selections were visible at the top and bottom of the screen. Thus, notifications resulted in an overlap with the primary task (see Figure 7.6).

As a secondary task, we ran a trivia quiz within the notification interface.

We also aimed to create an interesting secondary task for notification interactions that covers both interactive (that require user action, such as messaging apps or reminders) and non-interactive (that do not require user interaction; only deliver information) notifications. Therefore, we created a trivia quiz that was completely operational from within notifications. A notification appeared on-screen when a new question was ready to be answered. Depending on the condition, people could touch the notification or, with gaze-explicit notifications, shift their gaze toward them to enlarge them. In the enlarged state, buttons for two possible answers are visible and can be selected by tapping (see Figure 7.7). The subsequent queued notifications pre-

sented a short explanation of the solution to the previous quiz question. It did not require explicit interactions and only aimed to deliver information to the users. By varying the types of notification contents, we tried to cover both use cases of notifications: notifications with and without action. After a notification was dismissed, the system waited a random time interval between 15 and 30 seconds before presenting a new notification.

We used a within-subjects study design in which each participant used each condition at least 10 times.

In each condition, people received at least 10 notifications. The order of conditions was randomized for every participant. Before the first notification of every condition, the app explained the current condition to the participants. Additionally, the instructor explained how the notifications will be presented, enlarged, and how one could interact with them. After the task, participants ranked their agreement to statements about their experience on Likert scales. They also expressed their impressions in a follow-up interview during which they were still allowed to test the systems again if needed. One study run took around 50 minutes.

7.4.2 Variables

Participants ranked their *Agreement* to statements on their experience with each TECHNIQUE.

We used TECHNIQUE [*gaze-implicit*, *gaze-explicit*, *touch-attentive*, *baseline*] as the **independent variable**. As we already measured response times in the preliminary study, our goal in this study was to learn more about the experience our participants had while operating the phone. Therefore, we measured ***Agreement*** scores on 5-point Likert scales. Options were labeled from *totally disagree* (–2) to *totally agree* (2).

7.4.3 Results

We analyzed the participant ***Agreement*** data using Friedman tests and Dunn-Bonferroni post hoc tests.

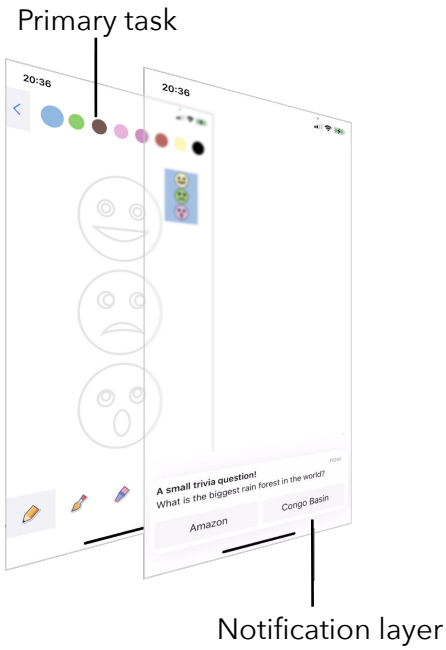


Figure 7.6: In the study, notifications could either be displayed along the top or bottom screen edge, resulting in an overlap of either the tool or color selection UI.

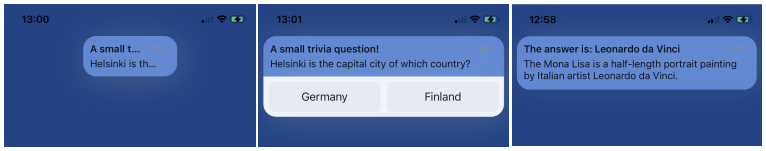


Figure 7.7: In the main study, gaze-explicit notifications used a small-size medium-contrast style (left) expanded once the participant shifted her gaze toward the notification. Expanded interactive notifications (middle) provided two buttons, while read-only notifications (right) did not.

Attention to Primary Task. First, participants rated their agreement to the statement "I could easily keep my attention on the drawing while notifications were presented on-screen." On average, our participants were indifferent about this statement in the baseline condition (M = .1) and slightly agreed in the gaze-explicit condition (M = .6). There was, however, no significant effect in the responses

Our participants could focus on the drawing slightly better using gaze-explicit notifications.

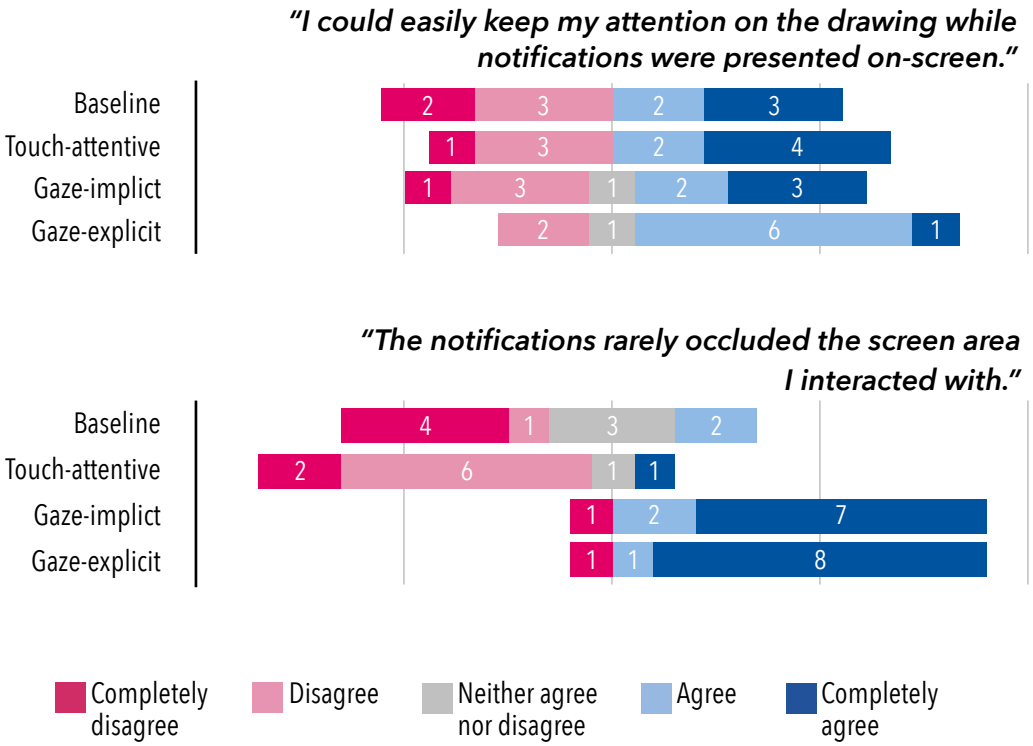


Figure 7.8: participants were asked to specify their agreement to statements on 5-point Likert scales after the study. The responses regarding distraction (left) and occlusion (right) caused by the notifications in the four conditions are presented in stacked charts. It is apparent that the visual representation of gaze-explicit notifications might benefit distraction. Our participants felt that both gaze-attentive conditions greatly reduce undesired occlusion effects.

($\chi^2(3) = 1.691$, $p = .639$). The responses of participants are depicted in Figure 7.8. Moreover, our participants agreed to the statement "I could easily return to drawing after I dismissed a notification" with the same average across all conditions (M = 1.3, SD = 1.2).

Baseline and touch-attentive notifications often occluded screen ares participants interacted with.

Disruption Caused by Notifications. Regarding the perceived disruption, participants rated the statement "The notifications rarely occluded the screen area I interacted with." There was a significant difference between the conditions ($\chi^2(3) = 18.357$, $p < .001$). Participants disagreed using the baseline (M = –0.7) and the touch-attentive (M =

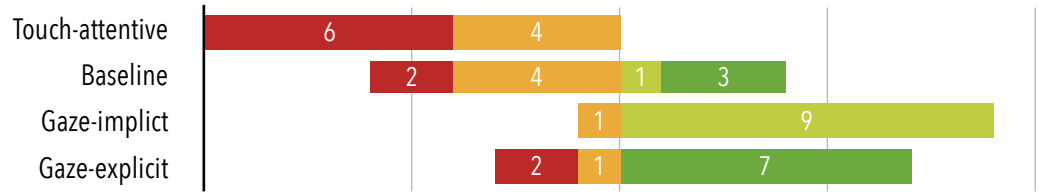


Figure 7.9: Both gaze-attentive notification designs lead the forced ranking of our participants. The possible ranks are encoded in color from most favorite (green) to least favorite (red). Those participants who did not like the gaze-explicit notifications feared accidental activations.

–0.8) conditions. On the other hand, participants agreed favorably with both gaze-implicit (M = 1.4) and gaze-explicit (M = 1.5) notification styles. Both gaze-based techniques were rated significantly better than the other two ($p < .02$). In addition, our participants agreed to the statement "It was easy to finish my current drawing action while a notification was displayed" using gaze-explicit notifications (M = 1.1, SD = 1.1). They only slightly agreed using gaze-implicit (both M = .5, SD = 1.5) and were indifferent in the baseline condition (M = .1, SD = 1.6).

Both gaze-explicit and gaze-implicit designs resulted in significantly fewer content overlaps.

Preference Ranking. After using all four designs, our participants were asked to "rank the experience the different designs offered from 1–4 with 1 being their favorite". There was a significant effect ($\chi^2(3) = 11.160$, $p = .011$) with participants preferring gaze-explicit (M = 1.8), and gaze-implicit notifications (M = 2.1) the most. Both of them were ranked significantly better than touch-attentive notifications (M = 3.6, $p < .02$). While not significant, even the baseline condition received a better average rating (M = 2.5). The results of this ranking are depicted in Figure 7.9.

Our participants liked the gaze-explicit and gaze-implicit notifications significantly better than the other two conditions.

7.5 Discussion

The follow-up interviews provide explanations for these results. The major source of disruption our participants identified was, in fact, not that the notification required attention while they were performing another task. Instead,

The main cause of notification disruption was impeding the primary task.

Our participants valued the smaller footprint of gaze-explicit notifications.

notifications impeded them from continuing their drawing by occluding tool selections or appearing right when they were about to tap on a different color, resulting in accidental activation. Eight participants mentioned that they experienced undesired content overlaps themselves in gaming and streaming apps. Gaze-explicit notifications not only provided a benefit by staying away from the area the user was about to interact with, but they also had a smaller footprint than the other notification styles: *"Especially the small size was useful. For example, I can easily notice which app sends me the notification [which is] enough to communicate the type of information provided." "The short version of the information in the small size notifications is totally enough to decide whether I need to pay attention to the notification or not."* This is in line with the findings of Klauck et al. [2017], which we confirmed here for the mobile setting.

Gaze tracking can reliably predict the area of user's interest. On the other hand, using the recent touch data worsened our study's occlusion problem.

Interestingly, while participants valued the system awareness of their interactive area, six out of ten rated the touch-attentive version as the worst design tested: *"The notifications based on the last touches were later than my gazing, so notifications often covered the toolbar at the moment when we wanted to change the tool"* and *"The touch-attentive version covered the color palette many times. It was very annoying that notifications covered [the UI] just before I tried to select a different color"*. This is especially surprising as we chose a drawing task specifically for fairness, as fingers and eyes move in parallel during drawing. Thus, mobile gaze tracking is clearly a better input than touch to identify the phone's interactive screen area (RQ1).

The disruption caused by notifications likely worsens with more complex real-world primary tasks.

Regarding their capability to pay attention to the drawing task, participants slightly favored gaze-explicit notifications, mainly due to the reduced content overlap. In our study, the type of disruption caused by notifications was perceived rather pragmatically than cognitively. The task was not cognitively challenging, and thus, our participants also had no issues returning to the drawing task after they dismissed a notification across all conditions. With device usage in the wild, however, this would likely change: With complex tasks, the effects of disruption worsen as more time is required to resume with the primary task [Czerwinski et al., 2004].

When explaining their overall ranking, our participants explained that *"gaze-explicit notifications were comfortable because they [...] avoid where [we] want to interact with"*. *"I really liked that notifications disappeared automatically if I looked at another part of the screen. I did not have to move my finger to close the notifications and I could quickly resume drawing."* However, we saw that gaze-explicit notifications had the most variance in their rankings. This was partially because it had a different design than the well-known default look but also because it introduces the Midas touch problem: *"if this was a chat program, it would have sent a read receipt despite me not wanting to mark it as read yet."* Thus, our participants were only concerned about undesirably activating a notification, whereas they enjoyed automatic dismissal. Future versions of gaze-explicit notifications should therefore only resize the notification accommodate more text without triggering the enlarged state of the notification instead to increase acceptance (RQ2). Regarding privacy, one participant criticized that even if the camera system is activated only shortly when a notification appears, she *"could not cover the camera with a sticker anymore."*

Our participants were afraid of Midas touch effects they assumed to occur with gaze-explicit notifications.

7.6 Conclusion

This chapter serves as a first exploration of possible designs for gaze-attentive notifications that reduce unnecessary distraction and content overlaps. In two user studies, we identified suitable parameters for implementing gaze-attentive notifications. We also collected valuable feedback from participants that helps to shape refined iterations of gaze-attentive notifications.

We explored novel interaction designs for mobile notifications that use gaze tracking.

The key findings of these studies are as follows:

1. Our participants enjoy notification styles that raise information awareness without distracting them from their current tasks.
2. They liked the gaze-explicit notifications for introducing low visual distraction and eliminating additional input for dismissal.

