Towards Interactive Desk Workspaces

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

vorgelegt von

Diplom-Informatiker
Simon Frederik Völker
aus Köln

Berichter: Prof. Dr. Jan Borchers
Prof. Dr. Johannes Schöning

Tag der mündlichen Prüfung: 23. September 2016

Contents

List of Figures

List of Tables

Abstract

Touch input is the dominant input method on mobile devices these days. Interactive desk workspaces are a way to bring touch to the workplace using large interactive touch surfaces that allow users to directly manipulate digital content by touching it. In contrast to its success on mobile platforms, touch input has not explored its full potential on desktop workspaces. We believe this is because of two main reasons: First, similar to touch on mobile devices, interactive desktop workspaces lack haptic feedback when interacting with the surface. Second, large interactive surfaces pose ergonomic challenges. If oriented vertically, like the existing displays in a desk workspace, direct interaction is cumbersome and fatiguing since it requires users to constantly hold their arms in the air. However, if the touch surfaces are oriented horizontally, constantly looking at them is uncomfortable and leads to neck pain.

This thesis aims to explore interactive desk workspaces by overcoming both of these issues. To overcome the ergonomic challenges of large interactive surfaces we present BendDesk, an interactive desk workspace that combines a horizontal and a vertical interactive surface into one large desk workspace. In our in-depth analysis, we show that combining both surfaces has numerous benefits, but the issues with direct interaction on a vertical surface still prevail. Based on the results of this analysis, we improve our system by bringing the concept of indirect touch to the desk workspace. This interaction technique allows users to comfortably interact with digital content on the vertical surface through touch input on the horizontal surface, maintaining most of the benefits of touch input. Furthermore, we utilize the user's gaze to allow them to easily switch between direct and indirect touch input. To overcome the limited haptic feedback of touch screens, we introduce PERCs tangible, physical objects that allow users to manipulate digital content displayed on capacitive touch screens without the need to look at it.

Überblick

Heutzutage ist Touch-Eingabe die meist genutzte Eingabemethode auf Mobilgeräten. Um diese Eingabetechnologie in das normale Arbeitsumfeld einzubringen, können interaktive Schreibtische genutzt werden. Diese bestehen aus großen Touchscreens, die es dem Nutzer erlauben, digitale Objekte durch Berührung direkt zu manipulieren. Im Gegensatz zum Erfolg von Touch-Eingabe bei Mobilgeräten konnte sich dieses Prinzip am Standard-Arbeitsplatz bisher nicht durchsetzen. Dahinter können zwei Gründe stecken: Zum einen fehlt diesen interaktiven Schreibtischen ähnlich wie den Mobilgeräten mit Touch das haptische Feedback. Zum anderen stellen große Touchscreens die Nutzer vor ergonomische Probleme. Sind die Bildschirme für die Eingabe vertikal aufgebaut, ist die direkte Interaktion damit unangenehm und wirkt auf Dauer ermüdend. Bringt man die Bildschirme dagegen horizontal an, so müssen die Nutzer ständig nach unten sehen, was Nackenprobleme verursachen kann.

Das Ziel dieser Arbeit ist es daher, diese beiden Probleme zu lösen. Um die ergonomischen Schwierigkeiten zu überwinden, wird BendDesk vorgestellt, ein interaktiver Schreibtisch, der eine vertikale und eine horizontale interaktive Oberfläche zu einem großen kontinuierlichen Arbeitsplatz verbindet. In detaillierten Studien wird gezeigt, dass diese Kombination viele Vorteile birgt, aber die Probleme der Interaktion mit vertikalen Oberflächen weiterhin bestehen bleiben. Basierend auf diesen Ergebnissen wird das System um das Konzept der indirekten Touch-Eingabe erweitert. Dies erlaubt dem Nutzer bequem durch Interaktion auf der horizontalen mit digitalen Objekten auf der vertikalen Oberfläche zu interagieren. Dabei bleiben die meisten Vorteile der Touch-Eingabe erhalten. Außerdem wird die Blickrichtung des Nutzers dazu genutzt, zwischen der direkten und der indirekten Eingabemethode zu wechseln. Um das Problem des fehlenden haptischen Feedbacks zu lösen, werden PERCs vorgestellt: kleine physikalische Objekte, die es dem Nutzer erlauben, mit digitalen Objekten auf einem kapazitiven Touchscreen ohne Blickkontakt zu interagieren.

Acknowledgements

Writing a PhD thesis is an immense amount of work, which would not have been feasible without the contributions and the support of a lot of people.

My first thanks go to my advisor, Jan Borchers, who would always be ready to give advice or encouragement—whatever was needed most at the moment. More importantly even, he created the best working environment an aspiring PhD student could hope for; his group is a place of great individual freedom, noncompeting collaboration, and unquestioning support for interesting ideas.

I also want to thank Johannes Schöning, who was kind enough to act as my second advisor on this thesis. He was always open for questions, and also helped a lot with structuring my thesis.

In the past six years, I had a great time at i10, mostly because I had great colleagues who always gave good advice and helped me more than a few times. I want to especially thank Malte Weiss, Thorsten Karrer, Chat Wacharamanotham, Jan-Peter Krämer, Moritz Wittenhagen, Leonhard Lichtschlag, and Florian Heller for not only being great colleagues but also being good friends. It was a joy working with you guys! I also want to thank my new colleagues Christian Corsten, Christian Cherek, and Nur Al-huda Hamdan for helping out in several projects.

