3D Pointing with Everyday Devices: Speed, Occlusion, Fatigue

by

Krzysztof Pietroszek

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2015

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contribution

This dissertation includes first-authored peer-reviewed material that has appeared in conference proceedings published by the Association for Computing Machinery (ACM). The publisher's policy on reuse of published materials in a dissertation is as follows[1]:

> Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included.

The following list serves as a declaration of the Versions of Record for works included in this dissertation:

Portions of Chapter 4

Krzysztof Pietroszek, Anastasia Kuzminykh, James R. Wallace, and Edward Lank. 2014. Smartcasting: a discount 3D interaction technique for public displays. *In Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: the Future of Design (OzCHI '14).* ACM, New York, NY, USA, 119-128.

`DOI=http://doi.acm.org/10.1145/2686612.2686629`

Section on "Spatial correspondence" in Chapter 6

Krzysztof Pietroszek and Edward Lank. 2012. Clicking blindly: using spatial correspondence to select targets in multi-device environments. *In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services (MobileHCI '12).* ACM, New York, NY, USA, 331-334.

`DOI=http://doi.acm.org/10.1145/2371574.2371625`

[1] `http://authors.acm.org/main.html`, retrieved: July 21, 2015

Portions of Chapter 6

Krzysztof Pietroszek, James R. Wallace, and Edward Lank. 2015. Tiltcasting: 3D Interaction on Large Displays using a Mobile Device. *In Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA.

`DOI=http://doi.acm.org/10.1145/2807442.2807471`

Abstract

In recent years, display technology has evolved to the point where displays can be both non-stereoscopic and stereoscopic, and 3D environments can be rendered realistically on many types of displays. From movie theatres and shopping malls to conference rooms and research labs, 3D information can be deployed seamlessly. Yet, while 3D environments are commonly displayed in desktop settings, there are virtually no examples of interactive 3D environments deployed within ubiquitous environments, with the exception of console gaming. At the same time, immersive 3D environments remain – in users' minds – associated with professional work settings and virtual reality laboratories. An excellent opportunity for 3D interactive engagements is being missed not because of economic factors, but due to the lack of interaction techniques that are easy to use in ubiquitous, everyday environments.

In my dissertation, I address the lack of support for interaction with 3D environments in ubiquitous settings by designing, implementing, and evaluating 3D pointing techniques that leverage a smartphone or a smartwatch as an input device. I show that mobile and wearable devices may be especially beneficial as input devices for casual use scenarios, where specialized 3D interaction hardware may be impractical, too expensive or unavailable. Such scenarios include interactions with home theatres, intelligent homes, in workplaces and classrooms, with movie theatre screens, in shopping malls, at airports, during conference presentations and countless other places and situations.

Another contribution of my research is to increase the potential of mobile and wearable devices for efficient interaction at a distance. I do so by showing that such interactions are feasible when realized with the support of a modern smartphone or smartwatch. I also show how multimodality, when realized with everyday devices, expands and supports 3D pointing. In particular, I show how multimodality helps to address the challenges of 3D interaction: performance issues related to the limitations of the human motor system, interaction with occluded objects and related problem of perception of depth on non-stereoscopic screens, and user subjective fatigue, measured with NASA TLX as perceived workload, that results from providing spatial input for a prolonged time.

I deliver these contributions by designing three novel 3D pointing techniques that support casual, "walk-up-and-use" interaction at a distance and are fully realizable using off-the-shelf mobile and wearable devices available today. The contributions provide evidence that democratization of 3D interaction can be realized by leveraging the pervasiveness of a device that users already carry with them: a smartphone or a smartwatch.

Acknowledgements

First and foremost I would like to thank my supervisor Dr. Edward Lank and my co-supervisor Dr. James Wallace who provided me with guidance, mentorship and support throughout my doctoral studies.

I also owe special thanks to Dr. Daniel Vogel, Dr. Michael Terry and Dr. Edith Law for helpful advice and discussions during our HCI lab meetings. Among all Human Computer Interaction Lab members to whom I am thankful for creating a friendly and open-minded research environment, I would like to especially thank Adam Fourney for serving as an excellent human search engine that always knows the correct references to any question I may have, and Anastasia Kuzminykh for research collaboration and the drawings used both in my publications and in the dissertation.

Likewise, I would like to thank Dr. Ewa Gurbiel, who was my supervisor during my Master of Science in Computer Science studies and who introduced me to the wonderful multidisciplinary field of Human Computer Interaction, and Prof. Tereza Barta, who was my advisor during my Master of Fine Arts studies, and who opened my eyes to an artistic and intuitive – as opposed to rigorous and scientific – approach to design.

Finally, I would like to thank my dissertation committee, including Dr. Joseph J. LaViola, Dr. Ian McKillop and Dr. Neil Randall, for considering my candidacy for a doctoral degree. And I would like to thank the hundreds of volunteers who have participated in my pilots and research studies over the years.

Dedication

I dedicate the dissertation to my family, especially to:

my loving wife Mila for her encouragement, patience
and understanding of what it means for a husband and father
to pursue a doctoral degree in computer science;

my sons Krzysztof and Jan, for their unconditional love
and their sacrifice of playtime with Dad;

my Mom and Dad for instilling in me the belief in the importance of education
and continuous emotional support in my pursuit of undergraduate and graduate education.

and to my best friend, Dr. Ira Ashcroft – for support, encouragement
and for being a member of my family.

My research would not have been possible
without all your love, flexibility, and understanding.

Table of Contents

List of Figures

Chapter 1

Introduction

In 1991 Mark Weiser offered a vision of calm computing, where computing devices "weave themselves into the fabric of everyday life until they are indistinguishable from it" (Weiser, 1991, p.1). He predicted that in the future technology will be omnipresent, but non-invasive and will facilitate interactions that do not require learning or specialized input devices. In 2015 we are much closer to fulfilling that vision, largely thanks to the vast proliferation of mobile technologies and the pervasiveness of digital displays. A few years after their introduction to the consumer market, smartphone proliferation rose to 77% in North America (Dediu, 2015), and it is predicted that 84% of the world's population will be using some kind of mobile technology by 2018 (Radicati, 2014). Now we are on the verge of yet another revolution: smartwatches, which may proliferate to a similar extent and at a similar speed.

However, Weiser's vision is far from being fully realized, despite the fact that end-users frequently possess a device that is convenient and available for interaction – such as a personal smartphone or a smartwatch. Rather than forming a rich ecology, displays, mobile devices and wearables exist largely independent of each other. Some efforts have been made towards compatibility, but the everyday use of *interconnected* devices is still not fully exploited. There is insufficient support for interaction techniques that seamlessly utilize mobile and wearable devices for input, thus facilitating casual, intuitive, low effort interaction with any and all displays.

Researchers have recognized the benefits of using mobile and wearable devices as convenience input devices for computing environments (Ballagas et al., 2006). The advantages of mobile devices have been discussed previously by Ballagas et al., who note that:

> "Mobile phones' prevalence gives them great potential to be the default physical interface for ubiquitous computing applications... However, realizing this potential

> will require intuitive, efficient, and enjoyable interaction techniques for applications in the ubiquitous computing domain" (Ballagas et al., 2006, p.1)

Yet, while researchers have already exploited mobile devices (Boring et al., 2010, 2009; Katzakis, 2012; Medeiros et al., 2013) and wearables (De La Hamette et al., 2002; Kim et al., 2012; Haque et al., 2015; Houben et al., 2015) as input devices for 2D interaction in computing environments, mobile- and wearable-based interaction with 3D environments is an under-explored topic (Pietroszek et al., 2014). Because display technologies continue to evolve to the point where displays can be both non-stereoscopic and stereoscopic and 3D environments can be rendered realistically on almost any kind of display, the need for seamless 3D interactions is more critical than ever. From movie theatres and shopping malls to conference venues and research labs, 3D information can be deployed seamlessly, but cannot be seamlessly interacted with.

Among a myriad of 3D interaction that can be performed in 3D environments, the most elementary is the 3D target acquisition task. 3D target acquisition is a pre-requisite for further object manipulation and 3D pointing is one of the ways in which 3D target acquisition can be realized. In my dissertation, I focus on facilitating 3D target acquisition through distant pointing by using mobile and wearable devices for input.

1.1 Thesis Statement

Mobile and wearable devices are carried by everyone. At the same time, for many application domains the presentation of 3D content is already realized through off-the-shelf display technology, both stereoscopic and non-stereoscopic. Advances in 3D image rendering such as autostereoscopic, volumetric, fog and holographic displays imply that 3D environments will become even more commonly deployed in the future than they are today. Yet, while 3D environments can be and are deployed everywhere, users cannot point at them from a distance. Thus:

> *Mobile and wearable devices can serve the attendant, unfulfilled need to support pointing for 3D environments without a use of specialized input devices.*

While some progress has been made in the domain of pointing in 2D environments, the need for casual pointing techniques is apparent in the context of 3D environments. Pointing techniques are needed that will help the proliferation of interactive 3D environments within computing environments other than virtual reality laboratories and similar professional settings. My dissertation attempts to break the barrier to proliferation of 3D environments by leveraging current mobile and wearable devices as a platform of opportunity to interact with 3D environments rendered on ubiquitous displays.

1.2 Contributions

I address the lack of support for distant pointing in 3D environments with use of mobile and wearable devices by designing, implementing, and evaluating thee novel 3D pointing techniques that leverage a smartphone or a smartwatch. I show that mobile and wearable devices may be especially beneficial as pointing devices for casual use scenarios, where specialized 3D interaction hardware may be impractical, too expensive or unavailable. Casual interaction scenarios happen in settings such as home theatres, intelligent homes, in workplaces and classrooms, with the movie theatre screens, in shopping malls, at airports, or during conference presentations.

The contributions of my dissertation can be grouped into three categories:

1. Leveraging mobile and wearable devices for casual 3D pointing

While 3D environments are commonly displayed in desktop settings, there are virtually no examples of interactive 3D environments deployed within casual environments such as public and semi-public settings, with the exception of console gaming. At the same time, immersive 3D environments remain – in users' minds – associated with professional work settings and virtual reality laboratories. I hypothesize that an opportunity for 3D interactive engagements is being missed not because of economical factors, but due to the lack of interaction techniques are easy to use and that have no need for specialized equipment. As the first step toward easier casual interaction with 3D environments, I present three novel 3D pointing techniques that are implemented on an off-the-shelf mobile and wearable devices.

2. Enabling mobile and wearable pointing at a distance

Research on casual pointing often relates to interaction performed from close proximity to the display, such as touch interfaces. Work on display interaction seems to assume that the interaction *intensity* is a function of a distance from the display: the further from the display users are, the "less" interactivity is offered to them (Vogel and Balakrishnan, 2004). At the same time, technological limitations inherent to interaction at arm's length prevent large groups of users from simultaneously interacting with a single display. Despite these limitations of direct, arm's length pointing, mobile and wearable pointing at a distance is much less explored in the literature than multi-touch pointing. My dissertation shows to what degree 3D pointing at a distance can be realized with the support of a modern smartphone or smartwatch.

3. Identifying and addressing challenges of 3D pointing

Hardware developments in mobile and wearable technology open new opportunities in multi-modal interaction. For example, a touchscreen combined with motion sensors offers multi-modal – yet single-handed – input that can read touch events, recognize non-touch gestures and provide an additional screen. My dissertation shows how multimodality, when realized with mobile and wearable devices, helps to address challenges for 3D pointing: performance issues related to the limitations of the human motor system, selecting occluded objects and related problem of perception of depth on non-stereoscopic screens, and high fatigue that results from providing spatial input for a prolonged time.

I realize these contributions by designing and evaluating three novel 3D pointing techniques that are implemented on off-the-shelf mobile and wearable devices available today. The contributions provide evidence that pointing in 3D environments can be implemented by leveraging the pervasiveness of a device that users already carry with them: a smartphone or a smartwatch.

1.3 Definitions

Due to the interdisciplinary nature of human computer interaction, the terminology used in this dissertation comes from a number of fields, such as large display interaction, immersive environments, and 3D user interfaces. To clarify my terminology, I hereby provide an definition of essential terms.

Unless explicitly stated otherwise, I use the word "smartphone" and "mobile device" interchangeably, although mobile devices include larger form factor devices such as tablets and, more recently, phablets. Similarly, I use the word "wearable" as a synonym of smartwatch, although wearable devices come in various forms and include devices such as smart glasses, armbands, and smart rings.

The term "large display" evolved over time. A 17" computer monitor that was considered "large" twenty years ago is no longer is considered as such today. In the field of large display interaction, displays that are multi-view, and support more than one user simultaneously interacting with them, are usually designated as “large”, even if their diagonal size is only 50” or so, and that is how I use the term “large display”.

The term “fatigue” may be understood as a quantitative measure of consumed endurance (Hincapié-Ramos and Guo, 2014) of muscles, or a subjective “feeling” of fatigue. Because I am interested in the subjective perception of the comfort or discomfort felt by users during 3D pointing task, throughout this dissertation I use an established NASA Task Load Index (Hart and Staveland, 1988) measure of perceived workload, instead of measuring the muscle’s fatigue directly.

I use the term “3D pointing” as a shortcut for “pointing at an object in a 3D environment”. Sometimes I use the words “pointing” and “selection” interchangeably, as is common in the literature, keeping in mind that pointing is just one of the methods that object selection may be initiated with, while selection itself additionally requires a selection confirmation (Bowman Doug A. et al., 1999).

When referring to “3D environment” in my dissertation, I mean a computer-generated graphics environment that preserves information on the depth of the object, in addition to its x and y location and size. For example, an image that creates an illusion of depth by using perspective or stereoscopic rendering does not, in my definition, constitute a 3D environment, while a 3D model rendered on a non-stereoscopic screen does constitute a 3D environment.

1.4 Overview of Research

The remainder of this dissertation is organized as follows.

In Chapter 2, I review relevant literature on 3D environment technologies and 3D interaction techniques, with a focus on 3D pointing and selection techniques. I discuss in detail techniques for pointing at a distance, as opposed to at arm’s length, as well as pointing techniques that leverage a mobile or wearable device for input. Finally, I review research where mobile or wearable devices are leveraged for 3D pointing. I identify areas the literature does not address comprehensively, including use of Bring Your Own Device (BYOD) approach to 3D pointing and interaction, support for interaction on non-stereo 3D environments, and designing for low perceived workload by utilizing multimodality of mobile and wearable devices.

In Chapter 3, I list challenges that 3D pointing techniques must address: speed-accuracy tradeoff, problems related to occlusion and depth perception, and high perceived workload. First, I discuss how the tradeoff between selection accuracy and speed is modelled with Fitts’s Law, and how hand tremor (Myers et al., 2002) and the Heisenberg effect (Bowman Doug A. et al., 2001) affect selection time. Second, I discuss three problems related to the perception of 3D environments. I start with the problem of occlusion, when a 3D target is hidden behind (or within) another object. Related to occlusion is the problem of depth identification: a user's inability to determine the z-position of a 3D target. Then I discuss the target disambiguation problem that is common in dense environments. I also discuss how 3D pointing techniques often result in high perceived workload and the related gorilla-arm effect (Schultz, 1988). The list of challenges identified in Chapter 3 provides a framework against which all my novel 3D pointing techniques are evaluated. I describe an experimental design that I use through this dissertation.

In Chapter 4, I develop a 3D pointing technique called Smartcasting (Pietroszek et al., 2014). Smartcasting is a smartphone-based interaction technique based on Raycasting (Liang and Green, 1994) that allows for the manipulation of a 3D cursor with the use of two modalities: a wrist rotation and a touch input. Smartcasting comes in two versions: a basic design where selection of 3D objects does not require manipulation of depth position of a cursor, but also does not support dense, occluded 3D environments, and an advanced version that allows for fully addressable 3D pointing (accessing any point in a 3D control space) and provides an occlusion removal mechanism that enables selecting targets hidden behind other objects. I formally evaluate Smartcasting and verify that it performs on a par with a Raycasting implementation for specialized input devices, such as a WiiMote. I conclude that a smartphone can replace specialized input hardware for casual 3D interaction.

In Chapter 5, I extend my work to wearable devices, such as a smartwatch, by designing and evaluating a technique called Watchcasting, I validate the extent to which a wearable device can be used in place of a smartphone, or other specialized devices, such as Thalamic Myo (www.thalmic.com), in order to perform 3D pointing tasks. While Watchcasting borrows from Smartcasting, its design is tailored for wrist-worn devices. I show that Watchcasting performs on a par with both Myo-based and smartphone-based 3D pointing.

In Chapter 6, I describe the design of a novel 3D interaction technique called Tiltcasting. Tiltcasting is a smartphone-based 3D interaction technique that takes full advantage of a modern smartphone's modalities. Tiltcasting's design is an extension of my work on the Smartcasting, but it also incorporates elements of spatial correspondence targeting (Pietroszek and Lank, 2012), a smartphone-based 2D interaction paradigm. My goal in developing Tiltcasting was to design a 3D pointing technique that both outperforms specialized low cost input hardware and addresses the challenges of 3D interaction. Through a formal evaluation I find that Tiltcasting speeds up selection of 3D targets in dense occluded environments, improves the target disambiguation process, offers perspective and depth cues that eliminate depth confusion, reduces the Heisenberg effect and hand tremor inaccuracy and reduces perceived workload.

In Chapter 7, I reflect on the proposed interaction design and techniques, pointing at the limitations of human motor system and the constraints imposed by mobile and wearable devices. I also discuss approaches to design that may improve future interaction design for 3D environments leveraging mobile and wearable devices such as a smartphone or a smartwatch.

Finally, in Chapter 8, I conclude my dissertation by summarizing the work presented and discussing the future directions in which the work can be extended.

Chapter 2

Literature Review

To identify what is needed to advance Weiser's vision, in this Chapter I review previous research on pointing techniques that utilize mobile and wearable devices and identify gaps in the literature that I bridge in my dissertation. I start by providing a brief overview of 3D environments and display technologies that enable rendering 3D content. Next, I provide an in-depth review of the relevant literature on interaction technologies, and the division of interaction techniques into two categories: interaction at arm's length and interaction at a distance. I focus on "at a distance" pointing as these are most relevant to the 3D pointing techniques I propose in the later chapters of this dissertation. Because I have identified mobile and wearable devices as *the* convenience device for casual pointing, I review research on mobile and wearable device-based pointing techniques, noting limited research on 3D pointing techniques that leverage these types of devices. I summarize my related studies chapter by discussing how gaps in the previous work have motivated my dissertation.

2.1 3D Environments

3D environments are computer-generated graphic environments that preserve the information about the depth dimension of presented objects. Such defined 3D environments are common – to different degrees of pervasiveness – in all kinds of settings, from home desktop computers to public 3D cinema screens. Each setting poses different requirements and constraints on the pointing techniques that are commonly used. While in some settings the user may expect to be equipped with or have access to a specialized 3D interaction device, such a requirement is not always feasible. For example, in public settings users usually do not carry specialized 3D input devices, while installation of alternative 3D input hardware in such settings may be too expensive or may not work well due to occlusion of some users by other users.

Desktop interaction with 3D environments is quite common and varies from 3D modelling software, Computer-Aided Design (CAD) software, scientific visualization software to 3D entertainment such as games and movies. The technology facilitating the 3D interaction includes desktop 3D mice, 3D joysticks, and game controllers such as a WiiMote. Desktop 3D environments can be rendered on consumer displays that support active or passive 3D stereoscopy. Polarized TVs that create a 3D rendering effect through passive polarized glasses use similar technology to the one used in many 3D movie theatres. More recently, head-mounted displays like Oculus Rift are entering the consumer market, especially for gaming applications. Earlier, active shutter glasses (NVIDIA 3D Vision) and anaglyph 3D rendering were commonly used to create the illusion of depth on the 2D surface of a monitor or projection screen.

For specialized applications such as flight simulation, modelling, scientific visualization, exhibitions, or virtual reality research, 3D environments are often realized in immersive systems, such as CAVE, or spherical (360°), or semi-spherical projection systems. Such environments usually come with specialized input devices. Use of a mobile device for these environments was criticized by Medeiros et al. (2012), who argue that engineering applications require precision tests that cannot be performed easily for mobile devices. Yet, it is possible that some 3D applications that do not require high precision, such as architectural or interior design walk-throughs for customers, could benefit from the cost-to-quality ratio and ease of use that mobile devices afford when used as inputs for 3D environments.

The settings in which the 3D environments are deployed are important for my dissertation, because they constrain and guide the design of the 3D pointing techniques that I am proposing in the next chapters. Although smartphones and smartwatches as convenience devices may be useful in every kind of setting, I expect that they will be used primarily for casual interaction, and thus should support "walk-up-and-use" scenarios. Users who often interact with 3D environments (e.g. architects, graphic artists, scientists) may be expected to have a specialized 3D input device. However, even in these settings, smartphones and smartwatches may still be useful as input devices for economical reasons. Those users who are interested in exploring 3D environments, but want to avoid the additional expense of specialized 3D input devices, may be interested in using the device they already own as long as it performs on a par with a specialized device.

2.2 Rendering 3D content

Recent advances in display technologies have rendered all kinds of displays cost effective, accessible, and mass-deployable. Specialized displays, such as obstructive head-mounted displays (Bowman Doug A. et al., 2004): the Oculus Rift (www.oculus.com) and Sony's "Project Morpheus", or non-obstructive holographic (Grossman and Balakrishnan, 2004), volumetric (Grossman and Balakrishnan, 2006), fog (Diverdi et al., 2006), autostereoscopic displays (Lee et al., 2008) and CAVE systems (DeFanti et al., 2009) are also slowly making their way from research labs to the consumer market, mainly for immersive gaming, modelling and scientific applications. Smaller displays are present on most electronic devices and commonly have direct interaction, through touch input or peripheral devices such as a mouse, a keyboard or a trackpad. Large displays are also more common than ever, available in both vertical and horizontal (tabletop) deployments and appearing in private (e.g. home theatre), semi-public (e.g. workspace) and public (e.g. digital signage) settings.

Large displays have attracted the attention of multiple researchers, who have noted productivity gains (Czerwinski et al., 2003) and improved collaborative interactions around displays (Russell et al., 2002; Wallace et al., 2009). Czerwinski et al. (2003) list the cognitive benefits of larger size displays, noting that they improve information recognition and peripheral awareness. These properties make large displays well-suited for many applications, such as command and control (Dudfield et al., 2001), automotive design (Buxton et al., 2000), geospatial imagery (Sandstrom et al., 2003), scientific visualization (Sandstrom et al., 2003), tele-medicine (Garner et al., 1997), collaboration in tele-immersive environments (Maimone and Fuchs, 2011), education and training (Lanir et al., 2008), and virtual reality applications (DeFanti et al., 2009). Another common application of large displays is their deployment as digital signage. Digital signage deployed in shopping malls, amusement parks, airports, stadiums, hospitals, city halls, shop windows, workspaces and building walls often aims to provide personalized or shared user experiences. Many research questions arise in this context, with recent research paying particular attention to interactivity awareness, territoriality, proxemics, and interaction techniques for these displays.

Rendering 3D environments can be realized on all the above displays, even those that are not able to render depth dimension. However, recent advances in stereoscopic, autostereoscopic (Ueda et al., 2014; Nii, 2013; Liao et al., 2011; Lee et al., 2008; Travis, 1990) and fog display technologies (Diverdi et al., 2006) enable rendering of 3D environments that presents depth information, often on large format displays and in non-private settings. While the presentation of 3D content can be realized, it comes with certain limitations. For example, stereoscopic displays require active or passive glasses to be worn by the user, with the exception of head-mounted displays that can provide separate images for each eye. Stereoscopic rendering is also known to create visual discomfort in humans (Lambooij et al., 2007). Autostereoscopic displays do not require glasses, but provide a 3D effect only from a limited number of “sweet spots”, resulting in disappearance of the 3D effect if the user moves his head or walks by the display.

2.3 Interaction Techniques for 3D Environments

Interaction with displays can be more engaging for users than a passive presentation of information. However, while interaction with 2D environments is well understood in research and supported by many input technologies, interaction with 3D environments continues to be a challenge (Bowman Doug A. et al., 2004). In general, interaction with 3D environments can be realized in two ways: at arm’s length or at a distance.

2.3.1 Interaction at Arm’s Length

Arm’s length interaction is possible by making the display itself interactive, a metaphor commonly realized through a multi-touch interface (Lee et al., 1985; Azad et al., 2012). Techniques developed for multi-touch displays usually implement multi-touch gestures, either similar to the gestures used on mobile devices such as smartphones and tablets, or specific to the size of the display (Voelker et al., 2013). Some techniques combine mid-air gestures close to the display with multi-touch gestures on the display itself, e.g. MirrorTouch (Müller et al., 2014). In “walk-up-and-use” (Izadi et al., 2003) interaction scenarios, interaction must support first-time users who have no previous experience with similar systems. One of the most comprehensive studies of large display deployment that supported this assmption was CityWall (Peltonen et al., 2007), yet it was a 2D environment study. Jacucci et al. (2010) implemented a version of CityWall that featured 3D spherical widgets to manipulate 2D photos. However, the study focused on observed user behavior rather than the interaction technique itself.

With the noted exception of immersive environments that are outside of the scope of this work, the use of arm’s length interaction techniques to select and manipulate objects in 3D environments is less common than it is for 2D environments. One method of direct multitouch interaction with 3D objects is to point at the 2D coordinates of the 3D target projection on the viewport (the 2D plane of the monitor). The position of the viewport can be also manipulated by changing the virtual camera position. Alternatively, a separate viewport may be provided for each of the x, y, and z axes. A single view is sufficient to select a 3D object via its 2D projection as long as objects are sparsely distributed.

When stereoscopy is used, an essential problem with touch-based direct selection is the very presence of the depth dimension: the z-axis point of any object that is positioned in non-zero parallax space (e.g. inside the displayed scene) by definition cannot be touched. Direct touch interaction within a 3D environment becomes a form of interaction at a distance. Due to the close distance of the user to the display, the problem here is a mismatch between the eye line and the imaginary extension of the finger into the depth dimension of the display. One solution is to adjust the pointing direction based on the eye-line direction (Möllers et al., 2012). Another solution was presented by Valkov et al. (2011), who provided a set of techniques in which 3D objects are shifted onto the 2D surface when the user touches an object at depth. The disadvantage of this solution is that the 3D environment is temporarily modified, complicating tasks such as translation.

When 3D objects are rendered using stereoscopy, direct pointing is technically challenging because stereoscopic effects rely on eye convergence (Reichelt et al., 2010) to convey 3D information. Perceiving stereoscopy requires a user to stand within a certain range from the display when interacting, thus limiting the possibility of touch interaction with large displays that require standing at a distance that excludes possibility of direct interaction. Another issue, called stereo fusion, arises when using 2D cursors to select 3D content rendered with stereoscopy (Argelaguet and Andujar, 2009), resulting in a depth mismatch between the cursor and the target object, thus preventing the user from "fusing" both objects in order to complete the selection.

Other challenges of direct interaction include multi-touch technologies that usually support a limited number of simultaneous touch points, thus limiting the number of users who can interact concurrently with the same display. The maximum number of users simultaneously interacting with a display is also limited by the number that can physically fit in front of a vertical or around a horizontal (tabletop) display. Another problem is that multi-touch interaction with content distributed across the entire display may not be possible, as large public displays may stretch beyond arm's reach.

When the display is out of arm's reach (e.g. it is mounted too high), or is shared between large number of users, all of whom would not fit in front of the display, or in settings when the position of user is fixed at a certain distance from the screen (e.g. in movie theatre), interaction at arm's length must be replaced by interaction at a distance, discussed next.

2.3.2 Interaction at a Distance

The distinction of interaction at a distance vs. interaction at arm's length is somewhat blurred as there exists a mixed approach that combines touch input with mid-air gestures in front of the display (Vogel and Balakrishnan, 2005; Müller et al., 2014) or above the display (Bruder et al., 2013). Thus, for the purpose of this dissertation, I define distal pointing as direct pointing performed at such a distance from the display that direct interaction is out of arm's reach. Such defined "pointing at a distance" techniques typically make use of two metaphors: virtual hand and virtual pointing (Poupyrev et al., 1998).

2.3.2.1 Virtual Hand Metaphor

In the virtual hand metaphor, objects are acquired and manipulated in a way that closely resembles real-world touching and grabbing. That is, users make use of a virtual hand that they control in order to acquire and reposition objects in the 3D world. Grabbing is too difficult to be reliably implemented without specialized equipment such as Leap Motion (Sutton, 2013), Virtual User Concept (VICON) motion capture system or a virtual glove (Bowman Doug A. et al., 2001). Some research towards achieving this goal was presented by Kim et al. (2012) and Quian et al. (2014), who showed how robust hand tracking can be realized using depth cameras. When specialized equipment is used, virtual hand techniques are prone to many of the same disadvantages as interaction in the real world such as limited reach and the potential for high perceived workload (Liao et al., 2011). Variants, such as the Go-Go technique (Poupyrev et al., 1996), address reach limitations by extending a user's arm using a nonlinear transformation: The user's hand is represented as a virtual hand in the 3D environment, that is positioned on a ray extending from the torso and intersecting their physical hand. This approach allows for extended – although not unlimited – reach within the control space of the 3D environment.

In general, the metaphor of the virtual hand falls within the Natural User Interfaces paradigm, which focuses on designing gestural interaction for both 2D and 3D environments in such a way that they resemble the way that people interact with everyday objects. Norman criticized that approach, stating that natural interfaces are not at all "natural" (Norman, 2010). He argues that the interaction research should focus on designing for ease of use and learning, for low mental demand and low perceived workload interaction that results in *better than natural* interaction, instead of trying to imitate natural interactions, which are expressions of our physical and biological limitations rather than the best possible solutions. Although it may be argued that the metaphor of grabbing is easy to use due to its resemblance to the way that people interact with objects in the real world, it is known to cause fatigue (Schultz, 1988) due to prolonged use of the same group of muscles (Hincapié-Ramos and Guo, 2014).

2.3.2.2 Virtual Pointing Metaphor

The pointing metaphor closely resembles real-life pointing with a finger: the user points at the object and selects it with a hand gesture or button click. One of the earliest examples of such an interaction technique was called “Put-that-there” (Bolt, 1980). This interaction technique combines voice input with mid-air gestures, allowing objects to be moved on a large display deployed in a private environment, such as a media room, and shows how multimodality can enhance pointing.

A recent survey of 3D pointing techniques by Angelaguet and Andujar (2013) proposes to classify selection techniques into three categories: point cursor, Raycasting and curve. The most basic 3D pointing technique is to freely move a 3D point cursor (Argelaguet and Andujar, 2008) within a 3D environment. The technique resembles a mouse cursor, but requires manipulation of the cursor position along the z-axis. Moreover, the 3D cursor is usually represented not as an arrow, but as a sphere or a crosshair. In this technique, the input device’s position in motor space is directly translated into the cursor position in the control space of 3D environments. Thus, to reach the target, the user moves the input device. This simple solution comes with a number of problems related to accuracy and speed tradeoff and occlusion. A detailed discussion on these issues, including related literature, is presented in Chapter 3.

One of the first virtual pointing techniques designed is Raycasting (Liang and Green, 1994), where an object is selected when a user points at that object using an input device such as a tracker, a glove, or in a freehand manner. Raycasting is a technique similar to laser-pointing described in the 2D interaction section. The difference is that in Raycasting, the ray travels into the depth dimension of the 3D display.

Raycasting exists in many variations (Argelaguet and Andujar, 2013), developed to address various challenges in 3D interaction, that are discussed in Chapter 3. One of the common modifications is to adjust the shape of the selection cursor, or the ray. For example, in 3D BubbleCursor (Vanacken et al., 2007), the size of the spherical cursor expands or shrinks automatically to reach the object closest to its centre. Similarly, ApertureSelection (Forsberg et al., 1996) allows for the manual adjustment of the the selectable area’s apex angle. Some techniques combine a ray with a 3D point cursor that is contained along the ray. For example, in Depth Ray (Grossman and Balakrishnan, 2006). In Depth Ray, from all objects intersected by the ray the one that is closest to the 3d cursor that can move along the ray, is the one that is selected. Another option is to zoom in the area surrounding the target (Cashion et al., 2012).

A significant shortcoming of many Raycasting techniques is that their performance degrades in dense environments due to ambiguity in targeting. Specifically, a cast ray represents an infinite set of candidate points along a single line, and when targets are grouped closely together it may not be clear which candidate a user wishes to interact with. Moreover, the technique does not allow for the selection of distant/small objects due to the angular accuracy required. That is, small angular rotations (a tremor) of the user’s hand can result in large movements at a distance on screen (Myers et al., 2002; Steed, 2006). Despite these disadvantages, Raycasting is widely used for 3D selection (Steed, 2006), and is publicly known for its use in devices such as Nintendo's WiiMote. The popularity of these techniques can be attributed to their simplicity and intuitiveness, arising from a "natural" extension of the user's finger.

2.3.3 Virtual 3D Pointing

Most interaction tasks in 3D environments start with a selection of a target of interest, thus target selection is the fundamental task in 3D user interfaces (Bowman Doug A. et al., 2004). Bowman et al. (1999) note that a selection technique must facilitate the indication of an object, confirmation of its selection, and should provide feedback during the selection task (visual, haptic or audio). One way selection can be realized is through virtual pointing, other methods include grabbing, or manually entering the 3D coordinates of objects. However, unlike virtual hand and direct touch techniques, virtual pointing allows the user to select objects beyond their reach.

Consequently, multiple user studies have found that virtual pointing results in higher selection effectiveness (Zhai et al., 1997). For example, an evaluation by Bowman et al. (1999) compared Raycasting efficiency with that of the virtual hand technique over a wide range of object distances, sizes, and environment densities. They found Raycasting to perform better, because the target could be reach by the ray at infinite distance, as opposed to a limited reach of virtual hand techniques. From the ergonomic perspective the most common implementation of selection is done by simply changing the pitch and yaw of the wrist, thus requiring relatively less physical movement in comparison with direct touch input and virtual hand techniques, that both require large movements of the hand. For that reason, Bowman recommends to "use Raycasting techniques if speed of remote selection is a requirement" (Bowman Doug A., 2002).

Another distinction that differentiates 3D pointing is its realization as a direct pointing or indirect pointing. While indirect 2D pointing at a distance have received a lot of attention from the research community (Pietroszek and Lank, 2012; Nancel et al., 2013), indirect 3D pointing at a distance is rarely addressed outside of the context of immersive environments. Rare examples of an indirect 3D pointing for ubiquitous displays include CubTile (Hachet et al., 2013) and Toucheo (Hachet et al., 2011).

2.4 Input Technologies for 3D Pointing

Input for 3D interaction can be realized in a number of ways. For desktop 3D applications the most common way of interacting with 3D environments is a traditional, 2D mouse and keyboard. Professional users of 3D modelling or CAD software may use more specialized input devices such as a 3D mouse or 3D joystick. In immersive 3D environments one method is to use specialized handheld devices, that I discuss in detail below. Finally, other methods of input, such as image processing and gaze-based input were developed and evaluated and have gained increased popularity the recent years thanks to technological advances.

Regardless of the input technology used for interaction with 3D environments, it will usually support more degrees of freedom (DoFs) than input technologies for 2D environments. Although it is possible to use 2DoF input technology, such as a regular mouse, for interaction with 3D environments, the missing degree of freedom slows down interaction (Takemura and Tomono, 1988). Thus, to support fully addressable 3D pointing and translation, the input technology should allow for manipulation of at least 3 DoFs in order to provide values for any of the x, y, or z axes to define position within a Cartesian 3D coordinate system. Raycasting techniques utilize five DoFs: three to determine the ray's origin and two to determine its orientation. For further 3D object manipulation such as rotation, an additional 3 DoFs may be provided, thus most input devices support six DoFs (Zhai, 1995). However, for ease of use, it is recommended to minimize the number of DoFs used in the interaction technique and match the number of degrees of freedom with the technique's requirements (Bowman Doug A. et al., 2004).

2.4.1 Handheld Input Devices

As mentioned, 3D pointing can be realized with specialized handheld input devices, such as magnetic trackers (Zhai, 1998), gyroscopic mice (MacKenzie and Jusoh, 2001), Soap (Baudisch et al., 2006), or wearables such as interactive gloves (Bowman Doug A. et al., 2001). However, research on the performance of virtual pointing versus the most commonly used mouse pointing is inconclusive. The performance and usability results depend on both the input technologies used and the interaction technique. On one hand, when MacKenzie and Jusoh (MacKenzie and Jusoh, 2001) compared a regular mouse with two air mice – a GyroPoint, with gyroscope-based cursor movement, and RemotePoint with a joystick for moving a cursor – they reported that both GyroPoint and RemotePoint was slower than the regular mouse. Also, both input devices had higher error rates than a regular mouse. On the other hand Takemura and Tomono (1988) found the mouse to be slower and more error-prone than a magnetic tracker. Jota et al. (2009) compared the performance of grabbing, pointing and mouse cursor. Their study finds that pointing was the fastest and the least tiresome technique, although it was performed using a motion-tracking input with high precision.

Pointing may be also accomplished using technologies other than magnetic trackers, air mice or 3D joysticks. In a 2D interaction context researchers considered using a laser pointer as an input device (Eckert and Moore, 2000; Kirstein and Muller, 1998) and performing selection by a

1s dwell time. The Stanford iRoom took the idea further by using a laser pointer to draw gestures and interact with pie menus (Winograd and Guimbretière, 2003). Others tested a variety of laser pointers for menu selection, scrolling, and text entry (Olsen and Nielsen, 2001).

2.4.2 Input via Image Processing

Another option is to use image processing, which can be used in conjunction with passive objects such as wands or labels. A example of such a project is VisionWand (Cao and Balakrishnan, 2003), which is a passive wand marked at its ends with two colours and tracked by two cameras. The movement of the markers is reflected on the screen in real time. Combinations of the wand's rotation and its distance from the screen allow not only for pointing and selection, but also for rich interaction including pan-zoom gestures, displaying additional information about the objects, or performing pie menu selections.