3. Resizing the gaze-explicit notification during interaction reduced distraction without lowering its information content.

4. While gaze can effectively be used to determine a suitable placement of notifications, this is not the case with touch inputs. In fact, the touch-attentive placement of notifications felt quite random to our participants.

7.7 Future Work

Future notification designs could apply layout changes to prevent content occlusions completely.

Based on these findings, our next steps are refining gaze-attentive notifications and conducting a more extensive study to analyze their impact. The feedback we obtained from the study participants made clear to us that occlusion is even more of a problem than we already expected. Alternative notification designs could completely resolve occlusion problems. For instance, slightly decreasing the screen's viewport on the screen edge of the notification and presenting it vertically next to the actual content of the current app.

Users expect Midas touch effects. Notification designs should convey the consequences of gazing at a notification via feedforward.

Secondly, we learned that enlarging the notification when gazing at it is an effective way to display more textual content when the user wants to pay attention to it. However, directly expanding the notification and providing buttons, chat options, etc., leads to user reservations. Future notification designs should, therefore, take care that no Midas touch effects exist and that this is appropriately communicated to the users.

Notifications on tablets could be placed dynamically inside the user's visual field.

Moreover, adapting gaze-attentive notifications to tablets will provide new challenges: With tablet computers, more screen space moves even further into peripheral vision, so determining a notification size or contract level based on the gaze location becomes an interesting factor.

The current limitations of this work are the small studies and that they were only conducted in a lab setting. Moreover, while the drawing task was intended to introduce

fairness between the touch and gaze conditions, it might not have been cognitively challenging enough. Therefore, we want to test the refined gaze-attentive notification design with a more extensive in-the-wild study to capture actual usage across different contexts.

A larger and more complex study design should be used for future evaluations.

In the last five chapters, we introduced new interaction techniques for mobile devices. We used facial tracking to augment touch input in our explicit interaction techniques. In contrast, our implicit interaction techniques used facial tracking to adapt the on-screen content to what the user is looking at. In the next chapter, we summarize the thesis and draw a conclusion. We will also look at potential future interactions with mobile devices that use inputs from facial tracking.

In the next chapter, we summarize the thesis and look at future perspectives of facial tracking on mobile devices.

Chapter 8

Summary and Future Perspectives

"The eyes are the window to your soul."

—William Shakespeare

Common interaction paradigms in HCI neglect many human aspects of their user. Visually tracking the eyes and head provides an additional communication channel.

Humans are experts in reading the eyes of others. Your eyes, combined with facial expressions, give away your mood. They tell others whether you are happy or sad and even change based on your concentration level or tiredness. Furthermore, head movements over time result in gestures whose cultural meaning is already known to children. Visual cues like these provide richness in face-to-face communication between humans.

In today's computing interfaces—actually, in HCI as a whole domain—we usually reduce the users to 2D input locations that fit into the interaction model. This results in a lack of context, making interactions shallow or poorly blended with their environment. Yet, the increased processing power and high-resolution cameras in mobile devices allow us to track the user visually and enable novel interaction techniques. This work aimed to better understand the possible applications of visually tracking the user's eyes and head orientation on mobile devices across different use cases.

8.1 Summary and Conclusions

We designed interaction techniques for mobile devices that rely on the increased communication bandwidth from adding facial tracking.

In this thesis, we investigated how increasing the communication bandwidth with facial tracking of the eyes and head can benefit mobile interactions. To better understand the application areas for this new input type, we designed techniques targeting either screen-space or world-space content. The former denotes interactions with the GUI elements displayed directly on the user's handheld screen; the latter is our umbrella term for content residing outside these bounds, e.g., remote devices or virtual content placed inside the room. In addition, we investigated interaction techniques that map the facial tracking input either implicitly or explicitly.

Headbang is an interaction technique that allows users to perform quick actions from subtle back-and-forth gestures with their heads.

We began with explicit uses of facial tracking with discretized head input in **Chapter 3**. The *Headbang* interaction technique used head tilt to trigger quick actions in GUIs. The number of elements in context menus increases with more functionalities in mobile apps. Extensive context menus require more screen space, resulting in content overlap and difficulties reaching them with the thumb for touch input. With Headbang, upon touching an element of interest, users specified their intended action through head tilting. We highlighted this action in a compact radial menu. One could even change the selection by changing the head orientation and confirm the selection by lifting the finger.

Making discrete selections in context menus using *Headbang* can be faster than touch input depending on the menu size.

In our first study, we saw that Headbang could reliably be used while sitting and walking, making it suitable for mobile interactions. In Study 2, we evaluated Headbang against tilt and touch to operate context menus. Here, Headbang provided reliable accuracy with appropriate speed. Depending on the number of menu entries, our Headbang menu was even faster than conventional touch list menus.

In **Chapter 4**, we again used head tracking as input, but this time focussed on continuous tracking. While smartphones have become bigger in recent years, our thumbs have not grown simultaneously. The resulting reachability issues come with unergonomic phone use and reduced

grip stability, increasing the likelihood of dropping the device while using it. Researchers tried to extend touch reach with various techniques, such as MagStick and BezelCursor. However, operating such reachability techniques adds some overhead to the interaction, making selections slower than direct touch. Touch-controlled cursor techniques also do not scale well, as the touchscreen area required to operate the cursors grows with larger device sizes. With our *Head + Touch* technique, users first control the cursor location using head tilting. They can then lock the cursor in place and refine the selection using touch.

Our *Head + Touch* reachability technique scales better to larger screens than previous approaches. This is possible by head tilting to control a cursor on the screen coarsely.

Again, we evaluated both sitting and walking situations and found that a head-controlled cursor (*Pure Head*) provided no feasible replacement for existing reachability techniques. However, by combining head tracking with touch, for instance, in our *Head + Touch* technique, our participants could reliably select items outside their thumb's reach 20% faster than using BezelCursor. Using head tracking allowed us to eliminate the time overhead of the reachability interaction to under 100 ms while also allowing users to maintain a stable grip.

With an overhead under 100 ms, *Head + Touch* as reachability technique is significantly faster than touch-based reachability techniques.

When people are not on the move, e.g., in seated multi-device collaboration sessions, sensor noise in visual tracking is notably smaller, and gaze tracking becomes feasible for novel interaction techniques. In **Chapter 5**, we took a step toward gaze as a real-world input modality with the *GazeConduits* system. Working entirely with off-the-shelf devices and without the requirement for calibration, GazeConduits tracked the users' gazing to identify which devices or collaborators they were looking at. Our quantitative studies showed that the system could accurately select one of 20 simultaneous tablets on a table and one of four collaborators using gaze.

GazeConduits uses gaze tracking in a cross-device setup to identify what or who users are looking at.

Building on this easy and fast input mode, our envisioned interaction techniques foster collaboration, e.g., by removing the need to physically reach for distant devices or trying to control cursors across different devices. We presented different interaction scenarios that benefit from gaze-at-device tracking, gaze-at-user tracking, and user awareness provided by GazeConduits. GazeConduits functioned as a

Gaze-at-device tracking, gaze-at-user tracking, and user awareness enable new interaction techniques that foster collaboration.

proof-of-concept that gaze-supported cross-device interactions are possible with current handheld devices and provide benefits to the interaction.

Our *User-Aware Rendering* tracks the user's eyes to calculate the camera frustum of handheld AR dynamically. This results in a better alignment between device viewport and peripheral vision.

We also explored cases where eye tracking is implicitly used to enhance the interaction. In **Chapter 6**, we presented a novel rendering technique for handheld augmented reality called *User-Aware Rendering*. In AR, the handheld device serves as a portal to see the virtual world-space content surrounding us. Therefore, it should be a goal to increase this immersion by providing a natural and suitably sized viewing experience. User-perspective rendering is one approach to achieving realism, but it comes at the cost of a narrow FOV. The commonly used device-perspective rendering, however, comes with a mismatch of what users see on and around the device. Our approach used head tracking to calculate a camera frustum similar to UPR. Yet, by virtually extending the device borders for this calculation, we also increased the FOV compared to UPR.

User-Aware Rendering preserves the enhanced depth perception of UPR with a more useful FOV at typical usage postures.

In two studies, we evaluated UAR with different magnification factors. We found that a 1.5x device magnification provided a sweet spot of showing enough content on the screen while remaining visually authentic and providing good depth perception. By doing so, UAR combined the strengths of DPR and UPR in a single technique while also requiring less tracking accuracy than UPR.

Attentive Notifications appear at the horizontal screen edge that is furthest away from what the user is looking at.

We presented an implicit use of gaze tracking for screen-space contents with our two *Attentive Notification* techniques in **Chapter 7**. Notification banners on mobile devices appear at the upper screen edge. This screen area is also used to display toolbars and, thus, frequently tapped UI elements. Therefore, each displayed notification can result in undesired content overlap and possible accidental inputs. To mitigate this issue, our two gaze-attentive techniques display notification banners as far away from the screen area the user is looking at.

We compared these two techniques against a touch-attentive version and the default behavior of iOS in a user study. Our participants found that the gaze-attentive resulted in significantly rarer undesired content occlusions.

Interestingly, only gaze tracking allowed a timely estimation of the user's area of interest. In contrast, the touch-attentive notification placement felt quite random to our participants. 80% of our participants valued the smaller footprint found in one of our designs. However, they feared accidental activations when gaze shifts are used to interact with notifications directly.

This application of gaze tracking allows to significantly decrease distracting content occlusions.

8.1.1 Contributions and Benefits

In conclusion, we made five artifact contributions that contain novel interaction techniques enabled by the tracking of facial features visible in the front-facing camera. These techniques benefit handheld users in different ways across a variety of mobile use cases.

We presented five interaction techniques that increase the expressiveness and efficiency of handheld devices through facial tracking.

- The *Headbang* technique discretizes the head orientation to trigger actions in menus. It reduces content occlusions and makes inputs more efficient, as it performs faster than touch depending on the menu size **(H1)**.
- With the *Head + Touch* reachability technique, one-handed use of smartphones becomes more ergonomic as users do not need to stretch their thumbs to reach distant targets. At the same time, this technique removes the time overhead found in touch-controlled reachability techniques **(H1)**.
- *GazeConduits* provides reliable gaze tracking for cross-device interactions across devices and collaborators. This enhances collaboration through simple specification of intended target and action **(H3)**.
- Handheld AR using *User-Aware Rendering* calculates a viewing frustum based on head tracking. It provides a more usable camera frustum than UPR or DPR, conveying depth information and a good scene overview. **(H2)**.
- *Attentive Notifications* reduce distractions and unwanted content overlaps by placing notifications as

far as possible away from what the user is currently looking at. The timely cost of placing notifications closer to the peripheral vision is less than 100 ms **(H4)**.

8.1.2 Reflection

We listed a few take-home messages for our readers below.

While these techniques benefit smartphone users on their own, they can also inform future research with general conclusions that can be derived from them. Concretely, researchers can find evidence for the following themes in our evaluations:

Head tracking is reliable without calibration.

State of facial tracking on mobile devices. Our implementations were based on ARKit. Thus, we used current off-the-shelf smartphones and their standard AR library for facial tracking. We assured that these tracking capabilities were reasonably accurate in our preliminary studies. Our evaluations of *Headbang* and *Head + Touch* prove that head tracking interactions do not require calibration and are robust against users walking.

Currently, researchers should add a calibration step if they require precise gaze estimations.

However, gaze tracking is less accurate. Its quality deteriorates further when users move. Without calibration, we could identify areas of interest of 20×25 cm via gaze. This large size is an effect of gaze tracking having unique offsets with individual faces and the postures in which they hold the device. Therefore, researchers should consider calibrating their system for each participant. For our *Attentive Notifications*, a quick nine-point calibration was sufficient to reduce the error of gaze tracking to under 1 cm.

Users can comfortably control their heads over longer periods to create continuous input.

Avoiding the double role of eye gaze. As looking at objects at an angle is uncomfortable, humans follow their gaze with their heads. While this means that head tracking can be used as a substitute for gaze tracking for some use cases [Stiefelhagen et al., 1999], we found that using head tracking over gaze for *Headbang* and *Head + Touch* provided unique benefits: Not only can the head be controlled consciously and independently of the eyes, but unlike them,

it makes no subconscious saccades. This made it easy for our study participants to control interface elements using their head orientation while they could still look at arbitrary screen locations.

Ideally, only use gaze data as passive information. If gaze control is required, make selection explicit by adding a different modality and keep interactions as short as possible.

We therefore recommend prioritizing head tracking for active controls. On the other hand, eye tracking works well as a passive input stream, like in the case of *User-Aware Rendering* and *Attentive Notifications*. When users should actively control parts of an interactive system with their gaze, we recommend making these interactions explicit through the use of an additional modality, e.g., touch. Researchers should ensure these gaze interactions remain short, as consciously controlling the eyes will quickly feel unnatural. One example of this is our target selection in *GazeConduits*, in which users look at the tablet they want to control and directly perceive feedback on this exact device.

Head tracking suits mobile context well. Our interaction techniques were significantly faster than their touch-controlled counterparts.