I am fortunate to have worked with many extraordinary students. Rene Linden, Jan Thar, Hauke Scharper, and Florian Busch, it was my privilege to work with you and to see you mature intellectually.

Kjell Ivar Øvergard allowed me to be a part of the MACS project and gave me the freedom to explore my ideas.

I also want to thank my good friend Christian Remy for the late night discussions about research.

Thanks to all the proofreaders who provided indispensable feedback: Shangning Postel-Heutz, Philipp Wacker, Jeff Blum, Christian Remy, and Chat Wacharamanotham.

Last but not least, I want to thank my wife, Melanie. During the entire time, she supported me emotionally and always gave me the strength and joy to keep up with my research. Also I want to thank my daughters Judith and Nora for providing distraction from work and reminding me of the important things in live.

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Definitions of technical terms or short excursus are set off in coloured boxes.

EXCURSUS:
Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition: *Excursus*

The whole thesis is written in American English.

Chapter 1

Introduction

In the 14th century the first desks were built to explicitly support reading and writing tasks. Figure 1.1.a shows an early example of a desk used by scholars to read books. Back then, before the invention of the printing press, these often massive and stationary desks had a tilted surface and were primarily used in monasteries and universities to both reading as well as manually copy books. During this time books were very heavy and consisted of large parchment volumes that were difficult to transport. The tilted surface of these massive desks provided a surface on which users could rest their arms, and also elevated the book to maintain a comfortable neck position, which is about 12 to 25° downwards [Lange, Windel, 2008].

The first desks were built in the 14th century to support reading and writing tasks.

After the introduction of the printing press, books became smaller, lighter, and cheaper, since they no longer had to be manually copied. Therefore, these massive and stationary desks evolved into portable boxes with a tilted writing surface. For centuries these boxes were used as desks, and were fitted with small drawers that could store several books along with various writing tools such as pen and ink [Banham, 1997]. One famous example of these portable desks is Henry VIII's writing desk from approximately 1525, shown in Figure 1.1.b.

During the 16th century, desks were mostly portable boxes.

During the 17th century, the writing desk became a piece of furniture that was used to write letters. (Fig. 1.1.c). It was

Figure 1.1: The evolution of desks: from the 14th century to today: **a)** A medieval writing desk from the 14th century. **b)** Henry VIII's portable writing desk from about 1525 [Wikimedia, 2008]. **c)** A writing desk from about 1700 [Bridge, 1989]. **d)** A secretary desk from 1790 [Hopfengaertner, 1790]. **e)** An office desk from 1910 [Bridge, 1989]. **f)** A modern office desk with a vertical display.

extended by stands and a larger horizontal writing surface.

Since the 17th century, desks evolved into large horizontal surfaces.

With the introduction of the secretary desk (Fig. 1.1.d) at the end of the 18th century, desks were mostly used for storing personal files. To be able to store more and more documents, desks became larger and included shelving or other storage space above the writing surface [Schaechter, 2010]

During the 20th century, desks became a commonly used workspace.

During the industrial revolution in the late 19th and early 20th century, the number of white-collar workers rapidly increased and the purpose of the desk changed from being a piece of furniture that belongs to an individual person into being a workspace in an office. These workers had to work with many documents such as contracts, reports, or other business documents, and desks had to be modified to support these tasks. During that time the basic form of a desk that we use today was designed: A large horizontal surface to hold a large number of documents on it, with a few drawers below the surface, shown in Figure 1.1.e [Banham, 1997]. In comparison to the tilted surface of the first desks from the 14th century, the horizontal surface of modern desks is rather optimised to store multiple documents and for writing tasks rather than for reading tasks. They allow users to rest their arms while writing, but force them to bend their neck in an uncomfortable position while reading [Sommerich et al., 2001].

With the introduction of desktop computers, desks were extended by a vertical surface.

Interaction with the vertical surface is limited by the capabilities of input devices such as mouse and keyboard.

With the introduction of the first affordable desktop computers such as the IBM PC and the Apple II in the early 1980s, the desk was extended to include a computer display placed vertically on top of the horizontal surface (Fig. 1.1.f). This design has the benefit of splitting input and output in a way that is generally more ergonomic than just a horizontal surface. Since the display is situated vertically in front of the user's gaze, it allows the worker to keep their head in a comfortable position [Fostervold et al., 2006] while resting their arms on the desk. However, this ergonomic advantage comes at the price of indirect input methods such as the keyboard and mouse. Most of these devices somehow limit how users can interact with digital content, and the display. For example, the mouse provides only a two dimensional position and binary buttons [Card et al., 1990] that can be used for interaction. Compared to the output that can be produced by a human hand with its "30 degrees