Yet another idea is to process the perspective distortion of the user's shadow (Shoemaker et al., 2007). Another study where image processing was used for interaction with a large screen was presented by Malik et al. (2005), who designed a finger-movement-based interaction technique. In the public screen context, the limitation of this technique is that the users had to be seated in order to perform the interaction. Hamette et al. (2002) designed a system called Fingermouse, where input from wearable cameras is analyzed in order to determine a finger pointing direction.

High precision pointing can be achieved using a motion-capture system, such as Virtual User Concept (VICON) (Kopper et al., 2010). Vogel and Balakrishnan (2004) designed a number of high-accuracy, high performance, and high comfort freehand gestures using a VICON system. Although their study focuses on 2D pointing, the techniques presented could be extended for non-occluded 3D environments, because, as noted by Bowman et al. (2002), in non-occluded environments, 3D selection is essentially reduced to 2D selection. Lower precision interaction can be achieved using low cost equipment such as webcams or Kinect in place of VICON, and WiiMote game controller (Yang and Li, 2011) or Myo armband (Haque et al., 2015) in place of a hardware tracker.

2.4.3 Gaze-control Input

A large number of studies have examined 3D pointing through gaze-control interfaces, where the eye's focus is tracked in order to determine the pointing vector, including depth. In the context of 3D gaming, Castellina and Corno (2008) propose to control camera view and keyboard through a gaze-based input. Kwon et al. (2006) offer a depth estimation method for gaze-based pointing on stereoscopic displays, while Ki et al. (2008) provide a similar method and evaluate it for autostereoscopic displays. In general, gaze-based pointing techniques suffer from the Midas touch (Jacob, 1991), that is an unintentional selection of object resulting from leaving the cursor pointed at an object for the period of a dwell time without the intention of selecting it.

2.4.4 Limitations of 3D Input Technologies

As exemplified by the studies listed above, many current technologies for 3D pointing at a distance require specialized, often expensive input hardware. Moreover, using trackers, gloves, or WiiMotes prerequisites possession of such devices by the interacting users. In some contexts, such as a casual interaction, users cannot be expected to carry specialized 3D interaction hardware with them. On the other hand technologies such as motion-capture cameras are also prohibitively expensive for most applications and require labeling passive interaction objects or augmenting users with special markers. Technologies that utilize image processing and can support interaction without providing a device to a user do not provide enough precision and robustness for many applications. For example, Microsoft Kinect's input can be easily confused by dynamically changing lighting conditions, background or occlusion of camera view on the user.

In contexts where other input technologies are either too expensive or not readily available, a smartphone or a smartwatch emerges as *the* convenience device for "at a distance" interaction, including 3D pointing. This observation motivates the focus of my dissertation on smartphones and smartwatches as input devices for pointing in 3D environments. However, before discussing mobile-based interaction techniques, I present a review of previous work on using smartphones and smartwatches for 3D interaction with displays.

2.5 Interacting with a Smartphone

Historically, mobile devices were used for pointing not at virtual 3D environments, but as a remote controller for the objects in the physical world. Fitzmaurice (1993) was one of the first to propose mobile devices – a palmtop – for selection and retrieval of information from physical objects, such as active maps or computer-augmented libraries. One of the first high-fidelity implementations of this concept was presented by Rekimoto and Nagao in a system called NaviCam (Rekimoto, 1995). It combined a mobile device with a digital camera that could read labels attached to physical objects, similar to QR codes. Another project used mobile-based image/object recognition techniques to facilitate retrieval of additional information in a museum (Foeckler et al., 2005). Valkkynen and Tuomisto (2003) built a mobile pointing solution that could retrieve information from printed posters. They used light-sensor-triggered RFID tags that were illuminated using a built-in infrared beam (from a close distance) or using a laser beam attached to the mobile device.

Research at the ACM Conference on Human Factors in Computing Systems and related conferences has also explored the use of portable and mobile devices to enable interactions, often collaborative in nature, with nearby displays (Elwart-Keys et al., 1990; Stewart et al., 1999). For example, Myers et al.'s seminal Pebbles Project (Myers et al., 1998) explored the use of Personal Digital Assistants, precursors to today's smartphones, as input devices to a large, shared display. An example of a direct pointing for a tabletop display using a mobile device was investigated by Schmidt et al. (2010). They proposed a pick-and-drop style technique that allows for acquisition of targets by touching the display with a mobile phone. At the moment of contact the phone either acquires or drops the target. The technique was designed for 2D interaction, but could be extended for 3D interaction, if combined with the technique proposed by Valkov et al. (2011). While this work has established the feasibility of supporting interaction via a user's personal device, several challenges have also been observed, including logistics such as facilitating connectivity between devices (Hinckley et al., 2004; Lucero et al., 2012).

Yet another project, ARC-Pad (Mccallum and Irani, 2009), is an example of a system that implements an absolute + relative pointing on a modern smartphone's touchscreen. Absolute pointing means that the touchscreen is mapped 1-to-1 to the display, while relative pointing means that there is no relation between the position of the finger on a touchscreen and the position of the cursor, but the cursor moves in the direction of the finger movement on the touchscreen.

3D pointing usually requires at least 3 DoFs. However, by projecting a 3D environment onto a 2D display, and assuming that no target in such a view is fully occluded by other objects, 3D pointing can be realized with 2 DoFs (Takemura and Tomono, 1988; Bowman Doug A., 2002). Thus, an example of a pointing interaction technique provided by Boring et al. (2009) who used a mobile phone's sensors to move a mouse cursor on a large public display, is helpful in determining the usability of mobile devices for 3D pointing. Their technique shows that moving a cursor by reading accelerometer data is possible. Although their implementation results in a high selection error rate, this finding may have been a result of the hardware limitations of the smartphones used or lack of application of sensor fusion techniques. Selection techniques that used only the tilt (up and down movement) of a mobile device was presented by Rahman et al. (2009).

Another interesting form of a 3D selection, mediated through a mobile phone camera, is presented by Boring et al. (2010). In their Touch Projector project, users interact with the 3D environments continuously captured by a smartphone's built-in camera. When a display is within view of the smartphone's camera, any within-view object presented on this display can be acquired and dragged within that display or into another display. To achieve that, the position of the smartphone is continuously determined using optical flow analysis. This requirement can be problematic in low lighting conditions or where other users and passersby can occlude the camera view, thus changing the reference background.

Steinicke et al. (2008) discuss the possibilities that mobile devices offer for general interaction with 3D environments that are presented on stereoscopic screens. They suggest using touch input as a remote controller e.g. to adjust the parallax effect. Although the work does not discuss the pointing techniques in detail, it suggests that technique similar to Image Plane (Pierce et al., 1997) could be implemented using a mobile device's 2D touchscreen.

The use of mobile devices as input devices for virtual environments was also proposed by Medeiros et al. (2013), who combined gyroscope and accelerometer data and touch input from a tablet to facilitate the selection of objects in a CAVE environment. The technique leverages a tablet for control of a virtual camera represented in the 3D environment as a truncated, semi-transparent pyramid. The camera view defines an area of the control space that is selectable. The size of the pyramid is adjusted in a manner proposed in 3D Bubble Cursor technique (Vanacken et al., 2007), defining the group of objects that are selectable. Once the list of selectable objects (those within the cone of the camera view) is established, the camera view is projected onto the tablet's 2D surface, where they can be selected through touch input. Although the technique provides an interesting application of mobile device multimodality, the technique is not formally evaluated and it is not clear how it performs in comparison with other 3D selection methods.

In a casual setting, an example of a 3D interaction technique using a mobile device was presented by Katzakis et al. (2012). The technique translates the position of the phone in 3D space into the control space of the 3D environment. The cursor can be moved on the touchscreen, but its actual movement in 3D control space is also a function of the phone's position and rotation. Katzakis does not provide a formal evaluation of the technique and I expect that the technique will have inconsistent selection performance, because it allows for 3 DoF movement of the smartphone touchscreen thus making it difficult to perform the movement of the cursor at majority of smartphone's positions in the 3D space. Also, because the interaction plane travels in space with the smartphone movement, reaching some areas of the control space may result in awkward hand positions, causing high perceived workload and gorilla-arm effect.

While the above research provides evidence that mobile devices may be a feasible option for 3D interaction, it does not provide clarity towards whether 3D pointing can be efficiently realized using off-the-shelf smartphones. None of the techniques proposed in the literature were shown to provide similar performance and functionality to techniques realized with specialized 3D input hardware. One of the contributions of my dissertation is to bridge that gap in the literature.

2.6 Interacting with a Smartwatch

Smartwatches are emerging as a ubiquitous digital companion, complementary to other devices such as smartphones, tablets, traditional PCs, and embedded displays. As wearable technologies, a smartwatch is always present, contains a advanced sensors such as a gyroscope, magnetometer, and accelerometer, and provides access to high-power computational and graphics processors, and high-speed networking. Where early work leveraged wearable devices for personal health informatics, such as measurement of sleep patterns and heart rate (Bieber et al., 2013), a recent focus has been understanding how these capabilities can serve as a convenience device for more active tasks such as managing notifications (Schirra and Bentley, 2015), supporting navigation (Lim et al., 2015; Kerber et al., 2014), or as a platform for assistive technologies (Kearns et al., 2013; Porzi et al., 2013; Twyman et al., 2015).

When interacting with a smartwatch, users are often restricted to interacting with a smartwatch's powerful CPU and sensors through a small touchscreen. The physical size of human fingers and wrists are often obstacles to interaction on such small devices, a design constraint known as the 'fat finger problem' (Albinsson and Zhai, 2003; Siek et al., 2005). Research has sought to overcome the fat finger problem to enable interaction with a watch. For example, projects such as TouchSense (Huang et al., 2014), SplitBoard (Hong et al., 2015) and Beats (Oakley et al., 2015) have explored alternative models for touch interaction on a small watch display. Others have sought to move interactions away from the display entirely. SkinWatch enables input to on-watch applications through manipulations of the watch's body to enable rotation and zooming (Ogata and Imai, 2015). Funk et al. (2014), Knibbe et al. (2014) and Burstyn et al. (2015) explore the use of specialized watch-bands for input. While this work has enabled more powerful input to a smartwatch, it does so at the cost of learnability or the need for additional hardware, and constrains interaction to the smartwatch itself.

Smartwatches were also explored for a use as a digital companion. Notably, Porzi et al. (2013) explore the use of smartwatch, in tandem with a worn smartphone's camera, to assist visually impaired users to avoid obstacles while walking. After developing an initial prototype that supports interaction via a limited set of two gestures, follow-up work improved recognition of up to 19 gestures (Costante et al., 2014).

Previous work has also explored use of smartphone's sensors for mid-air gesture recognition. Protractor (Li, 2010) provides fast and reliable gesture recognizer for a set of gestures that can be both orientation-invariant or -sensitive. Gesture Watch (Kim et al., 2007) presents a custom-made watch that recognizes gestures, thus allowing for control of other devices in computing environments. More recently, Kim et al. (2012) presented a custom-made wrist-mounted device that is equipped with depth camera and allows for more fine-grained gesture recognition. Xu et al. (2015) exploit in-watch sensors to recognize 37 finger and whole arm gestures, and the writing of english characters on a nearby surface. In fine-tuning their recognizers, Xu et al. demonstrate recognition rates of 98% and 95%, respectively, for these tasks, and establish the feasibility of smartwatch-based gestural input in the general case. However, their work focuses solely on the development of these specialized recognizers, and does not investigate the appropriateness of gestural interaction in real-world settings.

Specialized wearable devices similar to smartwatches have been previously leveraged for pointing (De La Hamette et al., 2002; Kim et al., 2012; Haque et al., 2015). The most related work on 2D pointing with a smartwatch is WatchConnect (Houben et al., 2015). WatchConnect provides a framework for smartwatch-centric cross-device applications. The research mentions a possibility of using a smartwatch's touchscreen as an input sensor for another screen. Finally, in the project called "Duet", Chen et al. (2014) explore using a smartwatch to enrich the interactions performed on a smartphone. For example, by using the smartphone's screen in conjunction with a smartwatch, the user is able to perform gestures, such as a "knuckle" or touching the screen with a side of the finger, that otherwise could not be recognized by the smartphone's touchscreen. The literature review did not revealed any examples of previous work where a smartwatch is used an input device for interaction with 3D environments.

2.7 Where Current Research Falls Short

The literature review indicates the pervasiveness of displays in computing environments. Among their various types are specialized displays that are often used for 3D environments. Yet, as discussed, 3D environments can be also presented on non-stereo displays that are most common in ubiquitous computing environments. I am focusing on the latter type of displays in my dissertation, when designing and evaluating smartphone-based and smartwatch-based 3D pointing techniques.

Use of displays depends on the settings of their deployment. While professional users of desktop 3D environments may be expected to have a specialized input device for 3D interaction, because they use 3D environments on a frequent basis, users who interact with 3D environments casually can benefit from using a mobile device they already own. In public settings the benefits of mobile-based interaction techniques are even more apparent. Passersby cannot be expected to carry specialized input devices with them, while related work on freehand techniques that do not require such devices show that these input methods are either too expensive to deploy, or do not work well in some environments. Thus, users may greatly benefit from the BYOD (Bring Your Own Device) approach to 3D pointing and interaction.

The process of designing a 3D pointing technique does not start in a void and many previous techniques were designed and evaluated. Among the types of interaction techniques I discussed are “arm’s length” and “at a distance” techniques. While direct techniques that utilize touch input directly on the display have many advantages, in the context of 3D environments they pose a number of problems. For example, they are difficult to use with stereoscopic displays, and even when no stereoscopy is needed, direct techniques pose problems such as the paradox of reaching into the third dimension of the display (the depth), while using 2 DoF touch input. Consequently, I argued that using interaction at a distance is a more natural choice for 3D pointing and manipulation tasks than direct interaction through touch input. Many options for interaction at a distance can be categorized into two metaphors: virtual hand and virtual pointing. The literature review provides evidence that virtual pointing is considered – at least for non-immersive 3D interaction – a preferable option, especially in terms of user fatigue. This result further motivates my choice of using mobile devices for virtual pointing.

The literature review reveals many input technologies used for 3D interaction from those that use input devices to those that utilize image processing or gaze-control for input. Yet in many settings, users cannot be expected to carry or have access to specialized input devices, and in the same settings the other input techniques may be prohibitively expensive or unreliable. The lack of easy-to-use, reliable and economically viable input technology for 3D interaction not only points at the limitations of the current 3D input technologies, but also further motivates my choice of using a smartphone and a smartwatch as a convenience devices for 3D environments.

As I am not the first to notice the potential of mobile and wearable devices for interaction with displays, I have reviewed previous work on mobile- and wearable-device-based interactions, showing how mobile and wearable devices were already recognized by researchers to be a powerful and useful input device. I have discussed previous work that used mobile and wearable device input for both direct as well as indirect pointing. I have noted previous work on 3D interaction using mobile and wearable devices, stressing that the topic is under-explored and that to my knowledge no previous work focused on designing 3D pointing techniques that are easy to use, work in a casual setting and result in low perceived workload. My dissertation bridges that gap in the literature. I show through the design and evaluation of three novel interaction techniques that in ubiquitous computing environments users can benefit from using a smartphone or a smartwatch as an input device. Yet, before going into details on the design of these techniques, I need first to discuss the challenges that 3D environments pose for 3D pointing.

Chapter 3

3D Pointing Challenges

Universal smartphone or smartwatch-based 3D pointing techniques must address a number of known 3D interaction challenges, both technical and ergonomic. If not carefully addressed, these challenges will limit the use of pointing techniques to specific 3D environments (e.g. Raycasting for sparse 3D environments) or specific conditions (e.g. short-term use due to high perceived workload). Thus, before embarking on the design of novel mobile device-based 3D pointing techniques, I first discuss challenges for 3D interaction. These challenges guide the design of interaction techniques in Chapters 4-6 and provide evaluation criteria against which each they are validated.

Hinckley et al. (1994) present a survey of issues that occur in the context of 3D pointing, including high received workload, the need of recalibration, clutching, as well as motion and orientation. Problems related to the perception of a 3D environment include difficulties with discovering and selecting occluded objects (Elmqvist and Tsigas, 2006, 2008; Elmqvist, 2005; Zhai et al., 1996a), distinguishing between nearby objects in dense environments (Grossman and Balakrishnan, 2006, 2005; Steed, 2006; Steed and Parker, 2004; Wyss et al., 2006), and accurately perceiving an object's depth (Cipiloglu et al., 2010; Cook et al., 2008; Kytö et al., 2013; Lee et al., 2008; Rogers and Graham, 1979).

As discussed in Chapter 2, the literature offers techniques for 3D pointing that address some 3D challenges using specialized hardware or input devices. However, many of the proposed solutions work only in certain settings (e.g. private or semi-public), are prohibitively expensive and do not support casual, casual interaction with 3D environments. One of the main contributions of my dissertation is to bridge the gap between the increased pervasiveness of displays, the technological advances of mobile devices, and the lack of interaction techniques that leverage these technological advances in support of 3D pointing.

To provide a frame of reference against which interaction design can be validated in this context, in this Chapter I identify a number of 3D pointing challenges. The first one, speed-accuracy trade-off, is a consequence of Fitts's Law (MacKenzie, 1992) which states that the index of difficulty depends on the logarithmic ratio between the distance to the target and the target width. Speed of selection is also affected by the natural hand tremor that influences the accuracy with which a target can be selected. A related problem, known as the Heisenberg effect (Bowman Doug A. et al., 2001), occurs at the moment of selection, when pressing a button may cause small movements, resulting in a missed selection.

The second group of challenges relates to perceptual issues in presenting 3D environments on a flat display. The first challenge in this group is occlusion: if the target is hidden behind another object, it has a strong negative influence on the performance of most 3D pointing techniques. A problem related to target disambiguation and common in dense environments is the target disambiguation problem that occurs in dense 3D environments. When objects are close to each other, the performance of many 3D interaction techniques slows down, and additional means need to be employed to distinguish between the target and its neighbours. The last perceptual problem is the problem of depth identification. Even in stereoscopic rendering, when the depth information is displayed, some users have problems with identifying the depth position of an object. In the case of 2D displays, the problem is more severe, including the effect of so-called depth illusion in which the user mis-guesses the position of the objects due to perceptual assumptions.

Finally, high perceived workload also affects the proliferation of 3D environments (Hinckley et al., 1994; Bowman Doug A. et al., 2004). While 2D pointing with a mouse or a trackpad can be performed for hours without significant user fatigue, pointing techniques for 3D environments result in fatigue and gorilla-arm effect (Hincapié-Ramos and Guo, 2014).

While some of these problems have been addressed individually in the literature (see below), I am not aware of a 3D pointing technique that performs well against all of the above challenges. Thus, the quest for such a technique remains open. When proposing novel 3D pointing techniques in Chapters 4, 5 and 6, I report on the extent that each of my techniques addresses the following 3D interaction challenges.

3.1 Performance

Users of interactive systems are bounded by the limitations of the human motor system. Although individual users differ in their ability to perform 3D pointing tasks, every user is subject to a natural trade-off between the speed and accuracy of a selection. Performance is also tightly coupled with the muscle groups involved (Poupyrev et al., 1997; Casiez et al., 2008), where smaller muscle groups (fingers, wrist) achieve higher motor precision than bigger ones (arms, shoulders) (Zhai et al., 1996b). Thus, input devices relying on smaller muscle groups should be employed for pointing tasks that require high accuracy (Argelaguet and Andujar, 2013).

One of the most successful in its predictive power human motor models is proposed by Fitts (1964). Fitts's Law accurately models the time required to perform aiming movements, such as the movements that the user needs to perform in order to select a target directly, or using an input device. In human computer interaction, the formulation of Fitts's Law that is most commonly used was proposed by MacKenzie (1992).

Predicting pointing performance is more difficult for the family of Raycasting techniques that do not directly translate the position of the input device from the motor space to the 3D environment's control space. Instead, while the positioning of the ray's origin can be modelled with Fitts's Law, the actual selection movement is usually realized as a two DoF angular movement of the wrist. The distance of the cursor movement on the screen depends on the distance of the user from the screen and the angular distance that the input device has travelled. MacKenzie's formulation of Fitts's Law does not work well for this situation. The problem was better described by Kopper et al. (2010), who developed an angular version of Fitts's Law (MacKenzie, 1992), showing that the speed of the pointing task depends both on the distance of the user from the screen, the angular size of the target and the angular amplitude of movement, both measured from the user's position.

3.1.1 Target-aware vs. Target-agnostic 3D Environments

The predictive power of Fitts's Law, in its original as well as its angular formulation, allows for the development of guidelines that improve the performance of 3D pointing tasks (Argelaguet and Andujar, 2013). However, the application of those guidelines depends on the type of 3D environments, whether it is target-aware or target-agnostic. Target-agnostic 3D environments allow for a fully addressable control space. In target-agnostic environments the input signal does not have information about the structure of 3D environment at the time of selection (e.g. MRI model of a brain tissue). In such 3D environments, the user interface designer must take measures to ensure the user's ability to select any 3D point in the control space. If increased precision is needed, zooming, repositioning the camera, or mapping a fragment of the 3D environment onto a 2D plane can be used, thus reducing the problem of 3D pointing to that of 2D pointing. Smartcasting, Watchcasting and Tiltcasting all allow for pointing in target-agnostic 3D environments.

Smartcasting, Watchcasting and Tiltcasting can be also adapted for interaction in target-aware 3D environments, that is 3D environments where the information about the position and shape of the target is available and can aid the pointing task. Such optimization may be beneficial, given that in target-aware environments the targets may be large and, as per Fitts's law, the size of the target influences selection time.

Optimizations of selection time are usually realized by increasing the size of the target. Bowman recommends temporarily increasing the target size itself during the selection process (Bowman Doug A., 2002). Another option is to zoom in the area surrounding the target (Cashion et al., 2012). Other 3D pointing techniques, instead of rescaling the target itself, increase the *effective* size (Forsberg et al., 1996; Pierce et al., 1997) of the target (that is the size of target's selection zone). For example, the 3D Bubble Cursor (Vanacken et al., 2007) divides the target-aware environment into a set of Voronoi regions, which equally distribute the space between targets. The Bubble Cursor changes its selection volume such that the object closest to the cursor's centre can always be selected, even if the centre of the cursor is far away from it. In sparse 3D environments the size of the Bubble Cursor – and consequently the effective width of the target's selection area – is much larger than the target width, thus significantly decreasing the selection time as per Fitts's Law.

3.1.2 Hand Tremor

Another aspect of human motor behavior that affects the speed and accuracy of selection is hand tremor: the unintentional, rhythmic muscle movement (oscillations) involving a hand, wrist or fingers (Elble and Randall, 1978). Myers et al. (2002), who tested a laser-based pointing metaphor on small and distant targets reported that hand jitter negatively affects target selection. Solutions to this problem, developed in 2D pointing, but which are applicable in 3D pointing include a dynamic recursive low pass filter by Vogel and Balakrishnan (2005). Other techniques include the application of a two-stage mean filter based on angular velocity (Wilson and Pham, 2003) or Kalman filters (Oh and Stuerzlinger, 2002).

Another approach called ARM/ZELDA (Kopper et al., 2008) enables users to magnify an area of interest, however entering this mode slows down selection. Yet, in the context of 2D pointing, a similar magnifying technique proposed by Nancel et al. (2013) was shown to perform on a par with or faster than alternative pointing techniques, while enabling selection of targets as small as 4mm on a 5.5m wide display. Importantly, Nancel et al.'s solution uses a tablet as an input device and is realized as an indirect pointing technique.

3.1.3 The Heisenberg Effect

Another issue, the so-called Heisenberg effect, that affects pointing performance, is the sudden movement of the cursor to the area outside the target that happens at the moment of selection.

The term "Heisenberg effect" (Bowman Doug A. et al., 2001) draws upon the eponymous uncertainty principle in physics, where the mere act of measuring a phenomenon interferes with one's ability to accurately observe it. For virtual pointing, the Heisenberg effect occurs when the user's physical actions, such as pressing a button on a pointing device, interfere with their ability to accurately select onscreen targets through the introduction of angular jitter. For small targets, particularly when the user is located far from the display, a cast ray may momentarily move outside of the target, resulting in a missed selection.

One solution to reduce the Heisenberg effect that can be applied in target-aware 3D environments, is to interpret the recent history of a cursor's movement (Bowman Doug A. et al., 2001). If the selection button was pressed but no object was selected the algorithm checks whether the cursor intersected any target recently, and if so, applies a correction. However, it is important to note that in dense environments this approach may lead to false positives and does not eliminate incorrect selections resulting from the cursor moving onto a nearby target. Moreover, this solution does not work for target-agnostic environments, when there is no information about the possible targets that the user may have tried to select. A solution that addresses, to a certain degree, the Heisenberg effect and that works for target-agnostic environments is proposed in the design of Smartcasting in Chapter 4.

3.2 Occlusion and Depth

Most displays, even stereoscopic ones (with exception of volumetric and holographic displays), are only capable of displaying a 2D image. Stereoscopy is an illusion and as such, it does not support a realistic reach into the displayed 3D environment. In the context of immersive environments some advances have been made to create a better illusion of the third dimension. For example, objects presented in the positive parallax are perceived as if they could be touched directly. Yet, the level of detail that immersive environments are able to present does not match the real world and thus does not create a feeling of presence that matches reality. This disparity between expectation and the physical fact that 3D environments are not *really* three-dimentional results in a number of challenges for interaction with these 3D objects in 3D environments.

The most important perceptual challenge that results from technological limitation of non-volumetric displays is the Problem of Occlusion, where users are unable to interact with objects that are behind other objects (Elmqvist and Tsigas, 2006, 2008; Elmqvist, 2005; Zhai et al., 1996a). Related to occlusion is the target disambiguation problem, where a user is unable to distinguish nearby objects in dense environments (Grossman and Balakrishnan, 2006, 2005; Steed, 2006; Steed and Parker, 2004; Wyss et al., 2006) and the depth identification problem, where a user is unable to accurately perceive an object's depth (Cipiloglu et al., 2010; Cook et al., 2008; Kytö et al., 2013; Lee et al., 2008; Rogers and Graham, 1979).

Below I discuss each of those problems in detail, briefly indicating how smartphone- or a smartwatch-based techniques can address them.

3.2.1 Occlusion Management

In 3D environments, the occluded target problem arises when, from the perspective of the user, a target is obstructed by another object or objects, thus inhibiting a user's ability to select such objects (Figure 3.1). Elmqvist and Taigas (2008) identify four object interactions that may cause occlusion: proximity, intersection, enclosure, and containment. They also present a comprehensive list of fifty occlusion management mechanisms (Elmqvist and Tsigas, 2008), that is a set of strategies that allow users to reach occluded targets. Smartcasting and Watchcasting use virtual X-ray, while Tiltcasting uses object removal as the occlusion management mechanism.

Most 3D pointing techniques require the object to be visible in order for it to be selectable. In cases where a target is partially occluded, most techniques will allow for selection, but will suffer from speed-accuracy tradeoffs resulting from the reduced target width in accordance with Fitts's Law (MacKenzie, 1992; Vanacken et al., 2009). Also, for ray-based 3D pointing techniques, occlusion is closely related to target disambiguation, because a ray cast into a 3D control space may intersect more than one target while the first intersected target may be occluding the other intersected candidate targets.

A common solution to the occlusion problem is to modify the scene by hiding or removing the occluding objects, or by repositioning the viewport so that a target becomes visible (Argelaguet and Andujar, 2013). If occluder objects need to be first selected and then removed one by one, this approach may significantly slows down the selection process, especially in dense 3D environments. To address this issue researchers have explored other occlusion management mechanisms. One idea is to interactively distort the space (Cipiloglu et al., 2010; Elmqvist, 2005) or the viewing projection (Elmqvist and Tsigas, 2006), in order to unhide the candidate target. Another option is to prompt the user about the presence of the target with haptic, audio or visual feedback (Vanacken et al., 2009).

A number of explicit occlusion removal mechanisms have been developed. A group of techniques were developed around the idea of a depth marker, or variable length ray (Dang et al., 2003). In the most basic case, the user controls the end of the ray point and can select the target that is closest to the endpoint (this technique uses a tracked wand as an input device). The previously discussed Depth Ray technique also introduces a depth marker that travels between the

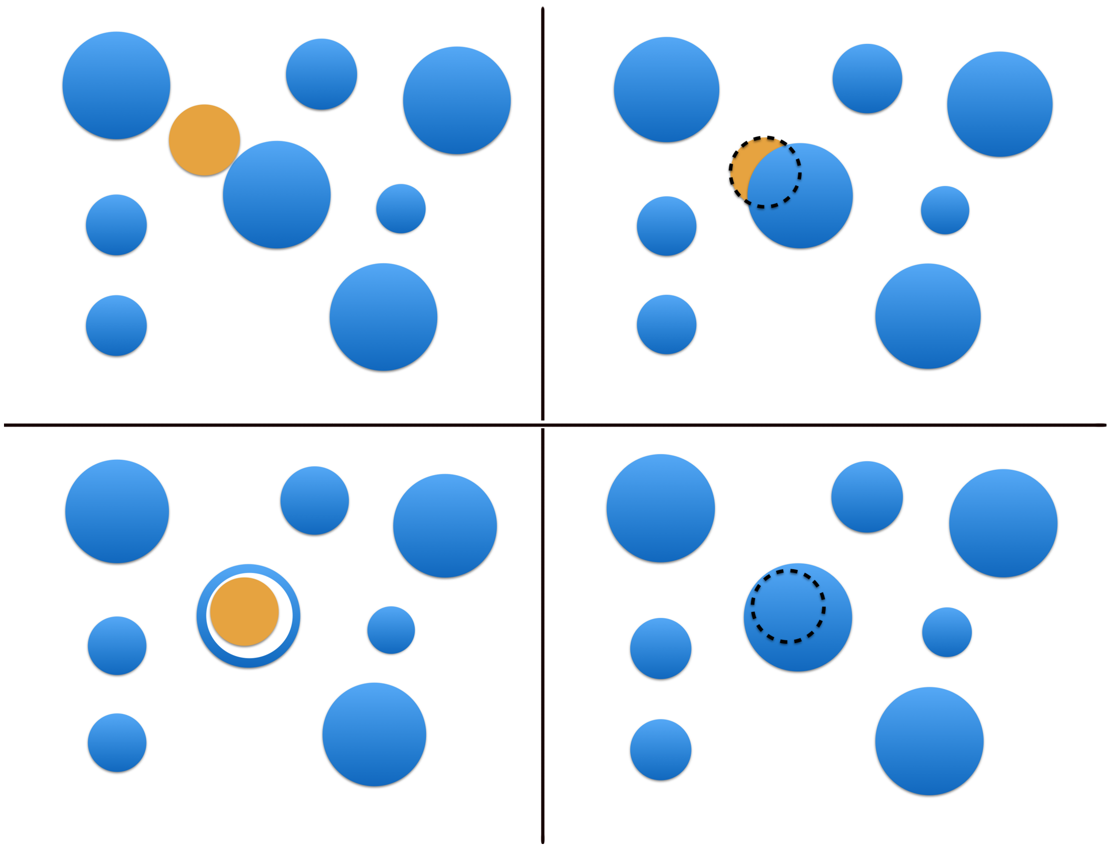

Figure 3.1: Four Types of Occlusion. Elmqvist and Taigas (2008) identify four possible object interactions that may cause occlusion: proximity – target is close to the occluder, so that from a certain perspective it is behind the occluder, intersection – target intersects the occluder, enclosure - target is surrounded by an occluder, containment -target is within the occluder

objects when the user moves the input device toward or away from the screen. Objects between the ray's origin and the depth marker are rendered semitransparent, so the occluded object closest to the depth marker can be seen. The problem with this solution is that it limits the movement along the z-axis to the user's reach, a limitation that is common for virtual hand techniques, but not for Raycasting techniques. More importantly, the movement changes the origin of the ray, thus requiring further adjustments of the ray angle. The second problem is solved in a modified version of Depth Ray called Lock Ray (Grossman and Balakrishnan, 2006), in which the ray origin is locked before the depth marker can be moved along the ray, thus reducing confusion between pointing and disambiguation phases.

Some methods used for target disambiguation may also serve as occlusion management mechanisms. For example, menu selection techniques that provide candidate targets may include those that are hidden from the user's view. Examples of such techniques include Flower Ray (Grossman and Balakrishnan, 2006), Daisy (Liang and Green, 1993), a circular menu technique called Ring (Liang and Green, 1993), or a list menu technique called Floating menu (Ramos et al., 2006). Similar to the disambiguation problem, this form of occlusion management does not work well if the number of candidate objects to choose from is large.

Sometimes occlusion management mechanisms are built into the pointing techniques, In the iSith technique (Wyss et al., 2006) the user controls two rays. Objects closest to the intersection of two rays are selectable. This arrangement allows the user to reach targets that are behind other objects. Yet another technique, called Flexible Pointing (Olwal et al., 2003), enables users to bend the ray in order to reach partially occluded objects without intersecting the occluders.

While it slows down object selection and requires specific management mechanisms, occlusion may also have a positive side effect on the 3D environment: it helps with depth identification of the target position. An object that partially occludes the target, or that is partially occluded by the target provides a depth cue about the position of the target on the z-axis. Depth identification is another important challenge of 3D pointing, specific to the presentation of 3D environments on 2D displays.

3.2.2 The Target Disambiguation Problem

Related to occlusion is the target disambiguation problem that occurs when interacting with a target that has a dense neighborhood (Figure 3.2). In ray-based techniques, the problem is further complicated by the fact that more than one target may be crossed by the same ray. Basic Raycasting techniques usually allow the user to point only at the first intersected target (Balaa et al., 2014). However, often the desired target may lay behind other targets – in such cases the target disambiguation problem becomes the problem of occlusion.

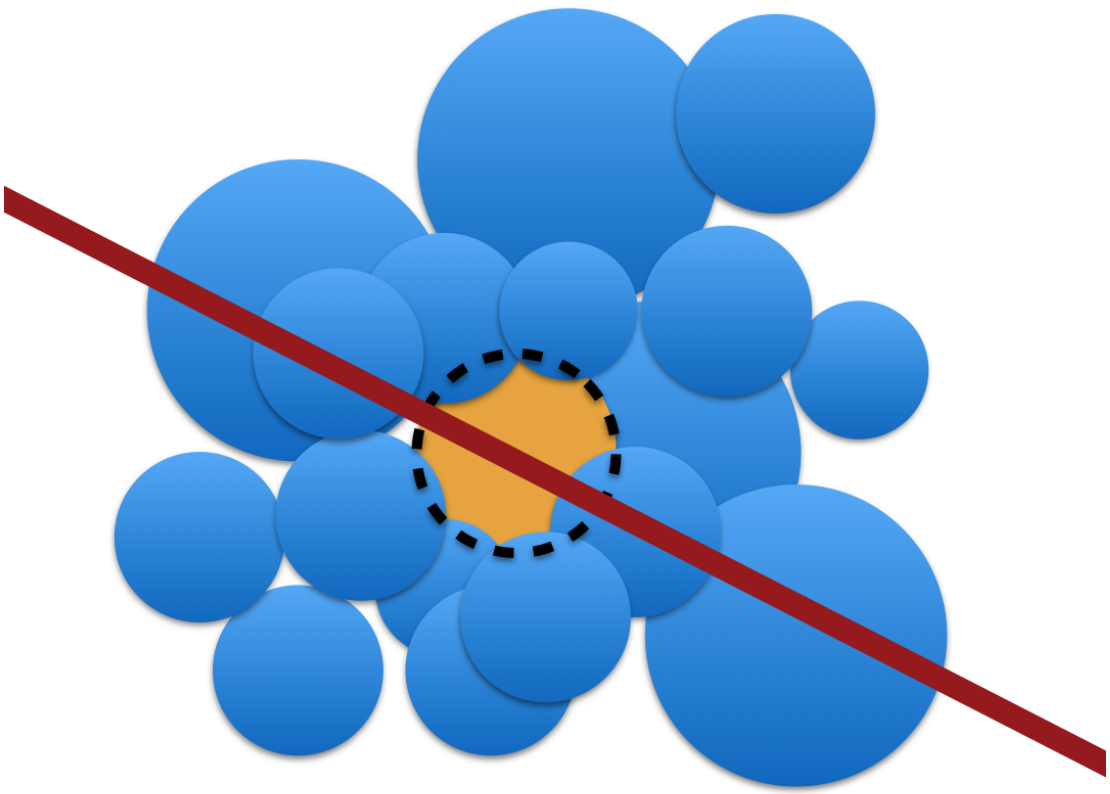

Figure 3.2: Targets that are close to each other and are crossed by a single ray result in a disambiguation problem, requiring a mechanism allowing the user to choose the desired target.

In response to target ambiguity, researchers have developed a number of disambiguation mechanisms. Argelaguet and Andujar (2013) classify the disambiguation techniques into three groups: heuristic, behavioral and manual. Kopper et al. (2011) note the tradeoff between the performance of the technique and the introduction of the target disambiguation mechanism. Target disambiguation mechanisms slow down the selection of easy targets, but speed up the selection of targets that are otherwise difficult to select. Yet, for many 3D application domains target disambiguation is necessary, thus I discuss some of the proposed mechanisms below.

One way to perform disambiguation is to attempt to interpret the user's target. In Flash Light (Liang and Green, 1994) the object closest to the axis of the flashlight cone is selected. This approach was improved by Schmidt et al. (2006) by introducing a probabilistic selection model. Finally, the behavioral approach takes into account cursor movements prior to the pointion and continuously ranks all objects in the vicinity of the cursor as probable targets. The data considered includes the volume of the target and its distance from the cursor or ray. For moving targets, the potential for intersection of the cursor's movement vector and the target's movement vector or the probability that the cursor is following the target increases the rank of the target. Examples of such techniques include IntenSelect (Haan et al., 2005) and SenseShapes (Olwal et al., 2003).