Faster interactions from higher bandwidths. Independent of technical inaccuracies in gaze tracking, the mobile usage context brings further challenges to the table: Especially while outside, people have to frequently shift their gaze away from their phones to check their surroundings. We, therefore, believe that head tracking suits the contexts of mobile interaction well. While head tracking might be slower than gaze in the lab [Kytö et al., 2018], we still measured significantly faster interaction times with head tracking in comparison to touch input in our *Headbang* and *Head + Touch* studies.

Head gestures speed up interactions as they can be performed parallel to existing inputs.

The success of these techniques shows how well the human processors work in parallel. The additional load on the motor processor from head control does not interfere with other processes of operating a mobile device. For instance, in the case of *Headbang*, users could tilt their heads to initiate the context menu while thinking about their desired action. With our *Head + Touch* technique, inputs were even more expressive as we combined using touch, force, and head control. Again, increasing the bandwidth of possible inputs worked out here.

Evaluations should evaluate mobile interaction techniques with moving users to provide external validity to their studies.

Realistic testing conditions and user attitudes. Across our evaluations of any of the five artifacts, we noticed how important realistic testing is in user evaluations. One example is our *Pure Head* reachability technique: While standing, it provided fast and reliable selections but failed while participants were walking. While we found related work that suggested impacts of walking in interaction techniques [Wilson et al., 2011], our impression is that researchers often omit an evaluation with moving users.

Interaction designers should know that even normal users are afraid of accidental activations in gaze interfaces. Also, our studies confirm that using touch for explicit confirmation works well in combination with facial tracking.

It was equally essential for us to understand the attitudes of our participants toward the new interaction techniques. For instance, we did not expect participants to consider the potential effects of accidental activations and methods to cancel unintended actions. With *Head + Touch* users confirm their selection by lifting the finger. This allowed them to check whether the cursor would hit the desired target. With our *Attentive Notifications*, however, our participants were afraid that the expanding behavior of gaze-explicit notifications could trigger unwanted read receipts for messages. This shows that normal users understand the Midas touch problem of gaze interactions, although they use different words to describe it. User precautions like these could be cleared with labels or guidance. However, interaction techniques that require input from an alternative modality for confirmation seem to achieve higher user acceptance.

Facial tracking is currently integrated into commercial systems as accessibility technology. Our work shows that it is promising for able-bodied users, too.

Overall, the techniques presented in this thesis show that facial tracking can provide rich interactions that go well beyond accessibility use cases. Speaking of accessibility, we want to remark that the idea of a head-controlled cursor we presented in Chapter 4 was integrated into macOS in the meantime. The *head pointer* allows users to control the cursor by tilting their head in front of the screen. Actions like clicking can be executed by facial expressions like winking or sticking out the tongue.

8.2 Future Perspectives

Future research on facial tracking can take different routes. On the one hand, exploring further application areas of vi-

sual tracking will complement more novel interaction techniques. On the other hand, enhancements to the tracking software can tackle the remaining limitations. Lastly, abstracting from mobile devices to larger screens provides opportunities for new interaction contexts.

Also, desktop work could benefit from new interactions using facial tracking.

In this thesis, we only investigated facial tracking on handheld devices. However, stationary work could also benefit from this type of tracking. For instance, one could tackle ergonomic issues when sitting at a computer desk. Many people report lower back pain from sitting at a desk for too long, also because it is hard for many people to maintain an upright position over an extended time. A computer could observe changes in posture by tracking the user's head position in front of the screen and provide incentives to change the seating position again.

Future work should also evaluate the impact of facial tracking on the battery life.

Also some technical concerns remain that need to be considered for the widespread adoption of interaction techniques based on facial tracking. Firstly, battery concerns. An activated camera will increase power consumption and performing computer vision to identify facial features increases CPU load. However, we must acknowledge that our smartphones have optimized pipelines and dedicated processing units for facial tracking. Moreover, the camera only needs to be activated for short timespans. For instance, a few seconds are sufficient with *Headbang* and our *Attentive Notifications*. In some cases, the camera might already be activated independently of facial tracking.

Future work should also identify which concerns users of such techniques have and tackle them appropriately.

Also, privacy concerns must be cleared sufficiently for this type of interaction. Our techniques processed facial data locally and did not store facial information on the user where it was not needed. However, when integrated into commercial software, we could expect users to be suspicious of additional usage of their facial data.

Once these challenges are overcome, facial tracking provides an exciting and expressive input modality that is fast and easy to control. We are excited to see which systems will be the next to make new interactions possible with facial tracking.

Bibliography

[1] Israel Abramov, James Gordon, and Hoover Chan. Color appearance in the peripheral retina: effects of stimulus size. *J. Opt. Soc. Am. A*, 8(2):404–414, February 1991. doi.org/10.1364/JOSAA.8.000404.

[2] Richard A Abrams, David E Meyer, and Sylvan Kornblum. Speed and accuracy of saccadic eye movements: characteristics of impulse variability in the oculomotor system. *Journal of Experimental Psychology: Human Perception and Performance*, 15(3):529, 1989.

[3] Piotr D. Adamczyk and Brian P. Bailey. If Not Now, When? The Effects of Interruption at Different Moments within Task Execution. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, page 271–278, New York, NY, USA, 2004. Association for Computing Machinery. doi.org/10.1145/985692.985727.

[4] Kenneth Alberto Funes Mora and Jean-Marc Odobez. Geometric Generative Gaze Estimation (G3E) for Remote RGB-D Cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[5] Daniel Andersen, Voicu Popescu, Chengyuan Lin, Maria Eugenia Cabrera, Aditya Shanghavi, and Juan Wachs. A Hand-Held, Self-Contained Simulated Transparent Display. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 96–101, 2016. doi.org/10.1109/ISMAR-Adjunct.2016.0049.

[6] Stuart Anstis. Picturing Peripheral Acuity. *Perception*, 27(7):817–825, 1998. doi.org/10.1068/p270817.

[7] Caroline Appert and Shumin Zhai. Using Strokes As Command Shortcuts: Cognitive Benefits and Toolkit Support. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2289–2298, New York, NY, USA, 2009. ACM. doi.org/10.1145/1518701.1519052.

[8] Jacopo M. Araujo, Guangtao Zhang, John Paulin Paulin Hansen, and Sadasivan Puthusserypady. Exploring Eye-Gaze Wheelchair Control. In *ACM*

Symposium on Eye Tracking Research and Applications, ETRA '20 Adjunct, New York, NY, USA, 2020. Association for Computing Machinery. doi.org/10.1145/3379157.3388933.

[9] Ravit Avraham Bahir, Yisrael Parmet, and Noam Tractinsky. Effects of Visual Enhancements and Delivery Time on Receptivity of Mobile Push Notifications. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, page 1–6, New York, NY, USA, 2019. Association for Computing Machinery. doi.org/10.1145/3290607.3312993.

[10] Mathias Baglioni, Eric Lecolinet, and Yves Guiard. JerkTilts: Using Accelerometers for Eight-Choice Selection on Mobile Devices. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, ICMI '11, pages 121–128, New York, NY, USA, 2011. ACM. doi.org/10.1145/2070481.2070503.

[11] G. Susanne Bahr and Richard A. Ford. How and why pop-ups don't work: Pop-up prompted eye movements, user affect and decision making. *Computers in Human Behavior*, 27(2):776–783, 2011. doi.org/10.1016/j.chb.2010.10.030.

[12] Monique Faye Baier and Michael Burmester. Not Just About the User: Acceptance of Speech Interaction in Public Spaces. In *Proceedings of Mensch Und Computer 2019*, MuC '19, page 349–359, New York, NY, USA, 2019. Association for Computing Machinery. doi.org/10.1145/3340764.3340801.

[13] Till Ballendat, Nicolai Marquardt, and Saul Greenberg. Proxemic Interaction: Designing for a Proximity and Orientation-Aware Environment. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, page 121–130, New York, NY, USA, 2010. Association for Computing Machinery. doi.org/10.1145/1936652.1936676.

[14] Yiwei Bao, Yihua Cheng, Yunfei Liu, and Feng Lu. Adaptive Feature Fusion Network for Gaze Tracking in Mobile Tablets. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9936–9943, 2021. doi.org/10.1109/ICPR48806.2021.9412205.

[15] Domagoj Baričević, Cha Lee, Matthew Turk, Tobias Hö, and Doug A. Bowman. A hand-held AR magic lens with user-perspective rendering. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 197–206, 2012. doi.org/10.1109/ISMAR.2012.6402557.

[16] Domagoj Baričević, Tobias Höllerer, Pradeep Sen, and Matthew Turk. User-Perspective Augmented Reality Magic Lens from Gradients. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, VRST '14, pages 87–96, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2671015.2671027.

[17] John Bateman, Janina Wildfeuer, and Tuomo Hiippala. *Multimodality. Foundations, Research and Analysis – A Problem-Oriented Introduction*. De Gruyter Mouton, Berlin, Boston, 2017. ISBN 9783110479898. doi.org/10.1515/9783110479898.

[18] Joanna Bergstrom-Lehtovirta and Antti Oulasvirta. Modeling the Functional Area of the Thumb on Mobile Touchscreen Surfaces. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 1991–2000, New York, NY, USA, 2014. ACM. doi.org/10.1145/2556288.2557354.

[19] Kenneth J Berry, Janis E Johnston, and Paul W Mielke Jr. An alternative measure of effect size for Cochran's Q test for related proportions. *Perceptual and motor skills*, 104(3_suppl):1236–1242, 2007.

[20] Patricia E.G. Bestelmeyer, Benjamin W. Tatler, Louise H. Phillips, Gillian Fraser, Philip J. Benson, and David St.Clair. Global visual scanning abnormalities in schizophrenia and bipolar disorder. *Schizophrenia Research*, 87(1): 212–222, 2006. doi.org/10.1016/j.schres.2006.06.015.

[21] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and Magic Lenses: The See-through Interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 73–80, New York, NY, USA, 1993. Association for Computing Machinery. doi.org/10.1145/166117.166126.

[22] Marc D. Binder, Nobutaka Hirokawa, and Uwe Windhorst, editors. *Saccade, Saccadic Eye Movement*, pages 3557–3557. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-29678-2. doi.org/10.1007/978-3-540-29678-2_5190.

[23] Laura Boccardo. Viewing distance of smartphones in presbyopic and non-presbyopic age. *Journal of Optometry*, 14(2):120–126, 2021. doi.org/10.1016/j.optom.2020.08.001.

[24] Richard A. Bolt. "Put-That-There": Voice and Gesture at the Graphics Interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '80, page 262–270, New York, NY, USA, 1980. Association for Computing Machinery. doi.org/10.1145/800250.807503.

[25] Richard A. Bolt. Gaze-Orchestrated Dynamic Windows. *SIGGRAPH Comput. Graph.*, 15(3):109–119, August 1981. doi.org/10.1145/965161.806796.

[26] Jan Borchers, Anke Brocker, Sebastian Hueber, Oliver Nowak, René Schäfer, Adrian Wagner, Paul Miles Preuschoff, and Lea Emilia Schirp. The Aachen Lab Demo: From Fundamental Perception to Design Tools. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI

EA '23, New York, NY, USA, 2023. Association for Computing Machinery. doi.org/10.1145/3544549.3583937.

[27] Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. The Fat Thumb: Using the Thumb's Contact Size for Single-Handed Mobile Interaction. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '12, pages 39–48, New York, NY, USA, 2012. ACM. doi.org/10.1145/2371574.2371582.

[28] Michael Braun, Jonas Schubert, Bastian Pfleging, and Florian Alt. Improving Driver Emotions with Affective Strategies. *Multimodal Technologies and Interaction*, 3(1), 2019. doi.org/10.3390/mti3010021.

[29] Frederik Brudy, Steven Houben, Nicolai Marquardt, and Yvonne Rogers. CurationSpace: Cross-Device Content Curation Using Instrumental Interaction. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*, ISS '16, pages 159–168, New York, NY, USA, 2016. ACM. doi.org/10.1145/2992154.2992175.

[30] Frederik Brudy, Christian Holz, Roman Rädle, Chi-Jui Wu, Steven Houben, Clemens Klokmose, and Nicolai Marquardt. Cross-Device Taxonomy: Survey, Opportunities and Challenges of Interactions Spanning Across Multiple Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, May 2019. doi.org/10.1145/3290605.3300792.

[31] William Buxton. Living in augmented reality: Ubiquitous media and reactive environments. *Video mediated communication*, pages 363–384, 1997.

[32] William Buxton, Ralph Hill, and Peter Rowley. Issues and techniques in touch-sensitive tablet input. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, page 215–224, New York, NY, USA, 1985. Association for Computing Machinery. doi.org/10.1145/325334.325239.

[33] Neil A Campbell, Jane B Reece, Lisa A Urry, Michael Lee Cain, Steven Alexander Wasserman, Peter V Minorsky, Robert B Jackson, and others. *Campbell Biology*, volume 9. Pearson, 2011. ISBN 9780321558237.

[34] Stuart K Card. *The Psychology of Human-Computer Interaction*. CRC Press, 1983. ISBN 0-89859-243-7.

[35] Çağla Çığ and Tevfik Metin Sezgin. Gaze-Based Virtual Task Predictor. In *Proceedings of the 7th Workshop on Eye Gaze in Intelligent Human Machine Interaction: Eye-Gaze & Multimodality*, GazeIn '14, page 9–14, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2666642.2666647.