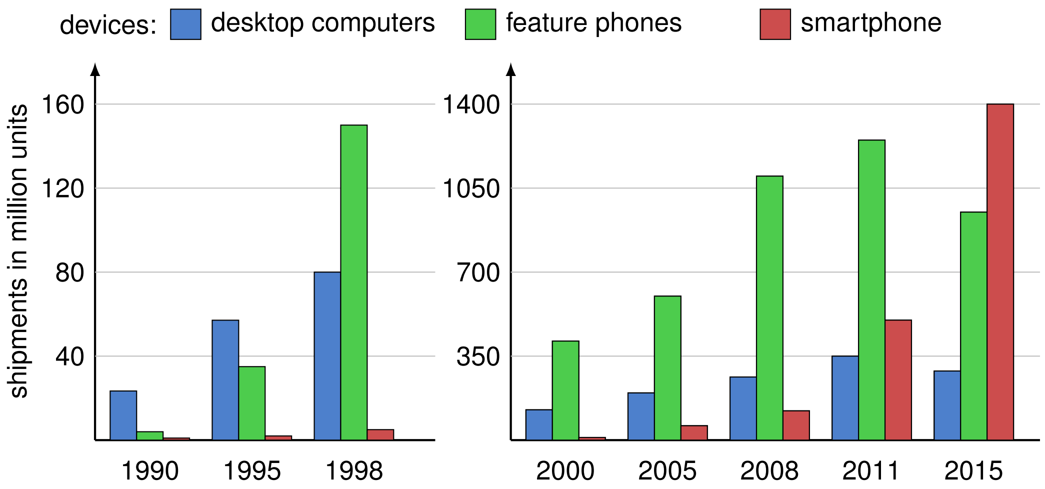


Figure 1.2: The number of worldwide shipped devices in units of millions between 1990 and 2015 [IDC, 2016; ITU, 2016].

of freedom" [Lin et al., 2000], these devices limit the user's expressiveness. While using these devices, users are forced to change their behaviour by executing commands sequentially rather than in parallel as they can when interacting with physical objects. For example, the mouse does not allow one to rotate, scale, and translate an object at the same time [Forlines et al., 2007].

Despite the fact that desktop computers are now used for many tasks including sketching images, 3D animations and complex simulations, the general input/output system remains largely unchanged over the last two decades.

The first mobile devices had two separate areas for input and output.

Interestingly, in contrast to computers, the interaction on mobile devices has changed dramatically, as shown in Figure 1.2. The first mobile phones such as the DynaTAC 8000x[1] had two separate areas: physical buttons for input and a small display for output. Over time, these *feature phones* became much smaller and their displays became larger. However, the basic concept of having separate areas for input and output persisted through the 1990s and the early 2000s, when these phones were the most common

---

[1] www.motorola.com/us/consumers/about-motorola-us/About_Motorola-History-Timeline/About_Motorola-History-Timeline.html

mobile devices. Even in 2015 these devices made up about 40% of all shipped mobile devices worldwide (Fig. 1.2). However, already in the early 1990s an alternative interaction concept for mobile devices was being introduced. The IBM Simon[2] (1992) is the predecessor of the modern smartphone, in that it incorporated a large touch-screen covering the entire device, used for both output and input. Users interacted with it by directly manipulating the graphical user interface (GUI) displayed on the screen using their fingers or special pens. During the 1990s and the early 2000s smartphonelike devices such as the Palm Pilot[3] were mostly used by business customers.

With the introduction of the Apple iPhone[4] in 2007, smartphones became consumer devices and replaced the feature phone as the dominant mobile device. In 2015 smartphones with large touchscreens made about 60% of all shipped mobile devices worldwide (Fig. 1.2).

Touch input became the dominant input technique for mobile devices.

One reason for the success of the smartphone over the feature phone is that the expressiveness of touch input is much larger than the expressiveness of a button interface [Card et al., 1990]. Furthermore, touch input allows the user to directly manipulate content in a very intuitive way. This directness has several benefits, the interaction is easy to learn, and users can immediately see if their actions are furthering their goals, and if not, simply change the direction of their activity [Shneiderman, 1982; Hutchins et al., 1985].

Touch input allows users to directly interact with digital content.

But touch-screens are not only beneficial for small devices; they also have many benefits for interacting with larger tablets or even desktop computers as shown by Forlines et al. [2007]. They demonstrated that in tasks which can be done with both hands, touch input users outperform users using the mouse. Furthermore, a study by Gindrat et al. [2015] suggests that constant touch-screen usage can improve hand and finger coordination. For these reasons, there have been some tentative steps towards touch input in the desk workspace. For example, systems such as the

Large touch-screens allow users to use bimanual interaction.

---

[2] time.com/3137005/first-smartphone-ibm-simon

[3] www.computerhistory.org/revolution/mobile-computing/18/321

[4] www.apple.com

HP Sprout[5] , which allows pen and touch input on the horizontal surface, and laptops such as the Lenovo Yoga[6] provide touch input in addition to the more typical trackpad or mouse input mechanisms. Nonetheless, indirect input devices such as mouse and keyboard are still the dominant input devices for interacting with desktop and laptop computers.

The DigitalDesk was the first touch-based interactive desk workspace.

Nevertheless, over the last 25 years the research community was actively developing touch-based systems that could be used in a desk workspace. As one of the first, Wellner [1991] presented a desk that combines the benefits of physical and digital documents. The DigitalDesk prototype was a conventional desk that was enhanced with a projector and a camera above the surface. It supported paper-based office work but also enabled users to interact with digital documents, which were projected onto the surface. In contrast to using the mouse and keyboard, digital content could be manipulated directly using pens or touch input. This work was the starting point of the research field on interactive tabletops.

Building these interactive tabletops was very complicated, and thus early publications largely focused on how to practically develop them. This changed in 2005, when J. Y. Han [2005] published a vision-based tracking technique that allowed practitioners to easily construct low-cost interactive tabletops. Since then subsequent papers were published that deal with engineering, interaction techniques, and applications of interactive tabletops. For example, the eLabBench project [Tabard et al., 2011] showed how a tabletop can be used to support work in a biology laboratory, and [Wigdor et al., 2007; Hardy, 2012] explored how an interactive tabletop can be used as a desk workspace.