Manual techniques require that the user decides on the target. At the expense of cognitive load and performance, manual techniques provide the greatest expressiveness and maximum flexibility, for example, by cycling through all indicated targets with a press of a button (Hinckley et al., 1994), but such methods only work if the number of indicated objects is relatively small. Alternatively, Grossman and Balakrishnan (2006), propose a FlowRay technique that presents all indicated objects in the form of a pie menu, thus creating additional, but relatively fast selection tasks. Kopper et al. (2011) proposed a progressive refinement approach called SQUAD, that divides the candidate targets into four groups and then asks the user to select a subgroup repeatedly, until each group has a maximum of one target candidate. This approach works for very dense environments as it exponentially reduces the number of steps required to select a target. In reference to the Raycasting technique, a number of manual disambiguation variants have been proposed. Expand, a technique that improves on SQUAD by keeping the contextual information intact, was proposed by Cashion et al. (2012), who show that it performs on par with SQUAD. Grossman and Balakrishnan (2006) and Wyss et al. (2006) explore techniques that augment traditional Raycasting with a depth component. In their implementation the user can control not only the position of the cursor along the x- and y-axes, but can also manipulate the position of the cursor on the z-axis by moving the 3D cursor (the depth mark) along the ray so that any target intersected by the ray can be selected.

One observation is that with increased dimensionality of the environments, the complexity of target disambiguation increases. Consequently, methods used in dense 2D environments, such as BubbleCursor (Grossman and Balakrishnan, 2004), do not generalize well, without modifications for 3D environments when objects are also densely distributed along the z-axis. One method that could reduce the number of candidate targets is therefore to reduce the dimensionality of the environment from 3D to 2D for the purpose of selection. Unlike many of the disambiguation mechanisms discussed above, dimensionality reduction, if integrated into the pointing mechanism, does not add another phase to the selection time. In Chapter 6, I introduce Tiltcasting that applies dimensionality reduction to a target disambiguation mechanism.

3.2.3 The Depth Identification Problem

Humans rely on various types of sensory input to identify the depth position of an object: stereoscopy, occlusion, mental models, and motion parallax all contribute to the way that we perceive our surroundings. One significant factor is stereoscopic vision, a quality that is lacking in traditional computer displays. For these displays, identifying the depth position of a target is often a challenging problem without additional contextual cues, and humans may misjudge the relative depth of two on-screen objects (Figure 3.3). Even when stereoscopic rendering is used, e.g. in immersive systems, most systems do not provide as many details and cues as those present in the real world, resulting in much slower selection of virtual objects than real-world physical objects (Plumert et al., 2005)

With simple Raycasting techniques that do not require movements of the cursor along the ray, the problem of depth identification is avoided by the fact that the ray intersects any object in its way, regardless of its depth position. This property of simple Raycasting implicitly removes the depth identification problem, reducing the selection into a 2D selection problem. However, as discussed in the previous sections, simple Raycasting is unable to select occluded targets, thus excluding its applicability in many 3D application domains.

The user's ability to identify the depth position of a target is essential for any 3D pointing technique that allows the user to manipulate the selection tool along the z-axis. For example, in the experiment described in Chapter 6, I observed users having difficulties placing the 3D cursor inside the target, when they could not determine the position of the target on the z-axis. The behaviour observed – moving the 3D cursor there and back multiple times – significantly slows down the pointing performance. Yet, many interaction techniques do not directly provide an on-screen cue for the depth position of an object, thus requiring stereoscopic or immersive displays to fully support 3D perception.

One of a few techniques that do provide a depth identification mechanism is the Silk Cursor technique (Zhai et al., 1994). In Silk Cursor, the cursor is represented as a semi-transparent cube. The occlusion levels are used to cue the relative depth positions of targets with no measure of how much they are spatially separated. Kyto et al. (2013) also discusses how occlusion can help

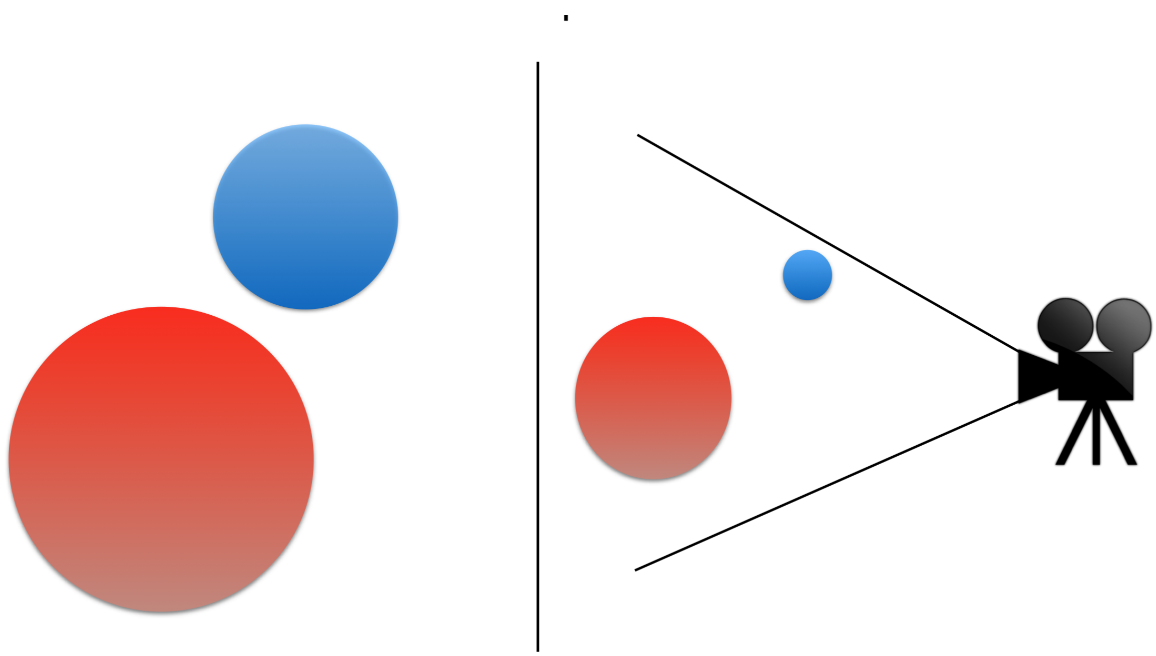

Figure 3.3: The blue sphere appears further from the user when viewed from the perspective of the user (a), an illusion caused by its relative proximity to the user and smaller size (b), resulting in depth identification problem

users to infer the depth position of the object. However, these approaches have serious limitations: inferring depth from occlusion requires scenes to include occluded objects, a requirement that excludes entire classes of sparse 3D environments. Other techniques attempt to imitate 3D rendering on 2D displays, by means of depth of field rendering (Mauderer et al., 2014) or linear perspective (Cook et al., 2008; Wanger et al., 1992), thus imposing a specific rendering estetics that may not be appropriate for all 3D environments.

Similar to an occlusion management mechanism, it is desirable to have the depth identification mechanisms built into the pointing technique, to avoid the added cost of separate depth identification phases. An example of such an approach is provided by (Wyss et al., 2006), who introduce a two-handed technique that uses the cross point of two cast rays as an indicator of the object position. Yet crossing rays is a bimanual technique that requires users to manipulate double the number of degrees of freedom at the same time, thus increasing the technique's complexity (Argelaguet and Andujar, 2013) and perceived workload. Another contribution of my thesis is the design of the Tiltcasting, presented in Chapter 6, that has the depth identification mechanism built into the first phase of selection.

3.3 Perceived Workload

High perceived workload resulting from prolonged use of almost all 3D pointing techniques is an important problem of 3D pointing and 3D interaction (Bowman Doug A. et al., 2004). Arguably, interaction with 3D environments resembles the way humans interact with the real world as compared to interaction with 2D environments. Yet, while 2D environments use techniques that cause low perceived workload (e.g. computer mouse), 3D pointing techniques fail to provide similarly comfortable solutions (Card et al., 1991). Mouse movements are made by movements of small and fast muscle groups, while 3D pointing often requires a complex arm movement involving larger and slower muscles (König et al., 2009). For that reason many 3D environments used in professional semi-public settings, such as CAD and 3D modelling applications, use 2D input techniques (e.g. mouse or trackpad), instead of 3D input.

One reason why 3D pointing techniques tend to produce high perceived workload is the use of 1-1 spatial input (Hincapié-Ramos and Guo, 2014). For example, in 3D Depth Cursor and related techniques, using a magnetic tracker requires that the users hold their hand raised in front of them for a prolonged time. If targets are positioned in upper sections of the 3D environment control space, users must reach into these sections with their hands. The same situation applies to virtual hand techniques. Performing mid-air gestures was determined to cause high perceived workload in the early stages of 3D pointing development, including the so-called "gorilla-arm" effect.

The "gorilla-arm" effect is a condition first reported by Schultz (Schultz, 1988). Perceived workload in general, and the gorilla-arm effect in particular, is difficult to quantify, as it is a subjective perception of the user. For the gorilla-arm effect, Ramos and Guo (2014) proposed a metric called "consumed endurance" that quantifies the effect as a ratio of the interaction time and the computed endurance time. Consumed endurance has its basis in sport sciences and ergonomics, using Rohnert's formation of the endurance model, which is a function of the value of force applied in relation to a maximum force of the muscle. Other methods of quantifying perceived workload were tried, including heart-rate (Sjogaard et al., 1988), oxygen level (Ferguson et al., 2011) or EMG (Peres et al., 2009) as well as more subjective assessments, such as the Borg CR10 scale (Borg, 1998).

While it might be desirable to leverage consumed endurance as a measure, much of the work described in this thesis was performed prior to the development of consumed endurance measure (Hincapié-Ramos and Guo, 2014). A more general way of quantifying perceived workload is to perform NASA Task Load Index (TLX) survey (Hart and Staveland, 1988). NASA TSX is a widely used method for quantifying a task's load index, although it is often criticized for its high scalar invariance, leading to biased mean scores, thus making the examination of mean differences misleading (Bustamante and Spain, 2008; Wiebe et al., 2010). However, the NASA TLX index is still useful for within-subject comparisons of task difficulty between interaction techniques. For that reason I use the NASA TLX as an additional metric of the quality of interaction techniques proposed in this dissertation.

The number of degrees of freedom used to control the 3D pointing technique are also an important factor contributing to the technique's overall comfort. The DoFs the user has to control is a measure of the complexity of the selection technique: the more DoFs, the more complex the control, but also the more expressive the technique (Argelaguet and Andujar, 2013). Consequently, the more degrees of freedom used to control the technique, the higher its mental demand, which contributes to the overall perceived workload.

In my dissertation, the issue of perceived workload is an essential consideration in the design of the techniques I propose. I hypothesize that a 3D pointing technique offering low perceived workload and thus increasing user comfort will be preferred by users over a high perceived workload technique, even if it offers lower time performance. Some evidence for this postulate is presented in Chapters 5 and 6, when I discuss the Tiltcasting and Watchcasting' workload. Additionally, in the case of the two ray-based techniques I propose, fixed-origin rays are used to reduce the number degrees of freedom involved to 3.

3.4 Validating 3D Pointing Techniques

3D pointing challenges related to speed, occlusion and perceived workload provide design guidelines for pointing techniques introduced in the next chapters. I used the same experimental setup in all studies. By comparing the performance of each technique against an established hardware device, I was able to quantitatively measure whether a smartphone or a smartwatch can in fact replace specialized equipment, as postulated by my thesis statement. Formal validation of each technique helped to identify strengths and weaknesses of smartphone- and smartwatch-based 3D pointing.

In all experiments I measured selection time and error rate. I have always included occlusion and target size as experimental conditions. To quantitatively measure perceived workload of each technique, I used the NASA Task Load Index (Hart and Staveland, 1988). Together, selection time, error rate and NASA TLX allowed me to evaluate to what degree my techniques address the 3D pointing challenges.

3.4.1 Experimental Environment

My experiment was heavily influenced by the experiment presented in Vanacken et al. (2007). By closely replicating their setup, I was able to perform a meta-analysis of each techniques' results in comparison with the performance results of Vanacken et al.'s Point Cursor technique. In Chapter 6, I discuss Vanacken et al.'s study results in the context of the performance of Tiltcasting.

3.4.2 Scene

The experiment consisted of an interactive 3D environment rendered on a black background. Each trial scene consisted of a start object, a target, and 43 distractors (Figure 3.4).

The start object was rendered as a yellow sphere, the target as a red sphere, and the distractors as blue spheres. Throughout the experiment the start object had a constant size of 1.5 cm and was displayed in a fixed position in the center of the display at a zero depth.

To preserve the same Index of Difficulty between all experiments, I have ensured that the target sizes and the 3D distance between the start point and the target was kept constant. The size of the target was either 1.5 cm or 3 cm and the 3D distance between the start object and the target was 40 cm in 3D space, resulting in $ID = 4.79$ and $ID = 3.84$, respectively. The index of difficulty (ID) was the same as in Vanacken et al. (2007). In their study target sizes were 0.75 cm and 1.5 cm and the distance in 3D space equal to 20 cm. While preserving the ID, I have

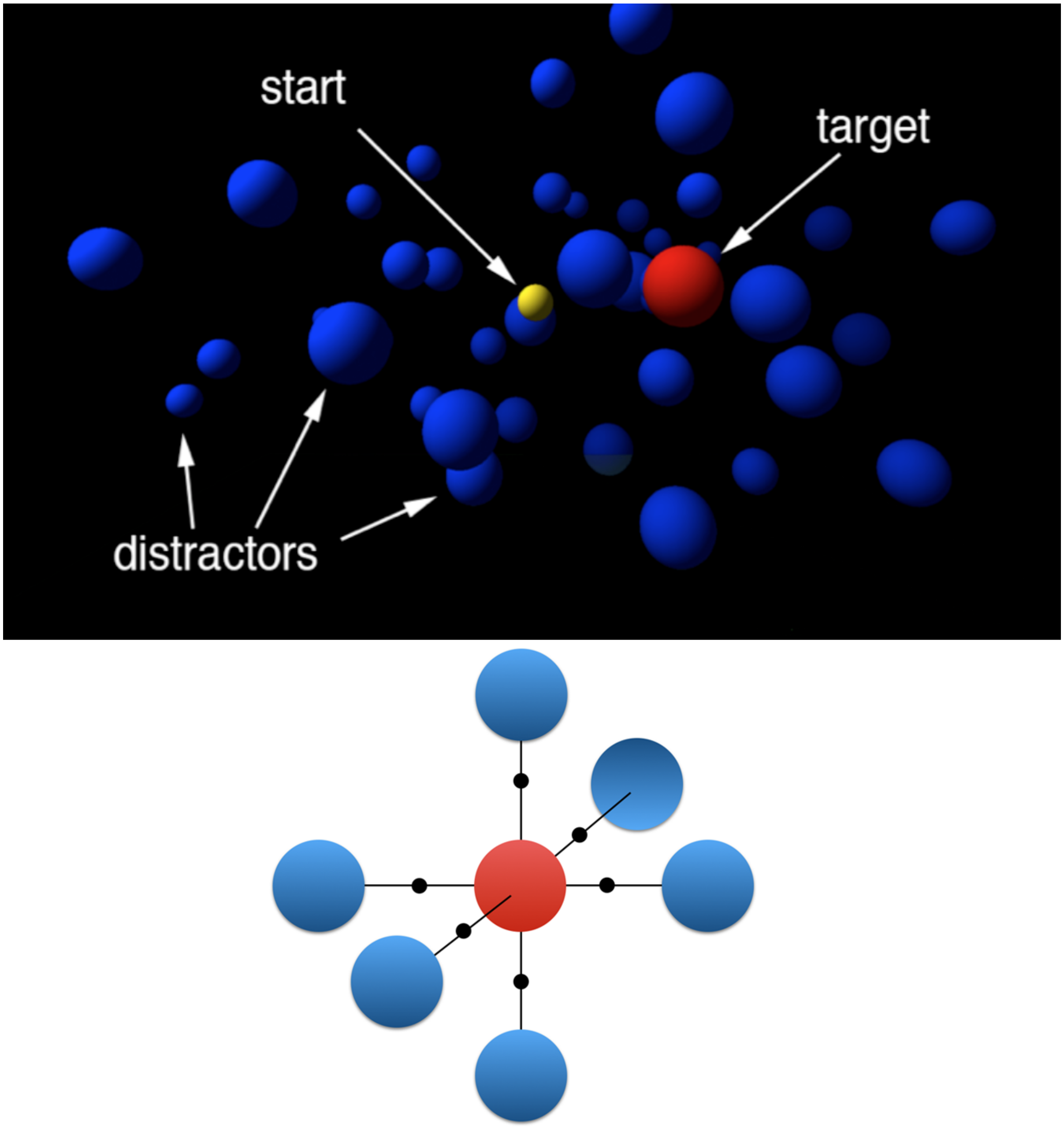

Figure 3.4: Experimental environment used in all experiments

doubled both the distance and the size of the target because of larger size of the displays used in my study.

The positions of the distractor targets were randomly determined, with constraints on their position ensuring that they did not intersect with each other, the start target, or the goal target. In each experimental block the distractor targets were randomly assigned sizes between 1.5 cm and 3 cm. The density of the scene was constant in the immediate area surrounding the target. Six distractor objects were carefully placed around the target forming a cube-shaped Voronoi region. For each trial the entire Voronoi region surrounding the target was further rotated by a random angle.

3.4.3 Task

For each trial, participants first selected the start object. Once the selection tool entered the start object, the object would disappear and the red target sphere would become selectable. Trying to select the target without first selecting the start object was not possible.

After selecting the start objects, participants moved the selection tool to intersect the target. Once that was done, participants had to confirm the selection of the target. The selection confirmation mechanism was technique-specific and is described in detail in each technique's chapter. Once the selection was confirmed, the task ended and the 3D environment was re-set for another trial.

For trials with occluded targets, participants were first presented with the target for $500ms$. Then the target would be completely hidden behind an occluder. This method ensured that visual search for the target was excluded from the selection time. Nevertheless, to reach the occluded target the participants had to use the occlusion management mechanisms. The occlusion removal mechanism was specific for each presented technique and is described in details in each techniques' chapters.

3.4.4 Independent Variables

In every experiment I used two target sizes (small and large), and two occlusion conditions (visible, occluded). While target sizes condition was measured within each block, with every block having equal number of small and large targets, the occlusion condition always formed a separate block of trials. The number of independent variables was study-specific.

The design of each experiment was balanced with partial or full latin square to reduce biasing influence of one blocks on another. A number of combinations that formed latin square design was specific for each experiment and is discussed in details in each technique's chapter.

3.4.5 Procedure

Participants were first asked to complete a brief demographic questionnaire. Before the experimental trials, each participant was briefed on the technique. Then the participants completed 10 training trials for each technique (in Tiltcasting) or input device (in Smartcasting and Watchcasting). These practice trials allowed participants to familiarize themselves with input devices used in the experiment. It was experimentally observed that after four to six trials the selection performance stabilized. The goal of these practice trials was to eliminate learning effects. In each of the following three chapters, I perform a block-by-block analysis of performance to ensure that learning effects were negligible during the experimental blocks.

Participants then completed a number of experimental trials in separate blocks. The number of trials and blocks varied between experiments depending on the number of experimental conditions. However, each block always consisted of an equal number of large and small targets that were always rendered at the same distance (in 3D space) from the start object.

After each block, participants completed a brief post-study questionnaire that examined perceived workload during their trials using the NASA Task Load Index survey. Each participant's commitment to the study lasted between 30 to 60 minutes, depending on the number of experimental conditions in a given experiment.

3.4.6 Data Recorded

All interactions made with the study software, including successful and erroneous selections, were logged to .csv computer files. Error rates were calculated automatically by counting the number of errors registered in each block's log file.

Selection time was the primary experimental measure. Selection time was defined as the time between entering the start position and confirming the selection of the target. If the user accidentally selected a distractor, the distractor's colour changed to light blue in order to indicate an erroneous selection, which was recorded by the system. However, the participant would continue the selection process until the target was successfully selected.

3.4.7 Outliers

In calculation of average selection time per given combination of independent variables I have removed those selection trials that had selection time that differed from the mean by three standard deviations. This decision was justified by the fact that, through qualitative observation of participants during the experiment I noticed that selections that took a particularly long time resulted from distraction in the middle of the trial, and not from the characteristics inherent to the trial.

3.4.8 Data Analysis

Selection times were read from the .csv files and, after calculation of means, converted to SPSS input files. Repeated Measures analysis of variance (RM-ANOVA) tests were conducted to examine differences in selection times between target sizes, target visibility, and depth rendering conditions.

NASA Task Load Index (TLX) (Hart and Staveland, 1988) data was transcribed into SPSS input files. Friedman tests were used to examine differences in perceived workload measures between conditions based on NASA TLX data. An alpha-value of .05 was used for all statistical tests.

3.5 Summary

In this Chapter I have identified a number of challenges of 3D pointing:

1. The speed and accuracy trade-off, that points at the limits of the human motor system and its consequences for 3D pointing performance

2. User perception-related problems, including: the problem of occlusion, which explains the difficulties of selecting a target that is hidden behind other objects, the depth identification problem, which has its strongest manifestation when 3D environments are presented on 2D displays that do not show the depth position of the object, and the target disambiguation problem, which happens in dense environments where the user must choose *the* target from many nearby candidate targets,

3. Perceived workload, which is a reason (Schultz, 1988) why 3D environments are less commonly used than 2D environments.

Addressing these challenges is essential in designing a usable 3D pointing technique. The degree to which the challenges listed in this section are successfully addressed validate the design of three novel 3D pointing techniques offered in my dissertation: Smartcasting, Watchcasting, and Tiltcasting, presented in Chapters 4, 5 and 6, respectively. Each technique is evaluated using the experimental set-up described in the previous section.

Each of the three techniques address the 3D interaction challenges to a different degree. For example, Smartcasting (Chapter 4) reduces user's perceived workload by fixing a ray's origin and thus allowing for manipulation of the cursor's depth position through touch input, in turn reducing the number of degrees of freedom required to control the technique. Smartcasting also introduces a novel solution to cope with the Heisenberg effect for target-agnostic environments, increasing accuracy and reducing the target disambiguation problem. Yet, Smartcasting does not improve on previous work on depth identification, and it uses an occlusion management mechanism previously reported in the literature.

Watchcasting, presented in Chapter 5, offers the first step toward using wearable devices, such as a smartphone, to facilitate pointing in 3D environments. I find that, while Watchcasting provides a freehand pointing technique that does not require specialized input devices or hardware, further work is needed to improve Watchcasting usability in terms of perceived workload, and the gorilla-arm effect (Schultz, 1988).

A final technique, Tiltcasting, is reported in Chapter 6. Tiltcasting seeks to address all of the 3D pointing challenges listed in this chapter. It offers a low perceived workload 3D pointing technique, eliminating the gorilla-arm effect. It reduces hand tremor through a design that involves both hands in stabilizing the cursor position. Using the same method as Smartcasting, it also reduces the Heisenberg effect. It introduces a novel occlusion management mechanism that is integrated into the selection technique, and does not add a new, costly phase to the selection process. It offers a depth identification mechanism that eliminates depth confusion by providing depth cues that allow the relative depth position of objects to be identified, even in target-agnostic environments. It reduces the target disambiguation problem through both an occlusion management mechanism and the reduction of the 3D pointing problem to a 2D pointing problem. It also decreases overall selection time by integrating 3D environment exploration (search for an occluded target) into the selection mechanism. Moreover, by introducing a reversed 3-state Buxton model, it allows for easy integration of selection tasks with further touch-based rotation and translation techniques, as recommended by Bowman et al. (2002).

Chapter 4

Smartcasting

The first technique I designed to support 3D pointing in computing environments is called Smartcasting. In order to verify whether a smartphone can support 3D pointing, I implemented two baseline virtual pointing metaphors: Raycasting and Depth Cursor. Implementing Raycasting (Liang and Green, 1994) on a smartphone demonstrates that smartphone hardware is capable of supporting basic 3D pointing. The Smartcasting implementation also serves as a baseline reference for the comparative studies performed in Chapters 5 (Watchcasting) and 6 (Tiltcasting).

However, Raycasting does not support the selection of occluded targets or allow for control over the depth position of the selection tool. Yet, universal 3D pointing should support selection of both non-occluded as well as occluded targets, because both types of selection may be required in interaction with 3D environments in many applications. To address this requirement, I also implement Depth Cursor, an extension of Raycasting in which the user can control a position of Depth Cursor that travels along the ray.

The design of Smartcasting was an iterative process, driven by pilot studies, which I describe in detail in Section 4.1 of this Chapter. Once I arrive at the final design, I compare Smartcasting's performance with WiiMote-based Raycasting and depth-ray in a formal experiment. My goal is to implement an equivalent to specialized input device pointing for 3D environments using a smartphone.

4.1 Designing Smartcasting

Objects in 3D environments can be positioned and rotated around six degrees-of-freedom. Such objects can be then translated in space along any of the x-, y-, or z-orthogonal axes to define position within a space mapped via a Cartesian 3D coordinate system. Furthermore, even without translation, objects within a 3D space can be rotated around each of the axes, defining three orthogonal rotations typically referred to as pitch, yaw, and roll. Orthogonal rotations are particularly easy to address using a smartphone as an input device: accelerometers can provide either an isometric or elastic technique for controlling the rate of rotation, similar to the way in which an isometric joystick controls cursor speed on laptop computers so equipped. On the other hand, less is known about how well a commercially available smartphone can be used to perform 3D pointing.

A smartphone can be used as an input device by implementing either virtual hand or virtual pointing techniques. However, as discussed in Chapter 2, virtual pointing techniques have a number of advantages over virtual hand techniques. In particular, virtual hand techniques require wide movements of hands, and thus quickly cause high perceived workload and the so called gorilla-arm effect.

Of the various virtual pointing techniques, the smartphone is most suited to a Raycasting with Depth Cursor, because smartphone's form factor and hardware is more similar to a tracker than a virtual glove. A Depth Cursor is analogous to a 3D Cursor manipulated with a mouse, where an on-screen indictor (a pointer) serves as a virtual proxy for a user's on-screen location. I therefore designed and implemented a Depth Cursor technique, where the angle of the ray intersecting the display is controlled via the yaw and pitch of a smartphone and "depth" is specified using the touchscreen of the smartphone. I dub this technique Smartcasting, illustrated in Figure 4.1.

When a user holds a smartphone in his or her hand, it may be moved around even when 3D pointing technique is not being performed. Additionally, users may switch between using a smartphone to control their Depth Cursor and using it to access information. As a result, I need an elegant mechanism to move between the three states typical of input devices: an out-of-range state where the input device is not being tracked; a tracking state where the movement of the input device maps to on-screen cursor movement; and a dragging state where acquired targets are repositioned on the display. To address this requirement, Smartcasting's behaviour can be characterized using a 3-state model (Buxton, 1990): it begins in an "out of range state" where its inputs are ignored by the 3D environment. After placing a finger on the smartphone's touch screen, the device shifts into a tracking state where orientation information is relayed as input to the large display. While tracking, a finger up and down (i.e. a "reverse" click) will select an on-screen target, and move interaction into a "drag" state allowing for further object manipulations. The full interaction model is depicted in Figure 4.2.

In conventional Raycasting techniques the ray should appear to emanate directly from the smartphone in a straight line from the the device. To do this, I must map the yaw and pitch of the smartphone onto the world coordinate system, projecting out from the user. Smartphones

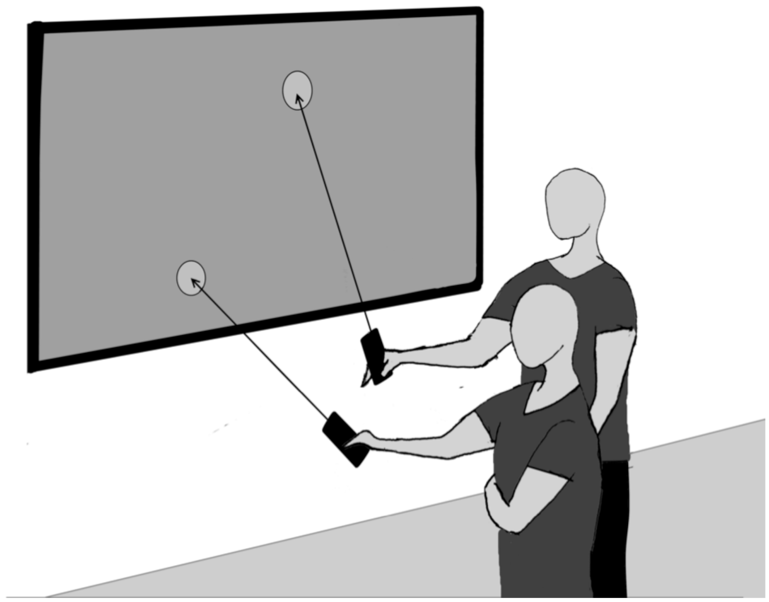

Figure 4.1: Through Smartcasting, users can interact with 3D content on nearby large displays using their personal mobile device.

contain a gyroscope and an accelerometer. Using the force of gravity, an accelerometer can provide accurate pitch data, and using the gyroscope, a smartphone can sense changes in its yaw angle. However, in order to measure cursor location and angle of the ray on the display accurately, I must know both the yaw and pitch of the device and the user's distance from the display. Furthermore, gyroscope readings for yaw are subject to drift introducing additional imprecision in the horizontal/x-axis location.

Before embarking on an aggressive design exercise to correct for yaw drift or identify the location of a user via computer vision, I first wanted to explore the severity of the problem. Given the similarity between the Depth Cursor represented by on-screen 3D pointer (i.e. a "cursor") and a 2D cursor manipulated by a computer mouse, it might be the case that the relative movement, controlled by yaw and pitch, rather than the absolute mapping of yaw and pitch might be sufficient to allow control of the Depth Cursor. In other words, the smartphone might function as a relative, not absolute, input device (Figure 4.3).

To test relative versus absolute input, I informally piloted my technique with four graduate students. I found that, for pitch, there exists a significant tolerance for variations between device pitch and y-axis location on the display. Even a coarse-grained location estimated from the device camera can be sufficient to position the initial ray, and relative mappings caused few

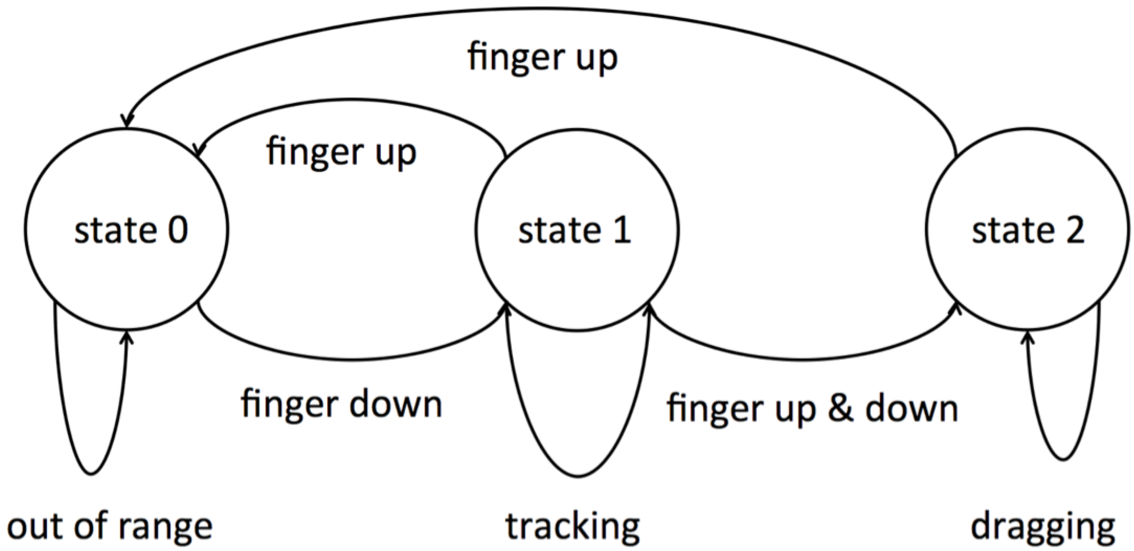

Figure 4.2: 3-state model for Smartcasting based on that of Buxton (1990). Finger up and down (reversed tap) triggers the transition from state 1 to state 2.

problems with y-location control. Additionally, initial inaccuracies were insignificant for yaw. If one assumes that the phone is pointed at the cursor when movement begins, the yaw angle inaccuracies are easily overlooked during one targeted movement. In case of drift, the yaw angle can be re-set each time a user transitions into state 0, the out-of-range state.

Beyond the (x, y) location of the ray on the 3D display, I also need to control the depth of the cursor using movement of the contact finger on the touchscreen. Two options present themselves: direct mapping of finger position to depth, and relative mapping of finger position to depth. In direct mapping of finger to depth, I assume that the y-axis of the smartphone display maps to depth along the ray.

I evaluated each of the two options for finger position to depth mapping. For direct mapping, I found that touch accuracy on a smartphone is limited due to the "fat finger" problem (Albinsson and Zhai, 2003; Siek et al., 2005), making it difficult to acquire small targets for my participants. In contrast, with relative mapping, users needed to clutch (i.e. release and move their finger back on the touchscreen to increase cursor reach) to control the depth of the target. However, based on feedback collected from users, the cost of clutching on a touchscreen is relatively small when compared to the cost of an inability to target. One user noted that my use of the touchscreen was analogous to using a touchpad, where clutching is a frequent, acceptable action to move the

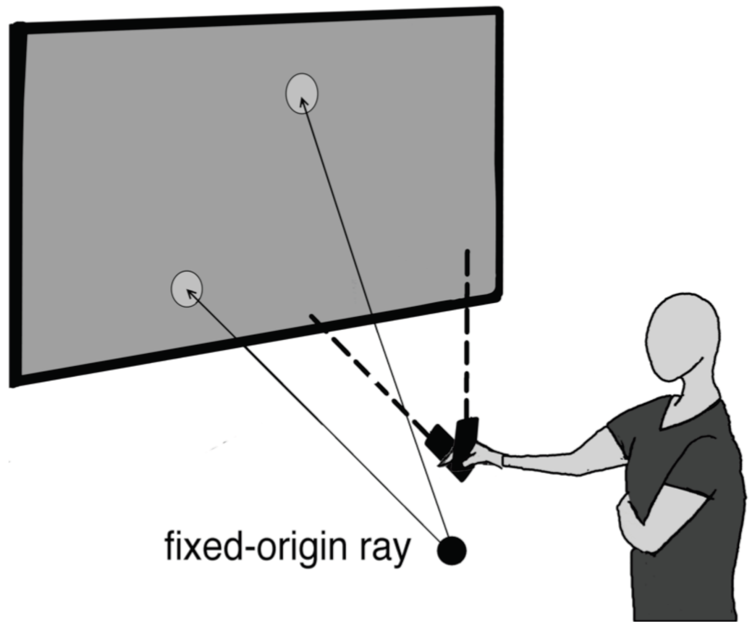

Figure 4.3: In Smartcasting, position of the phone is not known; the origin of ray is fixed

cursor longer distances with lower control to display gain.

4.1.1 Coordinates of Fixed Ray Origin

Initially, I fixed the origin of the ray at a distance of 1 m on a line originating from the center of the display and perpendicular to it. For displays mounted at eye level such placement closely matched the 3D position of the smartphone in front of the users and pointing at the centre of the display. Such a ray origin simulates traditional Raycasting with the ray originating from the input device as long as the user stays in a specific position in front of the display.

In order to support 3D pointing from any position in front of the screen, I embarked on finding a position for the ray origin that did not impose or encourage any specific proxemic relation between the user and the screen. In a number of pilot studies, I asked users to manipulate the ray. For each trial, the ray origin was was placed at a different locations. I chose the ray origin position at the bottom of the screen, 170 cm in front of the screen. This position of ray did not require downward movement of the smartphone for selection of targets that were rendered in the lower half of the display: horizontal position of the smartphone allowed select the targets displayed at the bottom of the screen.

4.1.2 Non-isomorphic Mapping of Ray Rotation

I initially attempted to use isomorphic mapping between the ray movement and the movement of the smartphone. However, isomorphic mapping did not appear isomorphic. When the users stood at a distance of 170 cm from the screen, the cursor angular gain appeared to match the smartphone rotation. When the user stood very close to the display, seemingly small movements of the smartphone appeared as large moments of the cursor on the screen. I realized that this was because with isomorphic mapping user perceived the ray as originating from the device. This was confirmed by pilot users commenting that the ray originated from the smartphone only when standing at a certain distance for the screen.

In the next iteration I ensured that the gain did not mach the cursor movement, thus immediately informing the user that the mapping between the cursor movement and the smartphone's rotation is non-isomorphic. Further, a pilot study revealed that users preferred non-isomorphic gain for the ray movement over the gain that closely matched isomorphic mapping. For example, users preferred smaller wrist movements that resulted in larger rotations of the fixed origin ray, as long as the gain allowed enough precision to select small targets. Too much gain increased the hand tremor, while too little gain required wide wrist movements. After a number of experiments, I set the gain to 1.5× the rotation of the input device, so that a rotation of the input device by an angle α resulted in a rotation of the ray by $1.5 \times \alpha$.

4.1.3 Interacting with Occluded Targets

To address the occluded target problem the target intersected by the Depth Cursor is visible, while all intersected objects that are in front of the target are temporarily made transparent. A target-occluding object is shown again when it becomes the furthest object intersected by the Depth Cursor or when the ray no longer intersects it. While a number of occlusion removal mechanisms exist (Elmqvist and Tsigas, 2006, 2007; Zhai et al., 1996a), hiding objects along the ray in front of the cursor seems to be most efficient solution (Vanacken et al., 2007). Moreover, the ability of casual users to easily understand the occlusion removal technique was an important consideration for my design.

4.1.4 Ray Visibility

I also considered whether the ray on which the Depth Cursor moves should be displayed. Implementations of Depth Cursor often do not show the ray (Bowman Doug A. and Hodges, 1997). Yet, showing the ray could help users to identify their Depth Cursor, particularly if other Depth

Cursors are present. Additionally, showing the ray might act as a perspective cue that helps users to identify the relative positions of targets and the Depth Cursor in situations where the 3D environment is presented on a non-stereoscopic screen. On the other hand, rays can clutter the display. Given the strengths of both options, I conducted an iterative prototyping exercise. Pilot testing revealed no differences between a displayed or hidden ray for selection time, or for the perceived workload of pilot participants as measured using the NASA Task Load Index (Hart and Staveland, 1988). Ultimately, I decided to display the ray during my experimental evaluation.