[36] Xi-Jing Chang, Fang-Hsin Hsu, En-Chi Liang, Zih-Yun Chiou, Ho-Hsuan Chuang, Fang-Ching Tseng, Yu-Hsin Lin, and Yung-Ju Chang. Not Merely Deemed as Distraction: Investigating Smartphone Users' Motivations for Notification-Interaction. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. doi.org/10.1145/3544548.3581146.

[37] Youli Chang, Sehi L'Yi, and Jinwook Seo. Reaching Targets on Discomfort Region Using Tilting Gesture. In *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST'14 Adjunct, page 115–116, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2658779.2658803.

[38] Youli Chang, Sehi L'Yi, Kyle Koh, and Jinwook Seo. Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1499–1508, New York, NY, USA, 2015. ACM. doi.org/10.1145/2702123.2702425.

[39] Olivier Chapuis, Jean-Baptiste Labrune, and Emmanuel Pietriga. DynaSpot: Speed-dependent Area Cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1391–1400, New York, NY, USA, 2009. ACM. doi.org/10.1145/1518701.1518911.

[40] Kuan-Wen Chen, Yung-Ju Chang, and Liwei Chan. Predicting Opportune Moments to Deliver Notifications in Virtual Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. doi.org/10.1145/3491102.3517529.

[41] Xiang 'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott Hudson. Around-Body Interaction: Sensing & Interaction Techniques for Proprioception-Enhanced Input with Mobile Devices. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*, MobileHCI '14, page 287–290, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2628363.2628402.

[42] Xiang 'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott E. Hudson. Air+Touch: Interweaving Touch & In-Air Gestures. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 519–525, New York, NY, USA, 2014. ACM. doi.org/10.1145/2642918.2647392.

[43] Shiwei Cheng, Qiufeng Ping, and Tiyong Liu. A New Eye-tracking Method with Image Feature Based Model for Mobile Devices. In *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communi-*

cations, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta), pages 1902–1909, 2022. doi.org/10.1109/SmartWorld-UIC-ATC-ScalCom-DigitalTwin-PriComp-Metaverse56740.2022.00275.

[44] Alasdair D. F. Clarke, Aoife Mahon, Alex Irvine, and Amelia R. Hunt. People Are Unable to Recognize or Report on Their Own Eye Movements. *Quarterly Journal of Experimental Psychology*, 70(11):2251–2270, 2017. doi.org/10.1080/17470218.2016.1231208.

[45] Christian Corsten, Simon Voelker, Andreas Link, and Jan Borchers. Use the Force Picker, Luke: Space-Efficient Value Input on Force-Sensitive Mobile Touchscreens. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery. doi.org/10.1145/3173574.3174235.

[46] Christian Corsten, Marcel Lahaye, Jan Borchers, and Simon Voelker. ForceRay: Extending Thumb Reach via Force Input Stabilizes Device Grip for Mobile Touch Input. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery. doi.org/10.1145/3290605.3300442.

[47] D. A. Craig and H. T. Nguyen. Wireless Real-Time Head Movement System Using a Personal Digital Assistant (PDA) for Control of a Power Wheelchair. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 772–775, New York, NY, USA, January 2005. IEEE. doi.org/10.1109/IEMBS.2005.1616529.

[48] Andrew Crossan, Mark McGill, Stephen Brewster, and Roderick Murray-Smith. Head Tilting for Interaction in Mobile Contexts. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 6:1–6:10, New York, NY, USA, 2009. ACM. doi.org/10.1145/1613858.1613866.

[49] Carolina Cruz-Neira, Daniel J. Sandin, Thomas A. DeFanti, Robert V. Kenyon, and John C. Hart. The CAVE: Audio Visual Experience Automatic Virtual Environment. *Commun. ACM*, 35(6):64–72, June 1992. doi.org/10.1145/129888.129892.

[50] James E. Cutting and Peter M. Vishton. Chapter 3 - Perceiving Layout and Knowing Distances: The Integration, Relative Potency, and Contextual Use of Different Information about Depth. In William Epstein and Sheena Rogers, editors, *Perception of Space and Motion*, Handbook of Perception and Cognition, pages 69–117. Academic Press, San Diego, 1995. doi.org/10.1016/B978-012240530-3/50005-5.

[51] Mary Czerwinski, Eric Horvitz, and Susan Wilhite. A Diary Study of Task Switching and Interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, page 175–182, New York, NY, USA, 2004. Association for Computing Machinery. doi.org/10.1145/985692.985715.

[52] Nicholas S. Dalton, Emily Collins, and Paul Marshall. Display Blindness?: Looking Again at the Visibility of Situated Displays Using Eye-tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 3889–3898. ACM, 2015. doi.org/10.1145/2702123.2702150.

[53] Arindam Dey, Mark Billinghurst, Robert W. Lindeman, and J. Edward Swan. A Systematic Review of 10 Years of Augmented Reality Usability Studies: 2005 to 2014. *Frontiers in Robotics and AI*, 5:37, 2018. doi.org/10.3389/frobt.2018.00037.

[54] Catherine Diaz, Michael Walker, Danielle Albers Szafir, and Daniel Szafir. Designing for Depth Perceptions in Augmented Reality. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 111–122, 2017. doi.org/10.1109/ISMAR.2017.28.

[55] Tiffany D. Do, Joseph J. LaViola, and Ryan P. McMahan. The Effects of Object Shape, Fidelity, Color, and Luminance on Depth Perception in Handheld Mobile Augmented Reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 64–72, 2020. doi.org/10.1109/ISMAR50242.2020.00026.

[56] David Drascic and Paul Milgram. Perceptual issues in augmented reality. In Mark T. Bolas, Scott S. Fisher, Mark T. Bolas, Scott S. Fisher, and John O. Merritt, editors, *Stereoscopic Displays and Virtual Reality Systems III*, volume 2653, pages 123–134. International Society for Optics and Photonics, SPIE, 1996.

[57] Andrew T Duchowski. A breadth-first survey of eye-tracking applications. *Behavior Research Methods Instruments and Computers*, 34(4):455–470, 2002.

[58] Rachel Eardley, Anne Roudaut, Steve Gill, and Stephen J. Thompson. Understanding Grip Shifts: How Form Factors Impact Hand Movements on Mobile Phones. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 4680–4691, New York, NY, USA, 2017. ACM. doi.org/10.1145/3025453.3025835.

[59] Augusto Esteves, David Verweij, Liza Suraiya, Rasel Islam, Youryang Lee, and Ian Oakley. SmoothMoves: Smooth Pursuits Head Movements for Augmented Reality. In *Proceedings of the 30th Annual ACM Symposium on User In-*

terface Software and Technology, UIST '17, pages 167–178, New York, NY, USA, 2017. ACM. doi.org/10.1145/3126594.3126616.

[60] Augusto Esteves, Yonghwan Shin, and Ian Oakley. Comparing selection mechanisms for gaze input techniques in head-mounted displays. *International Journal of Human-Computer Studies*, 139:102414, 2020. doi.org/10.1016/j.ijhcs.2020.102414.

[61] Joel E. Fischer, Nick Yee, Victoria Bellotti, Nathan Good, Steve Benford, and Chris Greenhalgh. Effects of Content and Time of Delivery on Receptivity to Mobile Interruptions. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, page 103–112, New York, NY, USA, 2010. Association for Computing Machinery. doi.org/10.1145/1851600.1851620.

[62] Joel E. Fischer, Chris Greenhalgh, and Steve Benford. Investigating Episodes of Mobile Phone Activity as Indicators of Opportune Moments to Deliver Notifications. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, page 181–190, New York, NY, USA, 2011. Association for Computing Machinery. doi.org/10.1145/2037373.2037402.

[63] Nicholas Fitz, Kostadin Kushlev, Ranjan Jagannathan, Terrel Lewis, Devang Paliwal, and Dan Ariely. Batching smartphone notifications can improve well-being. *Computers in Human Behavior*, 101:84–94, 2019. doi.org/10.1016/j.chb.2019.07.016.

[64] John M. Franchak, Kari S. Kretch, Kasey C. Soska, and Karen E. Adolph. Head-Mounted Eye Tracking: A New Method to Describe Infant Looking. *Child Development*, 82(6):1738–1750, 2011. doi.org/10.1111/j.1467-8624.2011.01670.x.

[65] Alexandra Frischen, Andrew P. Bayliss, and Steven P. Tipper. Gaze cueing of attention: visual attention, social cognition, and individual differences. *Psychological bulletin*, 133(4):694, 2007. doi.org/10.1037/0033-2909.133.4.694.

[66] C. Furmanski, R. Azuma, and M. Daily. Augmented-reality visualizations guided by cognition: perceptual heuristics for combining visible and obscured information. In *Proceedings. International Symposium on Mixed and Augmented Reality*, pages 215–320, 2002. doi.org/10.1109/ISMAR.2002.1115091.

[67] Yulia Gizatdinova, Oleg Špakov, Outi Tuisku, Matthew Turk, and Veikko Surakka. Gaze and Head Pointing for Hands-Free Text Entry: Applicability to Ultra-Small Virtual Keyboards. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, New York, NY, USA, 2018. Association for Computing Machinery. doi.org/10.1145/3204493.3204539.

[68] Dmitry O. Gorodnichy and Gerhard Roth. Nouse 'Use your Nose as a Mouse' Perceptual Vision Technology for Hands-Free Games and Interfaces. *Image and Vision Computing*, 22(12):931–942, October 2004. doi.org/10.1016/j.imavis.2004.03.021.

[69] G. Goswami, M. Vatsa, and R. Singh. RGB-D Face Recognition With Texture and Attribute Features. *IEEE Transactions on Information Forensics and Security*, 9(10):1629–1640, October 2014. doi.org/10.1109/TIFS.2014.2343913.

[70] Saul Greenberg, Nicolai Marquardt, Till Ballendat, Rob Diaz-Marino, and Miaosen Wang. Proxemic Interactions: The New Ubicomp? *Interactions*, 18(1):42–50, January 2011. doi.org/10.1145/1897239.1897250.

[71] Jens Emil Grønbæk, Mille Skovhus Knudsen, Kenton O'Hara, Peter Gall Krogh, Jo Vermeulen, and Marianne Graves Petersen. Proxemics Beyond Proximity: Designing for Flexible Social Interaction Through Cross-Device Interaction. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–14, New York, NY, USA, 2020. Association for Computing Machinery. doi.org/10.1145/3313831.3376379.

[72] Jens Grubert and Matthias Kranz. HeadPhones: Ad Hoc Mobile Multi-Display Environments Through Head Tracking. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3966–3971, New York, NY, USA, 2017. ACM. doi.org/10.1145/3025453.3025533.

[73] Ankit Guleria and Ramandeep Kaur. Unintended Notification Swipe Detection System. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1614–1620. IEEE, 2021. doi.org/10.1109/ICIRCA51532.2021.9544898.

[74] Edward T. Hall. *The Hidden Dimension*, volume 609. Anchor, 1966.

[75] Peter Hamilton and Daniel J. Wigdor. Conductor: Enabling and Understanding Cross-device Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2773–2782, New York, NY, USA, 2014. ACM. doi.org/10.1145/2556288.2557170.

[76] Chris Harrison and Anind K. Dey. Lean and Zoom: Proximity-Aware User Interface and Content Magnification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 507–510, New York, NY, USA, 2008. Association for Computing Machinery. doi.org/10.1145/1357054.1357135.

[77] Khalad Hasan, Junhyeok Kim, David Ahlström, and Pourang Irani. Thumbs-Up: 3D Spatial Thumb-Reachable Space for One-Handed Thumb Interaction on Smartphones. In *Proceedings of the 2016 Symposium on Spatial*

User Interaction, SUI '16, pages 103–106, New York, NY, USA, 2016. ACM. doi.org/10.1145/2983310.2985755.

[78] Alex Hill, Jacob Schiefer, Jeff Wilson, Brian Davidson, Maribeth Gandy, and Blair MacIntyre. Virtual transparency: Introducing parallax view into video see-through AR. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 239–240, 2011. doi.org/10.1109/ISMAR.2011.6092395.

[79] Ken Hinckley. Synchronous Gestures for Multiple Persons and Computers. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 149–158, New York, NY, USA, 2003. ACM. doi.org/10.1145/964696.964713.

[80] Ken Hinckley, Gonzalo Ramos, Francois Guimbretiere, Patrick Baudisch, and Marc Smith. Stitching: Pen Gestures That Span Multiple Displays. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, pages 23–31, New York, NY, USA, 2004. ACM. doi.org/10.1145/989863.989866.

[81] Ken Hinckley, Morgan Dixon, Raman Sarin, Francois Guimbretiere, and Ravin Balakrishnan. Codex: A Dual Screen Tablet Computer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1933–1942, New York, NY, USA, 2009. ACM. doi.org/10.1145/1518701.1518996.

[82] Uta Hinrichs and Sheelagh Carpendale. Gestures in the Wild: Studying Multi-Touch Gesture Sequences on Interactive Tabletop Exhibits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 3023–3032, New York, NY, USA, 2011. ACM. doi.org/10.1145/1978942.1979391.

[83] Leila Homaeian, Nippun Goyal, James R. Wallace, and Stacey D. Scott. Group vs Individual: Impact of TOUCH and TILT Cross-Device Interactions on Mixed-Focus Collaboration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA, 2018. Association for Computing Machinery. doi.org/10.1145/3173574.3173647.