Commercial tabletops, such as Samsung SUR40[7]or the models by SMART tables[8], have also been released. However, most of these systems use a vision-based approach to detect touch points. These use infrared sensors that can

---

[5] www8.hp.com/us/en/sprout/home.html
[6] www.lenovo.com/DE/Yoga
[7] www.samsung.com
[8] smarttech.com

detect objects such as a human hand that are very close to the surface by sensing their infrared light reflections. As described by Schöning et al. [2010], these systems suffer from several problems. They are often bulky and extremely sensitive to external light conditions such that most of these systems cannot be used in normal lighting conditions. Therefore, interactive tabletops such as the Microsoft Surface Hub[9]and 3M displays instead use capacitive touch tracking. Capacitive touch displays have the advantage that they detect touches by creating an electric field above their surface, which makes the touch tracking extremely robust. Furthermore, they provide a very smooth and pleasant touch experience.

More and more companies develop large capacitive touch displays.

Given the many benefits of touch input, the technical advances, and the fact that current input methods for desktop computers constrain interactions in important ways, we come to the main problem that we address in this thesis: What problems do we have to overcome to exploit the full potential of touch interaction for desktop workspaces? We hypothesize that two main reasons prevented this development:

1. Ergonomic challenges of touch interaction on large surfaces.
2. The limited haptic feedback in touch-based systems.

1) In contrast to small mobile devices, large touch-based surfaces in a desk workspace have several ergonomic challenges. If oriented vertically, like the existing displays in a desk workspace, direct interaction is cumbersome and fatiguing since it requires one to constantly hold their arms in the air as shown in Figure 1.3.a [Sears, 1991]. This effect is know as the *gorilla arm effect* [Hincapié-Ramos et al., 2014]. However, if the touch surfaces are oriented horizontally (Fig. 1.3.b), constantly looking at them is uncomfortable [Wigdor et al., 2007; Hardy, 2012] and can even lead to neck pain [Bachynskyi et al., 2015]. One proposal to address such ergonomic issues is presented by Tognazzini [1992]. His Starfire video prototype merges a horizontal

Directly interacting with a vertical surface leads to the gorilla arm effect.

Constantly looking at a horizontal surface leads to neck pain.

---

[9] www.microsoft.com/microsoft-surface-hub/en-us

Figure 1.3: The issues of vertical and horizontal touch-screens: Left: Directly interacting with the vertical screen is cumbersome and leads to the *gorilla arm effect* [Hincapié-Ramos et al., 2014]; Right: Constantly looking a the horizontal surface leads to neck pain [Bachynskyi et al., 2015].

Starfire merges a horizontal and a vertical surface into one large interactive surface.

and a vertical interactive surface into one large continuous workspace. This combination has the benefit that users can choose where to perform each task. For example, for a reading task the user can move digital documents to the vertical surface, while for a task that requires constant interaction the user can instead use the more comfortable horizontal surface. However, this system is only a video prototype and was not evaluated to determine its suitability as a desk workspace.

Users have to constantly look at their input since touch-screens do not provide haptic feedback.

2) Touch-based surfaces only provide the limited haptic feedback of touching a planar surface. This means that the user cannot "feel" the shape of on-screen objects, and does not receive haptic feedback when triggering actions. This requires users to constantly look at their input to make sure that they trigger the desired actions, otherwise they cannot tell whether they have hit the correct GUI object [Weiss et al., 2011]. For small mobile devices this is not an issue since the entire screen is always visible. However, for large tabletops, users are not able to see the entire surface within their field of view, which leads to the problem that if they are creating input in one area and the output is displayed elsewhere, they must constantly switch focus between the two areas [Weiss et al., 2011]. A common example for such a scenario is typing in a document where the on screen keyboard is displayed in one area of the surface and the doc-

Figure 1.4: Due to the lack of haptic feedback on interactive tabletops, users have to constantly look at their hands to make sure that they interact with the correct digital content. In this example the user cannot "feel" the keys. Therefore, he has to constantly switch his focus between the digital document (left) and the digital keyboard (right).

ument in another area. In a classical desk workspace, this problem is solved by the physical keyboard, which allows the users to feel the keys.

Tangibles on top of a touch-screen provide haptic feedback.

A common solution for improving the haptic feedback on interactive tabletops is to use tangibles on their surface [Fitzmaurice et al., 1995; Underkoffler, Ishii, 1999]. As shown in Figure 1.4, tangibles are physical objects that are detected by the tabletop while resting upon it. These tangibles allow the user to manipulate physical controls that directly interact with the digital content displayed on the surface. Weiss et al. [2009] presented a set of general purpose tangibles that mimic the classic control widgets of a GUI. In addition to providing haptic feedback, they showed that tangibles can also improve performance in specific tasks. For example, Jansen et al. [2012] showed that users are faster and more accurate using tangible sliders compared to touch-based sliders.

Detecting tangibles on capacitive touch-screens can only be done while a user is touching the tangibles.

However, until recently, tangibles could only be reliably used on visual tracking based tabletops, and therefore suffered from the disadvantages described above. There are some approaches such as Capstones [Chan et al., 2012] that attempted to bring tangibles to capacitive touch-screens. However, these approaches have the drawback that the tangibles are only detected when they are touched by a user. As soon as the user releases the tangible, the capacitance drops, and the system fails to detect the tangible—even if it remains on the surface. This makes it impossible to distinguish whether a tangible has been picked up and removed from the touch-screen, or whether the user has just let go of the tangible, leaving it on the touch-screen.