4.1.5 The Heisenberg Effect

I initially considered using a tactile volume button, present on most mobile devices, to perform selection in Smartcasting. However during pilot testing, I found that pressing and releasing a hardware button could cause the Heisenberg effect. Moreover, because my 3D pointing techniques are designed for casual interaction, I must assume that the technique should support any kind of 3D environment, including the target-agnostic one. Unfortunately, previously developed methods of addressing the Heisenberg effect assume that the 3D environment is target-aware, while I assume that both target-agnostic and target-aware 3D pointing should be supported by casual mobile-based technique. Thus, to reduce this involuntary motion in target-agnostic 3D environments, selection in Smartcasting is performed using a "reverse click", consisting of an up and down motion, on the touchscreen rather than using a physical button. My pilot testing suggested that this choice reduced the impact of the Heisenberg effect, and produced a less error-prone selection technique.

4.1.6 Establishing Connection

In the context of walk-up-and-use scenarios, it is important to consider the way that a smartphone will quickly and temporarily bind with any display, preferably ensuring that the connection is secure. Smartcasting connects a smartphone to a display through a unique URL that is opened in the smartphone's browser. The user can type in the unique URL, read it from a QR code, or use a multicast DNS service. The URL can be reached through LTE or 3G data connection – there is no need to connect the smartphone to a local wifi network. For further security, the connection between the screen and the phone is relayed through a Secure WebSocket with SSL certificates. Pilot testing revealed that the speed of this setup is sufficient for low-latency interaction. My implementation provided interaction latency lower than 50ms over a 3G cellphone network.

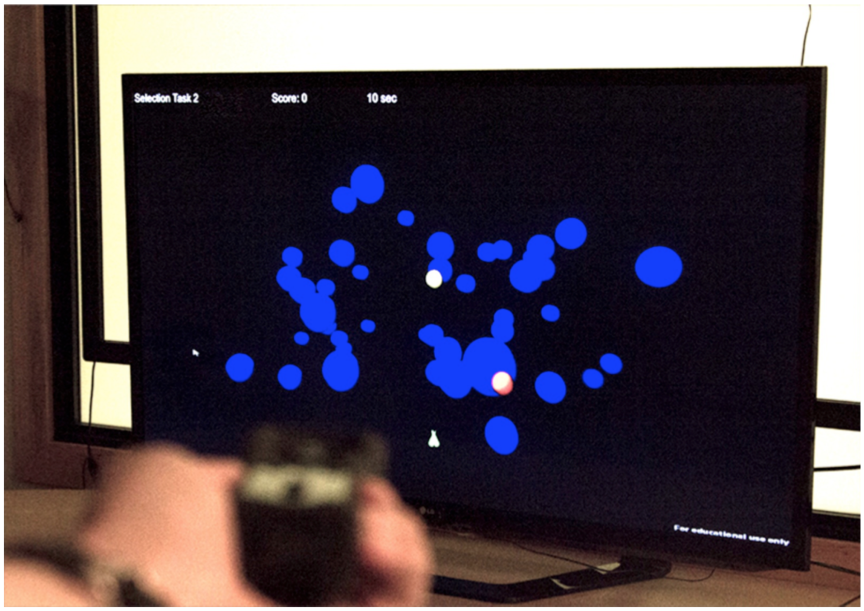

Figure 4.4: Smartcasting: Experimental Setup and Apparatus. 55-inch LG HDTV Cinema 3D and two input devices: WiiMote (shown) and iPhone 5 (not shown)

4.2 Empirical Validation

In order to validate Smartcasting's design, I compared the performance of Smartcasting using a smartphone against an implementation of Smartcasting that used a WiiMote to perform selection tasks in 3D scenes. My experiment investigated the efficacy of both techniques across different target sizes, and for occluded and non-occluded targets. By comparing the performance of Smartcasting with a smartphone against an established hardware device I was able to quantify the degree to which the use of a smartphone can replace more specialized equipment, and to identify any strengths and weaknesses inherent to smartphone-based 3D manipulations.

My evaluation followed the experimental setup discussed in Chapter 3. Below I describe the implementation and experimental design details.

4.2.1 Implementation Details

During the course of the study, participants were seated 3m in front of a 55-inch LG HDTV Cinema 3D circularly polarized stereoscopic display that was centred vertically and horizontally in relation to participant's eye line. All experimental software, except for the mobile client webapp, ran on a locally connected PC with an Intel i7 processor, 16GB RAM, and an NVidia GTX570. Implementation of Raycasting with and without Depth Cursor was identical for both input devices, thus ensuring that the implementation differences did not confound the comparison of results.

A baseline Raycasting technique was implemented using a Nintendo WiiMote Plus, which is equipped with a IMU and connected over Bluetooth. Smartcasting was implemented as a JavaScript webapp on an iPhone 5 that transmitted rotation and touch events at 10Hz over a local 802.11n wireless network. The resolution of the touch input was less than 0.07 mm.

Figure 4.4 illustrates the experimental setup and apparatus.

4.2.2 Participants

Twelve participants (10 males, 2 females) were recruited from local university to participate in the study. Participants' ages ranged from 24 to 30 (average = 26.8). Eleven participants were right-handed, one was left-handed, and all participants were screened on a stereoscopic display prior to the study for their ability to order objects by depth. Participants received $10 for their participation in the study.

4.2.3 Experimental Design

I used a 2 (INPUT DEVICE) $\times$ 2 (TARGET SIZE) $\times$ 2 (TARGET VISIBILITY) within-subjects design. The study utilized three independent variables: input device, target size, and occlusion. Participants completed trials using each of the smartphone and WiiMote implementations. For the target sizes, targets with either "small" (1.5 cm) or "large" (3.0 cm) sizes provided two levels of difficulty index based on Fitts's Law (MacKenzie, 1992). Finally, targets were either fully visible or fully occluded upon starting the trial. For the fully visible targets, no distractors occluded or partially occluded the goal target, whereas for the occluded version, the goal targets were hidden by the presence of distractor targets. The order of conditions was counterbalanced using a partial Latin square design.

My experimental design can be summarized as:

12 Participants

$\times$ 2 Input Device: Smartphone or WiiMote

$\times$ 2 Target Size: Small or Large

$\times$ 2 Target Occlusion: Occluded or Non-Occluded

$\times$ 14 Repetitions

For a total of 1344 trials.

4.2.4 Experimental Task

Participants performed the selection task described in Chapter 3. For each trial, participants first selected a start object in the center of the large display. Once the ray (or Depth Cursor) entered the start object, the start object would disappear and the destination target, another object on the display, would become selectable. Participants moved the ray to intersect the target. To select the goal target, participants released their finger from the touchscreen (for Smartcasting) or WiiMote "A" button (for Raycasting). Once the ray (or Depth Cursor) entered the goal target, the task ended and the screen was reset. If the user accidentally selected a distractor, the distractor's colour changed to light blue in order to indicate an error, which was recorded by the system. However, the participant could continue until the goal target was successfully reached.

For trials where occluded targets were present, a Raycasting with Depth Cursor (Grossman and Balakrishnan, 2006) implementation was provided for both the WiiMote and smartphone conditions, and participants were required to move the Depth Cursor to reach the start and goal objects. The Depth Cursor was manipulated by moving a finger on the touchscreen for Smartcasting and by the up-down buttons of the WiiMote's d-pad. As an occlusion management mechanism I used a variation of virtual X-ray: if the goal target was occluded, the distractors in front of the Depth Cursor that were intersected by the ray disappeared.

4.2.5 Procedure

Participants were first asked to complete a brief demographic questionnaire. Before the experimental trials, each participant was briefed on the Smartcasting, and screened for the ability to see depth. Then the participants completed 20 training trials: 10 with the WiiMote, and 10 with the smartphone input device. Participants then completed four blocks (two for each input device) of 28 experimental tasks, corresponding to 14 trials for each target size per block. Block orders were balanced with partial latin square design. After each block, participants completed a brief post-study questionnaire that examined perceived workload during their trials using the NASA Task Load Index. Each participant's commitment to the study totalled approximately 30 minutes.

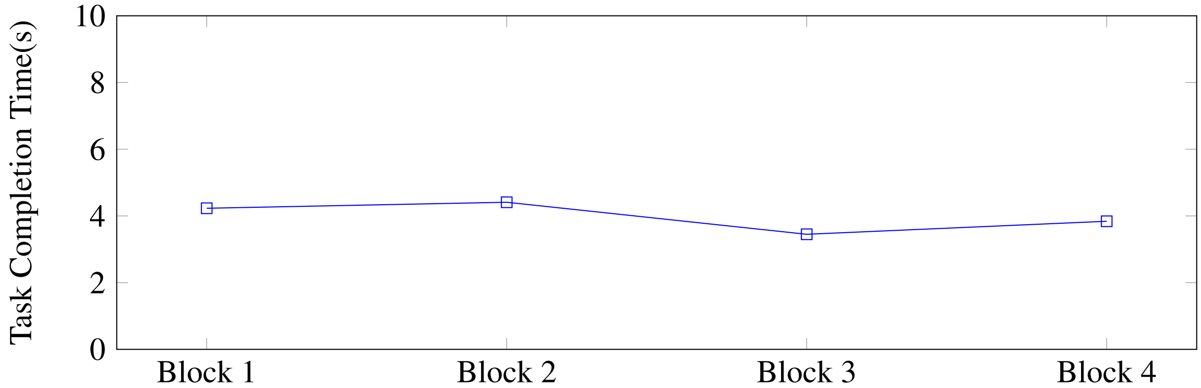

Figure 4.5: No learning effect was found between blocks

4.2.6 Data Collection and Analysis

All interactions made with the study software were logged to .csv files and transcribed for SPSS input files, as described in Chapter 3.

Repeated Measures analysis of variance (RM-ANOVA) tests were conducted to examine differences in selection times between target sizes, target visibility, depth rendering conditions and block ordering. Friedman tests were used to examine differences in perceived workload measures between conditions based on NASA TLX data. An alpha-value of .05 was used for all statistical tests.

4.3 Results

4.3.1 Learning Effect

No significant difference was found between blocks ($F_{1,3} = 0.360$, $p = .613$, $\eta_p^2 = .035$), indicating no learning effect between blocks (Figure 4.5). Because the data did not satisfy the test of sphericity, the reported effect size was corrected with Huynh-Feldt.

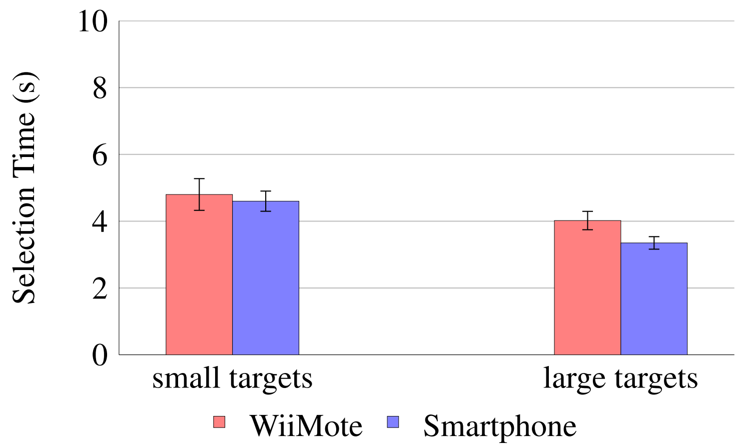

Figure 4.6: Mean selection times for smartphone vs. WiiMote for small and large targets

4.3.2 Performance

On average, participants completed each trial in $4.24s$ ($\sigma = 1.1$). I now consider each of my three independent variables: input device, target size, and occlusion. Average selection times were normally distributed.

4.3.2.1 Input Device

My analysis revealed no difference between the two input devices for target selection time ($F_{1,11} = 1.68, p = .221, \eta_p^2 = .13$). Smartphone selections took an average of $4.04s$ ($\sigma = .77$), and WiiMote selections took $4.43s$ ($\sigma = 1.54$) on average.

4.3.2.2 Target Size

As expected, my analysis revealed main effects for destination target size ($F_{1,11} = 84.04, p < .001, \eta_p^2 = .884$), with small targets ($4.75s, \sigma = 1.19$) taking longer to select than large targets ($3.72s, \sigma = 1.05$). An interaction effect was also found between target size and occlusion ($F_{1,11} = 5.61, p = .037, \eta_p^2 = .338$), where small, occluded targets ($6.78s, \sigma = 1.80$) took longer to select than large, occluded targets ($5.39s, \eta_p^2 = 1.75, p < .001$), however for non-occluded targets my analyses revealed no difference (small targets:$2.77s, \sigma = .759$; large targets: $2.04s, \sigma = .485$; $p <$

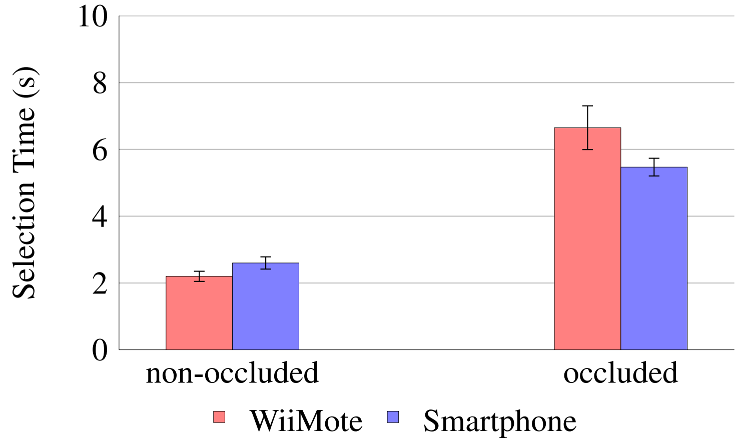

Figure 4.7: Mean selection times for smartphone vs. WiiMote for occluded and non-occluded targets

.001). No interaction effect was found between input device and target size ($F_{1,11} = .76, p = .403, \eta_p^2 = .064$) (Figure 4.6)

4.3.3 Occlusion

As expected, selection times for occluded targets were longer than for non-occluded targets ($F_{1,11} = 83.14, p < .001, \eta_p^2 = .88$). Occluded targets were selected in 6.06s on average ($\sigma = 1.75$), whereas non-occluded targets were selected in 2.4s on average ($\sigma = .585$). However, my analysis revealed an interaction effect between input device and occlusion ($F_{1,11} = 7.69, p = .018, \eta_p^2 = .412$). Occluded targets were selected in less time ($p = .005$) using the smartphone ($5.47s, \sigma = 1.06$) than using the WiiMote ($6.65s, \sigma = 2.62$), however no difference was found ($p = .234$) between smartphone ($2.60s, \sigma = .734$) and WiiMote ($2.20s, \sigma = .610$) selection times for non-occluded targets (Figure 4.7)

4.3.4 Error Rates

In Chapter 3, I defined an error as a confirmed selection of a distractor object. In the experiment no selections resulted in a selection error. This result is consistent with no errors reported for Depth Cursor in Vanacken et al. study (2007).

4.3.5 Perceived Workload

I also analyzed participant questionnaire responses for any identifiable trends in their perceived workload across all trials. My analyses revealed differences in the frustration ($p = .002$) and physical demand ($p = .004$) for trials in which occlusion was present. However, NASA TLX data revealed no other differences between the experimental conditions. A complete summary of the NASA-TLX data is presented in Figure 4.8.

4.4 Discussion

My results indicate that Smartcasting with a smartphone offers similar performance levels to those of my WiiMote implementation. This result validates my goal of supporting efficient interaction without requiring users to carry specialized input devices, and instead to rely only on interactions via a mobile device. My results also suggest that selections made with Smartcasting were faster for occluded targets when using a smartphone. When considered as a whole, the effect of the input device (smartphone versus WiiMote) accounted for a relatively small portion of the variance in my model ($\eta_p^2 = .13$), suggesting that smartphones perform similarly to WiiMotes across different target sizes and degrees of occlusion. Finally, my analysis of perceived workload revealed no differences between input devices.

4.4.1 Performance

My study validates many choices that I made while designing Smartcasting, and informs the design of cross-device interaction with ubiquitous displays. I now reflect on these decisions, and on how my findings affected Smartcasting's performance. In particular, I discuss my decisions related to enabling Raycasting interactions without real-time/precise spatial input data, and in leveraging a phone's touch screen to enable more powerful interactions.

4.4.1.1 Raycasting without Accurate Spatial Tracking

One of the compromises in limiting my design to existing smartphone technology was forgoing the ability to accurately track a user's position in front of the display, and instead capture only sensor data related to the smartphone's orientation in 3D space. My validation of Smartcasting demonstrates that such a choice may not be as significant a compromise as many might expect,

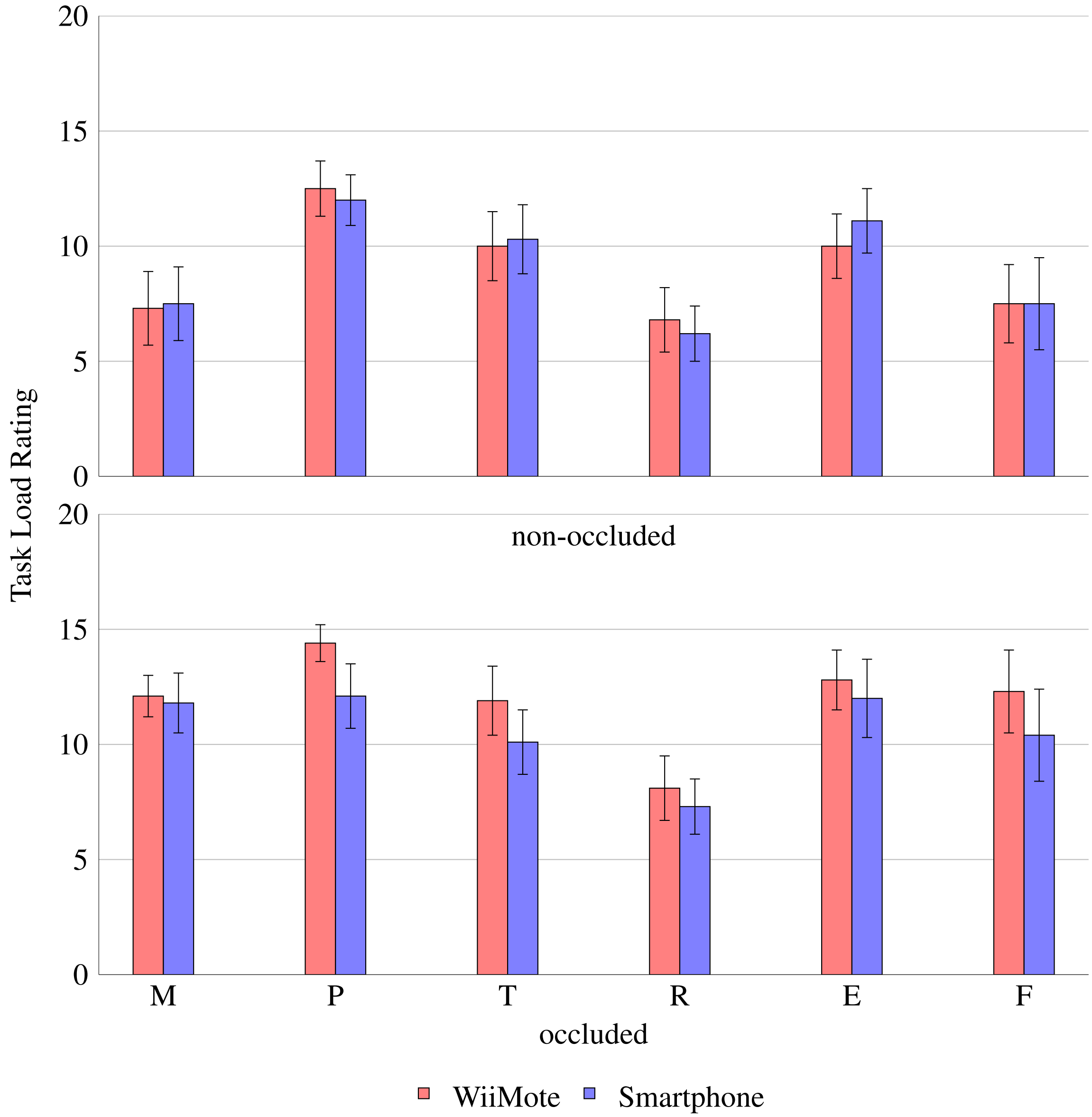

Figure 4.8: NASA TLX results for WiiMote vs. smartphone, with and without Depth Cursor: Mental Demand (M), Physical Demand (P), Temporal Demand (T), Performance (R), Effort (E) and Frustration (F) measures.

and that techniques matching the performance and perceived workload of traditional Raycasting implementations can be developed without such technology.

Previous research indicates that this design is similar to a form of Raycasting called "fixed-origin Raycasting" (Jota et al., 2010). Even though fixed-origin Raycasting techniques were shown to perform slightly slower than free origin techniques Raycasting (Jota et al., 2010), it is the former that should be recommended from the ergonomic perspective as they allow users to choose – and change – the position of their hand and they involve only wrist and finger movements. For casual interaction in computing environments, user comfort should take precedence over small performance differences.

Interestingly, this choice also provides insight into how Raycasting techniques can support accurate interaction at a distance. By sacrificing a precise measure of the user's position, and thus the ability to accurately draw an on-screen "origin" of the cast ray, I am also able to improve angular precision for users interacting at a distance. For example, when a user is in close proximity to the screen using a conventional Raycasting implementation, they have a stronger degree of angular control over the ray's position than when standing far away from the display. Smartcasting follows a different model, and is agnostic to the user's position relative to the large display. This model could be further instrumented to enable more fine-grained control over the mapping between the phone's angle and the on-screen ray's trajectory, thus providing users at a distance a means of more accurately interacting with on-screen artifacts.

4.4.1.2 Raycasting with a Touchscreen

I initially explored methods of enabling Raycasting using only a smartphone's gyroscope and physical volume buttons. However, my prototyping process revealed that this choice led to imprecision in terms of selection, as the physical act of pressing a button often interfered with a user's ability to select small on-screen targets. Consequently, I explored techniques that leveraged the smartphone's touch screen and developed a 3-state model that allowed for users to disengage from the display, in order to select and drag on-screen objects. My validation then confirmed that the developed technique could facilitate basic interaction with a nearby display.

On the other hand, because my evaluation was intended to verify my design choices, I methodologically chose to constrain my Smartcasting design in order to implement a fair comparison to the WiiMote by minimizing potential confounds in my experimental design. Thus, while my technique provides an example of what is possible for smartphone interaction on large displays, there remains a need to explore the smartphone's touch screen and additional sensory inputs more extensively. For example, it may be beneficial to explore the use of on-screen chording (Davidson and Han, 2006), multi-touch input (Steinicke et al., 2008; Valkov et al., 2011), or gestures (Vogel and Balakrishnan, 2005) to enable more powerful 3D interactions. Similarly, accelerometers may be used to enable motion gestures (Jeon et al., 2010) through the phone.

4.4.2 Occlusion and Depth

Although successful as the first step toward 3D interaction with ubiquitous displays, Smartcasting addresses some of the challenges listed in Chapter 3, while other challenges were not addressed any better than in previous studies. The performance of Smartcasting is on a par with, but not better than the performance of WiiMote-based 3D pointing. The occlusion problem is addressed in a similar way to that described in the literature, i.e. through transparency. Target disambiguation is realized in a manner similar to depth ray (Vanacken et al., 2007), while other techniques known from the literature and listed in Chapter 2 could also be used. In Chapter 6, I present Tiltcasting, a smartphone-based, multi-modal, two-handed technique that addresses those challenges that were not sufficiently addressed by Smartcasting.

4.4.3 Perceived workload

Lowering perceived workload was one of the main considerations when designing Smartcasting as a 3D pointing technique. Two observations are worth noting. First, for the task that take a longer time – i.e. selection of an occluded targets – users indicated a slight preference for smartphone over WiiMote in terms of frustration (p = .002) and physical demand (p = .004). I hypothesize that using a touchscreen for manipulation of depth is preferable over using the tangible buttons of a WiiMote when the task takes a longer time, although further study is required to confirm that hypothesis.

Second, because in Smartcasting the origin of the ray is fixed, the technique is controlled with only two (or three, when controlling depth), instead of five (or six, when controlling depth) degrees of freedom. A lower number of degrees of freedom results in less complex and less mentally demanding techniques. A minimal number of degrees of freedom required to enable a given interaction is therefore recommended (Bowman Doug A., 2002).

4.5 Summary

Smartcasting provides a novel 3D interaction technique that leverages the ubiquity of smartphones to enable interaction with ubiquitous displays. The design and evaluation of Smartcasting allows me to conclude that a smartphone device is capable of acting as an input device for

3D environments. Moreover, hardware limitations, such as difficulties in determining the exact position of the smartphone in space, do not significantly handicap the interaction. I also suggest that taking advantage of the multimodality of a smartphone, such as combining orientation with touch input, can benefit the technique's design. Together, the hardware constraints and the multimodality allow for the design of a technique that is less complex and thus easier to use than the alternatives.

The use of mobile, touch-enabled devices, combined with removing the position information constraints of conventional Raycasting techniques provide opportunities to explore new, engaging methods of serendipitously interacting with ubiquitous displays. Most importantly, Smartcasting eliminates the need for specialized input or display hardware, thus addressing a common barrier to serendipitous 3D interaction. In doing so, it provides a first step towards low-cost interaction with 3D environments.

In the next Chapter I introduce Watchcasting, a 3D pointing technique that takes virtual pointing in another direction: input via a smartwatch. In the design of Watchcasting I discuss how lessons learned in Smartcasting design can be applied, with modifications, in a design of 3D pointing with a smartwatch.

Chapter 5

Watchcasting

Smartcasting provides an efficient smartphone-based 3D interaction for public displays. As the goal of the previous Chapter was to verify the capabilities of an off-the-shelf smartphone for 3D pointing, the goal of this Chapter is to verify that an off-the-shelf smartwatch can also provide such 3D interaction. More specifically, I wish to explore the differences between the design space of smartphone and smartwatch-based Raycasting and to verify to what extent techniques such as Smartcasting can be leveraged on a smartwatch. In this Chapter, I extend Smartcasting to wearable devices. While smartphones may remain the predominant convenience device for enabling interaction in computing environments, freehand interaction may also be useful, and sometimes preferable, for 3D pointing in computing environments.

A smartwatch – as opposed to a smartphone – is a device mounted on a wrist that leaves the user's hand unoccupied. Thus, smartwatch-based 3D pointing can be categorized as a freehand technique. My goal is to show that even given these constraints, smartwatch-based interaction provides a feasible alternative for 3D interaction. Additionally, I wish to provide evidence that specialized devices such as Myo may not be necessary for interaction in computing environments. To demonstrate that, I design, develop, and validate a smartwatch-based 3D interaction technique called *Watchcasting*.

5.1 Designing Watchcasting

Recall that Smartcasting leverages two modalities of the smartphone: touch input and orientation. Although the smartwatch is also a multimodal device, it is not a handheld device. Consequently, interacting with a smartwatch's touchscreen requires engagement of the other hand. Yet, engaging a second hand to interact with the touchscreen of a smartwatch would be against my design goal of creating a freehand technique. For that reason, in my design of Watchcasting I aimed at avoiding the engagement of both hands in the interaction and decided to use only a fusion of its gyroscope and accelerometer data into an rotation vector.

5.1.1 Raycasting on a Smartwatch

Similar to Smartcasting, the design of Watchcasting began with implementing fixed-origin Raycasting (Jota et al., 2010). I initially mapped rotation around the y axis to cursor movement along the x-axis, and rotation around the z-axis to cursor movement along the y-axis, thus creating a simple Raycasting implementation without control over the depth position of the cursor.

I have tested the above mapping in a short pilot study with four participants. Participants were asked to move the cursor from its initial position at the center of the control space into a red sphere that was placed at a random position. Observation of user interaction revealed that this simple mapping does not perform as well as it did in the case of Smartcasting. In Smartcasting the centre of rotation of the wrist is very close to the centre of rotation of the phone, thus the ray controlled by the smartphone covers 180°angular distance by travelling a relatively short distance in space. This is not the case for Watchcasting, that relies on arm movements. For a smartwatch mounted just behind the wrist, the centre of rotation is either the elbow or in the shoulder (when the user's hand is straightened) – both positioned relatively far from the centre of the smartwatch. Thus, to rotate the smartwatch by 180°, the user's entire arm (or shoulder) needs to rotate 180°. Such a wide movement of the arm in space results in a much slower rotation of the ray and, consequently, the cursor on the screen, than was the case in smartphone-based Raycasting.

5.1.1.1 Increasing the Speed of Ray Movement

To improve the performance of Watchcasting, I piloted increasing the gain of the ray's angular movement relative to the arm's movement. I tried multiplying the input angles by a constant factor, so that smartphone rotation by angle α would result in cursor movement of $k * \alpha$. While this simple solution solves the problem of reach, it reduces ray movement precision by a factor of k.

Another possibility I tested was to place the smartwatch closer to the hand, before the wrist, so that wrist movements would rotate it. However, in this position the smartwatch is not well attached to the hand and tends to shift around its x axis during the interaction. It is also impractical, since it is unrealistic to expect that users will change the way they wear their smartwatches.

5.1.1.2 Pilot Study

Despite the above concerns, the pilot study suggested that smartwatch hardware is capable of facilitating simple 3D pointing. For many ubiquitous interactions high ray speed may not be necessary. Moreover, pilot users found the ability to interact in a freehand manner both enjoyable and potentially useful. The fact that interaction did not require them to take out their smartphone was also found to be valuable in the context of interaction in a ubiquitous environment.

Given the positive results of proof-of-concept Watchcasting implementation, I embarked on extending Watchcasting to include the manipulation of depth, while preserving the freehand nature of the interaction.

5.1.2 Depth Cursor on a Smartwatch

Recall that in Chapter 4, I described implementation of a Depth Cursor in Smartcasting that allows for pointing and selection in fully addressable 3D space. Such an implementation requires an additional degree of freedom to control the depth position of the selection tool. In the case of Smartcasting, depth is controlled by a finger movement on a touchscreen.

In the initial design of the Depth Cursor for Watchcasting, I piloted a two-handed technique with the same four participants, where movement along the z-axis was realized on the smartwatch's touchscreen and controlled by the other hand. I tried to provide touch input via both dominant and non-dominant hands, however pilot participants preferred to use non-dominant hands for the touch input. The pilot also revealed that a smartwatch touchscreen is too small for absolute mapping, resulting in low precision cursor movement along the z-axis of the control space. This issue arises because the smartwatch touchscreen is three to four times smaller than the smartphone touchscreen, resulting in a three to four times lower precision of touch input for any given user, regardless of each individual user's precise touch ability.

5.1.2.1 Clutching on a Smartwatch's Touchscreen

The low resolution of touch input in Watchasting can be addressed by clutching. As opposed to my initial, absolute mapping of the z-axis onto the smartwatch's touchscreen, with clutching each swipe across the smartwatch's display moves the cursor further in the direction of touch movement on the z-axis, instead of returning the cursor to its initial position on the z-axis. Clutching can dramatically increase the precision of movement along the z-axis, while significantly slowing down the cursor movement time along the ray due to the additional in-air clutching movements required, most importantly, due to the three to four times smaller size of the smartwatch's screen vs. the smartphone's screen, clutching needs to be repeated two to three times in order to move the cursor by the distance of a single swipe across the smartphone's screen (given the same mapping gain).

5.1.2.2 Using Hand Twist to Control Depth

In search of faster depth interaction I tried to map cursor movement to rotation around the smartwatch's x-axis, corresponding to the clockwise and counterclockwise twist of the user's arm, where twisting the arm clockwise moves the cursor away from the user, deeper into the display, and twisting the arm counterclockwise moves it toward the user. The advantage of this approach is that it results in a freehand technique that only used one hand – a desirable form of interaction in some computing environments, such as a public space (Peltonen et al., 2007).

In a pilot study I have experimentally discovered that users can comfortably twist their hand by a maximum of 100°. When the elbow is not bent users can also twist their arm counterclockwise by about 80°, but when the elbow is bent, the counterclockwise movement is very restricted. Thus, in practice, arm rotations are restricted to a 100°twist. In my implementation, I have uniformly mapped twist angles to depth position, 0°mapped to $z = 0$, and 100°(clockwise twist) mapped to the maximum depth.

The pilot study also revealed that such absolute mapping of hand twist to depth position provides precision of movement along the z-axis similar to that of Smartcasting using a touchscreen, while being performed faster than finger movement on the touchscreen. If higher precision is required, relative mapping can be used. In this case, a clutching movement needs to be implemented. One way to achieve that is to interpret slow arm rotation as a movement of the cursor along the z-axis, while fast rotation in the opposite direction is interpreted as a clutching movement. The exact thresholds for speed of clutching vs. movement on the z-axis could be determined experimentally.

A side effect of twisting the arm to control depth is that it affects the inertial system of the smartwatch (the position of its coordinate system versus the coordinate system of the control space) with the z and y axes no longer aligned with the x-axis and the y-axis of the fixed-origin ray. This fact has to be taken into account when mapping cursor movement to smartwatch rotation.

5.1.3 Triggering Selection

In Smartcasting, selection was confirmed by releasing the finger from the smartphone's touchscreen ("reversed click"). Similarly, in Watchcasting, selection could be confirmed with a touch of the smartwatch's touchscreen. Such a solution introduces the second hand to the interaction. I have piloted a touch-based selection confirmation, where selection is performed by tapping the smartwatch's touchscreen, with the same four users that participated in the other parts of the iterative design study. The pilot revealed that such a technique was found to be fatiguing for users. Selection confirmation was physically difficult for targets placed at the edge of the display, when the dominant arm was bent away from the user's body. In this position, reaching for the smartwatch touchscreen with the non-dominant hand is inconvenient at best, and becomes impossible at angles close to 90°.

This issue, similar to the problem of the wide arm movements required to control the ray angle, could be addressed by increasing the gain on the mapping of the smartwatch rotation to the rotation of the fixed-origin ray, such that smartphone rotation by Δ would result in a cursor movement of $k * \Delta$. With sufficiently large k the arm would not need to be rotated by an inconveniently large angle. However, as discussed above, this solution could negatively affect the angle movement precision by a factor of k.

Commercial SDKs for smartwatch application development, such as Android Wear, suggest using voice commands for smartwatches equipped with a built-in microphone. Watchcasting with selection confirmation realized through voice commands could be implemented in a manner proposed by Bolt (1980), in a classic study "Put-that-there". However, in computing environments, high levels of auditory noise can be expected, to which voice commands are susceptible.

Another possibility I piloted is to use a hand gesture to trigger selection. I piloted various gestures, among which users most preferred the *poking* and *grabbing*. I initially chose the grabbing gesture because it had been used successfully by Haque et al.(2015) in their 2D pointing study of the Myo armband (www.myo.com). Myo uses a myoscope to recognize finger movements and thus allows for recognition of a grabbing gesture, even when the gesture is performed slowly. Smartwatches are not equipped with a myoscope, and thus must rely on other sensors for gesture recognition, such as gyroscopes, magnetometers, and accelerometers. For a high recognition rate I found that the grabbing gesture had to be well pronounced, so as not to be confused with fast, non-grabbing hand movements. Yet, such a pronounced grabbing gesture results in a strong Heisenberg effect, because of rapid movement generated through the fist shaking the smartwatch.

Better results were achieved with the poking gesture; while it could be equally well recognized from the analysis of the sensor data, it had a less pronounced Heisenberg effect. Moreover, the direction of the cursor displacement at the moment of poking was uniformly upwards, thus allowing for easier correction of the effect. Thus, among the gestures tested, the poking gesture seems best suited for the Watchcasting, therefore I decided to use the poking gesture as a trigger of selection.

5.2 Empirical Validation

In developing my experimental design, I hypothesized that Watchcasting would demonstrate performance comparable to Smartcasting. To verify that, I empirically compared Watchcasting to both versions of Smartcasting (smartwatch-based Raycasting and Depth Cursor). Additionally, given that commercial smartwatch technology is relatively novel, it can be expected that smartwatch sensor capacities will improve in the next generations of smartwatches. For that reason, I also compared Watchcasting with the Raycasting and Depth Cursor functions implemented on the Myo armband (www.myo.com) that is a specialized arm-mounted input device. I expect that the quality of sensors in the Myo armband today may find its way into off-the-shelf smartwatches in the near future.

I identified these two input devices, the smartphone and the Myo armband, as the most appropriate devices for the comparative study I wanted to perform, given that the goal of my research is to provide 3D pointing techniques with mobile and wearable devices.

To simplify the comparison of performance between three different devices, of which one provides a touch-based selection confirmation, and the other recognizes gestures from a EMG read of muscles, I decided to use a dwell time of .5s. In practical applications, a dwell time may not be the best solution for confirmation of selection due to the Midas effect (Velichkovsky et al., 1997), that is an unintentional selection of object resulting from leaving the cursor pointed at an object for the period of a dwell time without the intention of selecting it. But in an experimental setting this decision allowed for separation of the time of the selection phase from the time required to confirm the selection, and it ensured that for each technique the selection confirmation time was constant.

5.2.1 Implementation Details

I used a 50-inch projection screen on which a 3D environment was rendered using a Macbook Pro at 60fps over HDMI 1.4a at a refresh rate of 60Hz using an NVidia GeForce GT750M and an Intel i7 processor with 16GB RAM. The same workstation also run a node.js server written in JavaScript responsible for relaying messages from the smartwatch and the smartphone to the 3D environment.