[84] Eric Horvitz. Lumiere project: Bayesian reasoning for automated assistance. *Decision Theory & Adaptive Systems Group, Microsoft Research. Microsoft Corp. Redmond, WA*, 1998.

[85] Eric Horvitz, Andy Jacobs, and David Hovel. Attention-Sensitive Alerting. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, page 305–313, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[86] Gang Hu, Derek Reilly, Mohammed Alnusayri, Ben Swinden, and Qigang Gao. DT-DT: Top-down Human Activity Analysis for Interactive Surface Applications. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ITS '14, pages 167–176, New York, NY, USA, 2014. ACM. doi.org/10.1145/2669485.2669501.

[87] Sebastian Hueber, Christian Cherek, Philipp Wacker, Jan Borchers, and Simon Voelker. Headbang: Using Head Gestures to Trigger Discrete Actions on Mobile Devices. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '20, New York, NY, USA, October 2020. Association for Computing Machinery. doi.org/10.1145/3379503.3403538.

[88] Sebastian Hueber, Eunae Jang, and Jan Borchers. Attentive Notifications: Minimizing Distractions of Mobile Notifications through Gaze Tracking. In *Proceedings of the 25th International Conference on Mobile Human-Computer Interaction*, MobileHCI '23 Companion, New York, NY, USA, September 2023. Association for Computing Machinery. doi.org/10.1145/3565066.3608695.

[89] Sebastian Hueber, Johannes Wilhelm, René Schäfer, Simon Voelker, and Jan Borchers. User-Aware Rendering: Merging the Strengths of Device- and User-Perspective Rendering in Handheld AR. *Proceedings of the ACM on Human-Computer Interaction*, Vol. 7(MHCI), September 2023. doi.org/10.1145/3604278.

[90] Norman Höller, Johann Schrammel, Manfred Tscheligi, and Lucas Paletta. The Perception of Information and Advertisement Screens Mounted in Public Transportation Vehicles-Results from a Mobile Eye-tracking Study. *Pervasive Advertising*, page 141, 2009.

[91] Mirja Ilves, Yulia Gizatdinova, Veikko Surakka, and Esko Vankka. Head movement and facial expressions as game input. *Entertainment Computing*, 5(3):147–156, August 2014. doi.org/10.1016/j.entcom.2014.04.005.

[92] Shamsi T. Iqbal and Eric Horvitz. Notifications and Awareness: A Field Study of Alert Usage and Preferences. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, CSCW '10, page 27–30, New York, NY, USA, 2010. Association for Computing Machinery. doi.org/10.1145/1718918.1718926.

[93] Shahram Izadi, Harry Brignull, Tom Rodden, Yvonne Rogers, and Mia Underwood. Dynamo: A Public Interactive Surface Supporting the Cooperative Sharing and Exchange of Media. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 159–168, New York, NY, USA, 2003. ACM. doi.org/10.1145/964696.964714.

[94] Robert J. K. Jacob. What You Look at is What You Get: Eye Movement-Based Interaction Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, page 11–18, New York, NY, USA, 1990. Association for Computing Machinery. doi.org/10.1145/97243.97246.

[95] Robert J. K. Jacob. Eye tracking in advanced interface design. *Virtual environments and advanced interface design*, 258(288):2, 1995.

[96] Thibaut Jacob, Gilles Bailly, Eric Lecolinet, Géry Casiez, and Marc Teyssier. Desktop Orbital Camera Motions Using Rotational Head Movements. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, SUI '16, pages 139–148, New York, NY, USA, 2016. ACM. doi.org/10.1145/2983310.2985758.

[97] Nuwan Nanayakkarawasam Peru Kandage Janaka, Shengdong Zhao, and Shardul Sapkota. Can Icons Outperform Text? Understanding the Role of Pictograms in OHMD Notifications. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. doi.org/10.1145/3544548.3580891.

[98] Hans-Christian Jetter and Harald Reiterer. Self-Organizing User Interfaces: Envisioning the Future of Ubicomp UIs. In *Workshop Blended Interaction (in conjunction with CHI ' 13 ACM SIGCHI Conference on Human Factors in Computing Systems)*. Universtiy of Konstanz, 2013.

[99] Haojian Jin, Christian Holz, and Kasper Hornbæk. Tracko: Ad-hoc Mobile 3D Tracking Using Bluetooth Low Energy and Inaudible Signals for Cross-Device Interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 147–156, New York, NY, USA, 2015. ACM. doi.org/10.1145/2807442.2807475.

[100] Wendy Ju. The Design of Implicit Interactions. *Synthesis Lectures on Human-Centered Informatics*, 8(2):1–93, 2015.

[101] Daniel Kahneman. *Attention and effort*, volume 1063. Citeseer, 1973.

[102] Amy K Karlson and Benjamin B Bederson. ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices. In *Human-Computer Interaction – INTERACT 2007*, pages 324–338, Berlin, Heidelberg, September 2007. Springer, Berlin, Heidelberg. doi.org/10.1007/978-3-540-74796-3_30.

[103] Amy K. Karlson and Benjamin B. Bederson. One-Handed Touchscreen Input for Legacy Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1399–1408, New York, NY, USA, 2008. ACM. doi.org/10.1145/1357054.1357274.

[104] Amy K. Karlson, Benjamin B. Bederson, and Jose L. Contreras-Vidal. Understanding One-Handed Use of Mobile Devices. *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*, 1(1):86–101, 2008. doi.org/10.4018/978-1-59904-871-0.ch006.

[105] Viktoria A. Kettner and Jeremy I. M. Carpendale. Developing Gestures for No and Yes: Head Shaking and Nodding in Infancy. *Gesture*, 13(2):193–209, January 2013. doi.org/10.1075/gest.13.2.04ket.

[106] Mohamed Khamis, Mariam Hassib, Emanuel von Zezschwitz, Andreas Bulling, and Florian Alt. GazeTouchPIN: Protecting Sensitive Data on Mobile Devices Using Secure Multimodal Authentication. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI '17, page 446–450. ACM, 2017. doi.org/10.1145/3136755.3136809.

[107] 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*, MobileHCI '18, pages 38:1–38:17, New York, NY, USA, 2018. ACM. doi.org/10.1145/3229434.3229452.

[108] 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*, CHI '18, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery. doi.org/10.1145/3173574.3173854.

[109] Sunjun Kim, Jihyun Yu, and Geehyuk Lee. Interaction Techniques for Unreachable Objects on the Touchscreen. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, OzCHI '12, pages 295–298, New York, NY, USA, 2012. ACM. doi.org/10.1145/2414536.2414585.

[110] Tracy L Kivell. Evidence in hand: recent discoveries and the early evolution of human manual manipulation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1682):20150105, 2015. doi.org/10.1098/rstb.2015.0105.

[111] R. 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, July 2001. doi.org/10.1109/RATFG.2001.938911.

[112] Michaela Klauck, Yusuke Sugano, and Andreas Bulling. Noticeable or Distractive? A Design Space for Gaze-Contingent User Interface Notifications. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, page 1779–1786, New York, NY, USA,

2017. Association for Computing Machinery. doi.org/10.1145/3027063.3053085.

[113] Paul B. Kline and Bob G. Witmer. Distance Perception in Virtual Environments: Effects of Field of View and Surface Texture at Near Distances. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 40(22): 1112–1116, 1996. doi.org/10.1177/154193129604002201.

[114] Clemens N. Klokmose, James R. Eagan, Siemen Baader, Wendy Mackay, and Michel Beaudouin-Lafon. Webstrates: Shareable Dynamic Media. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*, pages 280–290. ACM Press, 2015. doi.org/10.1145/2807442.2807446.

[115] Kristin Koch, Judith McLean, Ronen Segev, Michael A. Freed, Michael J. Berry, Vijay Balasubramanian, and Peter Sterling. How Much the Eye Tells the Brain. *Current Biology*, 16(14):1428–1434, 2006. doi.org/10.1016/j.cub.2006.05.056.

[116] Robert Kooima. Generalized Perspective Projection. *J. Sch. Electron. Eng. Comput. Sci*, 6, 2009.

[117] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye Tracking for Everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2176–2184, June 2016. doi.org/10.1109/CVPR.2016.239.

[118] Richard J. Krauzlis, Laurent Goffart, and Ziad M. Hafed. Neuronal control of fixation and fixational eye movements. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1718):20160205, 2017. doi.org/10.1098/rstb.2016.0205.

[119] Ernst Kruijff, J. Edward Swan, and Steven Feiner. Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pages 3–12, 2010. doi.org/10.1109/ISMAR.2010.5643530.

[120] Victor Kyriazakos and Konstantinos Moustakas. A User-Perspective View for Mobile AR Systems Using Discrete Depth Segmentation. In *2015 International Conference on Cyberworlds (CW)*, pages 69–72, 2015. doi.org/10.1109/CW.2015.67.

[121] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14, New York, NY, USA, 2018. Association for Computing Machinery. doi.org/10.1145/3173574.3173655.

[122] J. Lai and D. Zhang. ExtendedThumb: A Target Acquisition Approach for One-Handed Interaction With Touch-Screen Mobile Phones. *IEEE Transactions on Human-Machine Systems*, 45(3):362–370, June 2015. doi.org/10.1109/THMS.2014.2377205.

[123] R. Langner, T. Horak, and R. Dachselt. VisTiles: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):626–636, January 2018. doi.org/10.1109/TVCG.2017.2744019.

[124] Jeremy Lanman, Emilio Bizzi, and John Allum. The coordination of eye and head movement during smooth pursuit. *Brain Research*, 153(1):39–53, 1978. doi.org/10.1016/0006-8993(78)91127-7.

[125] Otto Lappi. Eye Tracking in the Wild: the Good, the Bad and the Ugly. *Journal of Eye Movement Research*, 8(5):1–21, October 2015. doi.org/10.16910/jemr.8.5.1.

[126] Talia Lavie and Joachim Meyer. Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies*, 68(8):508–524, 2010. doi.org/10.1016/j.ijhcs.2010.01.004.

[127] Huy Viet Le, Patrick Bader, Thomas Kosch, and Niels Henze. Investigating Screen Shifting Techniques to Improve One-Handed Smartphone Usage. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, NordiCHI '16, pages 27:1–27:10, New York, NY, USA, 2016. ACM. doi.org/10.1145/2971485.2971562.

[128] Jacek Lebiedź and Adam Mazikowski. Multiuser Stereoscopic Projection Techniques for CAVE-Type Virtual Reality Systems. *IEEE Transactions on Human-Machine Systems*, 51(5):535–543, 2021. doi.org/10.1109/THMS.2021.3102520.

[129] Yaxiong Lei, Shijing He, Mohamed Khamis, and Juan Ye. An End-to-End Review of Gaze Estimation and Its Interactive Applications on Handheld Mobile Devices. *ACM Comput. Surv.*, 56(2), September 2023. doi.org/10.1145/3606947.

[130] Wing Ho Andy Li and Hongbo Fu. BezelCursor: Bezel-initiated Cursor for One-handed Target Acquisition on Mobile Touch Screens. In *SIGGRAPH Asia 2013 Symposium on Mobile Graphics and Interactive Applications*, SA '13, pages 36:1–36:1, New York, NY, USA, 2013. ACM. doi.org/10.1145/2543651.2543680.

[131] Yang Li. Gesture Search: A Tool for Fast Mobile Data Access. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*,

UIST '10, pages 87–96, New York, NY, USA, 2010. ACM. doi.org/10.1145/1866029.1866044.

[132] Samuel J. Ling, Jeff Sanny, and William Moebs. *University Physics Volume 3*. OpenStax, Houston, TX, USA, September 2016. URL https://openstax.org/books/university-physics-volume-3/pages/2-5-the-eye.

[133] Jingjing May Liu, Gayathri Narasimham, Jeanine K. Stefanucci, Sarah Creem-Regehr, and Bobby Bodenheimer. Distance Perception in Modern Mobile Augmented Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 196–200, 2020. doi.org/10.1109/VRW50115.2020.00042.

[134] Markus Löchtefeld, Christoph Hirtz, and Sven Gehring. Evaluation of Hybrid Front- and Back-of-Device Interaction on Mobile Devices. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, MUM '13, pages 17:1–17:4, New York, NY, USA, 2013. ACM. doi.org/10.1145/2541831.2541865.

[135] Markus Löchtefeld, Phillip Schardt, Antonio Krüger, and Sebastian Boring. Detecting Users Handedness for Ergonomic Adaptation of Mobile User Interfaces. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, MUM '15, pages 245–249, New York, NY, USA, 2015. ACM. doi.org/10.1145/2836041.2836066.

[136] Edmund LoPresti, David M. Brienza, Jennifer Angelo, Lars Gilbertson, and Jonathan Sakai. Neck Range of Motion and Use of Computer Head Controls. In *Proceedings of the Fourth International ACM Conference on Assistive Technologies*, Assets '00, pages 121–128, New York, NY, USA, 2000. ACM. doi.org/10.1145/354324.354352.

[137] Tao Lu, Kui Yuan, Huosheng H. Hu, and Pei Jia. Head Gesture Recognition for Hands-Free Control of an Intelligent Wheelchair. *Industrial Robot: The International Journal of Robotics Research and Application*, 34(1):60–68, January 2007. doi.org/10.1108/01439910710718469.

[138] Andrés Lucero, Jussi Holopainen, and Tero Jokela. Pass-them-around: Collaborative Use of Mobile Phones for Photo Sharing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1787–1796, New York, NY, USA, 2011. ACM. doi.org/10.1145/1978942.1979201.