1.1 Contributions

In this thesis, we address the two main challenges that currently hinder that the full potential of touch and tangibles is exploited in desk workspaces, as described above. We address the ergonomic challenges (1) by presenting and analysing two desk workspaces that combine horizontal and vertical touch surfaces. Furthermore, we also address the issue of limited haptic feedback of touch surfaces (2) by introducing tangibles that can be reliably tracked on modern capacitive touch-screens. Specifically, our contributions are as follows:

- C 1 We address the basic ergonomic issues of a touch-based workspace by developing and exploring the *BendDesk* system, a curved interactive desk workspace based on the Starfire video prototype.
  - C 1.1 We explain the technical development of the BendDesk and how the system is able to project digital content and track touch input on a curved surface.
  - C 1.2 We analyse how users execute fundamental interactions such as dragging operations, and explain the differences between interacting with planar and non-planar surfaces.
  - C 1.3 We further analyse how users perceive such a curved system, and explain how this differs from planar surfaces.
  - C 1.4 In addition, we explore how users perform more complex gestures such as flicking and explain how this differs from planar surfaces.

- C 2 We address the ergonomic issue of directly interacting with a vertical surface by introducing the concept of indirect touch that allows users to use touch input on the horizontal surface to interact with digital content on the vertical surface.
  - C 2.1 We designed the basic interaction concept of indirect touch.
  - C 2.2 We compared multiple interaction techniques that overcome several issues of the interact touch concepts.
  - C 2.3 We analysed how gaze can be used to create an interactive workspace that combines indirect touch on vertical surfaces with simultaneous direct touch on horizontal surfaces.
- C 3 We address the tracking problems of tangibles on modern capacitive touch-screens by presenting PERCs, tangibles that can be reliably detected on capacitive touch-screens.
  - C 3.1 We developed the *PERC* tangibles.
  - C 3.2 We present a detailed technical evaluation of the PERC tangibles that shows that they can be reliably detected on a variety of different capacitive touch-screens.
  - C 3.3 We present a user study that shows that the haptic feedback provided by the PERC tangibles improves user performance in direct and indirect interactions.

1.2 Structure

The thesis is structured as follows:

- Chapter 2 presents the BendDesk system, its development, and describes five user studies that explore how users interact and perceive such a curved interactive desk. It concludes with a discussion of its drawbacks and limitations. In this chapter we present the contributions C 1.1 – C 1.4. Parts of this chapter have been published in the Diploma thesis of the author [Voelker, 2010], at ITS 2010 [Weiss et al., 2010a] and CHI 2012 [Voelker et al., 2012].
- Chapter 3 introduces the concept of indirect touch input and how this concept allows users to use the horizontal surface to create touch input on the vertical surface. Furthermore, it presents and evaluates a prototype that uses the user's gaze to combine simultaneous direct and indirect touch input. In this chapter we present the the contributions C 2.1 – C 2.3. Parts of this chapter have been published at CHI 2013 [Voelker et al., 2013d] and SUI 2015 [Voelker et al., 2015b].
- Chapter 4 explains the PERC tangibles, their development and how they are detected by a capacitive touch display. It describes an extensive technical evaluation of the PERC tangibles that shows that they can reliably be detected on a variety of modern capacitive touch displays. Finally, the chapter presents a user study that confirms that the haptic feedback provided by tangibles improves user performance. In this chapter we present the the contributions C 3.1 – C 3.3. Parts of this chapter have been published at ITS 2013 [Voelker et al., 2013c], ITS 2015 [Voelker et al., 2015c], and UIST 2015 [Voelker et al., 2015a].
- Chapter 5 concludes the thesis. We summarize our contributions and provide an outlook on future work.

Chapter 2

Exploring Interactive Desk Workspaces

The typical desk workspace consists of a horizontal and a vertical surface.

As described in the introduction, a typical computer workplace integrates horizontal and vertical surfaces into a workspace. It encompasses one or more vertical displays plus a larger horizontal area that contains input devices. Each of these surfaces has different ergonomic advantages and is used for different tasks. While the horizontal surface allows the user to rest their arms while interacting with objects such as a mouse, keyboard, paper-based documents, and everyday objects, the vertical surface allows the user to look at digital content displayed on it, maintaining a comfortable neck position [Lange, Windel, 2008].

Several systems have been proposed that combine vertical and horizontal interactive touch-surfaces within a single desk environment (e.g., Coldefy, Louis-dit-Picard [2007] and Luff et al. [2006]). They provide a large interactive area

---

Publications: The work in this chapter is a collaboration with Malte Weiss and Christine Sutter. The author was responsible for developing the hardware, writing most of the software, designing the experiments, and analyzing data from the experiments. Part of this work was first published as a paper at the ITS 2010 conference [Weiss et al., 2010b], as a Diploma Thesis in 2010 [Voelker, 2010], and as a paper at the CHI 2012 conference [Voelker et al., 2012]. The author is one of the main authors of the ITS paper, the author of the Diploma thesis, and the main author of the CHI 2012 paper. Several sections of this chapter are taken from these publications.

Figure 2.1: The Sun Starfire video prototype envisioned an interactive desk workspace that consists of one continuos interactive surface [Tognazzini, 1992].