For Watchcasting input, an LG G Watch R smartwatch was used, first transmitting sensor and touch events at 10Hz over a bluetooth connection to an Android Nexus 5 smartphone and then over the local 802.11n wireless network to a PC workstation. The relay of messages via Android phone is necessary due to the architecture of the smartwatch and the Android Wear SDK, as it does not provide direct network connectivity. The client application was written in Android Wear SDK 20 and the Android application was written in Android SDK 20.

For Smartcasting input an iPhone 5 transmitted sensor and touch events at 10Hz over a local 802.11n wireless network. The client web application was identical to the one used in Smartcasting experiment, presented in the previous chapter. It was written in html and JavaScript.

The Myo armband transmitted accelerometer and gyroscope data over a bluetooth connection at a sampling rate of 50Hz, without relaying the messages through the node.js server. During the course of the experiment, participants were standing 3m in front of the projection screen, which was centred vertically and horizontally in relation to the participant's eye line.

5.2.2 Participants

11 participants (six males, three females) participated in the study, whose ages ranged from 19 to 27 ($\bar{x}$ = 23.2). All 11 participants were right-handed and worn the watch on the dominant hand. Two participants were unable to complete all blocks of the experimental trials (one participant gave up participating in the experiment in the middle, the other was excluded due to a technical issue with the Myo armband) and their data was excluded from my final analysis. Each participant received $10 remuneration. Participants were recruited from a local university.

5.2.3 Experimental Design

I used a 3 (INPUT DEVICE) × 2 (TARGET VISIBILITY) × 2 (TARGET SIZE) within-subjects design. The study utilized two independent variables: target visibility and target size. Targets with either 'small' (0.5°) or 'large' (1.0°) sizes provided two levels of index of difficulty based on Fitts's Law (MacKenzie, 1992). My experimental design can be summarized thus:

9 Participants

× 3 Input Devices: Smartwatch, Smartphone, Myo armband

× 2 Target Visibility: Occluded, Non-Occluded

× 2 Target Size: Small or Large

× 8 Repetitions

For a total of 864 trials.

5.2.4 Experimental Task

Participants performed the 3D selection task discussed in Chapter 3. For each trial, participants first selected a start object in the center of the projection screen. Once the ray (or Depth Cursor) entered the start object, the start object would disappear and the destination target, another object on the projection screen would become selectable. Participants moved the ray to intersect the target. To select the goal target, participants released their finger from the touchscreen (for Smartcasting) or performed a dwell with a smartwatch or a Myo armband. Once the ray (or Depth Cursor) entered the goal target, the task ended and the screen was reset. If the user accidentally selected a distractor, the distractor's colour changed to light blue to indicate an error which was recorded by the system. However, the participant could continue until the goal target was successfully reached or a timeout of 45 seconds elapsed.

For trials where occluded targets were present, a Depth Cursor implementation was provided for all three input devices: a smartphone, a smartwatch and Myo armband, and participants were required to reduce or extend the length of the ray to reach the start and goal objects. The Depth Cursor was manipulated by moving a finger on the touchscreen for a smartwatch and by twisting an arm clockwise and counterclockwise for a smartwatch and Myo armband. Similarly to Smartcasting's occlusion management mechanism, if the goal target was occluded, the distractors in front of the Depth Cursor that were intersected by the ray disappeared.

5.2.5 Procedure

As in the Smartcasting study, participants were asked to complete a brief demographic questionnaire. Before the experimental trials, each participant was briefed on the use of smartphone, smartwatch and Myo armband. Then the participants completed 20 training trials: 10 with the smartwatch, and 10 with the smartphone input device.Participants then completed six blocks (two for each input device) of 16 experimental tasks, corresponding to eight trials for each target size per block. The order of blocks was balanced with partial latin square design. After each block, participants completed a brief post-study questionnaire that examined perceived workload during their trials using the NASA Task Load Index. In total, each participant's commitment to the study was approximately 30 minutes.

5.2.6 Data Collection and Analysis

As in the Smartcasting study, all interactions made with the study software were logged to computer files. Selection time was the primary experimental measure, defined as the time taken

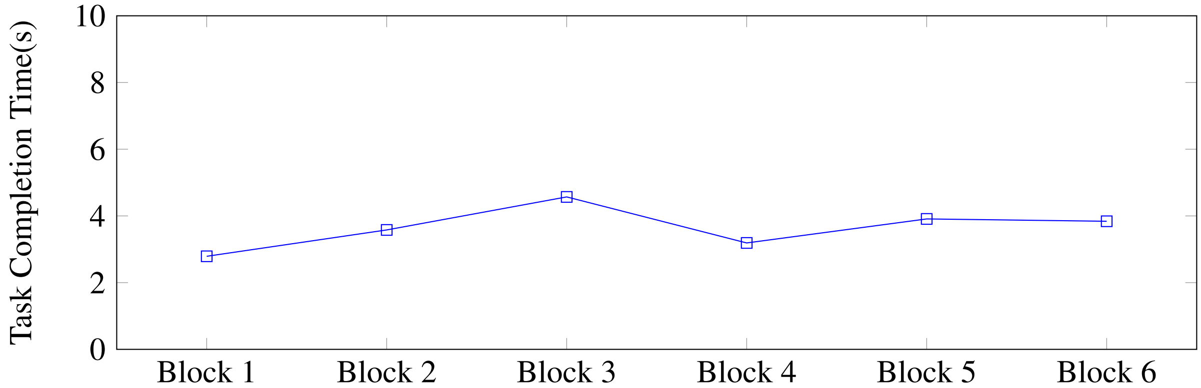

Figure 5.1: No learning effect was found between blocks

between entering the start position and reaching the destination target. NASA Task Load Index (TLX) (Hart and Staveland, 1988) data was transcribed into statistical analysis software.

Repeated Measures Analysis of Variance (RM-ANOVA) tests were conducted to examine differences in selection times between techniques and target sizes. Friedman tests were used to examine differences in perceived workload measures. An alpha- value of .05 was used for all statistical tests.

5.3 Results

5.3.1 Learning Effect

No significant difference was found between blocks ($F_{1,5} = 0.441, p = .745, \eta_p^2 = .052$), indicating no learning effect between blocks (Figure 5.1). Data did not satisfy the test of sphericity, thus the reported effect size used Huynh-Feldt corrected.

5.3.2 Error rates

For error rate defined as a confirmed selection of a distractor object, no selection resulted in a selection error.

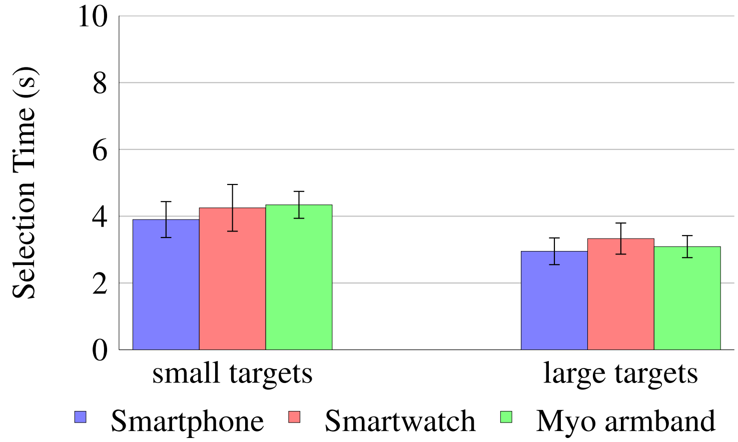

Figure 5.2: Mean selection times for Smartphone vs. Smartwatch vs. Myo armband for small and large targets

5.3.3 Performance

On average, participants completed each trial in $3.65s$ ($\sigma = .99$). The trial completion time excludes 0.5s dwell time for all input techniques. My analysis included all results, including those trials, in which selection tool entered (but not selected) a distractor. Experimental data was normally distributed ($sk = 1.78$).

5.3.3.1 Input Device

My analysis revealed no difference between the three input devices for target selection time ($F_{1,8} = .250, p = .782, \eta_p^2 = .030$). Smartwatch selections took on average $3.79s$ ($\sigma = 1.36$), smartphone selections took an average of $3.44s$ ($\sigma = 1.60$), and Myo selections took $3.72s$ ($\sigma = 1.01$) on average.

5.3.3.2 Target Size

As expected, my analysis revealed main effects (Figure 5.2) for destination target size ($F_{1,8} = 19.977, p = .002, \eta_p^2 = .714$), with small targets ($4.17s, \sigma = 1.167$) taking longer to select than large targets ($3.12s, \sigma = .936$). No interaction effect was found between target size and occlusion

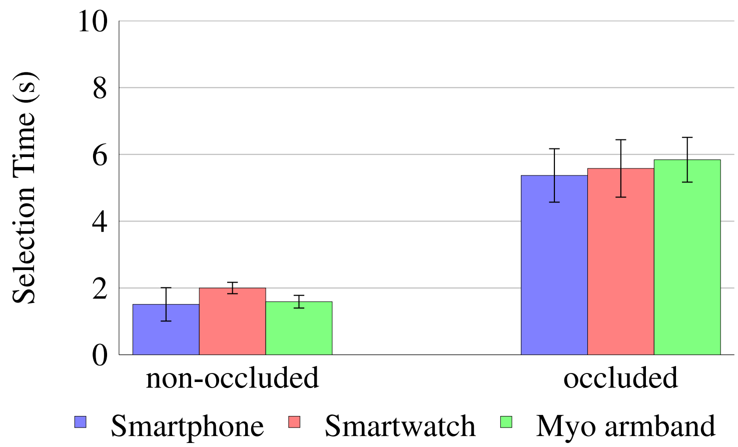

Figure 5.3: Mean selection times for Smartphone vs. Smartwatch vs. Myo armband for occluded and non-occluded targets

($F_{1,8} = 5.062, p = .055, \eta_p^2 = .388$). No interaction effect was found between technique and target size ($F_{1,8} = .191, p = .828, \eta_p^2 = .023$).

5.3.4 Occlusion

As expected, selection times for occluded targets were longer than for non-occluded targets ($F_{1,8} = 44.97, p \approx .000, \eta_p^2 = .849$). Occluded targets were selected in 5.60s on average (σ = 1.85), whereas non-occluded targets were selected in 1.70s on average ($\sigma = .282$). No interaction effect was found between the input device and occlusion ($F_{1,8} = .197, p = .823, \eta_p^2 = .024$). Results are summarized in Figure 5.3.

5.3.5 Perceived Workload

My analyses also revealed differences between input devices for perceived workload ($F_{2,7} = 5.499, p = .037, \eta_p^2 = .611$), where participants making selections with Smartcasting ($x = 7.39, \sigma = 1.48$) perceived less work than those making selections with Watchcasting ($x = 11.04, \sigma = .668$). However no other differences were found. As expected, I also found a main effect for occlusion on perceived workload ($F_{1,8} = 15.750, p = .004, \eta_p^2 = .663$), where occluded objects ($x = 11.84, \sigma = .814$) were reported as requiring more effort to select ($x = 7.583, \sigma = 1.193$) than non-occluded objects.

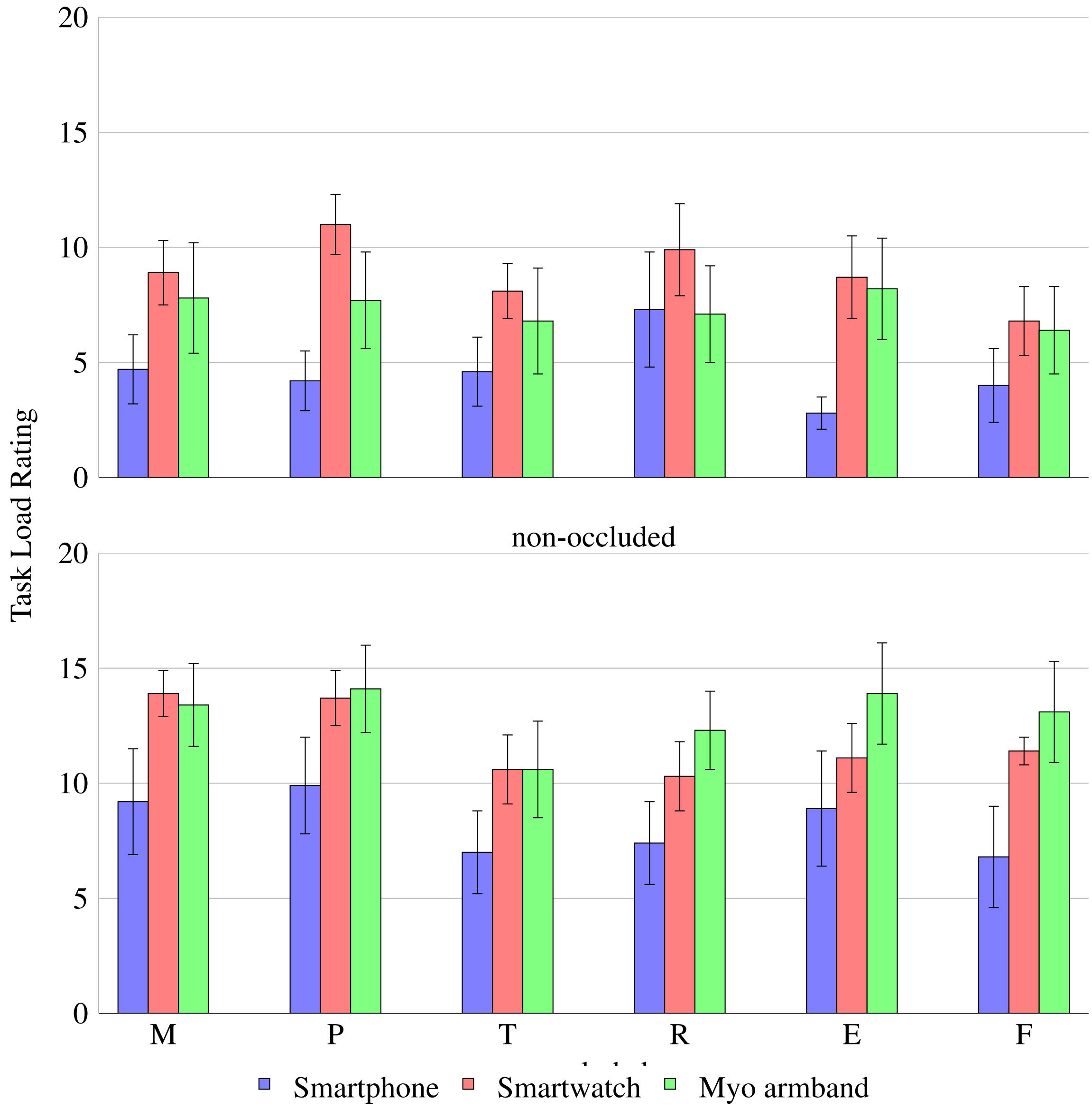

Figure 5.4: NASA TLX results for Smartphone vs. Smartwatch vs. Myo, with and without Depth Cursor: Mental Demand (M), Physical Demand (P), Temporal Demand (T), Performance (R), Effort (E) and Frustration (F) measures.

A complete summary of the NASA-TLX data is presented in Figure 5.4.

5.4 Discussion

5.4.1 Performance

The experiment results verified that Watchcasting performs on a par with both Smartcasting and the Myo armband. More specifically, my results demonstrate that Watchcasting effectively supports 3D pointing.

5.4.1.1 Selection Time and Accuracy

Even though Watchcasting performance, as expected, decreases with decreasing target size, for interaction in computing environments high precision selection may not be required, or even feasible, due to the distance of users from the screen. Small targets, even if they could be selected with high precision, such as those presented by Nancel et al. (2013) (4mm targets on 5.5m wide screen), may not be visible from this distance. Moreover, many applications such as gaming or modelling are target-aware environments. In such settings, using techniques such as Bubble Cursor may increase the effective width of the target, because the index of difficulty for selection depends on the size of the Voronoi regions (or the size of the cursor's bubble) that is a function of the (local) environment density, not the target size.

5.4.1.2 The Heisenberg Effect

Recall that in Smartcasting experiments the Heisenberg effect was not measured directly, but was indirectly inferred by comparing between techniques the rate of increase on selection time for smaller targets vs. larger targets. In the Watchcasting experiment the Heisenberg effect was expected to be much more pronounced, given that the same set of sensors (accelerometer, gyroscope) control the position of the cursor and the recognition of the selection gesture. Thus, to reduce the Heisenberg effect resulting from a pronounced hand gesture I have used a dwell time of $0.5s$ as the selection confirmation mechanism. To properly calculate the selection time, I have separated the selection task into two subtasks: the selection time, that is the time up to the moment of the cursor entering the target for the last time, before being selected, and the dwell time, that is the time from the moment the target was entered for the last time to the moment it was selected.

If I had used a gesture for selection triggering, given that Watchcasting uses the same sensors for the cursor movement and the selection triggering, the Heisenberg effect would have occurred. One way to address this issue is to use a different sensor. As described above, triggering selection by touch is one option, but it requires a two-hand technique. Another option is provided by the Myo armband. In addition to an accelerometer and gyroscope the Myo armband is equipped with a myoscope – a set of sensors that measure the electric resistance of muscles. The Myo SDK allows for recognition of hand gestures from this sensor, even if the gesture is performed slowly, thus it does not negatively affect the gyroscope and the accelerometer readings. To enable that, smartwatches could have a myoscope sensors built into the watch strap.

5.4.2 Occlusion and Depth

5.4.2.1 Occlusion Management

As discussed in Chapter 3 one of the main challenges of 3D interaction is selecting targets that are occluded by other objects. Watchcasting uses the same occlusion removal mechanism that was introduced in Smartcasting, with the difference that controlling depth by rotation was introduced in Watchcasting to replace controlling depth through touch input. The main problem lies in the fact that, as discussed earlier, a smartwatch's touchscreen is much smaller than a smartphone's touchscreen, thus reducing the precision of mapping touch events to cursor movements (without clutching).

5.4.2.2 Depth Identification

Watchcasting does not provide visual perspective cues that would facilitate depth identification. However, depth identification is indirectly suggested to user through the occlusion removal mechanism. When the user moves the Depth Cursor, any objects that the ray cuts through, that is any objects between the user and the Depth Cursor, disappear. By swiping the control space with a different length of Depth Cursor the user can build a mental model of the position of the objects in 3D space. This method provides depth identification cues precise enough to avoid user confusion. In dense environments the partial occlusion itself provides a depth cue: partially occluded objects are obviously further from the user than the occluding ones.

5.4.2.3 Target Disambiguation

Similar to Smartcasting, Watchcasting does not address the target disambiguation problem through design. However, it is important to note that, as shown by the experiment results, the selection of targets in a dense environment (the density was always constant across all experiments thanks to Voronoi alignment of occluders) was not slower for Watchcasting than for Smartcasting.

5.4.3 Perceived Workload

NASA TLX results confirms that Watchcasting produces higher perceived workload. Combined, these results provide some evidence that given the same number of degrees of freedom, a technique may benefit from using more than one modality (as is the case for Smartcasting) as compared to using single modality (as is the case for Watchcasting).

For 3D pointing with occlusion users reported differentrence in perceived workload between the smartphone, and smartphone and Myo. Given that selection times for occluded targets took on average over 5s, these results indicate the severity of gorilla-arm effect for a smartwatch and Myo armband, that are both input device requiring users to keep their hands raised. Participants preferred to make small wrist movements with a smartphone instead of raising their hand in a pointing gesture.

Although in the experiment the participants were standing, my pilot study revealed that the gorilla-arm effect may be less severe when the user interacts in a seated position, so that the elbow can be rested on a table or armchair and all rotation input happens within the forearm instead of the upper arm. As mentioned earlier, in this position rotation is limited to about 100°as opposed to 180°rotation when the arm is stretched, yet this precision may be sufficient for many interactions in computing environments, such as using a smartwatch as a remote controller.

5.5 Summary

Watchcasting demonstrates of how an off-the-shelf smartwatch can effectively facilitate casual 3D pointing, thus showing that techniques designed for mobile devices can be realized, with modifications, on wearable devices. The technique performed on a par with both specialized devices, the Myo armband, as well as the baseline Smartcasting.

However, use of arm-mounted wearables come at a cost of a strong Heisenberg effect and gorilla-arm effect. Thus, Watchcasting is a technique appropriate for short, casual interaction, or when users can rest their elbow on a support, such as an armchair; the technique cannot be recommended for prolonged use, such as a workspace. Yet, the advantage of Watchcasting, and wearable devices in general, is that users do not need to think of pulling out a mobile device – they are already augmented with a computing device that facilitates effective interaction with other computing devices nearby.

Although successful as the first step toward casual 3D pointing, neither Watchcasting, nor Smartcasting address all of the challenges listed in Chapter 3. Thus, in the next chapter, I present Tiltcasting, a smartphone-based multi-modal two-handed technique that addresses those challenges of 3D pointing that were not sufficiently explored by Watchcasting, Smartcasting, or any previous techniques that I am aware of.

Chapter 6

Tiltcasting

Smartcasting and Watchcasting verify that 3D pointing tasks can be realized with a smartphone or a smartwatch, achieving performance that is comparable to 3D pointing with a specialized input device. Yet, a universal 3D interaction technique for computing environments must perform well in any type of 3D environment, and thus has to address all 3D challenges. Thus, in designing the Tiltcasting metaphor, I address each of the challenges to 3D pointing listed in Chapter 3: selection of targets in target-agnostic environments, hand tremor, the Heisenberg effect, target disambiguation, selecting occluded targets, depth identification on non-stereoscopic screens, and high perceived workload. I iterated through a number of prototypes and conducted two empirical pilot studies. In particular, early prototypes of Tiltcasting explored three aspects of design: smartphones as a generic input device, degrees of freedom required for 3D pointing, and spatial correspondence (with cursor feedback, as opposed to original spatial correspondence) between a smartphone's touchscreen and the displays.

6.1 Designing Tiltcasting

The idea for the design of Tiltcasting came from an observation of users during the Smartcasting studies: after using Smartcasting for a few minutes, participants held the phone closer to their body and started tilting it as if manipulating a plane (Figure 6.2). I hypothesized that manipulating the smartphone as a plane reduces perceived workload because participants could hold the phone with both hands, close to the body, and rotate their wrists while accessing the touchscreen with their finger. This position is also typical when playing mobile games. This observation led me to contrast planar manipulations (i.e. Tiltcasting) with ray-based manipulation (i.e. Smartcasting). Thus before discussing in detail the design of Tiltcasting, I first report on my work on

2D planar interaction which, together with the lessons learned in designing the Smartcasting, led directly to the development of Tiltcasting.

6.1.1 Spatial Correspondence

Spatial correspondence (Pietroszek and Lank, 2012) targeting relies on a user's ability to map coordinates between two distinct surfaces (Figure 6.1). For example, artists, architects, interior designers, and engineers all engage in spatial correspondence targeting when beginning to create a painting, floor plan, or technical drawing where one surface (i.e. a subject, building, or room) is mapped to a corresponding replicate (i.e. a painter's canvas, blueprint, or sketch). Spatial correspondence is a component of spatial reasoning, and has been thoroughly examined in the literature, specifically with reference to enabling multi-device interaction (e.g. Gustafson et al., 2011). Tablets, the mouse, trackpad, and touchpad all make use of spatial correspondence. Based on the ubiquity of spatial correspondence, an initial "walk-up-and-use" interaction design was based on the premise that users could easily map coordinates on an attendant smartphone to those of a nearby large, public display; thereby enabling touch interaction on the public display.

In my previous work (Pietroszek and Lank, 2012), I conducted an empirical study to validate the use of spatial correspondence in the context of large display interaction in public spaces. I compared settings where participants selected targets both using a smartphone with the screen enabled (traditional targeting), and with the screen disabled but with touch events forwarded to the nearby large display (spatial correspondence). These conditions provided a contrast between use of the smartphone as a large display input device (spatial correspondence) to an ideal case where the user's attention is focused on the phone's display alone.

I found that spatial correspondence users were able to localize their targeting task to within 4% of the display area. The high level of spatial correspondence accuracy was quite surprising given the lack of visual feedback on the input device. For a generic smartphone plus large screen configuration, I reported that participants could target 25 distinct targets without any visual feedback on their smartphone device. Further, more complex interaction models that take advantage of relative finger position have been shown to enable more accurate selections (Holz and Baudisch, 2010). However, while spatial correspondence enables interaction with a small number of on-screen targets, many large display applications require 3D pointing in dense environments.

6.1.2 Final Design

Given spatial correspondence targeting results which show that absolute mapping of a smartphone touchscreen onto a large display can be performed with high accuracy and with small

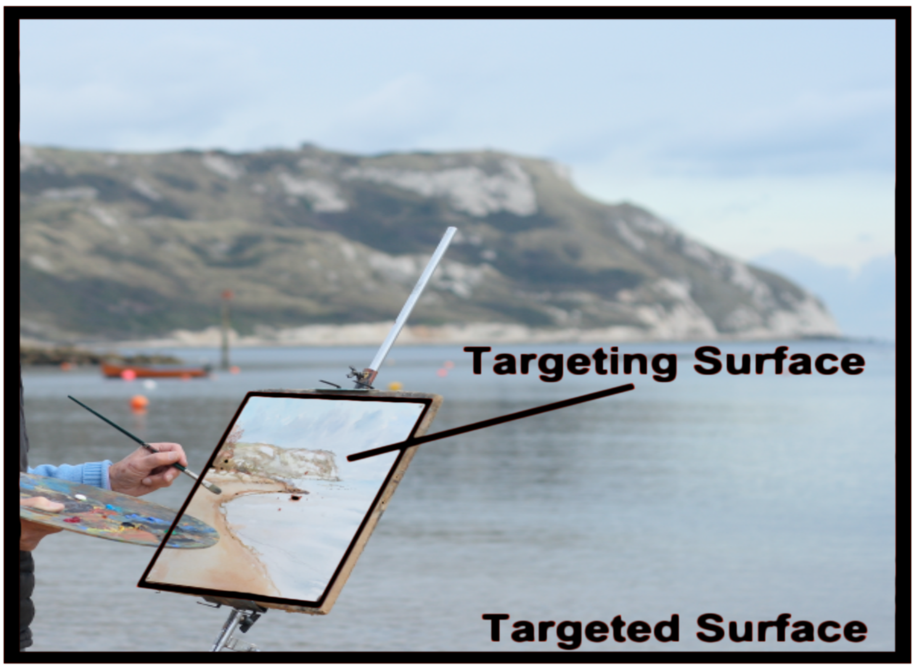

Figure 6.1: Many real-world tasks leverage spatial correspondence targeting. For example, landscape painting requires that an artist map the position of real-world features to their position on a canvas.

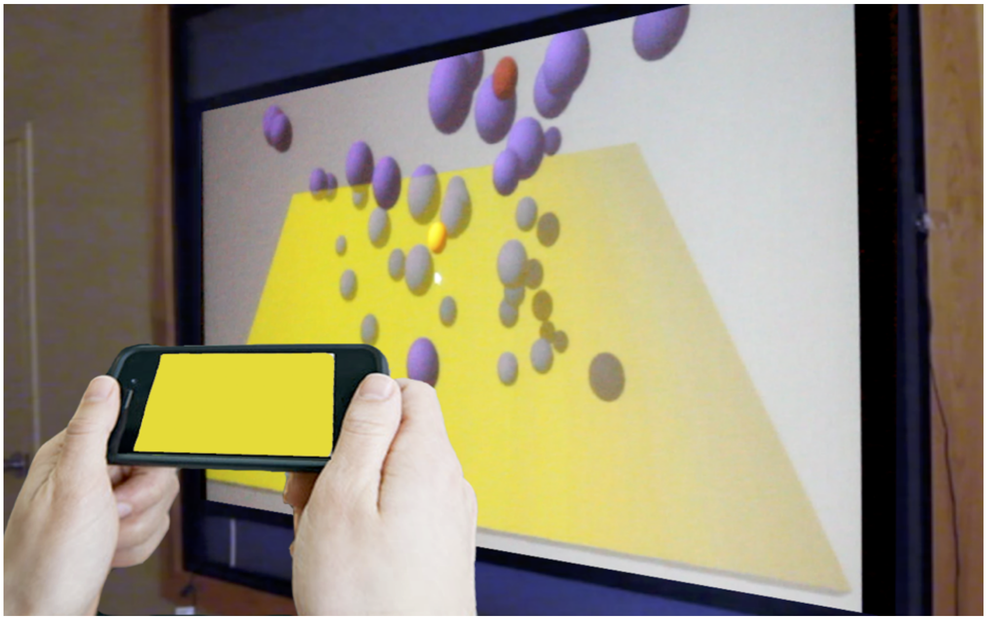

Figure 6.2: Tiltcasting defines a 2D interaction plane inside the 3D control space. The cursor moves on the plane. Only objects intersected by the plane can be selected.

bandwidth requirements (a condition important in ubiquitous deployments), I considered how the 2D plane interactions, such as spatial correspondence targeting, could be generalized for 3D interaction. Previous work (Valkov et al., 2011) tried extending a touchscreen interaction into depth dimension by mapping multitouch gestures to 3D space. I have instead considered using other modalities of the smartphone to control depth. In particular, I have considered whether the additional degree of freedom required to manipulate objects on the z-axis can be controlled via the smartphone's orientation.

6.1.2.1 Tiltcasting Metaphor

Tiltcasting metaphor defines a 2D interaction plane inside the 3D control space, dividing the control space into three distinct areas: 1) space behind the interaction plane, where objects are displayed to the user but are not selectable, 2) space intersected by the interaction plane, where objects are both visible and selectable by the user, and 3) space in front of the interaction plane, where objects are invisible and are not selectable by the user (Figure 6.3).

Users control the interaction plane within the 3D space via their phone's gyroscope, with rotations about the x-axis corresponding to a change in slope for the interaction plane. As the

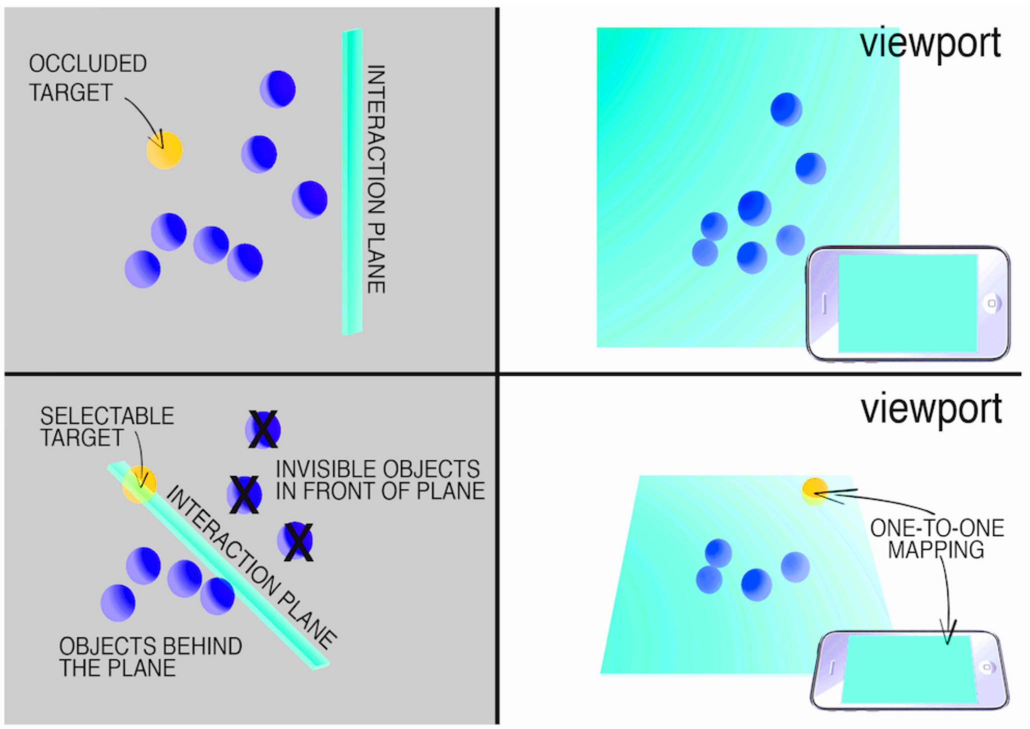

Figure 6.3: Tiltcasting's occlusion removal mechanism. a) Yellow sphere occluded by blue sphere, interaction plane is in vertical position. b) Tilting the interaction plane hides three blue objects, revealing the target.

phone rotates, the three defined regions encompass different areas of the 3D space, allowing users to view different parts of the space and to reveal occluded targets. When interacting with a target, users rotate their phone until the interaction plane intersects the target. Users select the target by touching the area on the touchscreen, which corresponds to the area of contact between the 3D object and the plane on the large screen.

Tiltcasting introduces a novel occlusion management mechanism: as a user tilts his/her smartphone, the interaction plane scans through 3D space, removes occluding objects (or object fragments), and reveals potential targets (Figure 6.3). For example, when a user lowers the angle of the phone, objects toward the lower portion of the interaction space may shift from being behind the plane, to intersecting the plane, to above the plane; in turn shifting from being visible, to visible and selectable, to not visible and not selectable. Since these interaction plane movements are an integral part of target selection, they do not add overhead to conventional interaction and provide a fast and accessible discovery mechanism. When the phone is held upright in a vertical position, users view the space in a similar way to the conventional viewport projection metaphor. A particularly useful feature is the ability to quickly scan the entire space by swiping the phone down and back up from the vertical position, revealing any occluded targets in the space.

It is important to note that in my design of Tiltcasting the aspect ratio of the interaction plane is the same as the aspect ratio of the smartphone's touchscreen. This constraint is not necessary, e.g. in case a of control space having different aspect ration (e.g. square).

6.1.2.2 Tiltcasting Limitations

By integrating occlusion management mechanism into selection process, tiltcasting provides a novel way of pointing in 3D environments. This feature comes at a cost of limiting the size of control space that can be addressed. While in Raycasting targets at any distance can be selected, Tiltcasting limits the control space to targets that can be intercut by the interaction plane. One way to address this limitation is to increase the size of the interaction plane, thus increasing the size of the control space. However, this method results in reducing the target size.

Another limitation of Tiltcasting is the existence of certain 3D environment configurations for which Tiltcasting may perform no better than ray-based techniques. For example, when selecting objects close to the bottom of the screen, the plane must be placed in a horizontal position. In such position, top part of the intersected target is hidden, while the lower part is shown. Moreover, objects closer to the user may occlude objects further from the user. This limitation can be addressed easily by moving the camera so that the plane is not in a horizontal position. However, a moving camera is not always beneficial as it disrupts the user's mental model of the 3D environment. Thus, alternatively, the occlusion removal mechanism used in

Smartcasting and Tiltcasting can be used for angles that are close to the horizontal position. In this case, objects that are below the cursor on a plane (or between the user and the cursor) would be made transparent, while objects above the cursor on the plane (or behind the cursor from the user's perspective) would stay visible.

Another limitation of the Tiltcasting is that the plane has a specific finite size, as opposed to an infinitely long ray in Raycasting technique. This limitation is typical for many 3D pointing techniques, such as Go-Go (Poupyrev et al., 1996) and Homer (Bowman Doug A. et al., 2004). In Tiltcasting, the size of the plane determines the size of the control space. Any object that cannot be reached by the tilting plane cannot be selected. One consequence of this limitation is that Tiltcasting should only be used when 3D environment's control space is finite. The amount of control space that is covered by the rotating plane can be manipulated by zooming in or out the 3D environment. On the other hand control space area can be moved by moving the camera together with the plane to a new position, thus redefining the control space.

6.2 Empirical Validation

Tiltcasting was motivated by creating a universal 3D technique that is capable of interacting with target-aware 3D environments as well as fully addressing 3D space. In developing my experimental design, I hypothesized that Tiltcasting would demonstrate performance gains for target-agnostic, occluded environments. However, to verify Tiltcasting's performance for more common use cases, I also chose to compare its performance against a target-aware technique, where selection can be performed without specifying target depth. In these cases, Tiltcasting's ability to fully address 3D space is disadvantageous, as it slows selection times. Thus, my validation provides both lower and upper bounds for target selection performance.

I empirically compared Tiltcasting to Smartcasting. For non-occluded target condition, I used target-aware Smartcasting with no depth manipulation, while for occluded target condition, I used target-agnostic Smartcasting with Depth Cursor (Pietroszek et al., 2014). I identified these Smartcasting implementations as the most appropriate baselines since their use of smartphones reduced potential hardware confounds and had already established performance levels comparable to common Raycasting implementations such as those on the WiiMote.

6.2.1 Apparatus

I used a 55-inch LG HDTV Cinema 3D circularly polarized stereoscopic display with a pair of passive circularly polarized LG glasses. Left and right eye images were provided at 60fps using a

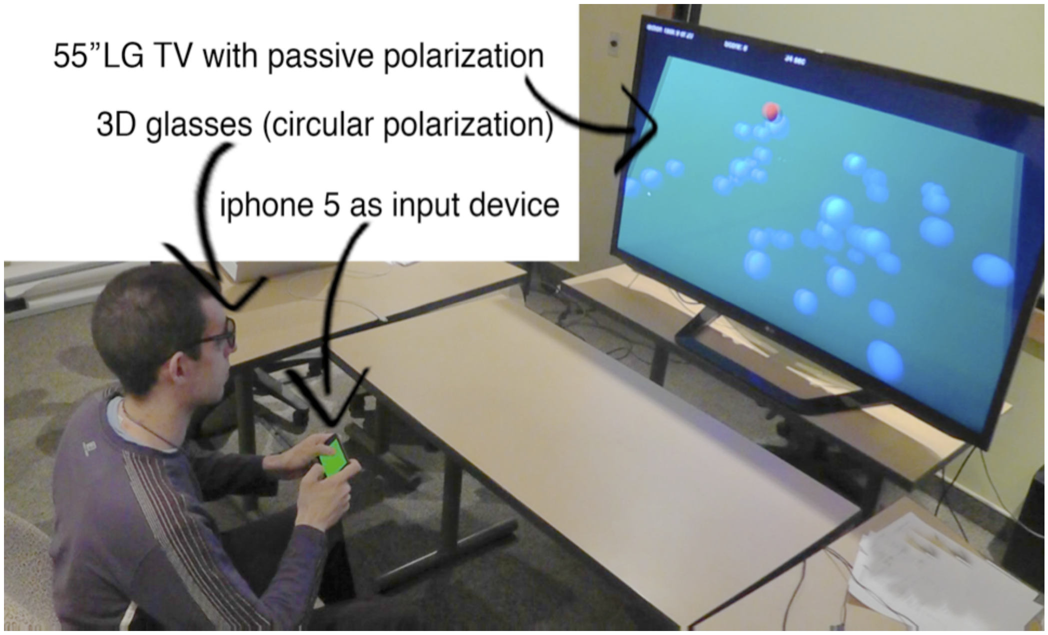

Figure 6.4: Experimental setup. 55-inch LG HDTV Cinema 3D circularly polarized stereoscopic display with an iPhone 5 used as an input device.

side-by-side HDMI 1.4a signal at a refresh rate of 60Hz using an NVidia GeForce GT760M and an Intel i7 processor with 16GB RAM. For input, an iPhone 5 transmitted gyroscope and touch events at 10Hz over a local 802.11n wireless network. The total latency of the input device over the network was $3ms$ on average, and never more than $10ms$. The resolution of the touch input was less than 0.07 mm, and was used to directly control the 3D cursor with an absolute one to one mapping, resulting in 1mm cursor resolution on the 55-inch display. During the course of the experiment, participants were seated 3m in front of the display, which was centered vertically and horizontally in relation to the participant's eye line in order to provide the optimal stereoscopic effect (Figure 6.4).