[139] Aristides Mairena, Carl Gutwin, and Andy Cockburn. Peripheral Notifications in Large Displays: Effects of Feature Combination and Task Interference. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery. doi.org/10.1145/3290605.3300870.

[140] Diako Mardanbegi, Dan Witzner Hansen, and Thomas Pederson. Eye-based Head Gestures. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 139–146, New York, NY, USA, 2012. ACM. doi.org/10.1145/2168556.2168578.

[141] Nicolai Marquardt, Ken Hinckley, and Saul Greenberg. Cross-device Interaction via Micro-mobility and F-formations. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 13–22, New York, NY, USA, 2012. ACM. doi.org/10.1145/2380116.2380121.

[142] Nicolai Marquardt, Frederik Brudy, Can Liu, Ben Bengler, and Christian Holz. SurfaceConstellations: A Modular Hardware Platform for Ad-Hoc Reconfigurable Cross-Device Workspaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14, New York, NY, USA, 2018. Association for Computing Machinery. doi.org/10.1145/3173574.3173928.

[143] Jonathan Samir Matthis, Jacob L. Yates, and Mary M. Hayhoe. Gaze and the Control of Foot Placement When Walking in Natural Terrain. *Current Biology*, 28(8):1224–1233.e5, April 2018. doi.org/10.1016/j.cub.2018.03.008.

[144] Evelyn Z. McClave. Linguistic Functions of Head Movements in the Context of Speech. *Journal of Pragmatics*, 32(7):855–878, June 2000. doi.org/10.1016/S0378-2166(99)00079-X.

[145] J. D. Mcdonald, A. T. Bahill, and M. B. Friedman. An adaptive control model for human head and eye movements while walking. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(2):167–174, March 1983. doi.org/10.1109/TSMC.1983.6313110.

[146] Abhinav Mehrotra, Mirco Musolesi, Robert Hendley, and Veljko Pejovic. Designing Content-Driven Intelligent Notification Mechanisms for Mobile Applications. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, page 813–824, New York, NY, USA, 2015. Association for Computing Machinery. doi.org/10.1145/2750858.2807544.

[147] Abhinav Mehrotra, Veljko Pejovic, Jo Vermeulen, Robert Hendley, and Mirco Musolesi. My Phone and Me: Understanding People's Receptivity to Mobile Notifications. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 1021–1032, New York, NY, USA, 2016. Association for Computing Machinery. doi.org/10.1145/2858036.2858566.

[148] Emiliano Miluzzo, Tianyu Wang, and Andrew T. Campbell. EyePhone: Activating Mobile Phones with Your Eyes. In *Proceedings of the Second*

ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds, MobiHeld '10, pages 15–20, New York, NY, USA, 2010. ACM. doi.org/10.1145/1851322.1851328.

[149] Peter Mohr, Markus Tatzgern, Jens Grubert, Dieter Schmalstieg, and Denis Kalkofen. Adaptive user perspective rendering for Handheld Augmented Reality. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 176–181, 2017. doi.org/10.1109/3DUI.2017.7893336.

[150] Louis-Philippe Morency and Trevor Darrell. Head Gesture Recognition in Intelligent Interfaces: The Role of Context in Improving Recognition. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, IUI '06, pages 32–38, New York, NY, USA, 2006. ACM. doi.org/10.1145/1111449.1111464.

[151] Takahiro Nagai, Kazuyuki Fujita, Kazuki Takashima, and Yoshifumi Kitamura. HandyGaze: A Gaze Tracking Technique for Room-Scale Environments using a Single Smartphone. *Proceedings of the ACM on Human-Computer Interaction*, 6(ISS), November 2022. doi.org/10.1145/3567715.

[152] Diederick C Niehorster, Thiago Santini, Roy S Hessels, Ignace TC Hooge, Enkelejda Kasneci, and Marcus Nyström. The impact of slippage on the data quality of head-worn eye trackers. *Behavior Research Methods*, 52:1140–1160, 2020. doi.org/10.3758/s13428-019-01307-0.

[153] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann, 1994. ISBN 0-12-518406-9.

[154] Tomi Nukarinen, Jari Kangas, Oleg Špakov, Poika Isokoski, Deepak Akkil, Jussi Rantala, and Roope Raisamo. Evaluation of HeadTurn: An Interaction Technique Using the Gaze and Head Turns. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, NordiCHI '16, pages 43:1–43:8, New York, NY, USA, 2016. ACM. doi.org/10.1145/2971485.2971490.

[155] I. Oakley and S. O'Modhrain. Tilt to scroll: evaluating a motion based vibrotactile mobile interface. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*, pages 40–49, New York, NY, USA, March 2005. IEEE. doi.org/10.1109/WHC.2005.138.

[156] Hiromu Ogawa, Kinji Matsumura, and Arisa Fujii. Appropriate Timing and Length of Voice News Notifications. In *ACM International Conference on Interactive Media Experiences*, IMX '21, page 194–198, New York, NY, USA, 2021. Association for Computing Machinery. doi.org/10.1145/3452918.3465488.

[157] Ji-young Oh and Hong Hua. User evaluations on form factors of tangible magic lenses. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 23–32, 2006. doi.org/10.1109/ISMAR.2006.297790.

[158] Dan O'Sullivan and Tom Igoe. *Physical Computing: Sensing and Controlling the Physical World with Computers*. Course Technology Press, Boston, MA, USA, 2004. ISBN 978-1592003464.

[159] Sharon Oviatt. Ten Myths of Multimodal Interaction. *Commun. ACM*, 42(11): 74–81, November 1999. doi.org/10.1145/319382.319398.

[160] D Paillé. Impact of new digital technologies on posture. *Points de Vue*, 72: 22–30, 2015.

[161] Laxmi Pandey, Khalad Hasan, and Ahmed Sabbir Arif. Acceptability of Speech and Silent Speech Input Methods in Private and Public. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. doi.org/10.1145/3411764.3445430.

[162] Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello, and Roy Want. TiltType: Accelerometer-Supported Text Entry for Very Small Devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*, UIST '02, page 201–204, New York, NY, USA, 2002. Association for Computing Machinery. doi.org/10.1145/571985.572013.

[163] Veljko Pejovic and Mirco Musolesi. InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, page 897–908, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2632048.2632062.

[164] Ken Pfeuffer and Hans Gellersen. Gaze and Touch Interaction on Tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 301–311, New York, NY, USA, 2016. Association for Computing Machinery. doi.org/10.1145/2984511.2984514.

[165] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. Gazetouch: combining gaze with multi-touch for interaction on the same surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, page 509–518, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2642918.2647397.

[166] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, Yanxia Zhang, and Hans Gellersen. Gaze-Shifting: Direct-Indirect Input with Pen and Touch Modulated by Gaze. In *Proceedings of the 28th Annual ACM Symposium on User Inter-*

face Software & Technology, UIST '15, page 373–383, New York, NY, USA, 2015. Association for Computing Machinery. doi.org/10.1145/2807442.2807460.

[167] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. GazeArchers: playing with individual and shared attention in a two-player look&shoot tabletop game. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia*, MUM '16, page 213–216, New York, NY, USA, 2016. Association for Computing Machinery. doi.org/10.1145/3012709.3012717.

[168] Andrea Phillipou, Larry A. Abel, David J. Castle, Matthew E. Hughes, Caroline Gurvich, Richard G. Nibbs, and Susan L. Rossell. Self perception and facial emotion perception of others in anorexia nervosa. *Frontiers in Psychology*, 6, 2015. doi.org/10.3389/fpsyg.2015.01181.

[169] Martin Pielot, Karen Church, and Rodrigo de Oliveira. An In-Situ Study of Mobile Phone Notifications. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*, MobileHCI '14, page 233–242, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2628363.2628364.

[170] Yue Qin, Chun Yu, Wentao Yao, Jiachen Yao, Chen Liang, Yueting Weng, Yukang Yan, and Yuanchun Shi. Selecting Real-World Objects via User-Perspective Phone Occlusion. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. doi.org/10.1145/3544548.3580696.

[171] Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ITS '14, page 45–54, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2669485.2669500.

[172] Roman Rädle, Hans-Christian Jetter, Mario Schreiner, Zhihao Lu, Harald Reiterer, and Yvonne Rogers. Spatially-aware or Spatially-agnostic? Elicitation and Evaluation of User-Defined Cross-Device Interactions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 3913–3922, New York, NY, USA, 2015. Association for Computing Machinery. doi.org/10.1145/2702123.2702287.

[173] Roman Rädle, Hans-Christian Jetter, Jonathan Fischer, Inti Gabriel, Clemens N. Klokmose, Harald Reiterer, and Christian Holz. PolarTrack: Optical Outside-In Device Tracking that Exploits Display Polarization. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–9, New York, NY, USA, 2018. Association for Computing Machinery. doi.org/10.1145/3173574.3174071.

[174] Jef Raskin. *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000. ISBN 0-201-37937-6.

[175] Keith Rayner. Eye movement latencies for parafoveally presented words. *Bulletin of the Psychonomic Society*, 11(1):13–16, 1978. doi.org/10.3758/BF03336753.

[176] Keith Rayner. The 35th Sir Frederick Bartlett Lecture: Eye movements and attention in reading, scene perception, and visual search. *Quarterly Journal of Experimental Psychology*, 62(8):1457–1506, 2009. doi.org/10.1080/17470210902816461.

[177] Jun Rekimoto. Tilting Operations for Small Screen Interfaces. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, page 167–168, New York, NY, USA, 1996. Association for Computing Machinery. doi.org/10.1145/237091.237115.

[178] Jun Rekimoto. Pick-and-drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, pages 31–39, New York, NY, USA, 1997. ACM. doi.org/10.1145/263407.263505.

[179] Jun Rekimoto, Yuji Ayatsuka, and Michimune Kohno. SyncTap: An Interaction Technique for Mobile Networking. In *Human-Computer Interaction with Mobile Devices and Services*, pages 104–115, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. doi.org/10.1007/978-3-540-45233-1_9.

[180] Christian Remy, Malte Weiss, Martina Ziefle, and Jan Borchers. A Pattern Language for Interactive Tabletops in Collaborative Workspaces. In *Proceedings of the 15th European Conference on Pattern Languages of Programs*, Irsee Monastery, Bavaria, Germany, July 2010. doi.org/10.1145/2328909.2328921.

[181] Donghao Ren, Tibor Goldschwendt, YunSuk Chang, and Tobias Höllerer. Evaluating wide-field-of-view augmented reality with mixed reality simulation. In *2016 IEEE Virtual Reality (VR)*, pages 93–102, 2016. doi.org/10.1109/VR.2016.7504692.

[182] Brian Rogers and Maureen Graham. Motion Parallax as an Independent Cue for Depth Perception. *Perception*, 8(2):125–134, 1979. doi.org/10.1068/p080125.

[183] Joan Sol Roo, Jean Basset, Pierre-Antoine Cinquin, and Martin Hachet. *Understanding Users' Capability to Transfer Information between Mixed and Virtual Reality: Position Estimation across Modalities and Perspectives*, pages 1–12.

Association for Computing Machinery, New York, NY, USA, 2018. ISBN 9781450356206. doi.org/10.1145/3173574.3173937.

[184] Ling Rothrock, Richard Koubek, Frederic Fuchs, Michael Haas, and Gavriel Salvendy. Review and reappraisal of adaptive interfaces: Toward biologically inspired paradigms. *Theoretical Issues in Ergonomics Science*, 3(1):47–84, 2002. doi.org/10.1080/14639220110110342.

[185] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. TapTap and MagStick: Improving One-Handed Target Acquisition on Small Touch-Screens. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, pages 146–153, New York, NY, USA, 2008. ACM. doi.org/10.1145/1385569.1385594.

[186] Rufat Rzayev, Sven Mayer, Christian Krauter, and Niels Henze. Notification in VR: The Effect of Notification Placement, Task and Environment. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '19, page 199–211, New York, NY, USA, 2019. Association for Computing Machinery. doi.org/10.1145/3311350.3347190.

[187] H. H. Sad and F. Poirier. Evaluation and Modeling of User Performance for Pointing and Scrolling Tasks on Handheld Devices Using Tilt Sensor. In *2009 Second International Conferences on Advances in Computer-Human Interactions*, pages 295–300, New York, NY, USA, February 2009. IEEE. doi.org/10.1109/ACHI.2009.15.

[188] Alireza Sahami Shirazi, Niels Henze, Tilman Dingler, Martin Pielot, Dominik Weber, and Albrecht Schmidt. Large-Scale Assessment of Mobile Notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 3055–3064, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2556288.2557189.

[189] Ali Samini and Karljohan Lundin Palmerius. A Perspective Geometry Approach to User-Perspective Rendering in Hand-Held Video See-through Augmented Reality. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, VRST '14, page 207–208, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2671015.2671127.

[190] Daniel J. Sandin, Todd Margolis, Jinghua Ge, Javier Girado, Tom Peterka, and Thomas A. DeFanti. The VarrierTM Autostereoscopic Virtual Reality Display. *ACM Trans. Graph.*, 24(3):894–903, July 2005. doi.org/10.1145/1073204.1073279.