Gaps between adjacent displays suggest isolated interactive areas, which leads to interaction issues.

and allow to move digital objects across multiple displays. However, such systems typically suffer from a lack of spatial continuity. According to the Gestalt Law of Closure [Wertheimer, 1923], gaps between adjacent displays suggest isolated interactive areas. Other design rules may be violated that are useful in screen design, e.g., the Law of Proximity, violated when objects belonging together are separated across the gap. This leads to users having problems finding objects, or else while focusing on one screen even forgetting that the other screen exists [Morris et al., 2008]. Furthermore, splitting objects across bezels impairs the users tunnel steering performance [Bi et al., 2010]. Finally, such setups limit the applicability of direct manipulation, as movement trajectories are interrupted when dragging a finger or pen from screen to screen.

To avoid this issue, Tognazzini [1992] envisioned the Starfire, an interactive desk workspace that consists of a single interactive surface. As shown in Figure 2.1, his proposed system consists of a horizontal and vertical surface that are merged by a curved area into one large workspace.

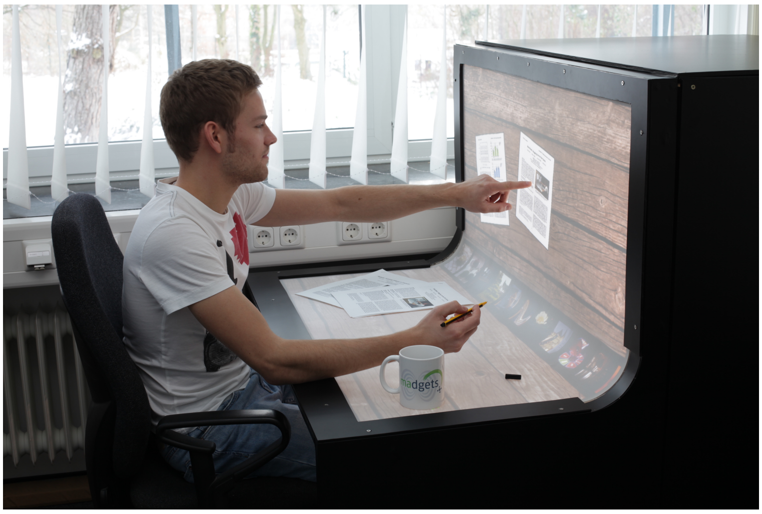


Figure 2.2: The BendDesk workspace combines a horizontal and a vertical surface into one large interactive surface.

In comparison to a system that consists of two separated screens, this approach has the benefit that users perceive this workspace as one continuous surface. Furthermore, the curve allows users to easily move objects from the horizontal to the vertical surface or vice versa by simply dragging them across the curve. However, as mentioned in the introduction, this system was only a video prototype and was never developed and evaluated. So it is unclear how a user would interact with such a system and whether such a system makes sense as a desk workspace.

The Starfire system is an interactive desk workspace that consists of one contiuous interactive surface.

To determine whether such a curved surface can be used as an interactive desk workspace, we developed the BendDesk system, an interactive desk workspace that merges a horizontal and a vertical multi-touch-surface into one large curved interactive surface. As shown in Figure 2.2, the BendDesk is very similar to the Starfire, since it provides a large interactive area within the user's reach and allows uninterrupted, seamless dragging gestures across the entire surface.

The BendDesk system is inspired by the Sun Starfire prototype.

In this chapter, we explain the technical development of the BendDesk system (C 1.1) followed by five studies that analyse the fundamental interactions with such a curved desk workspace. We first analyze basic motoric operation (C 1.2) and basic perceptual aspects (C 1.3). Finally, we evaluate how users perform more complex flicking gesture (C 1.4).

2.1 Related Work

We are not the first who developed a interactive desk workspace. In the following, we will give a overview over the related work in this area and point out the problems of the existing system. The publications are sorted into the following categories: "Single-Plane Interactive Desk Workspaces" , "Multi-Plane Interactive Desk Workspaces", and "Non-Planar Interactive Desk Workspaces".

2.1.1 Single-Plane Interactive Desks

Wellner developed the first interactive desk workspace in 1991.

As mentioned in the introduction, one of the early interactive prototypes is the DigitalDesk Calculator [Wellner, 1991]. This system projected digital documents using a projector that was positioned above a normal wooden desk, projecting downwards. Users could interact with digital documents by directly touching them. A camera above the desk was used to recognize the position of the users hand on the table. The main problem facing this system was determining the moment when the user touched an object, since from its point of view the camera could not distinguish between hover and touch. To solve this problem, they attached a small microphone under the table, that could detect the noise created by a touch.

Buxton suggested interactive displays should only be used for highly specialized tasks.

In a journal article by Buxton et al. [2000] the authors explored the use of large displays in automotive design processes and how they could be improved with, e.g., multi-touch displays. They showed that a traditional drafting table could be replaced by a multi-touch system. However, they also pointed out that each task in the automotive design process poses very different system requirements and that one interactive display for all tasks would be not suitable. They proposed to use several interactive desk workspaces, with each used for a highly specialized task.

In studies by Wigdor et al. [2007] and Hardy [2012], they observed one user using a horizontal multi-touch desk workspace as his workplace over a longer time period. Both studies showed that users used this desk not only as a computing device but also as furniture for placing everyday objects. Furthermore, they showed that the use of bimanual interaction saves time for some tasks. Wigdor et al. [2007] compared the user's email written on the multi-touch display with his emails written on a normal PC and found no significant differences in text length. This is in contrast to the study by Ryall et al. [2006] where they showed that using a soft keyboard on a multi-touch display is more error-prone and slower than classic keyboards for text entry. However, Ryall et al. [2006] also reported that exclusively using a horizontal surface is very uncomfortable, especially in long term usage.