6.2.2 Participants

Seventeen participants (11 males, 6 females) participated in the study, whose ages ranged from 19 to 38 ($\bar{x}$ = 25.2). Sixteen participants were right-handed, 1 was left-handed. One participant was unable to complete all experimental trials; his data was excluded from my final analysis. Each participant received $10 remuneration. Participants were recruited from a local university, and were screened for their ability to order objects by depth on the stereoscopic display prior to participation.

6.2.3 Experimental Design

I used a 2 (INTERACTION TECHNIQUE) × 2 (TARGET SIZE) × 2 (OCCLUSION) 2 × (STEREO RENDERING) within-subjects design. The study utilized four independent variables: technique, target size, occlusion and stereo. Targets with either 'small' (0.5°) or 'large' (1.0°) sizes provided two levels of index of difficulty based on Fitts's Law (MacKenzie, 1992). The experiment environment was rendered in stereo (with participants wearing passive 3D glasses) or without stereo. Finally, targets were either fully visible or fully occluded upon starting the trial. For fully visible targets, no distractors occluded or partially occluded the goal target, whereas for the occluded version, the goal targets were hidden by the presence of distractor targets. My experimental design is summarized as:

16 Participants

× 2 Technique: Tiltcasting, Smartcasting

× 2 Target Size: Small or Large

× 2 Target Occlusion: Occluded or Non-Occluded

× 2 Stereo Rendering: Stereo or Non-Stereo

× 16 Repetitions

For a total of 4096 trials.

6.2.4 Experimental Task

Execution of selection task for Smartcasting was identical to the one presented in Chapter 4. For Tiltcasting participants first tilted the plane to the vertical position, then placed the cursor at a start object. The starting position, with the interaction plane vertical put the Tiltcasting at a disadvantage compared to Smartcasting, because the optimal initial position for the plane would be a 45°angle (the length of the worst case scenario path would be 45°). The vertical position of the plane ensured that the target was always fully hidden behind occluders in the case of occluded-targets study condition.

The start object would disappear once the cursor entered it, and the destination target – another object on the display – would appear. Participants tilted the plane to move the cursor, keeping the finger in contact with the touch screen, until it reached the target destination. If the destination target was occluded, tilting the interaction plane supported the discovery. Once the cursor entered the destination target the user could release the finger, thus confirming the selection. In such case, the task ended and the screen was reset. If during the pointing process the cursor collided with any distractor and the finger was released, the distractor's colour changed to magenta to indicate an error which was recorded by the system. However, the participant could continue the trial until the destination target was successfully selected.

Start point, target and distractors sizes were preserved from previous experiments. The size of the control space, that is the size of the interaction plane, was 140 cm by 80cm in 3D space. Using camera perspective settings, the projection of the 3D environment was scaled by a factor of 0.85 to its projection onto the TV screen resulting in physical plane size of 122 cm by 69 cm.

6.2.5 Procedure

Participants were first asked to complete a brief demographic questionnaire. Before the experimental trials, each participant was briefed on each technique, and screened for the ability to see depth. Then the participants completed five training trials. Participants then completed 8 blocks of 32 experimental tasks, corresponding to 16 tasks for each target size per Block. After the experimental trials, participants completed a post-study questionnaire that examined perceived workload. In total, each session lasted approximately 60 minutes.

6.2.6 Data Collection and Analysis

All gyroscope and touch interactions were logged to computer files. Selection time was the primary experimental measure, defined as the time taken between entering the start position and reaching the destination target. Selection time thus also includes the time taken for participants to visually search the display. NASA Task Load Index (TLX) (Hart and Staveland, 1988) data was collected post-trial.

Repeated Measures Analysis of Variance (RM-ANOVA) tests were conducted to examine differences in selection times between target sizes, target visibility, and depth rendering conditions. Friedman tests were used to examine differences in perceived workload measures. An alpha-value of .05 was used for all statistical tests.

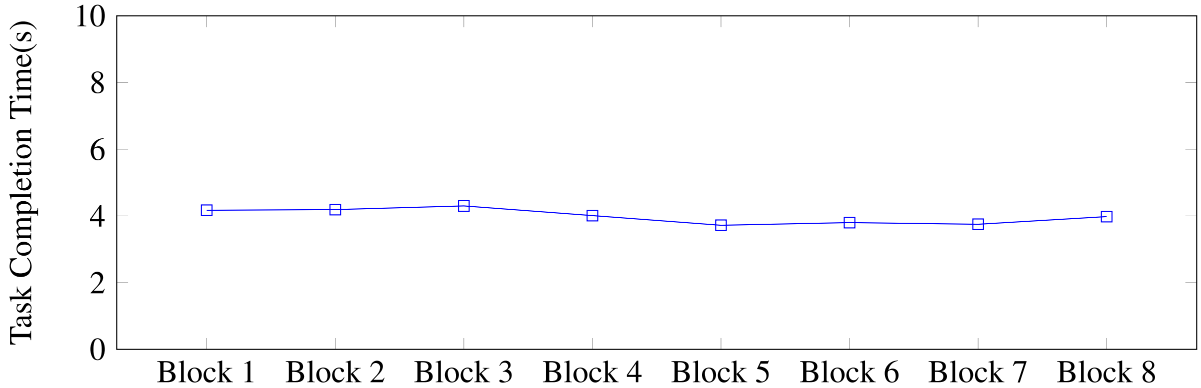

Figure 6.5: No learning effect was found between blocks

6.3 Results

6.3.1 Learning Effect

No significant difference was found between blocks ($F_{1,7} = 0.175$, $p = .936$, $\eta_p^2 = .012$), indicating no learning effect between blocks (Figure 6.5). Because the data did not satisfy the test of sphericity, the reported effect size was corrected with Huynh-Feldt.

6.3.2 Performance

Both previous experiments, discussed in Chapter 4 and 5, showed that the selection time in 3D pointing is strongly affected by the presence or absence of occlusion. This observation is confirmed by a significant difference between the selection time of non-occluded and occluded targets. For non-occluded targets participants completed each trial in $2.67s$ ($\sigma = 1.023$) on average, while for occluded targets, participants took $5.40s$ ($\sigma = 2.179$) on average to make selections. For that reason I separate my results between non-occluded and occluded techniques. That is, I compare *Smartcasting* and *Tiltcasting*, and *Smartcasting with Depth Cursor* and *Tiltcasting* separately. For each comparison, I evaluated three independent variables: technique, target size, and stereo rendering. Results are summarized in Figure 6.6 and Table 6.1.

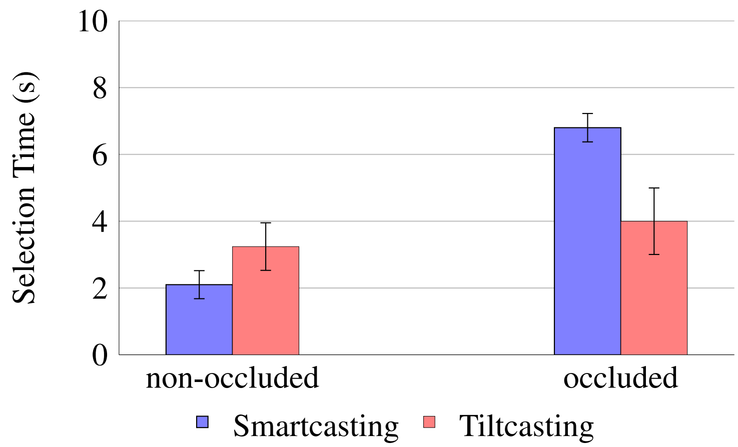

Figure 6.6: Mean selection times for Tiltcasting and Smartcasting

6.3.3 Occlusion and Depth

6.3.3.1 Visible Targets

Participants completed each trial in $2.67s$ ($\sigma = 1.023$) on average, and my analysis revealed a main effect for target selection time ($F_{1,15} = 94.499, p \approx .000, \eta_p^2 = .863$), where Tiltcasting selections took an average of $3.24s$ ($\sigma = .712$) and Smartcasting selections took $2.10s$ ($\sigma = 0.42$) on average. As expected, my analysis revealed main effects for target size ($F_{1,15} = 73.312, p \approx .000, \eta_p^2 = .830$), with small targets taking longer ($3.02s, \sigma = .668$) to select than large targets ($2.33s, \sigma = .416$). My analysis also revealed a main effect of stereoscopy on selection time ($F_{1,15} = 46.77, p \approx .000, \eta_p^2 = .757$), where selections made with stereoscopic rendering ($3.03s, \sigma = .660$) were slower than those with non-stereoscopic rendering ($2.33s, \sigma = .468$). An interaction effect was found between stereoscopic rendering and size ($F_{1,15} = 6.69, p = .021, \eta_p^2 = .031$), where selection for small targets was faster for non-stereoscopic rendering ($2.57, \sigma = .528$) than with stereoscopic rendering enabled ($3.47s, \sigma = .872$). For large targets the selection time difference was significant for stereoscopic ($2.58s, \sigma = .532$) vs. non-stereoscopic ($2.08s, \sigma = .424$) rendering. No interaction effect was found between stereoscopic rendering and technique ($F_{1,15} = .063, p = .805, \eta_p^2 = .004$).

6.3.3.2 Occluded Targets

For occluded targets, participants took 5.40s ($\sigma = 2.179$) on average to make selections. My analysis revealed a main effect for target selection time ($F_{1,15} = 51.781, p \approx .000, \eta_p^2 = .775$), where Smartcasting with Depth Cursor selections took an average of 6.8s ($\sigma = 1.7$), whereas Tiltcasting selections took an average of 4.0s ($\sigma = .922$). My analysis revealed a main effect of stereoscopy on selection time ($F_{1,15} = 68.62, p \approx .000, \eta_p^2 = .821$), where selections made with stereoscopic rendering ($5.95s, \sigma = 1.18$) were slower than those with non-stereoscopic rendering ($4.85s, \sigma = 1.012$). An interaction effect was found between stereoscopic rendering and size ($F_{1,15} = 7.37, p = .016, \eta_p^2 = .033$), where selection for small targets was faster for non-stereoscopic rendering ($5.2s, \sigma = 1.216$) than with stereoscopic rendering enabled ($6.6s, \sigma = 1.38$). For large targets the selection time difference was significant ($4.5s, \sigma = .952$) for stereoscopic rendering. No interaction effect was found for stereoscopic rendering and technique ($F_{1,15} = 1.284, p = .275, \eta_p^2 = .079$).

Condition	Occluded				Non-Occluded			
	Non-Stereoscopic		Stereoscopic		Non-Stereoscopic		Stereoscopic	
	Small	Large	Small	Large	Small	Large	Small	Large
Tiltcasting	3.55s (0.676)	3.18s (0.672)	4.80s (1.18)	4.49s (1.32)	3.15s (.748)	2.62s (.642)	4.05s (1.10)	3.15s (.648)
Smartcasting	—	—	—	—	1.99s (.361)	1.54s (.280)	2.88s (.924)	2.01s (.545)
Smartcasting with Depth Cursor	6.48s (2.07)	5.84s (1.48)	8.41s (2.18)	6.10s (1.44)	—	—	—	—

Table 6.1: Average selection times for Tiltcasting and Smartcasting (standard deviations in parentheses).

6.3.4 Error Rates

As discussed in chapter 3, error rate was defined as a confirmed selection of a distractor object. With such defined error, Tiltcasting and Smartcasting have similar accuracy in target selections: errors accounted for less than 0.5% of Tiltcasting trials, compared to 0.4% of Smartcasting trials.

6.3.5 Perceived Workload

My analysis did not reveal differences between Tiltcasting and target-aware Smartcasting for mental demand, physical demand, performance, effort or frustration. However, my analysis did reveal differences between techniques for occluded target condition (in both stereo and non-stereo), where participants expressed a preference for Tiltcasting in mental demand ($p \approx .000, \chi^2 = 15$), physical demand ($p = .001, \chi^2 = 11.267$), effort ($p = .001, \chi^2 = 11.267$) and fatigue($p = .001, \chi^2 = 10.267$). A complete summary of the NASA-TLX data is presented in Figure 6.7.

6.4 Discussion

6.4.1 Performance

My results demonstrate that Tiltcasting effectively supports 3D pointing. For occluded targets, selection times for Tiltcasting were on average 70% faster than those completed using Smartcasting with Depth Cursor. For non-occluded targets Tiltcasting performed only marginally worse than Smartcasting – a technique that benefitted from target-aware, non-occluded pointing, equivalent to 2D pointing (Bowman Doug A., 2002). Further, for occluded target selection, Tiltcasting was overwhelmingly preferred by participants, and was perceived as requiring less effort to use than Smartcasting with Depth Cursor. These results validate its design with respect to selection of targets in target-agnostic environments, reduction in hand tremor, elimination of the Heisenberg effect, disambiguation of targets, interaction with occluded targets, depth identification on non-stereoscopic screens, and perceived workload and the gorilla-arm effect. Below, I discuss each of these results in detail.

6.4.1.1 Target-aware vs. Target-agnostic 3D environments

The basic Tiltcasting design is target-agnostic and that implementation of Tiltcasting was used in the experiment. It is possible to improve the performance of Tiltcasting for target-aware 3D

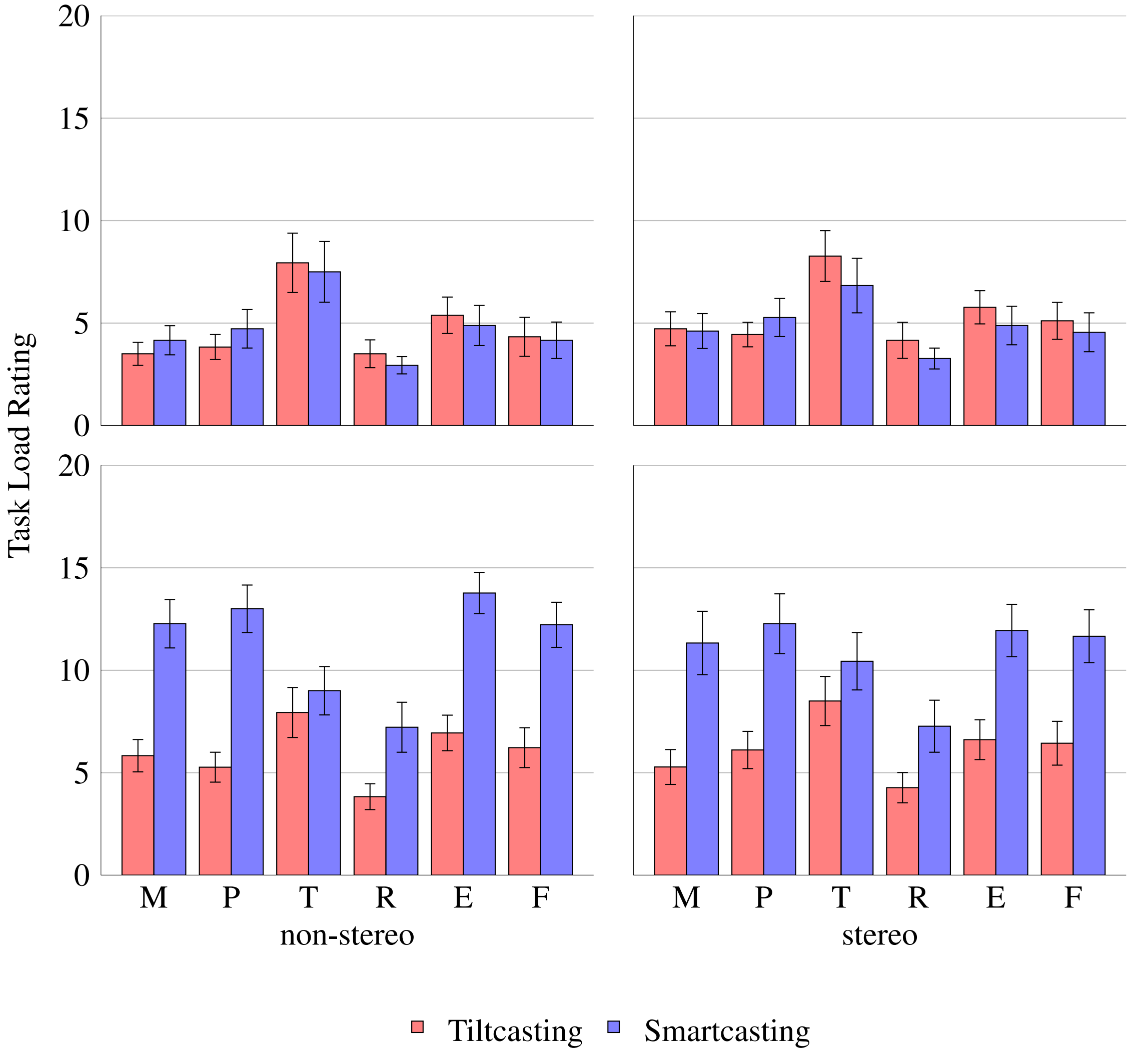

Figure 6.7: NASA TLX results for Mental Demand (M), Physical Demand (P), Temporal Demand (T), Performance (R), Effort (E) and Frustration (F) measures. For target-aware techniques, mental demand, physical demand, effort and frustration are significantly lower for Tiltcasting as compared to Smartcasting.

environments by implementing techniques such as 2D Bubble Cursor (Grossman and Balakrishnan, 2005) that would divide all targets intersected with the interaction plane into 2D Voronoi regions. Alternatively, Tiltcasting could be extended with a 3D Bubble Cursor implementation, by allowing for selections of targets *effective width* (Vanacken et al., 2007), that is the volume of its Voronoi regions. Yet, while these improvements may increase the performance, they complicate the technique and potentially increase the learning curve required. Given that Tiltcasting performs almost on a par with target-aware Smartcasting, I decided not to include such enhancements.

6.4.1.2 Hand Tremor and the Heisenberg Effect

Analyses of the interaction effect between technique and target size provides evidence that Tiltcasting was less susceptible to both than Smartcasting. Selections made using Smartcasting increased by 1.6*s* for small targets, while selections made with Tiltcasting increased by only 0.34*s*. Also, I have observed participants struggling to keep their hands stable when selecting small targets using Smartcasting, while no such behaviour was observed with the use of Tiltcasting. I attribute the reduced impact of the Heisenberg effect to a number of my design decisions, including: restriction of plane movement to 1 DoF and low-pass filtering of gyroscope readings. Finally, Tiltcasting primarily addresses the Heisenberg effect through the adoption of a two-handed technique instead of the more conventional one-handed pointing paradigm. The use of two hands adds stability to the user's pointing device, and reduces the occurrence of jitter upon target selection, while any jitter that does occur is reduced through low pass filtering of the smartphone's gyroscope input.

Evidence of the reduction of hand tremor in case of Tiltcasting vs. Smartcasting can be deducted from the accuracy-to-performance ratio. On one hand, my error rate results suggest that Tiltcasting and Smartcasting have similar accuracy in target selections: errors accounted for less than 0.5% of Tiltcasting trials, compared to 0.4% of Smartcasting trials. However, keeping in mind that the participants in my study were instructed to perform the task as quickly as they can, but *accurately* and given that the Tiltcasting selection time for smaller and occluded targets is much higher when the error rate is the same, it is possible to conclude that Tiltcasting reduced the number of times users aimed more than once at the target due to hand tremor. Evidence of that is provided by examining the number of times the cursor enters and re-enters the target in Tiltcasting vs. Smartcasting. In my experiment, for small occluded targets the Tiltcasting cursor re-entered the target 3.54 times less often than with Smartcasting.

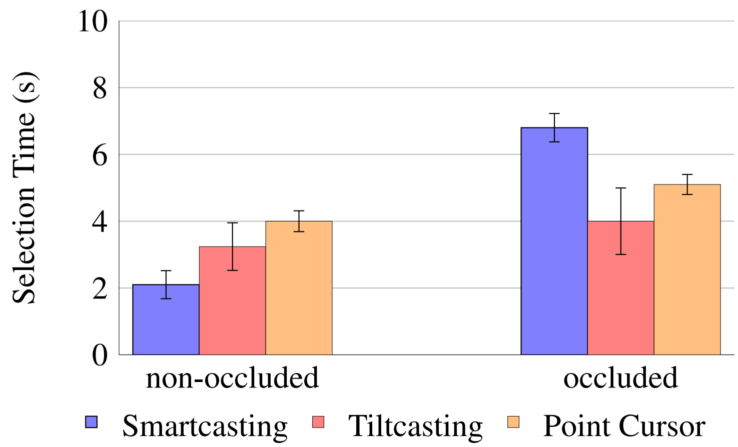

Figure 6.8: Meta-analysis of mean selection times for Tiltcasting, Smartcasting and Vanacken's Point Cursor

6.4.2 Meta-analysis

As discussed in Chapter 3, experimental evaluation of Tiltcasting vs. Smartcasting closely replicated the evaluation of magnetic-tracker-based techniques presented by Vanacken et al. (2007). Although the target size and the distance from the start point were in my experiment larger than the size and the distance in Vanacken et al.'s, their proportions were identical, resulting in the same Fitts's Index of Difficulty. However, two of three techniques presented by Vanacken et al, Depth Ray and 3D Bubble Cursor, are target-aware techniques and their "effective" width was larger than their actual width. Consequently, Fitts's Index of Difficulty for these target should have been calculated based on the "effective" width of the targets. Therefore, comparing performance of my technique with these two techniques is not possible. In the evaluation of the third technique, Point Cursor, the width of the target is the same as the "effective" width of the target and thus the comparison can be performed.

The average selection time for Point Cursor was 4.59s as compared to 3.62s for Tiltcasting. Vanacken et al. does not report exact selection times per target size and occlusion conditions; they can only be read from the bar chart. Selection of occluded targets took approximately 5.1s in average for Point Cursor, while it took 4.0s for Tiltcasting. Selection of non-occluded targets took in average approximately 4.0s for Point Cursor and 3.24s for Tiltcasting (Figure 6.8). Although the differences significance cannot be verified without access to Vanacken's et al. input data, Tiltcasting seems qualitatively faster than Point Cursor for the same Fitts's Index of Difficulty. However, Vanacken et al. reported no trial with selection error for Point Cursor, while

Tiltcasting selection had error rate of 0.4%.

6.4.3 Occlusion and Depth

6.4.3.1 Occlusion Management

Tiltcasting provides effective support for both occluded and non-occluded targets. I compared a single Tiltcasting against both Smartcasting implementations, with and without Depth Cursor. Tiltcasting also provided consistent selection times, averaging 3.6s regardless of whether the target was occluded. Further, two elements of my experimental design emphasize the importance of these differences. First, my experimental design target selection times include only the time taken to visually identify and the time taken to select targets. Often the visual search for occluded targets, called 'discovery' or 'exploration', is the most time-consuming part of a selection task (Carpendale et al., 1997). Thus, by shortening the exploration phase I expected to achieve improvements in overall selection time.

Indeed, the performance loss between occluded and non-occluded conditions was relatively small, and in practice a 15% increase in selection time may be a worthwhile tradeoff when compared to a threefold increase for Smartcasting. Second, I chose to compare a single Tiltcasting implementation against two Smartcasting implementations: one optimized for selection of non-occluded targets, and one for occluded targets. This choice was made to ensure that I held Tiltcasting to a high standard when assessing its performance, but this does not reflect compromises that would be made in practice when selecting a single virtual pointing implementation for deployment in a ubiquitous environment.

6.4.3.2 Target Disambiguation

By limiting interaction to a 2D plane within the interaction space, Tiltcasting limits the need to disambiguate nearby targets to those targets that are intersected by the plane. Thus, my planar interaction space seems to potentially increase the likelihood of the target disambiguation problem occurring compared to Raycasting, where only targets intersected by a line would require disambiguation.

Yet, for smaller targets that are most prone to the target disambiguation problem due to hand tremor, Raycasting must also disambiguate among targets in a nearby 3D space. While uncontrolled angular movement of the ray resulting from hand tremor increases the selectable space around the ray on both the x-axis and the y-axis, similar uncontrolled hand tremor increases the selectable space only along the vector perpendicular to the interaction plane angle. Given the same amount of hand tremor, the volume of such added space is smaller for Tiltcasting than it is for Raycasting. In addition, as discussed above, the hand tremor for Tiltcasting is smaller than for Smartcasting, because the former is a two handed technique, thus indirectly reducing the potential for the occurrence of th target disambiguation problem.

6.4.3.3 Depth Identification

Tiltcasting assists with depth identification since the interaction plane provides a depth cue through the linear perspective. Since this linear perspective is a feature of the interaction technique, it is always present. Further, as Tiltcasting relies on spatial correspondence (Pietroszek and Lank, 2012) between the smartphone's surface and the onscreen interaction plane, users can efficiently select targets via their projection onto the interaction plane.

Participants in my study struggled with stereoscopic rendering, and on average it imposed a near 1s penalty on selection times regardless of interaction technique. Selections made with Tiltcasting for non-stereoscopic targets were faster than those performed with stereoscopy (p = .003), despite additional depth information being available to participants when using the stereoscopic display. These findings suggest that stereoscopic rendering does not provide an advantage, regardless of the technique being used. However, my analysis suggests that Tiltcasting may provide benefits for more difficult cases of the depth identification problem. 'Large' targets rendered without stereoscopy represented a worst-case scenario in my study, where the depth identification problem was always present. That is, the target was always visibly larger than the start position sphere, while being further away from the user in 3D space, thus inducing the depth confusion effect. Nevertheless, I observed no interaction effect between the target size and stereoscopic rendering for Tiltcasting ($\eta_p^2 = .101$), indicating that the depth confusion effect may be diminished by Tiltcasting's perspective cue. The effect accounted for less than 10% of the variance in my model. Further, these results were supported by a user preference for Tiltcasting in non-stereo trials.

6.4.4 Perceived Workload

As indicated by the NASA TLX results, selecting occluded targets with Tiltcasting significantly reduces mental demand ($p \approx .000, \chi^2 = 15$), physical demand ($p = .001, \chi^2 = 11.267$), effort ($p = .001, \chi^2 = 11.267$) and perceived workload ($p = .001, \chi^2 = 10.267$) as compared with Smartcasting – all four factors together indicating better user comfort in Tiltcasting over Smartcasting. Also, users' informal comments indicate their preference for Tiltcasting. For example, P4 commented that Tiltcasting is "the easiest because it was easy to move the plane and you did not have to (search for) the ball". When commenting on Smartcasting with Depth Cursor, P5 stated that it was "*Frustrating* because it was difficult to judge distance".

6.5 Summary

In this Chapter I have presented the design and evaluation of a novel 3D interaction metaphor called Tiltcasting that supports smartphone-based 3D pointing for ubiquitous displays. I validated Tiltcasting for use with both stereoscopic and non-stereoscopic displays, and found that it provides effective support for interaction with both occluded and non-occluded targets. Further, my validation suggests that Tiltcasting provides support for depth identification and effective selections. This chapter contributes a deeper understanding of 3D pointing, particularly in the context of the occluded target and depth identification problems. Tiltcasting leverages the ubiquity of mobile, personal devices to enable 3D interaction with ubiquitous displays, and reduces barriers to data use on affordable, accessible, and commercially available displays.

Chapter 7

Reflections on Designing 3D Pointing with Mobile and Wearable Devices

The three pointing techniques presented in my dissertation, Smartcasting, Watchcasting and Tiltcasting, show that mobile and wearable devices can facilitate seamless and efficient 3D pointing in computing environments. Experimental evaluations of these techniques allow me to conclude that mobile and wearable devices perform on a par with devices that were designed for distant pointing, such as WiiMote and Myo armband. Moreover, the design, implementation and evaluation of Tiltcasting shows how the many modalities of mobile and wearable devices can be leveraged to address the challenges for 3D pointing listed in Chapter 3: the speed-accuracy trade-off, occlusion and depth, and perceived workload.

However, in human-computer interaction design there are always two factors: the human and the machine. While I have presented evidence that the hardware of mobile and wearable devices (i.e. the machines) is sufficient for 3D pointing tasks, I have not discussed the impact of the human factor. Thus, the following section I discuss the individual abilities of users and the ways in which the variability of the human factor was taken into account in my design.

7.1 Human Abilities vs. Device Constraints

Not all users are equally able to control 3 DoFs (Hegarty and Waller, 2005). In some professional 3D environments (e.g. military flight simulators), an individual's abilities can be assessed, and users' skills can be improved by training (Waller, 2000), or input devices can be adapted to serve the individual needs of each user (Charness et al., 2004). This is not the case for computing environments, where no assumptions about users' abilities or skills can be made by the designer of the interaction technique. In computing environments the interaction must be universal and cannot exclude groups of users based on their abilities.

Do participants in my experiments differ in their abilities to simultaneously control 3 DoFs? Such a claim is implied by the results of my experiments: Smartcasting, Tiltcasting and Watchcasting, as reported in Chapter 4, 5 and 6. In all experiments the standard deviation for selection times is large, when controlling for visibility and target size for each input device, indicating differences between users. Previous research shows that human spatial abilities vary between individuals (Hegarty and Waller, 2005; Charness et al., 2004). I hypothesize that individual differences in selection times may be related to differences in human spatial abilities.

The question is how and to what extent the individual differences in 3D pointing performance using mobile and wearable devices should impact the design of a technique for 3D pointing. Given that individual differences between users exist, it is important to design universal techniques that work well for a wide range of users' abilities. In the design of my pointing techniques I used two approaches to increase users' comfort with the technique regardless of individual abilities. The first avenue was to remap common interaction metaphors that the mobile and wearable devices were designed for. The second avenue was to take advantage of the computing power of mobile and wearable devices to improve interaction by performing in real-time, and also within the device, optimizations that would improve the quality of interaction. Both approaches were used together in the design of the Tiltcasting.

7.1.1 Remapping Common Interactions

In Chapters 4 and 5 I show that engaging the multimodality of mobile and wearable devices seems to be effective in improving interaction with 3D environments. Yet, while Tiltcasting and Smartcasting both use the device orientation data and the touchscreen, Tiltcasting was shown to be significantly faster and it was perceived as being less demanding to use (Figure 6.7). Moreover, in certain situations using multimodality may hurt the interaction, because, for example, it engages a second hand in a way that is not comfortable for a user, as discussed in Chapter 5,

The approach I found effective, as evidenced by users' preferences for Tiltcasting, was to use the multimodality of mobile and wearable devices in a way that mimics the way that users interact with a smartphone when performing tasks unrelated to 3D pointing. I call this approach "remapping common interactions". For example, in Tiltcasting the control of the interaction plane and the cursor moving on the plane closely resembles the way that users play some mobile games on smartphones. Games such as car racing games or other tilt-controlled games are controlled by holding the smartphone with two hands, close to the body, in a landscape position and interacting with the game through touch input and tilt. By using the smartphone in the same way it is used for mobile gaming, I implicitly take advantage of the users' familiarity with the interaction.

7.1.2 Using Computing Power

As noted by Zhai (1998), many factors related to the input device matter for the quality of interaction, including its form factor and input sampling rate. For that reason, specialized input devices are designed for a specific, often single purpose, such as a mouse for pointing or a keyboard for text input. Interaction design for 3D environments can and often does directly influence the design of the input device. For example a Rocking Mouse (Balakrishnan et al., 1997) was designed specifically for 3D pointing as a result of the observation that the regular mouse lacks the third degree of freedom required for a control of 3D cursor's depth position.

The case is different for mobile and wearable devices such as a smartphone and smartwatch which are designed as a universal portal for communication and for access to information. The functionalities that the smartphone and smartwatches are designed for include: making phone calls, showing the time, running apps, playing mobile games, notifications, browsing the Internet, reading emails and ebooks, listening to music and audiobooks or watching video streaming content. Mobile and wearable devices are not currently designed with distant pointing in mind, although this could change in the future. Moreover, the sensors in the mobile and wearable devices are not optimized for distant interaction in a way that specialized input devices are. For example, gyroscope and accelerometer sensors do not provide data at constant time intervals (Derawi et al., 2010), complicating the mapping of such input to cursor (or ray) movement. Thus, an interaction design that utilizes mobile and wearable devices must be approached differently than the design of interaction utilizing a specialized device. While the latter can and should influence the form factor and the hardware used in the input device, the designer of mobile and wearable devices needs to work within the constraints imposed by the device. These are constraints that interaction design using mobile and wearable devices must be aware of and actively consider.

One way of addressing this constraint is to remap the purpose of the interaction. In addition to this approach, one may take advantage of the mobile and wearable device's built-in computing power. Off-the-shelf mobile and wearable devices are powerful enough to execute – in real-time – a software that performs sensor fusion and noise removal, or simple machine learning algorithms such as Dynamic Time Warping (Derawi et al., 2010). In fact, some advanced filtering is built into the mobile and wearable devices' operating systems or SDKs. While previous studies reported low precision and high error-rates for mobile-based pointing experiments (Boring et al., 2010), my dissertation shows that this is no longer an issue: modern sensor data is reliable and sufficient enough for fast and accurate 3D pointing, when sufficient software-based filtering and optimization is applied.

I used many such optimizations, without which my techniques would not be usable. For example, in the implementation of Watchcasting, given that I could only achieve a 50 Hertz average sampling rate for orientation data, and that data did not arrive at equal time intervals, I used bezier curve smoothing for the cursor movement so that, while it introduced additional latency, the cursor appeared to move smoothly and thus provided a user experience closer to the experience provided by a WiiMote or a Myo armband. Other optimizations I used included buffering the sensor sampling on the device, performing low-pass filtering, artificially introducing equal time intervals for sensor data, thus smoothing the orientation data before sending such optimized input data out from the device over the network, to the ubiquitous environment. In my experience such implementation details make a difference to the performance of the interaction technique and should be described in detail to enable reproducibility of experimental results.

The discussion about hardware limitations that affect individual users' ability to use 3D pointing techniques leads to the question about the value of research into design interaction techniques for state-of-the-art hardware that I discuss next.

7.2 State-of-the-Art and Future Enhancements

All techniques presented in my dissertation use off-the-shelf mobile and wearable devices, which may mean that research may need to be replicated and the results modified when the devices evolve. Nevertheless, careful assessment of "where we are today" is useful for two reasons. First, it allows us to disseminate the knowledge to a broader community of researchers that can then use it to facilitate in-depth, domain-specific studies for various branches of human-computer interaction. For example, my dissertation, shows that off-the-shelf mobile and wearable devices can serve as efficient input devices for 3D interaction, and this could result in more research and deployments of specific 3D environments for ubiquitous displays. Second, the assessment often points at deficiencies, and therefore, the improvements in design, hardware and software that will be needed in the future.

An example of deficiency that affected my studies but is already being addressed by the industry is that most smartwatches are not standalone devices but peripheral devices for smartphones. For interaction with computing environments this limitation means another layer or information-relaying that introduces additional latency, complicates the design, and limits connectivity in computing environments, e.g. by requiring the user to first connect his phone to a local wi-fi network before his smartwatch can engage in interaction. While the wearables development community has already pointed at this limitation and the industry has responded with the first standalone watches (Samsung Gear W, or LG Watch Urbane running WebOS and having LTE connectivity), my dissertation provides additional arguments why standalone connectivity of a smartwatch should be a standard rather than exception.

An example of a deficiency of the current mobile and wearable devices is a lack of built-in fine-grained positioning data in 3D space. Currently, it is not possible to continuously track the smartphone or smartwatch movement in 3D space reliably without augmentation of the environment (Sachs, 2010). While fine-grain position of the device is possible through Virtual User Concept (VICON) or Wi-Fi-based Positioning System (WPS), and even the quality of GPS location is improving, a built-in fine-grained position of the device, for example, through built-in magnetic trackers would be beneficial. Fine-grained 3D positioning of the device would allow for multi-device interaction that is proxemics-based, allowing for use of an mobile and wearable device as a remote controller for the Internet of Things. For example, a seated user could point at the light switch, then at the TV and then at the coffee-machine in the kitchen.

The most important deficiency of current computing environments is that it is difficult to connect securely yet serendipitously and to configure device ecologies. While technologies such as iBeacon and multicast DNS provide opportunities for such configurations, support for interaction in computing environments where all devices communicate and interact with each other in a seamless and "calm" (not requiring user's attention) way will require more research, especially with in-the-wild deployments. If Weiser's vision of "calm technology" (Weiser, 1991) is to be brought to reality, research is needed to provide a similar contribution to those of my thesis, not in interaction design, but rather in the design of communication protocols, software frameworks and hardware sensors that will together create a symphony of interconnected and inter-operating devices.

Chapter 8

Summary

8.1 Contributions

My dissertation work bridges a gap in the research literature: a lack 3D pointing techniques that leverage mobile and wearable devices for input. I have presented design and evaluation of three 3D pointing techniques: Smartcasting, Watchcasting, and Tiltcasting. I now discuss each of these three contributions.