[191] Marian Sauter, Tobias Wagner, Teresa Hirzle, Bao Xin Lin, Enrico Rukzio, and Anke Huckauf. Behind the Screens: Exploring Eye Movement Visualization

to Optimize Online Teaching and Learning. In *Proceedings of Mensch Und Computer 2023*, MuC '23, page 67–80, New York, NY, USA, 2023. Association for Computing Machinery. doi.org/10.1145/3603555.3603560.

[192] Joey Scarr, Andy Cockburn, and Carl Gutwin. Supporting and Exploiting Spatial Memory in User Interfaces. *Foundations and Trends® in Human–Computer Interaction*, 6(1):1–84, 2013. doi.org/10.1561/1100000046.

[193] Nicu Sebe. Multimodal interfaces: Challenges and perspectives. *Journal of Ambient Intelligence and smart environments*, 1(1):23–30, 2009. doi.org/10.3233/AIS-2009-0003.

[194] Linda E. Sibert and Robert J. K. Jacob. Evaluation of Eye Gaze Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, page 281–288, New York, NY, USA, 2000. Association for Computing Machinery. doi.org/10.1145/332040.332445.

[195] Anton Sigitov, Ernst Kruijff, Christina Trepkowski, Oliver Staadt, and André Hinkenjann. The Effect of Visual Distractors in Peripheral Vision on User Performance in Large Display Wall Systems. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*, ISS '16, page 241–249, New York, NY, USA, 2016. Association for Computing Machinery. doi.org/10.1145/2992154.2992165.

[196] Adalberto L. Simeone, Julian Seifert, Dominik Schmidt, Paul Holleis, Enrico Rukzio, and Hans Gellersen. A Cross-device Drag-and-drop Technique. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, MUM '13, pages 10:1–10:4, New York, NY, USA, 2013. ACM. doi.org/10.1145/2541831.2541848.

[197] Daniel Spelmezan, Caroline Appert, Olivier Chapuis, and Emmanuel Pietriga. Side Pressure for Bidirectional Navigation on Small Devices. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 11–20, New York, NY, USA, 2013. ACM. doi.org/10.1145/2493190.2493199.

[198] India Starker and Richard A. Bolt. A Gaze-Responsive Self-Disclosing Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, page 3–10, New York, NY, USA, 1990. Association for Computing Machinery. doi.org/10.1145/97243.97245.

[199] Ian Stavness, Billy Lam, and Sidney Fels. PCubee: A Perspective-Corrected Handheld Cubic Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 1381–1390, New York, NY, USA, 2010. Association for Computing Machinery. doi.org/10.1145/1753326.1753535.

[200] Sophie Stellmach and Raimund Dachselt. Look & Touch: Gaze-Supported Target Acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 2981–2990, New York, NY, USA, 2012. Association for Computing Machinery. doi.org/10.1145/2207676.2208709.

[201] Sophie Stellmach and Raimund Dachselt. Investigating Gaze-supported Multimodal Pan and Zoom. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 357–360, New York, NY, USA, 2012. ACM. doi.org/10.1145/2168556.2168636.

[202] Sophie Stellmach and Raimund Dachselt. Look & Touch: Gaze-supported Target Acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2981–2990, New York, NY, USA, 2012. ACM. doi.org/10.1145/2207676.2208709.

[203] 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*, CHI '13, page 285–294, New York, NY, USA, 2013. Association for Computing Machinery. doi.org/10.1145/2470654.2470695.

[204] Rainer Stiefelhagen, Michael Finke, Jie Yang, and Alex Waibel. From Gaze to Focus of Attention. In *Visual Information and Information Systems: Third International Conference, VISUAL'99 Amsterdam, The Netherlands, June 2–4, 1999 Proceedings 3*, pages 765–772. Springer, 1999. doi.org/10.1007/3-540-48762-X_94.

[205] Cary Stothart, Ainsley Mitchum, and Courtney Yehnert. The attentional cost of receiving a cell phone notification. *Journal of Experimental Psychology: Human Perception and Performance*, 41(4):893–897, August 2015. doi.org/10.1037/xhp0000100.

[206] Leon Strapper, Robert Mertens, Sebastian Pospiech, Florian Bussmann, Arthur Grah, and Marius Mamsch. A Gaze Tracking Based, Multi Modal Human Computer Interaction Concept for Efficient Input. In *2017 IEEE International Symposium on Multimedia (ISM)*, pages 268–273, 2017. doi.org/10.1109/ISM.2017.46.

[207] Hans Strasburger. Seven Myths on Crowding and Peripheral Vision. *i-Perception*, 11(3):2041669520913052, 2020. doi.org/10.1177/2041669520913052.

[208] Qingkun Su, Oscar Kin-Chung Au, Pengfei Xu, Hongbo Fu, and Chiew-Lan Tai. 2D-Dragger: Unified Touch-Based Target Acquisition with Constant Effective Width. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16,

pages 170–179, New York, NY, USA, 2016. ACM. doi.org/10.1145/2935334.2935339.

[209] Xuetong Sun and Amitabh Varshney. Investigating Perception Time in the Far Peripheral Vision for Virtual and Augmented Reality. In *Proceedings of the 15th ACM Symposium on Applied Perception*, SAP '18, New York, NY, USA, 2018. Association for Computing Machinery. doi.org/10.1145/3225153.3225160.

[210] J. Edward Swan, II, Liisa Kuparinen, Scott Rapson, and Christian Sandor. Visually Perceived Distance Judgments: Tablet-Based Augmented Reality Versus the Real World. *International Journal of Human–Computer Interaction*, 33(7): 576–591, 2017. doi.org/10.1080/10447318.2016.1265783.

[211] Timothy D. Sweeny and David Whitney. Perceiving Crowd Attention: Ensemble Perception of a Crowd's Gaze. *Psychological Science*, 25(10):1903–1913, 2014. doi.org/10.1177/0956797614544510.

[212] Zhenyu Tang, Chenyu Yan, Sijie Ren, and Huagen Wan. HeadPager: Page Turning with Computer Vision Based Head Interaction. In *Computer Vision – ACCV 2016 Workshops*, Lecture Notes in Computer Science, pages 249–257, Cham, Switzerland, 2017. Springer International Publishing. doi.org/10.1007/978-3-319-54526-4_19.

[213] Vildan Tanriverdi and Robert J. K. Jacob. Interacting with Eye Movements in Virtual Environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, page 265–272, New York, NY, USA, 2000. Association for Computing Machinery. doi.org/10.1145/332040.332443.

[214] Dan Tasse, Anupriya Ankolekar, and Joshua Hailpern. Getting Users' Attention in Web Apps in Likable, Minimally Annoying Ways. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 3324–3334, New York, NY, USA, 2016. Association for Computing Machinery. doi.org/10.1145/2858036.2858174.

[215] Robert J. Teather and I. Scott MacKenzie. *Position vs. Velocity Control for Tilt-Based Interaction*, page 51–58. GI '14. Canadian Information Processing Society, 2014. ISBN 9781482260038.

[216] William B. Thompson, Peter Willemsen, Amy A. Gooch, Sarah H. Creem-Regehr, Jack M. Loomis, and Andrew C. Beall. Does the Quality of the Computer Graphics Matter when Judging Distances in Visually Immersive Environments? *Presence*, 13(5):560–571, 2004. doi.org/10.1162/1054746042545292.

[217] Feng Tian, Lishuang Xu, Hongan Wang, Xiaolong Zhang, Yuanyuan Liu, Vidya Setlur, and Guozhong Dai. Tilt Menu: Using the 3D Orientation Information of Pen Devices to Extend the Selection Capability of Pen-Based User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 1371–1380, New York, NY, USA, 2008. Association for Computing Machinery. doi.org/10.1145/1357054.1357269.

[218] Makoto Tomioka, Sei Ikeda, and Kosuke Sato. Approximated User-Perspective Rendering in Tablet-Based Augmented Reality. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 21–28, 2013. doi.org/10.1109/ISMAR.2013.6671760.

[219] Hsin-Ruey Tsai, Da-Yuan Huang, Chen-Hsin Hsieh, Lee-Ting Huang, and Yi-Ping Hung. MovingScreen: Selecting Hard-to-Reach Targets with Automatic Comfort Zone Calibration on Mobile Devices. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, MobileHCI '16, pages 651–658, New York, NY, USA, 2016. ACM. doi.org/10.1145/2957265.2961835.

[220] Jayson Turner, Andreas Bulling, and Hans Gellersen. Combining Gaze with Manual Interaction to Extend Physical Reach. In *Proceedings of the 1st International Workshop on Pervasive Eye Tracking & Mobile Eye-based Interaction*, PETMEI '11, pages 33–36, New York, NY, USA, 2011. ACM. doi.org/10.1145/2029956.2029966.

[221] Jayson Turner, Andreas Bulling, Jason Alexander, and Hans Gellersen. Cross-device Gaze-supported Point-to-point Content Transfer. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14, pages 19–26, New York, NY, USA, 2014. ACM. doi.org/10.1145/2578153.2578155.

[222] Jayson Turner, Jason Alexander, Andreas Bulling, and Hans Gellersen. Gaze+RST: Integrating Gaze and Multitouch for Remote Rotate-Scale-Translate Tasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 4179–4188, New York, NY, USA, 2015. ACM. doi.org/10.1145/2702123.2702355.

[223] Vincent van Rheden, Bernhard Maurer, Dorothé Smit, Martin Murer, and Manfred Tscheligi. LaserViz: Shared Gaze in the Co-Located Physical World. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '17, pages 191–196, New York, NY, USA, 2017. ACM. doi.org/10.1145/3024969.3025010.

[224] Klen Čopič Pucihar, Paul Coulton, and Jason Alexander. Evaluating Dual-View Perceptual Issues in Handheld Augmented Reality: Device vs. User Perspective Rendering. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 381–388, New York, NY, USA,

2013. Association for Computing Machinery. doi.org/10.1145/2522848.2522885.

[225] Bill Verplank. Interaction Design Sketchbook, 2009. URL http://www.billverplank.com/IxDSketchBook.pdf.

[226] Roel Vertegaal, Aadil Mamuji, Changuk Sohn, and Daniel Cheng. Media Eyepliances: Using Eye Tracking for Remote Control Focus Selection of Appliances. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1861–1864, New York, NY, USA, 2005. ACM. doi.org/10.1145/1056808.1057041.

[227] Simon Voelker, Andrii Matviienko, Johannes Schöning, and Jan Borchers. Combining Direct and Indirect Touch Input for Interactive Workspaces using Gaze Input. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, SUI '15, page 79–88, New York, NY, USA, 2015. Association for Computing Machinery. doi.org/10.1145/2788940.2788949.

[228] Simon Voelker, Sebastian Hueber, Christian Corsten, and Christian Remy. HeadReach: Using Head Tracking to Increase Reachability on Mobile Touch Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA, April 2020. Association for Computing Machinery. doi.org/10.1145/3313831.3376868.

[229] Simon Voelker, Sebastian Hueber, Christian Holz, Christian Remy, and Nicolai Marquardt. GazeConduits: Calibration-Free Cross-Device Collaboration through Gaze and Touch. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–10, New York, NY, USA, April 2020. Association for Computing Machinery. doi.org/10.1145/3313831.3376578.

[230] Chat Wacharamanotham, Jan Hurtmanns, Alexander Mertens, Martin Kronenbuerger, Christopher Schlick, and Jan Borchers. Evaluating Swabbing: A Touchscreen Input Method for Elderly Users with Tremor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 623–626, New York, NY, USA, 2011. ACM. doi.org/10.1145/1978942.1979031.

[231] Philipp Wacker, Adrian Wagner, Simon Voelker, and Jan Borchers. Heatmaps, Shadows, Bubbles, Rays: Comparing Mid-Air Pen Position Visualizations in Handheld AR. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–11, New York, NY, USA, 2020. Association for Computing Machinery. doi.org/10.1145/3313831.3376848.

[232] Tobias Wagner, Teresa Hirzle, Anke Huckauf, and Enrico Rukzio. Exploring Gesture and Gaze Proxies to Communicate Instructor's Nonverbal Cues in

Lecture Videos. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, New York, NY, USA, 2023. Association for Computing Machinery. doi.org/10.1145/3544549.3585842.

[233] Colin Ware and Harutune H. Mikaelian. An Evaluation of an Eye Tracker As a Device for Computer Input. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, CHI '87, pages 183–188, New York, NY, USA, 1987. ACM. doi.org/10.1145/29933.275627.

[234] Mark Weiser. The Computer for the 21 st Century. *Scientific American*, 265(3): 94–105, 1991.

[235] Elisabeth Wells-Parker, Jennifer Ceminsky, Victoria Hallberg, Ronald W Snow, Gregory Dunaway, Shawn Guiling, Marsha Williams, and Bradley Anderson. An exploratory study of the relationship between road rage and crash experience in a representative sample of US drivers. *Accident Analysis & Prevention*, 34(3):271–278, 2002. doi.org/10.1016/S0001-4575(01)00021-5.

[236] Brian Whitworth. Polite computing. *Behaviour & Information Technology*, 24 (5):353–363, 2005. doi.org/10.1080/01449290512331333700.

[237] Daniel Wigdor and Ravin Balakrishnan. TiltText: Using Tilt for Text Input to Mobile Phones. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, page 81–90, New York, NY, USA, 2003. Association for Computing Machinery. doi.org/10.1145/964696.964705.