Interactive desk workspaces that only consist of horizontal surfaces are uncomfortable in long term usage.

All of the systems and studies in this section show that a desk workspace consisting of multi-touch-surfaces could be useful. However, as shown by Ryall et al. [2006] the form factors such as the size and the arrangement of the system impact the usability of such interactive workspaces dramatically.

2.1.2 Multi-Plane Interactive Desks

In the last ten years, the specific properties of horizontal and vertical interactive surfaces have received great interest in the research community. Morris et al. [2007] showed that horizontal surfaces are more appropriate for annotation and pen-based note-taking, while vertical displays support reading and intensive writing tasks using keyboards. Since no display seems appropriate for all potential tasks, Morris et al. [2007] propose a hybrid system. In a follow up paper, Morris et al. [2008] report on a field study involving multiple horizontal and vertical screens. Although participants were enthusiastic about the extra space, almost all participants perceived the horizontal and vertical

Interactive surfaces in a desk workspace should fit into the ecologies of objects.

screens as isolated areas. Other studies [Morris et al., 2008; Müller-Tomfelde et al., 2008] indicate that interactive surfaces should allow tilting to increase comfort. However, Morris et al. [2008] also pointed out that desk environments should fit into the ecologies of objects. For example, a table should allow users to put down everyday objects, such as coffee mugs or documents. The authors point out the "dual use" of interactive tabletops as computing devices and as pieces of furniture. In their study, the participant tended to tilt the table at an angle that prevented objects from falling off the table.

Combining horizontal and vertical surfaces has been shown as useful for several tasks.

The combination of horizontal and vertical interactive surfaces has mostly been used in collaborative workspaces. While horizontal surfaces are suitable for face-to-face group work, vertical surfaces provide an overview of information shared among groups. Many systems have been developed that integrate vertical and horizontal interactive surfaces into collaborative workspaces in order to add digital capabilities [Everitt et al., 2006; Izadi et al., 2003; Rekimoto, Saitoh, 1999; Wigdor et al., 2009]. The combination of both surface types has also been applied to remote desk environments. For example, the Agora system [Luff et al., 2006] and DigiTable [Coldefy, Louis-dit-Picard, 2007] provide an interactive horizontal surface for a private document space and a vertical surface displaying a remote person via a video conferencing system. However, in most systems the vertical surface was only used as output.

Also, in all of these systems both surfaces are not spatially aligned which leads to the problem that users perceive these surfaces as two separated systems and not as a single connected one [Chang et al., 2002]. Finally, these setups limit the possibility of direct manipulation, as movement trajectories are interrupted when dragging a finger or pen from screen to screen. Bi et al. [2010] showed that even very small non-interactive strips between displays (bezels) can influence the users in their search strategies. Users tend to apply a display-by-display search strategy and even forget that the other display exists. Furthermore, they showed that bezels hinder the straight-tunnel steering performance across these bezels.

To improve the sense of visual as well as spatial continuity and connectivity across displays that are clearly spatially separated, Wigdor et al. [2006] proposed several techniques, such as the repeated patterns or techniques such as *World in Miniature (WIM)*. WIM presents a miniature version of the vertical display on the horizontal display, that allows users to control the vertical surface with this miniature version. Furthermore, this offers users an easy way to move objects between the displays by putting an object into or out of the miniature view.

Flux can be adapted to the current needs by tilting the entire system.

A completely different approach to combine a vertical and a horizontal area into one system is the FLUX system by Leitner et al. [2009]. They proposed a system that can be transformed from a vertical surface into a horizontal one by tilting it. They wanted to create a very flexible system, that can be used for many different tasks. They also established that many tasks have different requirements that cannot be fulfilled by one static surface. So they provide a system that can be adjusted for each task. But this implies that each time users change their tasks the system has to be adjusted.

2.1.3 Non-Planar Interactive Desks

The Curve system is a curved interactive desk workspace.

At the same time we presented BendDesk, Wimmer et al. [2010] presented Curve, a curved interactive workspace (Fig. 2.3). In their follow up studies they focused on basic interaction such as pointing and dragging. They showed that pointing tasks on these curved surfaces can be described by Fitts' Law [Hennecke et al., 2013a]. They also analysed different types of curves that can be used to connect the horizontal and the vertical surfaces and how these influence the dragging performance of the users [Hennecke et al., 2012]. In this study they showed that the dragging performance was not influenced by how the vertical and the horizontal surfaces are connected.

However, most of these studies focused on high level applications and explored for which kind of task these curved system can be used. Schwarz et al. [2012] presented a concept for how such a curved surface can be used for fo-

Figure 2.3: The Curve system combines a horizontal and a vertical surface into one large interactive surface. [Wimmer et al., 2010]

cus+context tasks. In this work they suggested to use the horizontal surface to display important information and to use the curve and the vertical surface as a viewport into a 3D scene that provides the context for the information displayed on the horizontal surface. In another work by Hennecke et al. [2013b], which was done in collaboration with the author of this thesis, they used the same concept to create an immersive video conferencing experience. A very similar approach was also presented by Benko et al. [2012]. In this work they used a curved surface to create an augmented reality workspace.

Curved surfaces have been shown useful especially for focus+context tasks.

Despite the fact that several studies investigated how users interact with these curved surface, there are still several open questions about how users perform basic operations such as dragging or how users perceive information on these non-planar desk workspaces.