8.1.1 Leveraging Mobile and Wearable Devices for 3D Pointing

In Chapter 4, I introduced Smartcasting, a novel 3D interaction technique that uses a smartphone as an input device. Smartcasting is a ray-based distant pointing technique that utilizes two modalities of a modern smartphone: the orientation sensors and the touch input for manipulation of a 3D cursor, including its depth position. Smartcasting implements basic Raycasting that is appropriate for non-occluded 3D environments and is equivalent to 2D distant pointing. It also implements a Depth Cursor that allows for fully addressable 3D pointing (accessing any point in a 3D control space) and provides an occlusion removal mechanism. I formally evaluate Smartcasting and verify that it performs on a par with Raycasting for specialized input devices, such as a WiiMote. I conclude that a smartphone can replace specialized input hardware for casual, everyday 3D interaction.

In Chapter 5 I introduce Watchcasting, that is a pointing technique implemented on a smartwatch. My goal in this Chapter was to assess to what degree a smartwatch can replace a Myo armband for use in 3D pointing. I have also assessed how smartwatch compares to a smartphone in terms of 3D pointing. The constraints imposed by using a smartwatch as input device, specifically the fact that it is a device mounted on an arm, lead to an interesting discovery that controlling depth by arm rotation does not slow down or lower the accuracy of selection compared to Smartcasting's depth control through a touchscreen.

In Chapter 6 I presented another novel 3D interaction technique, called Tiltcasting, that uses a smartphone as an input device. However, unlike Smartcasting, Tiltcasting leverages the smartphone as a two handed device. The added stability contributes to a performance gain of Tiltcasting in comparison with Smartcasting, as does the introduction of a spatial correspondence targeting paradigm into the design.

8.1.2 Enabling Interaction at a Distance

All three techniques presented in my dissertation enable 3D pointing at a distance. Smartcasting's design and empirical validation provides a baseline mobile-based 3D pointing technique and demonstrates that mobile devices can be used for casual "at a distance" pointing. Smartcasting also shows that specialized hardware is not necessary for casual 3D pointing, while Watchcasting provides a similar finding for a smartwatch.

This findings have important consequences for design of future techniques for casual interaction with 3D environments. First, the results of performance comparison with specialized input devices show that interaction at a distance can be efficiently realized with Bring Your Own Device (BYOD) paradigm. Second, the findings show that interaction at a distance can be realized with current mobile and wearable technology allowing for deployment of 3D content for out-of-arm's-reach displays. One example where such deployment could be beneficial is a movie theatre, where interaction could be realized by audience from a seated position. My dissertation provides the evidence that the first step of such interactions, that is target acquisition, can be realized on current mobile and wearable devices.

8.1.3 Identifying and Addressing Challenges for 3D Pointing

In Chapter 3 I described challenges for 3D pointing. Starting with the issue of tradeoff between accuracy and speed, I analyze how this issue differs depending on whether the 3D environment is target-aware of target-agnostic. I also show how hand tremor (Myers et al., 2002) and the related Heisenberg effect (Bowman Doug A. et al., 2001) both affect selection time. I have listed previous work that addressed the issue.

I have paid special attention to the problems of occlusion and depth that are related to users' perception of 3D environments. I identified the problem of occlusion – when the target cannot be reached, because it is hidden behind (or within) another object – as one of the main problems of 3D pointing. Related is the target disambiguation problem, when the target can be reached, but cannot be selected without a disambiguation mechanism. I also explained the problem of depth identification that is specific to non-stereoscopic presentations of 3D environments and which refers to a user's inability to determine the z-position of a 3D target.

I have also elaborated on the perceived workload (and the related gorilla-arm effect). I showed how 3D pointing techniques often result in high perceived workload and gorilla-arm effect (Schultz, 1988), and discussed how perceived workload can be quantitatively measured. Additionally, I have discussed a design of formal experiment that evaluated perceived workload and performance of 3D pointing in occluded and non-occluded conditions.

8.1.4 Impact of Contributions

Together, three contributions of my dissertation support the thesis that mobile and wearable devices are ready to be used for 3D pointing. More specifically, this finding opens new possibilities for casual pointing and interaction with 3D environments. For example, smartphone- and smartwatch-based 3D pointing can find its applications in public and semi-public settings, such as airports, museums, shopping malls or conference venues, where displays are mounted out of reach of the user. In these settings it is not possible to deploy interactive 3D environments and enable interaction with them using mobile and wearable devices. Interaction scenarios that can be supported include 3D gaming, exploring and navigating scientific visualizations in museums or participating in collaborative modelling sessions. Each scenario requires a design of interactions specific for a given domain, but my work provides the first step toward such design by showing that mobile and wearable hardware is sufficient. Also, because many interactions start with target acquisitions, 3D pointing techniques introduced in my thesis can serve as the entry techniques to be modified and extended for domain-specific purposes.

Another impact of my dissertation is to show that it is possible to address, to a certain degree, in a single technique all 3D pointing challenges listed in Chapter 3. In Chapter 4, I have shown how Smartcasting addresses some of the 3D interaction challenges – such as the Heisenberg effect and perceived workload – listed in Chapter 3 in a novel way: while still allowing for target occlusion and disambiguation to be addressed using methods previously developed in the literature. In Chapter 5, I have shown how the same level of performance can be achieved on a smartwatch. In Chapter 6, I have shown that Tiltcasting achieves better performance than Smartcasting in dense environments. Tiltcasting is also equipped with an occlusion removal mechanism that integrates the exploration and navigation phase into selection time, significantly increasing the overall performance of the technique, thus addressing the problem of occlusion. Additionally, Tiltcasting provides perspective cues when a 3D environment is presented on a non-stereoscopic screen, thus addressing the depth identification problem. Finally, unlike Smartcasting or Watchcasting, Tiltcasting does not cause the gorilla-arm effect. As a two-handed technique that uses the "remapping principle" introduced in Chapter 7, Tiltcasting eliminates the need for users to lift their hand for a prolonged time, thus causing less perceived workload.

8.2 Future Work

My dissertation opens a number of future work opportunities, which I am already addressing in my current research, or plan to address in the near future. One possible direction of future work is to revisit the design of the 3D pointing techniques, expand the interaction beyond pointing and selection and evaluate the techniques in the wild. Another direction is to optimize the techniques for target-aware 3D environments. Finally, research can be done in validating the techniques for mass interaction. I briefly describe some early results and research ideas below.

8.2.1 Revisiting Design of Smartphone- and Smartwatch-based 3D Pointing

8.2.1.1 Improving Smartcasting

The three pointing techniques presented in my dissertation were designed to the best of my abilities at the time of their design. Yet, they do not cover the entire design space, leaving many potentially fruitful paths unexplored. For example, as noted above, the Watchcasting experiment showed that controlling depth though rotation is surprisingly precise on smartwatches. Would such control be better than touchscreen depth manipulation for Smartcasting too? To verify that, I plan to perform a study that compares accuracy of depth manipulation when using touchscreen vs. hand rotation.

8.2.1.2 Improving Watchcasting

Watchcasting itself could also be expanded using the knowledge I gained through the Tiltcasting design, i.e. the "remapping principle" discussed in Chapter 7. The essential observation here is that the smartwatch's touchscreen is usually interacted with using the dominant hand, with the smartwatch mounted on the non-dominant hand. Thus, I could implement Tiltcasting on a smartwatch by engaging the non-dominant hand in controlling the rotation of the interaction plane, and the dominant hand in moving the cursor on the interaction plane mapped to the smartwatch's touchscreen. I hypothesize that such interaction, even if it does not achieve faster selection times than Watchcasting, may result in lower perceived workload.

8.2.1.3 Improving Tiltcasting

It is possible to imagine improvements of the Tiltcasting specifically for target-aware environments. One idea is to implement a Bubble Cursor (Grossman and Balakrishnan, 2005) on the interaction plane. Because Tiltcasting has already significantly reduced the number of targets that can be interacted with at any time, introduction of a Bubble Cursor would, by definition, increase the selection time by creating regions of selection larger than the crosscut area between the target and the cursor. Yet another improvement of Tiltcasting that is, in my opinion, worth further exploration in the context of target-aware environments is to automatically rotate the plane into a position that is optimal for a given subset of 3D control space. Optimization could be a function of precision and selection time, or other factors such as a maximum number of targets that can be interacted with.

8.2.2 Toward Mass Interaction

My dissertation work contributes a deeper understanding of 3D interaction using mobile and wearable devices, particularly in the context of the occluded target and depth identification problems. All my techniques leverage the ubiquity of mobile or wearable devices to enable new forms of interaction, and they reduce barriers to data usage on affordable, accessible, and commercially available displays. One direction I am particularly interested in studying next is *mass* 2D and 3D interactions that go beyond single-user or multi-user applications (Figure 8.1). A situation when masses of users interact with a single screen introduces new challenges, both technological as well as design-related.

For example, the basic paradigm of the users' ability to identify the cursor they are controlling fails when multiple users connect to a single screen. The question is how such interactions could be facilitated. I performed some initial work in this area in a context of mass gaming in a cinema. In a startup company created for just that purpose, my team and I implemented a game that can be played before the main feature is shown, using a mobile device as a game controller. The game, "Little Red Riding Hood" consisted of 2 interactive scenes, each using interaction designs that addressed the problem of a potentially unlimited number of users.

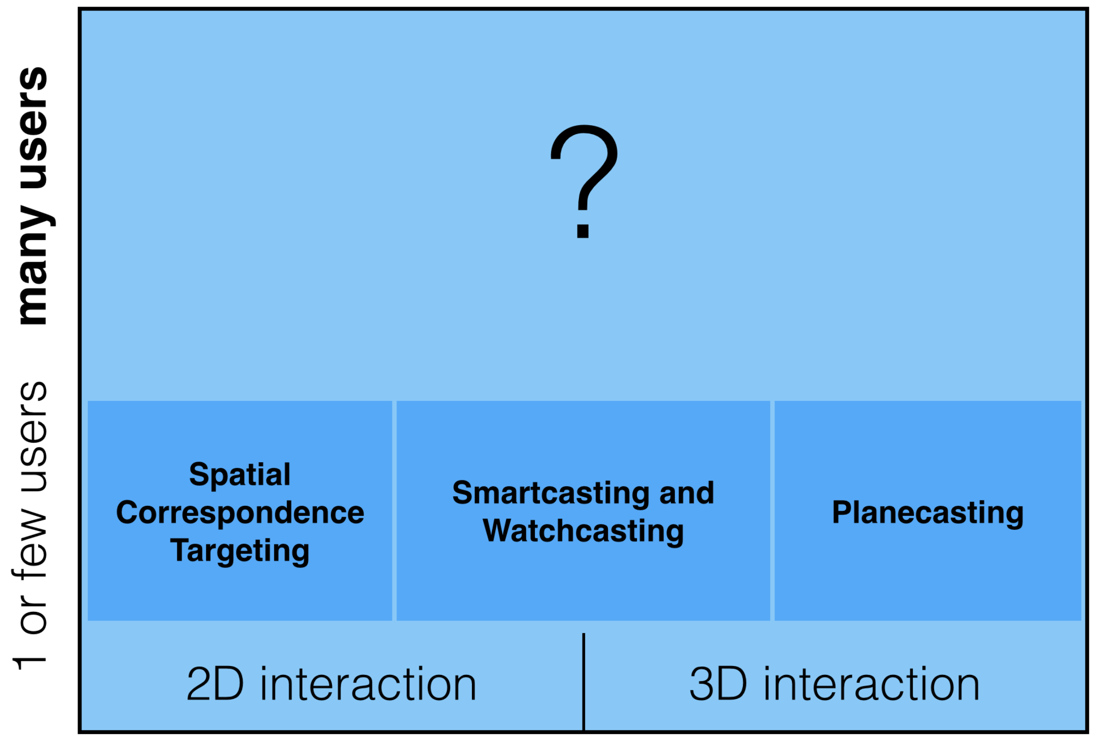

Figure 8.1: Mass interaction with a single large screen is a relatively unexplored area of large display interaction research

In the first scene, we used the spatial correspondence targeting discussed in Chapter 6 to allow the audience to collect fruits visible on the cinema screen, but not shown on their mobile devices. The screen could present only a limited number of randomly distributed fruits (around 50, in our case). Thus, to avoid having all fruits collected within seconds, given that movie theatre may have hundreds of concurrent players, we allowed one fruit to be collected by any number of users, as long as the collection event happened within a given time span. This simple solution can be scaled to any number of users.

In the second scene, Little Red Riding Hood was attacked by a wolf. Users could scare the wolf away by shooting the fruits toward the wolf using their mobile device as a slingshot (a technique similar to Smartcasting). Here, the problem was that allowing hundreds of users to shoot the fruits toward the wolf would inevitably clutter the screen. To avoid that, I grouped the vectors of shooting into 5 zones: regardless of the direction in which the user shot at, the fruit trajectory was corrected to one of the five directions. Moreover, we ensured that if more than one user shot the same type of fruit toward the wolf within a given time interval (in our case, $1s$), only one fruit would be shown flying onto the screen. This solution will also scale to any number of concurrent users.

The "Little Red Riding Hood" game was deployed "in-the-wild" in the local cinema to very positive user response (Figure 8.2). Although the game was not designed as a research project, I plan to perform formal experiments that involve mass gaming in cinemas using mobile devices, and to investigate how mass interaction can be facilitated by mobile and wearable devices.

8.2.3 Gesture-controlled Omnipresent Displays

Mobile and wearable devices provide a rich set of sensors combined with powerful computing. In my dissertation I used the characteristics of the mobile and wearable devices to facilitate 3D pointing. However, it is also possible to use them for other types of interactions in computing environments. One idea I am currently working on with colleagues in the Human Computer Interaction lab is to design a rich set of gestures that would be recognized with machine learning in real-time. The approach we have already tried is to implement Dynamic Time Warping (DTW) for a set of characters and graffiti gestures, achieving good recognition rates. The set of gestures could facilitate interaction with ubiquitous displays, such as browsing, text entry or controlling presentations.

The second step in the research is to apply gestural language to the control of displays whose position, size, shape and even texture could be user-controlled in a custom-built studio that allows for turning any surface into a display. I hypothesize that the universal presence of displays may be the way in which the problem of "fat finger" that affects mobile devices – and, to even greater

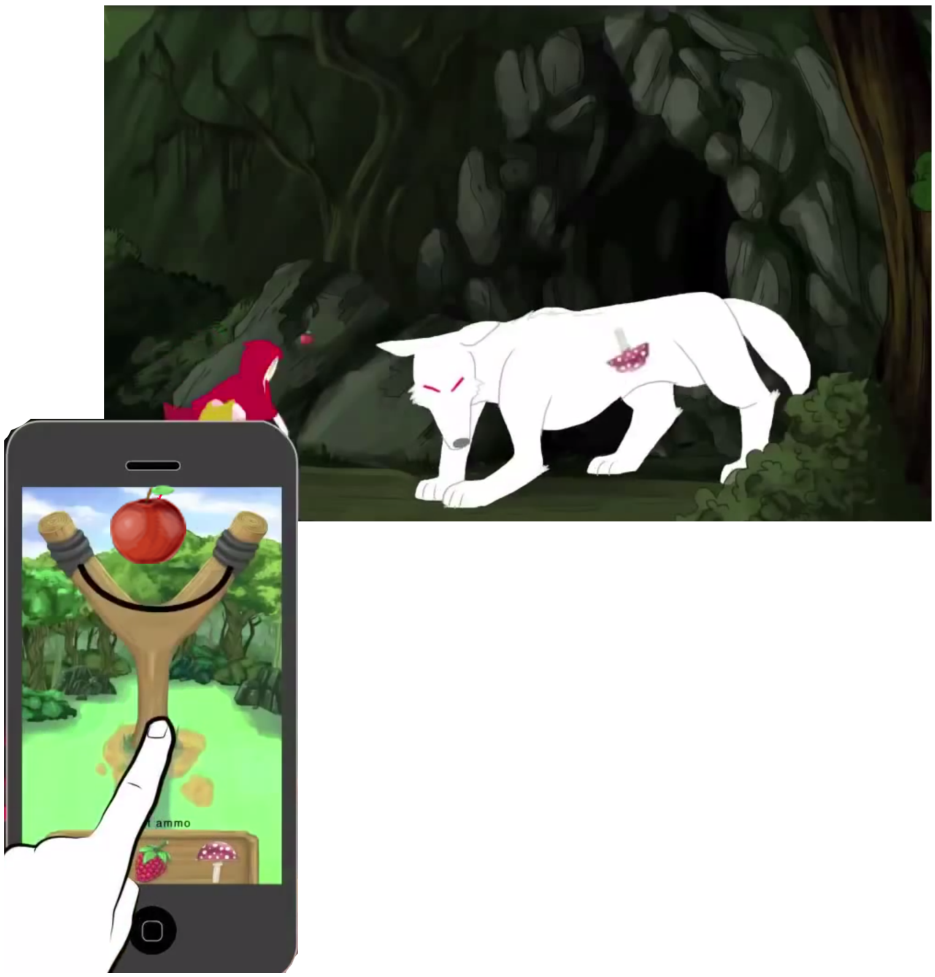

Figure 8.2: Mass interaction game, “Little Red Riding Hood”, deployed in a movie theatre.

degree, wearables – may be addressed. It is possible that the touchscreen of wearable devices will evolve to become an input (e.g. a touchpad) for any screen in a ubiquitous environment, transforming that screen to an extended (although temporary) private display of the wearable.

The potential of mobile and wearable devices is limited only by the imagination of the designers. In my dissertation I show that the current hardware is ready to facilitate Weiser's vision of "calm technology" and even go beyond that vision to a fully interconnected ubiquitous Internet of Things.

8.2.4 Limitations

My dissertation work shows that mobile and wearable devices can act as convenience devices in computing environments. The techniques designed and evaluated to support this claim were shown to work on par with specialized devices. Yet, neither the design of the techniques nor the experiments are free of limitations.

8.2.4.1 Measuring Perceived Workload vs. Fatigue

Another limitation of my dissertation is the measurement of perceived workload instead of fatigue. While perceived workload measurement provides useful information about the overall comfort of a 3D pointing technique, it is a subjective, self-reported measure. New measures where recently developed that may provide more objective quantitative data on pointing ergonomics. For example, a consumed endurance model (Hincapié-Ramos and Guo, 2014) provides a way of measuring fatigue. Because this method of measuring fatigue appeared after most of my experiments were already performed, for consistency and meta-comparison between the perceived workload results, I have kept using NASA TLX as the perceived workload measure.

8.2.4.2 Limitations of Evaluation

Certain limitations of my dissertation also result from using the same experimental setup through all studies. In order to allow for meta-analysis of my data, I have kept the same or similar experimental parameters throughout all experiments. Alternative solution would be to design more specific experiments for each technique and directly measuring other aspects of 3D pointing, such as Heisenberg effect or hand tremor. Yet, because the main goal of my experiment was to evaluate efficiency of casual 3D pointing, I did not specifically focus on Heisenberg effect or hand tremor effect. I plan to address Heisenberg effect in the future.

Also, given that all experiments were performed in a controlled environment, it is unclear how the techniques would perform in the wild. Given the goals of my studies, experimentation in a controlled environment was necessary to avoid many confounds present when deploying a technique in the wild. Yet, once the feasibility of the techniques has been established, I am interested in performing an in-the-wild study that would consider other aspects of mobile and wearable interaction with ubiquitous displays, such as the discoverability of the interaction and the ease of use when no training is provided.

One of the main limitations is that the performance results of my dissertation cannot be generalized and applied to the population of all users. For the purpose of experimental evaluation I chose participants who are familiar with mobile devices and who come from a specific environment: the university campus and who are thus represented a specific age range. Although the goal of my experiments was not to measure that facet, future research in this area would be useful.

Finally, the meta-analysis of Tiltcasting performance vs. Vanacken et al. (2007) techniques is limited for two reasons. First, while in Tiltcasting both wrist and finger movements are small, that is they don't cause large movements of hand in the 3D input space, that is not the case for Vanacken et al. Depending on the position of the target on the half-sphere, magnetic tracker has to be moved far (when value of randomly selected position on the z-axis approaches 20cm) or not at al. (when value of randomly selected position on the z-axis approaches 0). Thus, the distribution of selection times in Vanacken et al. may be biased toward the target positions that happen to have large z values, because that are the positions requiring large movement of hand.

8.2.4.3 Sensors Variability

Another limitation of my work comes from the fact that both mobile and wearable devices come in various forms and include sensors of varying quality. I did not measure the differences in quality between various models of smartphones and smartwatches; it is possible that some would perform better than others. Unfortunately, given the number of models available in the market such a study would not be feasible.

8.3 Concluding Note

Pervasive computing devices like smartphones and wearables are a reality, creating an opportunity for casual 3D interaction. While exploring the entirety of this ecology of devices is beyond the scope of any one dissertation or researcher, one important aspect of building device ecologies is understanding how computation can work together to support 3D pointing. The specific focus of this dissertation is leveraging personal and wearable computation as a platform-of-convenience for pointing with external displays, serendipitously encountered in the world. My work illustrates is the first step toward rich manipulation environments that can be realized today using current technology. It provides guidance on present-day 3D pointing design and on the sensor and infrastructure work needed to enhance casual 3D pointing.

Appendices

Appendix A

System Architecture

I present an overview of system architecture used in my experiments, thus allowing for easier reproducibility of my experimental results.

All three experiments presented in my dissertation were realized as a *client-server* architecture. For *server*, I used server script written in JavaScript that run on a node.js server. The server ran on a local PC workstation, the same one that was responsible for rendering the *client* 3D environment and recording the experimental data. This design choice eliminated the network latency of communication between the client 3D environment application and the node.js server, that could otherwise negatively affect the selection times. Because the PC workstation was equipped with a multicore CPU, I run the server process and 3D application on a separate core. This design choice ensured that there is no competition for processor resources between the node.js server and the client 3D application.

The *client* mobile device was realized as a web application that ran in a mobile browser. It consisted of three threads. The first thread read orientation data, that is an $< pitch, yaw, roll >$ orientation vector that results from a built-in sensor fusion of accelerometer, magnetometer and gyroscope data. It is important to note that by the time of implementation the orientation vector was imperfect and shown a horizontal drift. Newer versions of Android OS offer better orientation data in a form of rotation vector that virtually eliminates the drift. The orientation data was stored in a shared vector variable that could be accessed by other threads. The second thread was the built-in UI thread that read touchscreen events. The events were stored in the same shared vector variable, as the orientation data. For Smartcasting, I stored only the y coordinate of the touch, while Tiltcasting used the xy coordinates of the touch.

The communication protocol used was the standard WebSocket protocol. WebSocket simulates permanent connection over http using heartbeats messaging.

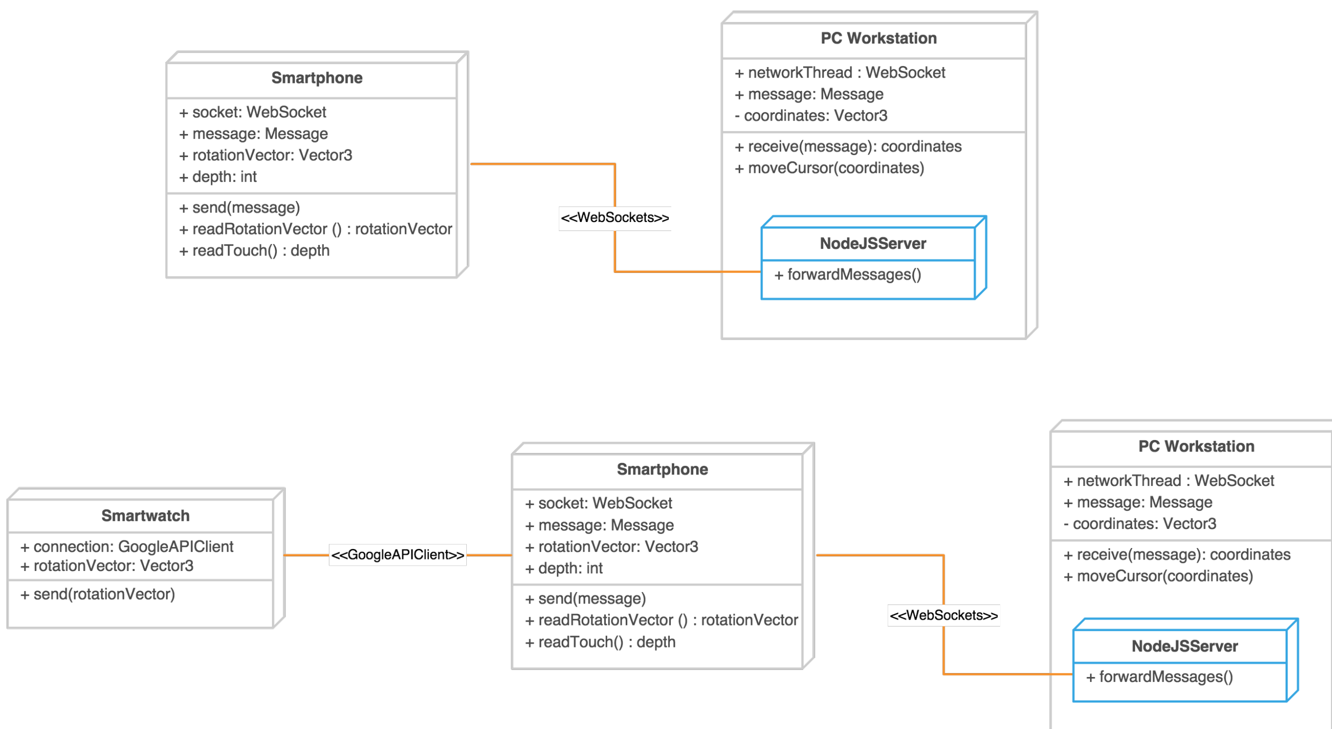

Figure A.1: UML Deployment Diagram

References

Pär-Anders Albinsson and Shumin Zhai. High Precision Touch Screen Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 105–112, New York, NY, USA, 2003. ACM. ISBN 1-58113-630-7. doi: 10.1145/642611.642631. URL `http://doi.acm.org/10.1145/642611.642631`.

Ferran Argelaguet and Carlos Andujar. Improving 3D selection in VEs through expanding targets and forced disocclusion. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5166 LNCS:45–57, 2008. ISSN 03029743. doi: 10.1007/978-3-540-85412-8_5.

Ferran Argelaguet and Carlos Andujar. Visual feedback techniques for virtual pointing on stereoscopic displays. *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology - VRST '09*, 2009. doi: 10.1145/1643928.1643966. URL `http://dl.acm.org/citation.cfm?id=1643928.1643966`.

Ferran Argelaguet and Carlos Andujar. A Survey of 3D Object Selection Techniques for Virtual Environments. *Computers and Graphics (Pergamon)*, 37(3):121–136, May 2013. ISSN 00978493. doi: 10.1016/j.cag.2012.12.003. URL `http://linkinghub.elsevier.com/retrieve/pii/S0097849312001793`.

Alec Azad, Jaime Ruiz, Daniel Vogel, Mark Hancock, and Edward Lank. Territoriality and Behaviour on and Around Large Vertical Publicly-Shared Displays. In *Proceedings of the Designing Interactive Systems Conference*, page 468, 2012. ISBN 9781450312103. doi: 10.1145/2317956.2318025. URL `http://dl.acm.org/citation.cfm?doid=2317956.2318025`.

Eliane Balaa, Mathieu Raynal, Youssef Bou Issa, and Emmanuel Dubois. Classification of Interaction Techniques in the 3D Virtual Environment on Mobile Devices. *Lecture Notes in Computer Science*, 8525:3–13, 2014.

Ravin Balakrishnan, Thomas Baudel, Gordon Kurtenbach, and George Fitzmaurice. The Rockin'Mouse: Integral 3D Manipulation on a Plane. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 311–318, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: 10.1145/258549.258778. URL `http://doi.acm.org/10.1145/258549.258778`.

Rafael Ballagas, Jan Borchers, Michael Rohs, and Jennifer G. Sheridan. The Smart Phone: A Ubiquitous Input Device. *IEEE Pervasive Computing*, 5:70–71, 2006. ISSN 15361268. doi: 10.1109/MPRV.2006.18.

Patrick Baudisch, Mike Sinclair, and Andrew Wilson. Soap: a pointing device that works in mid-air. *Proceedings of the 19th annual ACM symposium on User interface software and technology - UIST '06*, pages 43–46, 2006. doi: 10.1145/1166253.1166261. URL `http://dl.acm.org/citation.cfm?id=1166253.1166261`.

Gerald Bieber, Marian Haescher, and Matthias Vahl. Sensor requirements for activity recognition on smart watches. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, page 67. ACM, 2013.

Richard Bolt. Put-that-there. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques - SIGGRAPH '80*, pages 262–270, 1980. ISBN 0897910214. doi: 10.1145/800250.807503. URL `http://portal.acm.org/citation.cfm?doid=800250.807503`.

Gunnar Borg. *Borg's perceived exertion and pain scales.* Human kinetics, 1998.

Sebastian Boring, Marko Jurmu, and Andreas Butz. Scroll, Tilt or Move it: Using Mobile Phones to Continuously Control Pointers on Large Public Displays. In *Proceedings of 21st Annual Conference of the Australian Computer-Human Interaction (OzCHI'09)*, pages 161–168, 2009. ISBN 9781605588544. doi: 10.1145/1738826.1738853.

Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. Touch Projector: Mobile Interaction Through Video. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*, pages 2287–2296, 2010. ISBN 9781605589299. doi: 10.1145/1753326.1753671.

Bowman Doug A. Principles for the design of performance-oriented interaction techniques. *Handbook of Virtual Environments*, page 277, 2002.

Bowman Doug A. and Larry F. Hodges. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. In *Proceedings of the 1997 Symposium on Interactive 3D Sraphics (SI3D '97)*, pages 35–ff., 1997. ISBN 0897918843. doi: 10.1145/253284.253301. URL `http://portal.acm.org/citation.cfm?doid=253284.253301`.

Bowman Doug A., Db Johnson, and Lf Hodges. Testbed evaluation of virtual environment interaction techniques. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, volume 10, pages 26–33, 1999. doi: 10.1162/105474601750182333. URL `http://www.mitpressjournals.org/doi/abs/10.1162/105474601750182333$\delimiter"026E30F$nhttp://dl.acm.org/citation.cfm?id=323667`.

Bowman Doug A., C Wingrave, J Campbell, and V Ly. Using Pinch Gloves (tm) for Both Natural and Abstract Interaction Techniques in Virtual Environments. In *Proceedings of HCI International*, number 0106, pages 629–633, 2001. URL `http://eprints.cs.vt.edu/archive/00000547/`.

Bowman Doug A., Ernst Kruijff, Joseph LaViola, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Pearson Education, Inc., 2004. ISBN 0201758679.

Gerd Bruder, Frank Steinicke, and Wolfgang Stuerzlinger. Touching the void revisited: Analyses of touch behavior on and above tabletop surfaces. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8117 LNCS(PART 1):278–296, 2013. ISSN 03029743. doi: 10.1007/978-3-642-40483-2_19.

Jesse Burstyn, Paul Strohmeier, and Roel Vertegaal. DisplaySkin: Exploring Pose-Aware Displays on a Flexible Electrophoretic Wristband. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, pages 165–172, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3305-4. doi: 10.1145/2677199.2680596. URL `http://doi.acm.org/10.1145/2677199.2680596`.

E Bustamante and R Spain. Measurement Invariance of the Nasa TLX. *Human Factors and Ergonomics Society*, 52(19):1522–1526, 2008.

William Buxton. A Three-State Model of Graphical Input. *Human Computer Interaction*, 90: 449–456, 1990. URL `http://www.billbuxton.com/3state.html`.

William Buxton, George Fitzmaurice, Ravin Balakrishnan, and Gordon Kurtenbach. Large Displays in Automotive Design. In *Proceedings of IEEE Computer Graphics and Applications*, volume 20, pages 68–75, 2000. ISBN 0272-1716. doi: 10.1109/38.851753.

Xiang Cao and Ravin Balakrishnan. VisionWand: Interaction Techniques for Large Displays Using a Passive Wand Tracked in 3D. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, volume 23 of *UIST '03*, pages 173–182, New York, NY, USA, August 2003. ACM. ISBN 1-58113-636-6. doi: 10.1145/1015706.1015788. URL `http://doi.acm.org/10.1145/964696.964716http://doi.acm.org/10.1145/1015706.1015788`.

Sk Card, Jd Mackinlay, and Gg Robertson. A morphological analysis of the design space of input devices, 1991. URL `http://dl.acm.org/citation.cfm?id=128726`.

M.S.T. Carpendale, D.J. Cowperthwaite, and F.D. Fracchia. Extending Distortion Viewing from 2D to 3D. *IEEE Computer Graphics and Applications*, 17, 1997. ISSN 0272-1716. doi: 10.1109/38.595268.

Jeffrey Cashion, Chadwick Wingrave, and Joseph LaViola. Dense and dynamic 3D selection for game-based virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):634–642, 2012. ISSN 10772626. doi: 10.1109/TVCG.2012.40.

Gery Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. The Impact of Control-Display Gain on User Performance in Pointing Tasks. *Human-Computer Interaction*, 23(3): 215–250, 2008. ISSN 0737-0024. doi: 10.1080/07370020802278163.

E. Castellina and F. Corno. Multimodal gaze interaction in 3D virtual environments. *COGAIN 2008 Communication, Environment and Mobility Control by Gaze*, pages 1–5, 2008. URL `http://www.cad.polito.it/db/Cogain08.pdf`.

Neil Charness, Patricia Holley, Jeffrey Feddon, and Tiffany Jastrzembski. Light pen use and practice minimize age and hand performance differences in pointing tasks. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 46(3):373–384, 2004.

Xiang 'Anthony' Chen, Tovi Grossman, Daniel J. Wigdor, and George Fitzmaurice. Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, pages 159–168, 2014. doi: 10.1145/2556288.2556955. URL `http://dl.acm.org/citation.cfm?id=2556288.2556955`.

Zeynep Cipiloglu, Abdullah Bulbul, and Tolga Capin. A Framework for Enhancing Depth Perception in Computer Graphics. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization - APGV '10*, volume 1, page 141, 2010. ISBN 9781450302487. doi: 10.1145/1836248.1836276. URL `http://dl.acm.org/citation.cfm?id=1836248.1836276$\delimiter"026E30F$nhttp://dl.acm.org/citation.cfm?id=1836276$\delimiter"026E30F$nhttp://portal.acm.org/citation.cfm?doid=1836248.1836276`.

Norman Cook, Asami Yutsudo, Naoki Fujimoto, and Mayu Murata. Factors Contributing to Depth Perception: Behavioral Studies on the Reverse Perspective Illusion. *Spatial Vision*, 21: 397–405, 2008. ISSN 0169-1015. doi: 10.1163/156856808784532518.

Gabriele Costante, Lorenzo Porzi, Oswald Lanz, Paolo Valigi, and Elisa Ricci. Personalizing a smartwatch-based gesture interface with transfer learning. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 22nd European*, pages 2530–2534. IEEE, 2014.

Mary Czerwinski, Greg Smith, Tim Regan, Brian Meyers, George Robertson, and Gary Starkweather. Toward Characterizing the Productivity Benefits of Very Large Displays. *Proceedings of INTERACT*, 3(c):9–16, 2003.

Nguyen-Thong Dang, Hong-Ha Le, and M Tavanti. Visualization and interaction on flight trajectory in a 3D stereoscopic environment. In *Digital Avionics Systems Conference, 2003. DASC'03. The 22nd*, volume 2, pages 9—-A. IEEE, 2003.

Philip L. Davidson and Jefferson Y. Han. Synthesis and Control on Large Scale Multi-touch Sensing Displays. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 216–219, 2006. ISBN 2844263143. URL `http://portal.acm.org/citation.cfm?id=1142269`.

Patrick De La Hamette, Paul Lukowicz, Gerhard Tröster, and Toma Svoboda. Fingermouse: A Wearable Hand Tracking System. *Lecture Notes in Computer Science*, 3894:15–16, 2002. URL `http://www.ubicomp.org/ubicomp2002/ubicomp_adjunct_proceedings.pdf#page=19`.

Horace Dediu. Winning Against Non-consumption, 2015. URL `http://www.asymco.com/2015/05/11/winning-against-non-consumption/`.

Thomas A. DeFanti, Felipe Gomes De Carvalho, Daniel J. Sandin, Jurgen P. Schulze, Peter Otto, Javier Girado, Falko Kuester, Larry Smarr, and Ramesh Rao. The StarCAVE, a Third-Generation CAVE and Virtual Reality OptIPortal. *Journal Future Generation Computer Systems*, 25:169–178, 2009. ISSN 0167739X. doi: 10.1016/j.future.2008.07.015.

Mohammad O. Derawi, Claudia Nickely, Patrick Bours, and Christoph Busch. Unobtrusive user-authentication on mobile phones using biometric gait recognition. *Proceedings - 2010 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP 2010*, pages 306–311, 2010. doi: 10.1109/IIHMSP.2010.83.

Stephen Diverdi, Ismo Rakkolainen, Alex Olwal, Santa Barbara, Tobias Höllerer, and Alex Olwal. A Novel Walk-Through 3D Display. In *Proceedings of SPIE*, volume 6055, pages 428–437, 2006. ISBN 0819460958. doi: 10.1117/12.643286. URL `http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=727763$\delimiter"026E30F$nhttp://dx.doi.org/10.1117/12.643286http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=727763`.

H. J. Dudfield, C. Macklin, R. Fearnley, A. Simpson, and P. Hall. Big is Better? Human Factors Issues of Large Screen Displays with Military Command Teams. In *Human Interfaces in Contorl Rooms, Cockpits and Command Centres*, pages 304–309, 2001. ISBN 0537-9989. URL `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=942759&isnumber=20413`.

Richard R. Eckert and Jason A. Moore. The Classroom of the 21st Century: The Interactive Learning Wall. *SIGCHI Bulletin*, 32:33–40, 2000.