[238] Daniel Wigdor, Hao Jiang, Clifton Forlines, Michelle Borkin, and Chia Shen. WeSpace: The Design Development and Deployment of a Walk-up and Share Multi-surface Visual Collaboration System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1237–1246, New York, NY, USA, 2009. ACM. doi.org/10.1145/1518701.1518886.

[239] Julie R. Williamson, Stephen Brewster, and Rama Vennelakanti. Mo!Games: Evaluating Mobile Gestures in the Wild. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 173–180, New York, NY, USA, 2013. ACM. doi.org/10.1145/2522848.2522874.

[240] Graham Wilson, Stephen A. Brewster, Martin Halvey, Andrew Crossan, and Craig Stewart. The Effects of Walking, Feedback and Control Method on Pressure-Based Interaction. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 147–156, New York, NY, USA, 2011. ACM. doi.org/10.1145/2037373.2037397.

[241] Graham Wilson, Stephen A. Brewster, Martin Halvey, Andrew Crossan, and Craig Stewart. The Effects of Walking, Feedback and Control Method on

Pressure-based Interaction. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 147–156, New York, NY, USA, 2011. ACM. doi.org/10.1145/2037373.2037397.

[242] Bob G. Witmer and Wallace J. Sadowski, Jr. Nonvisually Guided Locomotion to a Previously Viewed Target in Real and Virtual Environments. *Human Factors*, 40(3):478–488, 1998. doi.org/10.1518/001872098779591340.

[243] Jacob O. Wobbrock and Julie A. Kientz. Research Contributions in Human-Computer Interaction. *Interactions*, 23(3):38–44, April 2016. doi.org/10.1145/2907069.

[244] Erroll Wood and Andreas Bulling. EyeTab: Model-based Gaze Estimation on Unmodified Tablet Computers. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14, pages 207–210, New York, NY, USA, 2014. ACM. doi.org/10.1145/2578153.2578185.

[245] Pawel Wozniak, Nitesh Goyal, Przemys\law Kucharski, Lars Lischke, Sven Mayer, and Morten Fjeld. RAMPARTS: Supporting Sensemaking with Spatially-Aware Mobile Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 2447–2460, New York, NY, USA, 2016. ACM. doi.org/10.1145/2858036.2858491.

[246] Chi-Jui Wu, Steven Houben, and Nicolai Marquardt. EagleSense: Tracking People and Devices in Interactive Spaces Using Real-Time Top-View Depth-Sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3929–3942, New York, NY, USA, 2017. ACM. doi.org/10.1145/3025453.3025562.

[247] Yukang Yan, Chun Yu, Xin Yi, and Yuanchun Shi. HeadGesture: Hands-Free Input Approach Leveraging Head Movements for HMD Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable, and Ubiquitous Technologies (IMWUT '18)*, 2(4):198:1–198:23, December 2018. doi.org/10.1145/3287076.

[248] Jing Yang, Shiheng Wang, and Gábor Sörös. User-Perspective Rendering for Handheld Applications. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 270–274, 2018. doi.org/10.1109/ISMAR-Adjunct.2018.00084.

[249] Xing-Dong Yang, Edward Mak, Pourang Irani, and Walter F. Bischof. Dual-Surface Input: Augmenting One-Handed Interaction with Coordinated Front and Behind-the-Screen Input. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 5:1–5:10, New York, NY, USA, 2009. ACM. doi.org/10.1145/1613858.1613865.

[250] Shanhe Yi, Zhengrui Qin, Ed Novak, Yafeng Yin, and Qun Li. GlassGesture: Exploring Head Gesture Interface of Smart Glasses. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, New York, NY, USA, April 2016. IEEE. doi.org/10.1109/INFOCOM.2016.7524542.

[251] Hyunjin Yoo, Jungwon Yoon, and Hyunsoo Ji. Index Finger Zone: Study on Touchable Area Expandability Using Thumb and Index Finger. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, MobileHCI '15, pages 803–810, New York, NY, USA, 2015. ACM. doi.org/10.1145/2786567.2793704.

[252] Neng-Hao Yu, Da-Yuan Huang, Jia-Jyun Hsu, and Yi-Ping Hung. Rapid Selection of Hard-to-Access Targets by Thumb on Mobile Touch-Screens. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 400–403, New York, NY, USA, 2013. ACM. doi.org/10.1145/2493190.2493202.

[253] Shumin Zhai. What's in the Eyes for Attentive Input. *Commun. ACM*, 46(3): 34–39, March 2003. doi.org/10.1145/636772.636795.

[254] 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*, CHI '99, page 246–253, New York, NY, USA, 1999. Association for Computing Machinery. doi.org/10.1145/302979.303053.

[255] Chi Zhang, Nan Jiang, and Feng Tian. Accessing Mobile Apps with User Defined Gesture Shortcuts: An Exploratory Study. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*, ISS '16, pages 385–390, New York, NY, USA, 2016. ACM. doi.org/10.1145/2992154.2996786.

[256] Xinyong Zhang, Xiangshi Ren, and Hongbin Zha. Improving Eye Cursor's Stability for Eye Pointing Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 525–534, New York, NY, USA, 2008. Association for Computing Machinery. doi.org/10.1145/1357054.1357139.

[257] Yanxia Zhang, Andreas Bulling, and Hans Gellersen. SideWays: A Gaze Interface for Spontaneous Interaction with Situated Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 851–860. ACM, 2013. doi.org/10.1145/2470654.2470775.

[258] Yanxia Zhang, Ken Pfeuffer, Ming Ki Chong, Jason Alexander, Andreas Bulling, and Hans Gellersen. Look together: using gaze for assisting co-

located collaborative search. *Personal and Ubiquitous Computing*, 21(1):173–186, February 2017. doi.org/10.1007/s00779-016-0969-x.

[259] M. Zimmermann. *The Nervous System in the Context of Information Theory*, pages 166–173. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989. ISBN 978-3-642-73831-9. doi.org/10.1007/978-3-642-73831-9_7.

[260] Oleg Špakov, Poika Isokoski, Jari Kangas, Jussi Rantala, Deepak Akkil, and Roope Raisamo. Comparison of Three Implementations of HeadTurn: A Multimodal Interaction Technique with Gaze and Head Turns. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI '16, pages 289–296, New York, NY, USA, 2016. ACM. doi.org/10.1145/2993148.2993153.

Index

Accessibility 33, 67
ARKit 35
Around-body interaction 16
Attentive Notifications 159–179, 184–185
Augmented Reality (AR) 126

BezelCursor 77, 85
BezelSpace 78

Camera frustum 27
Cave automatic virtual environment 29
Clippy 32
Collaboration 17
Communication bandwidth 4, 187
Contrast 166
CornerSpace 78
Cross-device interactions 106–108

Depth perception 132–133
Device-perspective rendering 126
Dimensionality of user interfaces 5
Distraction 162
Double role of eye gaze 26, 186
Dual-view problem 133
Dwell time 21, 169

Ergonomics 189

ExtendedThumb 77
Eye 19
- Acuity 19, 22, 24
- Movements 6, 20
EyePhone 34

Facial expressions 2
Facial tracking 8, 10–11, 34–35
Field of view (FOV) 133
Fixation 20
ForceRay 77

Gaze tracking
- Challenges 18–21
- Collective gaze 21
- Indicator of attention 6, 21
- Selection times 7, 26
Gaze-explicit notifications 169
Gaze-implicit notifications 169
GazeConduits 103–124, 183–184
Gesture Search 49

Head Area + Touch Selection (HA) 84
Head gestures 6, 51
Head tracking
- Head follows gaze 21
- Selection times 26
- Stability 25–26
- Visibility of users 23
Head + Touch Selection (HT) 82
Headbang 45–70, 182
HeadNod, HeadPager 52
HeadReach 71–101, 182–183
HeadTurn 52

How the computer sees us 2
HuddleLamp 17
Human information processor 2

Intelligent user interfaces 31–33
InterruptMe 163
iPad 112, 141
iPhone 37, 55, 80, 110, 165
iRecipe 51

Lean and Zoom 15

Magic lens 30, 130
MAGIC pointing 18
MagStick 77
Midas touch problem 7, 20
Motion parallax 129
Multimodality 3

Notifications 162

Occlusion 129, 165

pCubee 30
Phone case 111
Pick-and-Drop 107
Pie menu 59
Pointing input 2
PolarTrack 108
Privacy 120, 177, 189
Proxemics 14–15
Pure Head Selection (PH) 80
Put that There 3

Quasi-mode 80

Reachability technique 74–78

Research questions 10

Saccade 20
Screen-space content 9
Shadow 132
Sheep 127
Sliding Screen 75
Smartphone 8
Smooth pursuit 21
Stereoscopy 22, 126
SurfaceConstellations 107

TapTap 76
ThumbSpace 76
TiltCursor 77
Tilting input 50
TiltReduction 75
Touch input 7
- Combining gaze with touch 26
- Missing hover state 9
Touch-attentive notifications 170
Transfer function 81–82

Ubiquitous computing 14
Usability Engineering 5
User-Aware Rendering 125–157, 184
User-perspective rendering 28

Varrier 30
VisTiles 106
Visual angle 24
Visual field 22
Voronoi grid 113

World of Windows 6
World-space content 9

Own Publications

Papers (Peer-reviewed, archival)

Sebastian Hueber, Johannes Wilhelm, René Schäfer, Simon Voelker, and Jan Borchers. User-Aware Rendering: Merging the Strengths of Device- and User-Perspective Rendering in Handheld AR. *Proceedings of the ACM on Human-Computer Interaction*, Vol. 7(MHCI), September 2023. doi.org/10.1145/3604278

Contribution and Benefits: Presents an AR technique that calculates the frustum based on user's head position. It combines a large FOV with scene alignment and motion parallax. The visual outcome provides users strong depth perception with good scene overview.

Acceptance Rate: 38%

Sebastian Hueber, Christian Cherek, Philipp Wacker, Jan Borchers, and Simon Voelker. Headbang: Using Head Gestures to Trigger Discrete Actions on Mobile Devices. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '20, New York, NY, USA, October 2020. Association for Computing Machinery. doi.org/10.1145/3379503.3403538

Contribution and Benefits: Presents an interaction technique for mobile devices that uses head gestures to activate shortcuts and perform menu selections. Head gestures can be operated faster than touch input.

Acceptance Rate: 24%

Simon Voelker, **Sebastian Hueber**, Christian Holz, Christian Remy, and Nicolai Marquardt. GazeConduits: Calibration-Free Cross-Device Collaboration through Gaze and Touch. In *Proceedings of the 2020 CHI Conference on Human Factors in*

Computing Systems, CHI '20, page 1–10, New York, NY, USA, April 2020. Association for Computing Machinery. doi.org/10.1145/3313831.3376578

Contribution and Benefits: Presents a calibration-free ad-hoc mobile cross-device interaction concept using gaze and touch input. Collaboration is fostered through tracking capabilities of user presence, and gazing at tablets and gaze interactions with collaborators and tablets.

Acceptance Rate: 24%

Simon Voelker, **Sebastian Hueber**, Christian Corsten, and Christian Remy. HeadReach: Using Head Tracking to Increase Reachability on Mobile Touch Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA, April 2020. Association for Computing Machinery. doi.org/10.1145/3313831.3376868

Contribution and Benefits: Compares head tracking-based techniques to increase thumb reach during one-handed smartphone use. Two of the techniques yield higher success rates for selecting targets compared to direct touch input with the thumb.

Acceptance Rate: 24%

Posters (Peer-reviewed, semi-archival)

Sebastian Hueber, Eunae Jang, and Jan Borchers. Attentive Notifications: Minimizing Distractions of Mobile Notifications through Gaze Tracking. In *Proceedings of the 25th International Conference on Mobile Human-Computer Interaction*, MobileHCI '23 Companion, New York, NY, USA, September 2023. Association for Computing Machinery. doi.org/10.1145/3565066.3608695

Contribution and Benefits: Presents interaction techniques for mobile notifications. Gaze-explicit notifications appear away from the user's gaze point and enlarge when shifting gaze to them. Study results suggest this reduces occlusions and distractions.

★ *Best Late-Breaking Work Award*

Acceptance Rate: 45%

Demos (Juried, non-archival)

Jan Borchers, Anke Brocker, **Sebastian Hueber**, Oliver Nowak, René Schäfer, Adrian Wagner, Paul Miles Preuschoff, and Lea Emilia Schirp. The Aachen Lab Demo: From Fundamental Perception to Design Tools. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, New York, NY, USA, 2023. Association for Computing Machinery. doi.org/10.1145/3544549.3583937

Contribution and Benefits: Presents a collection of ongoing research projects in interactive demos.

Theses

Sebastian Hueber. Seeing Analog Forests and Digital Trees? — Impact of Digital Devices on the Construal Level. Master's Thesis, RWTH Aachen University, March 2018.

Contribution and Benefits: Presents two field studies (n=120) on the conceptual and perceptual construal level when conducted on either a digital or analog platform. Results indicate that digital displays have no inherent negative impact on their user's construal level.

Sebastian Hueber. Back-of-device Tactile Landmarks for Eyes-free Touch Input. Bachelor's Thesis, RWTH Aachen University, September 2015.

Contribution and Benefits: Presents different tactile landmark arrangements on the back of a smartphone for eyes-free touch inputs on the front-facing screen. A user study shows that too many landmarks reduce accuracy, while a close mapping of visual to tactile landmarks enhances accuracy.

Typeset November 4, 2024