Rodger J. Elble and James E. Randall. Mechanistic components of normal hand tremor. *Electroencephalography and clinical neurophysiology*, 44(1):72–82, 1978.

Niklas Elmqvist. BalloonProbe : Reducing Occlusion in 3D using Interactive Space Distortion. In *Proceeding VRST '05 Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '05, pages 134 – 137, New York, NY, USA, 2005. ACM. ISBN 1-59593-098-1. doi: 10.1145/1101616.1101643. URL `http://doi.acm.org/10.1145/1101616.1101643`.

Niklas Elmqvist and Philippas Tsigas. View projection animation for occlusion reduction. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 471–475, 2006. ISBN 1-59593-353-0. doi: http://doi.acm.org/10.1145/1133265.1133361.

Niklas Elmqvist and Philippas Tsigas. View-projection animation for 3D occlusion management. *Computers and Graphics (Pergamon)*, 31:864–876, 2007. ISSN 00978493. doi: 10.1016/j.cag.2007.09.006.

Niklas Elmqvist and Philippas Tsigas. A taxonomy of 3D occlusion management for visualization. *IEEE transactions on visualization and computer graphics*, 14(5):1095–109, 2008.

ISSN 1077-2626. doi: 10.1109/TVCG.2008.59. URL http://www.ncbi.nlm.nih.gov/pubmed/18599920.

Mary Elwart-Keys, David Halonen, Marjorie Horton, Robert Kass, and Paul Scott. User Interface Requirements for Face to Face Groupware. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 295–301, New York, NY, USA, 1990. ACM. ISBN 0-201-50932-6. doi: 10.1145/97243.97299. URL http://doi.acm.org/10.1145/97243.97299.

Sue A. Ferguson, W. Gary Allread, Peter Le, Joseph D. Rose, and William S. Marras. Shoulder Muscle Oxygenation during Repetitive Tasks. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 55(1):1039–1041, 2011. doi: 10.1177/1071181311551217. URL http://pro.sagepub.com/content/55/1/1039.abstract.

Paul M. Fitts. Perceptual-motor skill learning. *Categories of human learning*, 47:381–391, 1964.

George Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, 36(7), 1993.

Paul Foeckler, Thomas Zeidler, Benjamin Brombach, Erich Bruns, and Oliver Bimber. PhoneGuide: Museum Guidance Supported by On-Device Object Recognition on Mobile Phones. In *International Conference on Mobile and Ubiquitous Computing (MUM'05)*, 2005.

Andrew Forsberg, Kenneth Herndon, and Robert Zeleznik. Aperture Based Selection for Immersive Virtual Environments. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, pages 95–96, New York, NY, USA, 1996. ACM. ISBN 0-89791-798-7. doi: 10.1145/237091.237105. URL http://doi.acm.org/10.1145/237091.237105.

Markus Funk, Alireza Sahami, Niels Henze, and Albrecht Schmidt. Using a Touch-sensitive Wristband for Text Entry on Smart Watches. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 2305–2310, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2474-8. doi: 10.1145/2559206.2581143. URL http://doi.acm.org/10.1145/2559206.2581143.

P. Garner, M. Collins, S. M. Webster, and D. A. D. Rose. The application of telepresence in medicine. *BT TECHNOLOGY JOURNAL*, 15:181–187, 1997. ISSN 0265-0193. doi: 10.1023/A:1018699900499.

Tovi Grossman and Ravin Balakrishnan. Pointing at Trivariate Targets in 3D Environments. *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*, 6(1):447–454, 2004. doi: 10.1145/985692.985749. URL `http://portal.acm.org/citation.cfm?doid=985692.985749`.

Tovi Grossman and Ravin Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, page 281, 2005. ISBN 1581139985. doi: 10.1145/1054972.1055012. URL `http://portal.acm.org/citation.cfm?doid=1054972.1055012`.

Tovi Grossman and Ravin Balakrishnan. The design and evaluation of selection techniques for 3D volumetric displays. *Proceedings of the 19th annual ACM symposium on User interface software and technology - UIST '06*, page 3, 2006. doi: 10.1145/1166253.1166257. URL `http://portal.acm.org/citation.cfm?doid=1166253.1166257http://dl.acm.org/citation.cfm?doid=1166253.1166257`.

Sean Gustafson, Christian Holz, and Patrick Baudisch. Imaginary Phone: Learning Imaginary Interfaces by Transferring Spatial Memory from a Familiar Device. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, pages 283–292, 2011. ISBN 9781450307161. doi: 10.1145/2047196.2047233. URL `http://dl.acm.org/citation.cfm?id=2047196.2047233`.

Gerwin De Haan, Michal Koutek, and Frits H Post. IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In *In Virtual Environments 2005*, pages 201–209, 2005.

Martin Hachet, Benoit Bossavit, Aurélie Cohé, and Jean-Baptiste Baptiste De La Rivière. Toucheo: multitouch and stereo combined in a seamless workspace. In *Proceedings of the 24th*, pages 587–592, 2011. ISBN 9781450307161. doi: 10.1145/2047196.2047273. URL `http://dl.acm.org/citation.cfm?id=2047273`.

Martin Hachet, Jean-Baptiste Baptiste De La Rivière, Jérémy Laviole, Aurélie Cohé, and Sébastien Cursan. Touch-Based Interfaces for Interacting with 3D Content in Public Exhibitions. *Computer Graphics and Applications, IEEE*, 33(2):80–85, 2013. ISSN 02721716. doi: 10.1109/MCG.2013.34.

Faizan Haque, Matieu Nancel, and Daniel Vogel. Myopoint: Pointing and Clicking Using Forearm Mounted Electromyography and Inertial Motion Sensors. In *Proc. of CHI'15, the 33rd Conference on Human Factors in Computing Systems*, 2015.

Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988.

Mary Hegarty and D Waller. Individual differences in spatial abilities. *The Cambridge handbook of visuospatial thinking*, pages 121–169, 2005.

Jd Hincapié-Ramos and Xiang Guo. Consumed Endurance: A metric to quantify arm fatigue of mid-air interactions. *Proceedings of the 32nd ...*, pages 1063–1072, 2014. doi: 10.1145/2556288.2557130. URL `http://dl.acm.org/citation.cfm?id=2557130`.

Ken Hinckley, Randy Pausch, John C Goble, and Neal F Kassell. A Survey of Design Issues in Spatial Input. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST '94, pages 213–222, New York, NY, USA, 1994. ACM. ISBN 0-89791-657-3. doi: 10.1145/192426.192501. URL `http://doi.acm.org/10.1145/192426.192501`.

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. ISBN 1-58113-867-9. doi: 10.1145/989863.989866. URL `http://doi.acm.org/10.1145/989863.989866`.

Christian Holz and Patrick Baudisch. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, pages 581–590, 2010. ISBN 9781605589299. doi: 10.1145/1753326.1753413. URL `http://portal.acm.org/citation.cfm?doid=1753326.1753413`.

Jonggi Hong, Seongkook Heo, Poika Isokoski, and Geehyuk Lee. SplitBoard: A Simple Split Soft Keyboard for Wristwatch-sized Touch Screens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1233–1236, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702273. URL `http://doi.acm.org/10.1145/2702123.2702273`.

Steven Houben, Nicolai Marquardt, and Gower Street. WatchConnect: A Toolkit for Prototyping Smartwatch-Centric Cross-Device Applications. In *Proc. of CHI'15, the 33rd Conference on Human Factors in Computing Systems*, 2015.

Da-Yuan Huang, Ming-Chang Tsai, Ying-Chao Tung, Min-Lun Tsai, Yen-Ting Yeh, Liwei Chan, Yi-Ping Hung, and Mike Y Chen. TouchSense: Expanding Touchscreen Input Vocabulary

Using Different Areas of Users' Finger Pads. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 189–192, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557258. URL http://doi.acm.org/10.1145/2556288.2557258.

Shahram Izadi, Harry Brignull, T Rodden, Y Rogers, and M 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)*, volume 5, pages 159–168, 2003. ISBN 1-58113-636-6. doi: 10.1145/964696.964714. URL http://discovery.ucl.ac.uk/1324299/.

Robert J. K. Jacob. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems*, 9(2):152–169, 1991. ISSN 10468188. doi: 10.1145/123078.128728.

Giulio Jacucci, Ann Morrison, Gabriela T. Richard, Jari Kleimola, Peter Peltonen, Lorenza Parisi, and Toni Laitinen. Worlds of information: Designing for engagement at a public multi-touch display. *SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*, pages 2267–2276, 2010. doi: 10.1145/1753326.1753669.

Seokhee Jeon, Jane Hwang, Gerard J. Kim, and Mark Billinghurst. Interaction with large ubiquitous displays using camera-equipped mobile phones. *Personal and Ubiquitous Computing*, 14:83–94, 2010. ISSN 16174909. doi: 10.1007/s00779-009-0249-0.

Ricardo Jota, João M Pereira, and Joaquim a Jorge. A Comparative Study of Interaction Metaphors for Large-scale Displays, 2009.

Ricardo Jota, Miguel Nacenta, Joaquim a Jorge, Sheelagh Carpendale, and Saul Greenberg. A Comparison of Ray Pointing Techniques for Very Large Displays. *GI '10 Proceedings of Graphics Interface 2010*, pages 269–276, 2010. ISSN 07135424. URL http://grouplab.cpsc.ucalgary.ca/grouplab/uploads/Publications/Publications/2010-RayPointing.GI.pdf.

Nicholas Katzakis. Plane-Casting: 3D Cursor Control With A SmartPhone. *The 3rd Dimension of CHI, Workshop at ACM SIGCHI*, 2012.

William Kearns, Jan M Jasiewicz, James L Fozard, Paul Webster, Steven Scott, Jeffrey Craighead, Mary E Bowen, and Joseph McCarthy. Temporo-spacial prompting for persons with cognitive impairment using smart wrist-worn interface. *Journal of Rehabilitation Research & Development*, 50(10):vii, 2013.

Frederic Kerber, Antonio Krüger, and Markus Löchtefeld. Investigating the Effectiveness of Peephole Interaction for Smartwatches in a Map Navigation Task. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services*, MobileHCI '14, pages 291–294, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3004-6. doi: 10.1145/2628363.2628393. URL `http://doi.acm.org/10.1145/2628363.2628393`.

Jeongseok Ki and Yong-Moo Kwon. 3D Gaze Estimation and Interaction. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008*, pages 373–376, May 2008. doi: 10.1109/3DTV.2008.4547886.

David Kim, O. Hilliges, Shahram Izadi, Alex Butler, J. Chen, I. Oikonomidis, and P Olivier. Digits: Freehand 3D Interactions Anywhere Using a Wrist-worn Gloveless Sensor. *Proceedings of User Interfaces and Interactive Systems 2012*, pages 167–176, 2012. doi: 10.1145/2380116.2380139. URL `http://dl.acm.org/citation.cfm?id=2380139$\delimiter"026E30F$npapers://c80d98e4-9a96-4487-8d06-8e1acc780d86/Paper/p9825`.

Jungsoo Kim, Jiasheng He, Kent Lyons, and Thad Starner. The Gesture Watch: A wireless contact-free Gesture based wrist interface. In *Proceedings - International Symposium on Wearable Computers, ISWC*, pages 15–22, 2007. ISBN 9781424414536. doi: 10.1109/ISWC.2007.4373770.

C. Kirstein and H. Muller. Interaction with a projection screen using a camera-tracked laser pointer. *Proceedings 1998 MultiMedia Modeling. MMM'98 (Cat. No.98EX200)*, 1998. doi: 10.1109/MULMM.1998.723001.

Jarrod Knibbe, Diego Martinez Plasencia, Christopher Bainbridge, Chee-Kin Chan, Jiawei Wu, Thomas Cable, Hassan Munir, and David Coyle. Extending Interaction for Smart Watches: Enabling Bimanual Around Device Control. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 1891–1896, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2474-8. doi: 10.1145/2559206.2581315. URL `http://doi.acm.org/10.1145/2559206.2581315`.

Werner A. König, Jens Gerken, Stefan Dierdorf, and Harald Reiterer. Adaptive pointing-design and evaluation of a precision enhancing technique for absolute pointing devices. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5726 LNCS(PART 1):658–671, 2009. ISSN 03029743. doi: 10.1007/978-3-642-03655-2.

Regis Kopper, M. G. Silva, Ryan P. McMahan, and Bowman Doug A. Increasing the precision of distant pointing for large high-resolution displays. Technical report, 2008. URL `http://eprints.cs.vt.edu/archive/00001024/`.

Regis Kopper, Bowman Doug A., Mara G. Silva, and Ryan P. McMahan. A human motor behavior model for distal pointing tasks. *International Journal of Human-Computer Studies*, 68(10):603–615, October 2010. ISSN 10715819. doi: 10.1016/j.ijhcs.2010.05.001. URL `http://linkinghub.elsevier.com/retrieve/pii/S1071581910000637`.

Regis Kopper, F Bacim, and Bowman Doug A. Rapid and accurate 3D selection by progressive refinement. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 67–74, March 2011. doi: 10.1109/3DUI.2011.5759219.

Ym Kwon, Kw Jeon, Jeongseok Ki, Qm Shahab, Sangwoo Jo, and Sk Kim. 3D Gaze Estimation and Interaction to Stereo Dispaly. *Ijvr*, 5(3):41–45, 2006. doi: 10.1109/3DTV.2008.4547886. URL `http://pdf.aminer.org/000/330/085/virtual_d_interactions_between_d_real_multi_views.pdf`.

Mikko Kytö, Aleksi Mäkinen, Jukka Häkkinen, and Pirkko Oittinen. Improving Relative Depth Judgments in Augmented Reality with Auxiliary Augmentations. *ACM Trans. Appl. Percept.*, 10(1), March 2013. ISSN 1544-3558. doi: 10.1145/2422105.2422111. URL `http://doi.acm.org/10.1145/2422105.2422111`.

Marc T. M. Lambooij, Wijnand A. Ijsselsteijn, and Ingrid Heynderickx. Visual Discomfort in Stereoscopic Displays : A Review. *SPIE*, 6490(May 2010):1–13, 2007.

Joel Lanir, Kellogg S Booth, and Anthony Tang. MultiPresenter: A Presentation System for (Very) Large Display Spaces. *Human Factors*, pages 519–528, 2008. doi: 10.1145/1459359.1459428.

C.M.G. Lee, A.R.L. Travis, and R. Lin. Flat-panel autostereoscopic 3D display, 2008. ISSN 17518768.

S. K. Lee, William Buxton, and K. C. Smith. A Multi-Touch Three Dimensional Touch-Sensitive Tablet. In *Proceedings of CHI 1985*, number April, pages 21–25, 1985. ISBN 0897911490.

Yang Li. Protractor: a fast and accurate gesture recognizer. *Proceedings of the 28th international conference on Human factors in computing systems*, pages 2169–2172, 2010. doi: 10.1145/1753326.1753654. URL `http://dl.acm.org/citation.cfm?id=1753654`.

J. Liang and M. Green. Geometric modeling using six degrees of freedom input devices. *3rd International Conference on CAD and Computer Graphics*, 54:217–222, 1993. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Geometric+Modeling+Using+Six+Degrees+of+Freedom+Input+Devices#0.

J. Liang and M. Green. JDCAD: A highly interactive 3 D modeling system. *Computers & Graphics*, 18:449–506, 1994. URL http://cat.inist.fr/?aModele=afficheN&cpsidt=3361677.

Hongen Liao, Takeyoshi Dohi, and Keisuke Nomura. Autostereoscopic 3D Display with Long Visualization Depth Using Referential Viewing Area-Based Integral Photography. *IEEE Transactions on Visualization and Computer Graphics*, 17:1690–1701, 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2010.267.

Hyunchul Lim, YoonKyong Cho, Wonjong Rhee, and Bongwon Suh. Vi-Bros: Tactile Feedback for Indoor Navigation with a Smartphone and a Smartwatch. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pages 2115–2120, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3146-3. doi: 10.1145/2702613.2732811. URL http://doi.acm.org/10.1145/2702613.2732811.

Andrés Lucero, Tero Jokela, Arto Palin, Viljakaisa Aaltonen, and Jari Nikara. EasyGroups: Binding Mobile Devices for Collaborative Interactions. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 2189–2194, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1016-1. doi: 10.1145/2212776.2223774. URL http://doi.acm.org/10.1145/2212776.2223774.

I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7:91–139, 1992. ISSN 0737-0024.

I. Scott MacKenzie and Shaidah Jusoh. An Evaluation of Two Input Devices for Remote Pointing. *Interaction, Engineering for Human-Computer Lecture Notes in Computer Science Volume*, 2254:235–250, 2001.

Andrew Maimone and Henry Fuchs. A first look at a telepresence system with room-sized real-time 3d capture and life-sized tracked display wall. *Proceedings of ICAT 2011, to appear*, pages 4–9, 2011. URL http://www.cs.unc.edu/~fuchs/kinect_ICAT_2011.pdf.

Shahzad Malik, Abhishek Ranjan, and Ravin Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. *Proceedings of the 18th annual ACM symposium on User interface software and technology - UIST '05*, page 43, 2005. doi: 10.1145/1095034.1095042. URL http://portal.acm.org/citation.cfm?doid=1095034.1095042.

Michael Mauderer, Simone Conte, Miguel Nacenta, and Dhanraj Vishwanath. Depth perception with gaze-contingent depth of field. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, pages 217–226, 2014. doi: 10.1145/2556288.2557089. URL http://dl.acm.org/citation.cfm?doid=2556288.2557089.

David C Mccallum and Pourang P. Irani. ARC-Pad : Absolute + Relative Cursor Positioning for Large Displays with a Mobile Touchscreen. *Human Factors*, pages 153–156, 2009. doi: 10.1145/1622176.1622205. URL http://dl.acm.org/citation.cfm?id=1622205.

Daniel Medeiros, Felipe Carvalho, Lucas Teixeira, Priscilla Braz, and Alberto Raposo. Proposal and evaluation of a tablet-based tool for 3D virtual environments. *SBC Journal on 3D Interactive Systems*, 4(2):30–42, 2013.

L. Medeiros, D. P. S., Teixeira and Alberto Raposo. Navigation methods in engineering models using mobile devices. In *Proceedings of the 13th Symposium on Virtual and Augmented Reality 2012*, 2012.

M. Möllers, Patrick Zimmer, and Jan Borchers. Direct Manipulation and the Third Dimension: Co-Planar Dragging on 3D Displays. *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces - ITS '12*, (i), 2012. doi: 10.1145/2396636.2396639. URL http://hci.rwth-aachen.de/materials/publications/moellers2012b.pdf.

Jörg Müller, Gilles Bailly, Thor Bossuyt, and Niklas Hillgren. MirrorTouch: Combining Touch and Mid-air Gestures for Public Displays. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services (Mobile HCI '14)*, 2014. ISBN 9781450330046. doi: 10.1145/2628363.2628379.

Brad Myers, Herb Stiel, and Robert Gargiulo. Collaboration Using Multiple PDAs Connected to a PC. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, CSCW '98, pages 285–294, New York, NY, USA, 1998. ACM. ISBN 1-58113-009-0. doi: 10.1145/289444.289503. URL http://doi.acm.org/10.1145/289444.289503.

Brad Myers, Rishi Bhatnagar, Jeffrey Nichols, Choon Hong Peck, Dave Kong, Robert Miller, and Long Chris. Interacting at a distance: measuring the performance of laser pointers and other devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves (CHI'*, number 4, pages 33–40, 2002. ISBN 1581134533. URL internal-pdf://10.1.1.15.9445-2119101966/10.1.1.15.9445.pdf$\delimiter"026E30F$nhttp://doi.acm.org/10.1145/503376.503383.

Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Pourang P. Irani, and Michel Beaudouin-Lafon. High-precision pointing on large wall displays using small handheld devices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, page 831, 2013. doi: 10.1145/2470654.2470773. URL http://dl.acm.org/citation.cfm?doid=2470654.2470773.

Hideaki Nii. Wide area projection method for active-shuttered real image autostereoscopy. *ACM SIGGRAPH 2013 Posters*, page 4503, 2013. doi: 10.1145/2503385.2503443. URL http://dl.acm.org/citation.cfm?id=2503443.

Donald A Norman. Natural Interfaces are not Necesserily Natural. *ACM Interactions*, 2010. URL http://www.jnd.org/dn.mss/natural_user_interfa.html.

Ian Oakley, DoYoung Lee, M D Rasel Islam, and Augusto Esteves. Beats: Tapping Gestures for Smart Watches. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1237–1246, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702226. URL http://doi.acm.org/10.1145/2702123.2702226.

Masa Ogata and Michita Imai. SkinWatch: Skin Gesture Interaction for Smart Watch. In *Proceedings of the 6th Augmented Human International Conference*, AH '15, pages 21–24, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3349-8. doi: 10.1145/2735711.2735830. URL http://doi.acm.org/10.1145/2735711.2735830.

J Oh and W Stuerzlinger. Laser pointers as collaborative pointing devices. *In Graphics Interfaces*, pages 141–149, 2002. ISSN 07135424.

Dan R Olsen and Travis Nielsen. Laser pointer interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 01:17–22, 2001. doi: 10.1145/365024.365030. URL http://doi.acm.org/10.1145/365024.365030.

A. Olwal, Hrvoje Benko, and S. Feiner. SenseShapes: using statistical geometry for object selection in a multimodal augmented reality. *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pages 0–1, 2003. doi: 10.1109/ISMAR.2003.1240730.

Peter Peltonen, Antti Salovaara, Giulio Jacucci, Tommi Ilmonen, Carmelo Ardito, Petri Saarikko, and Vikram Batra. Extending Large-scale Event Participation with User-created Mobile Media on a Public Display. *Proceedings of the 6th International Conference on Mobile and Ubiquitous Multimedia (MUM '07)*, pages 131–138, 2007. doi: 10.1145/1329469.1329487. URL `http://portal.acm.org/citation.cfm?doid=1329469.1329487`.

S Camille Peres, Vickie Nguyen, Philip T. Kortum, Magdy Akladios, S Bart Wood, and Andrew Muddimer. Software ergonomics. *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems - CHI EA '09*, page 3949, 2009. doi: 10.1145/1520340.1520599. URL `http://portal.acm.org/citation.cfm?doid=1520340.1520599`.

Jeffrey S Pierce, Andrew Forsberg, Matthew J Conway, Seung Hong, Robert Zeleznik, and Mark R Mine. Image Plane Interaction Techniques in 3D Immersive Environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, I3D '97, New York, NY, USA, 1997. ACM. ISBN 0-89791-884-3. doi: 10.1145/253284.253303. URL `http://doi.acm.org/10.1145/253284.253303`.

Krzysztof Pietroszek and Edward Lank. Clicking Blindly: Using Spatial Correspondence to Select Targets in Multi-device Environments. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, number Figure 1 in MobileHCI '12, pages 331–334, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1105-2. doi: 10.1145/2371574.2371625. URL `http://doi.acm.org/10.1145/2371574.2371625`.

Krzysztof Pietroszek, Anastasia Kuzminykh, James R. Wallace, and Edward Lank. Smartcasting: A Discount 3D Interaction Technique for Public Displays. In *Proceedings of OzCHI '14*, 2014. ISBN 9781450306539.

J M Plumert, J K Kearny, J F Cremer, and K Recker. Distance perception in real and virtual environments. *ACM Transactions on Applied Perception*, 2(3):216–233, 2005.

Lorenzo Porzi, Stefano Messelodi, Carla Mara Modena, and Elisa Ricci. A Smart Watch-based Gesture Recognition System for Assisting People with Visual Impairments. In *Proceedings of the 3rd ACM International Workshop on Interactive Multimedia on Mobile & Portable Devices*, IMMPD '13, pages 19–24, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2399-4. doi: 10.1145/2505483.2505487. URL `http://doi.acm.org/10.1145/2505483.2505487`.

Ivan Poupyrev, Mark Billinghurst, and S Weghorst. The go-go interaction technique: non-linear mapping for direct manipulation in VR. *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80, 1996. doi: 10.1145/237091.237102. URL `http://portal.acm.org/citation.cfm?id=237091.237102&type=series`.

Ivan Poupyrev, Suzanne Weghorst, Mark Billinghurst, and Tadao Ichikawa. A framework and testbed for studying manipulation techniques for immersive VR. *Proceedings of the ACM Virtual Reality Systems and Technology Conference - VRST'97*, pages 21–28, 1997. doi: 10.1145/261135.261141. URL `http://dl.acm.org/citation.cfm?id=261135.261141`.

Ivan Poupyrev, T Ichikawa, S Weghorst, and Mark Billinghurst. Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques. *Computer Graphics Forum*, 17:41–52, 1998. ISSN 0167-7055. doi: 10.1111/1467-8659.00252. URL `http://doi.wiley.com/10.1111/1467-8659.00252`.

Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and Robust Hand Tracking from Depth. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014. ISSN 10636919. doi: 10.1109/CVPR.2014.145.

Sara Radicati. No Mobile Statistics Report, 2014-2018. Technical report, 2014.

Mahfuz Rahman, Sean Gustafson, Pourang P. Irani, and Sriram Subramanian. Tilt techniques: investigating the dexterity of wrist-based input. *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, page 1943, 2009. doi: 10.1145/1518701.1518997. URL `http://portal.acm.org/citation.cfm?id=1518701.1518997`.

Gonzalo Ramos, George Robertson, Mary Czerwinski, Desney Tan, Patrick Baudisch, Ken Hinckley, and Maneesh Agrawala. Tumble! Splat! helping users access and manipulate occluded content in 2D drawings. In *Proceedings of the working conference on Advanced visual interfaces - AVI '06*, page 428, 2006. ISBN 1595933530. doi: 10.1145/1133265.1133351. URL `http://dl.acm.org/citation.cfm?id=1133265.1133351`.

Stephan Reichelt, Ralf Haussler, Gerald Fütterer, and Norbert Leister. Depth cues in human visual perception and their realization in 3D displays. In *Three Dimensional Imaging, Visualization, and Display 2010*, pages 76900B–76900B–12, 2010. ISBN 9780819481542. doi: 10.1117/12.850094.

Jun Rekimoto. The Magnifying Glass Approach to Augmented Reality Systems. *International Conference on Artificial Reality and Telexistence icat*, 95:123–132, 1995. ISSN 10547460.

doi: 10.1.1.50.5835. URL http://www.vrsj.org/ic-at/ICAT2003/papers/95123.pdf.

B Rogers and M Graham. Motion parallax as an independent cue for depth perception. *Perception*, 8:125–134, 1979. ISSN 0301-0066. doi: 10.1068/p080125.

Daniel M Russell, Clemens Drews, and Alison E. Sue. Social Aspects of Using Large Public Interactive Displays for Collaboration. *UbiComp 2002 Ubiquitous Computing*, 2498: 229–236, 2002. ISSN 03029743. doi: 10.1007/3-540-45809-3_18. URL http://www.springerlink.com/content/fpupwgk001dhqk04/.

David Sachs. Sensor fusion on android devices: A revolution in motion processing. *Google Tech Talks*, 2010.

T. A. Sandstrom, C. Henze, and C. Levit. The hyperwall. *Proceedings of The Coordinated & Multiple Views in Exploratory Visualization*, 2003. URL http://people.nas.nasa.gov/~creon/papers/hyperwall.01215010.pdf.

Steven Schirra and Frank R Bentley. "It's Kind of Like an Extra Screen for My Phone": Understanding Everyday Uses of Consumer Smart Watches. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pages 2151–2156, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3146-3. doi: 10.1145/2702613.2732931. URL http://doi.acm.org/10.1145/2702613.2732931.

Dominik Schmidt, Fadi Chehimi, Enrico Rukzio, and Hans Gellersen. PhoneTouch: A technique for direct phone interaction on surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology (UIST'10)*, 2010. ISBN 9781605587455. doi: 10.1145/1866029.1866034. URL http://eprints.lancs.ac.uk/42612/.

G. Schmidt, Y. Baillot, D. G. Brown, E. B. Tomlin, and J. E. Swan. Toward Disambiguating Multiple Selections for Frustum-Based Pointing. In *3D User Interfaces, 2006. 3DUI 2006. IEEE Symposium on*, pages 87–94, March 2006. doi: 10.1109/VR.2006.133.

Jan Schultz. A history of the PROMIS technology: an effective human interface. In *A history of personal workstations*, pages 439–488. ACM, 1988.

Garth B. D. Shoemaker, Anthony Tang, and Kellogg S Booth. Shadow Reaching: A New Perspective on Interaction for Large Wall Displays. *Proc. UIST*, pages 53–56, 2007. doi: 10.1145/1294211.1294221. URL http://portal.acm.org/citation.cfm?id=1294221.

Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. Fat finger worries: How older and younger users physically interact with PDAs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3585 LNCS, pages 267–280, 2005. ISBN 3540289437. doi: 10.1007/11555261_24.

Gisela Sjogaard, Gabrielle Savard, and Carsten Juel. Muscle blood flow during isometric activity and its relation to muscle fatigue. *European Journal of Applied Physiology and Occupational Physiology*, 57(3):327–335, 1988. ISSN 0301-5548. doi: 10.1007/BF00635992. URL http://dx.doi.org/10.1007/BF00635992.

Anthony Steed. Towards a General Model for Selection in Virtual Environments. *Proceedings of 3D User Interfaces 2006*, pages 103–110, 2006. doi: 10.1109/VR.2006.134. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1647515.

Anthony Steed and Chris Parker. 3D Selection Strategies for Head Tracked and Non-Head Tracked Operation of Spatially Immersive Displays. In *Proceedings of 8th International Immersive Projection Technology Workshop*, pages 13–14, 2004.

Frank Steinicke, Klaus Hinrichs, Johannes Schöning, and Antonio Krüger. Multi-Touching 3D Data: Towards Direct Interaction in Stereoscopic Display, Environments Coupled with Mobile Devices. In *Proceedings of Advanced Visual Interfaces Workshop AVI2008*, pages 1–4, 2008. URL http://www.johannesschoening.de/website/Publications_files/MultiTouching3DData.pdf.

Jason Stewart, Benjamin B Bederson, and Allison Druin. Single Display Groupware: A Model for Co-present Collaboration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 286–293, New York, NY, USA, 1999. ACM. ISBN 0-201-48559-1. doi: 10.1145/302979.303064. URL http://doi.acm.org/10.1145/302979.303064.

Jeremy Sutton. Air Painting with Corel Painter Freestyle and the Leap Motion Controller: a Revolutionary New Way to Paint! In *Proceedings of ACM SIGGRAPH 2013 - Studio Talks*, page 1, 2013. ISBN 9781450322614. doi: 10.1145/2503673.2503694. URL http://doi.acm.org/10.1145/2503673.2503694.

Haruo Takemura and a Tomono. An Evaluation of 3-D Object Pointing Using a Field Sequential Stereoscopic Display. *Graphics Interface'88*, pages 112–118, 1988.

A.R.L. Travis. Autostereoscopic 3-D Display. *Applied Optics*, 29:4341–4342, 1990. ISSN 0018-9162. doi: 10.1364/AO.29.004341.

Marlon Twyman, Joe Mullenbach, Craig Shultz, J Edward Colgate, and Anne Marie Piper. Designing Wearable Haptic Information Displays for People with Vision Impairments. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, pages 341–344, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3305-4. doi: 10.1145/2677199.2680578. URL http://doi.acm.org/10.1145/2677199.2680578.

Yuta Ueda, Karin Iwazaki, Mina Shibasaki, Yusuke Mizushina, Masahiro Furukawa, Hideaki Nii, Kouta Minamizawa, and Susumu Tachi. HaptoMIRAGE Mid-air Autostereoscopic Display for Seamless Interaction with Mixed Reality Environments. In *ACM SIGGRAPH 2014 Emerging Technologies*, page 2012, 2014. ISBN 9781450329583.

Pasi Välkkynen, Ilkka Korhonen, J Plomp, and T Tuomisto. A User Interaction Paradigm for Physical Browsing and Near-Object Control Based on Tags. In *Proceedings of Physical Interaction Workshop on Real World User Interfaces*, pages 28–37, 2003. URL http://www-alt.medien.ifi.lmu.de/en/events/pi03/papers/pi03-ws-proceedings.pdf#page=31.

Dimitar Valkov, Frank Steinicke, Gerd Bruder, and Klaus Hinrichs. 2d Touching of 3d Stereoscopic Objects. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems - CHI '11*, page 1353, 2011. ISBN 9781450302289. doi: 10.1145/1978942.1979142. URL http://portal.acm.org/citation.cfm?doid=1978942.1979142.

Lode Vanacken, Tovi Grossman, and Karin Coninx. Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments. In *2007 IEEE Symposium on 3D User Interfaces*, pages 117–124. Ieee, 2007. ISBN 1-4244-0907-1. doi: 10.1109/3DUI.2007.340783. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4142854.

Lode Vanacken, Tovi Grossman, and Karin Coninx. Multimodal Selection Techniques for Dense and Occluded 3D Virtual Environments. *International Journal of Human-Computer Studies*, 67(3):237–255, March 2009. ISSN 10715819. doi: 10.1016/j.ijhcs.2008.09.001. URL http://linkinghub.elsevier.com/retrieve/pii/S1071581908001109.

Boris Velichkovsky, Andreas Sprenger, and Pieter Unema. Towards gaze-mediated interaction: Collecting solutions of the Midas touch problem. In *Human-Computer Interaction INTERACT97*, pages 509–516. Springer, 1997.

Simon Voelker, Chat Wacharamanotham, and Jan Borchers. An Evaluation of State Switching Methods for Indirect Touch Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, page 745, 2013. ISBN 9781450318990. doi: 10.1145/2470654.2470759. URL `http://dl.acm.org/citation.cfm?doid=2470654.2470759`.

Daniel Vogel and Ravin Balakrishnan. Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, volume 6, pages 137–146, 2004. ISBN 1581139578. doi: 10.1145/1029632.1029656. URL `http://portal.acm.org/citation.cfm?doid=1029632.1029656`.

Daniel Vogel and Ravin Balakrishnan. Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In *Proceedings of the 18th annual ACM Symposium on User Interface Software and Technology*, volume pp, page 33, 2005. ISBN 159593023X. doi: 10.1145/1095034.1095041. URL `http://portal.acm.org/citation.cfm?doid=1095034.1095041`.

James R. Wallace, Stacey Scott, Taryn Stutz, Tricia Enns, and Kori Inkpen. Investigating Teamwork and Taskwork in Single- and Multi-Display Groupware Systems. *Personal and Ubiquitous Computing*, 13:569–581, 2009. ISSN 16174909. doi: 10.1007/s00779-009-0241-8.

David Waller. Individual differences in spatial learning from computer-simulated environments. *Journal of Experimental Psychology: Applied*, 6(4):307, 2000.

L.R. Wanger, J.A. Ferwerda, and D.P. Greenberg. Perceiving Spatial Relationships in Computer-Generated Images. *IEEE Computer Graphics and Applications*, 12, 1992. ISSN 0272-1716. doi: 10.1109/38.135913.

Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.

Eric N Wiebe, Edward Roberts, and Tara S Behrend. An Examination of Two Mental Workload Measurement Approaches to Understanding Multimedia Learning. *Comput. Hum. Behav.*, 26 (3):474–481, May 2010. ISSN 0747-5632. doi: 10.1016/j.chb.2009.12.006. URL `http://dx.doi.org/10.1016/j.chb.2009.12.006`.

Andrew Wilson and Hubert Pham. Pointing in intelligent environments with the World Cursor. *Proceedings of the 9th conference on Human-Computer Interaction - Interact '03*, pages 495–502, 2003.

Terry Winograd and François Guimbretière. Visual Instruments for an Interactive Mural. In *CHI '99: Extended Abstracts on Human Factors in Computing Systems*, pages 1–2, 2003. ISBN 1581131585. doi: 10.1145/632856.632861.

HP Wyss, Roland Blach, and Matthias Bues. iSith-Intersection-Based Spatial Interaction for Two Hands. In *3D User Interfaces (3DUI'06)*, pages 59–61, 2006. ISBN 1424402255. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1647507`.

Chao Xu, Parth H Pathak, and Prasant Mohapatra. Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition Using Smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, HotMobile '15, pages 9–14, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3391-7. doi: 10.1145/2699343.2699350. URL `http://doi.acm.org/10.1145/2699343.2699350`.

Yonggao Yang and Lin Li. Turn a Nintendo WiiMote into a Handheld Computer Mouse. *IEEE Potentials*, 30(January):12–16, 2011. ISSN 02786648. doi: 10.1109/MPOT.2010.938621.

Shumin Zhai. Human Performance in Six Degree of Freedom Input Control. *Industrial Engineering*, page 179, 1995. doi: 10.1145/274644.274689. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.71.2326&rep=rep1&type=pdf`.

Shumin Zhai. User Performance in Relation to 3D Input Device Design, 1998. ISSN 00978930.

Shumin Zhai, William Buxton, and Paul Milgram. Investigating The Silk Cursor: Transparency for 3D Target Acquisition. *Human Factors in Computing Systems*, pages 459–465, 1994. doi: 10.1145/191666.191822.

Shumin Zhai, William Buxton, and Paul Milgram. The Partial-Occlusion Effect: Utilizing Semi-transparency in 3D Human-Computer Interaction. *ACM Transactions on Computer-Human Interaction*, 3:254–284, 1996a. ISSN 10730516. doi: 10.1145/234526.234532.

Shumin Zhai, Paul Milgram, and William Buxton. The influence of muscle groups on performance of multiple degree-of-freedom input. In *Proceedings of the SIGCHI conference on Human factors in computing systems common ground - CHI '96*, pages 308–315, 1996b. ISBN 0897917774. doi: 10.1145/238386.238534. URL `http://portal.acm.org/citation.cfm?doid=238386.238534`.

Shumin Zhai, Paul Milgram, and A. Rastogi. Anisotropic Human Performance in Six Degree-of-Freedom Tracking: an Evaluation of Three-Dimensional Display and Control Interfaces. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 27, 1997. ISSN 1083-4427. doi: 10.1109/3468.594917.