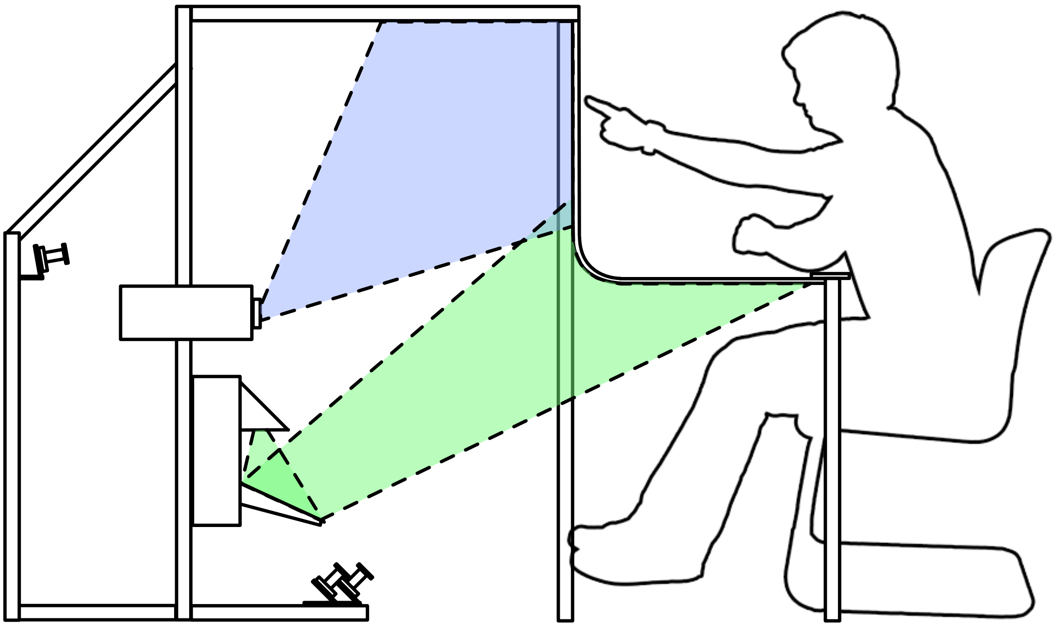


Figure 2.4: The placement of projectors and cameras inside the BendDesk. The colored cones show the projection areas of both projectors [Voelker, 2010].

2.2 Hardware Setup

As mentioned above the form factor of an interactive desk workspace has a very large influence on the usability of such a system. Therefore, we envisioned the BendDesk as an interactive desk workspace that supports interaction with digital documents but also respects the nature of traditional desks. As with a traditional desk, the user should be able to sit in a comfortable position and reach the entire input area without much effort. We conducted preliminary user tests [Voelker, 2010] on an adjustable table prototype to find the appropriate depth for the vertical surface. In these tests, users performed pointing and dragging tasks where the depth of the vertical surface was varied. The results of this tests suggest that the most comfortable and usable position for the vertical surface is about 50 cm away from the desk edge.

The BendDesk should allow user to comfortable sit in front of it.

Although there is evidence that tilted surfaces yield high acceptance for specific tasks as suggested by Buxton et al.

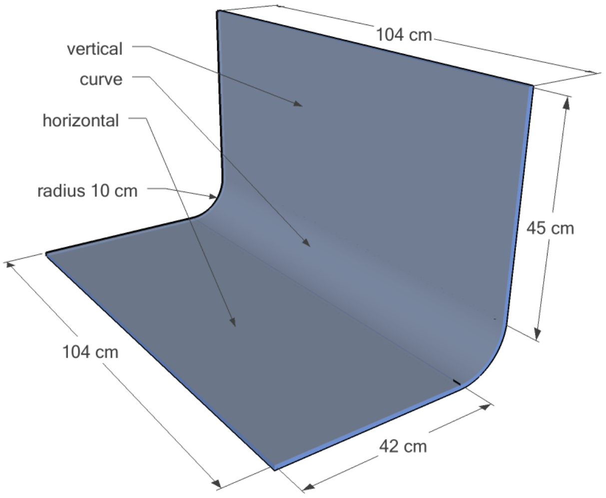


Figure 2.5: BendDesk's curved surface with its three areas: the horizontal surface, the curve, and the vertical surface [Voelker, 2010].

We avoided a tilted vertical surface for ergonomic reasons.

[2000], we intentionally avoided them for two reasons: Firstly, we consider the support of the ecology of (everyday) objects as crucial. With the exception of special purpose desks, such as drawing tables, office desks are usually horizontal because people need to place physical objects on them. In contrast, the possibilities of placing objects onto a tilted surface, even at small angles, are limited. Secondly, tilting the vertical surface backwards would reduce its reachability at the top.

The interactive surface of the BendDesk system is a curved one square meter large acrylic surface.

Figure 2.4 shows the hardware setup of the BendDesk. It consists of one 104 cm × 104 cm acrylic surface that is bent to yield a horizontal and a vertical surface, seamlessly merged by a curve. As shown in Figure 2.5, this curve separates the interactive surface into three interactive areas: a vertical surface (100 cm × 43 cm), the *curve* (100 cm × 16 cm) with a radius of 10 cm, and the horizontal surface (100 cm × 40 cm). We choose a radius of 10 cm to provide a large planar interactive surface while allowing a comfortable dragging through the curve [Voelker, 2010]. These measurements provide a large reachable area while maintaining a comfortable sitting position. We also added

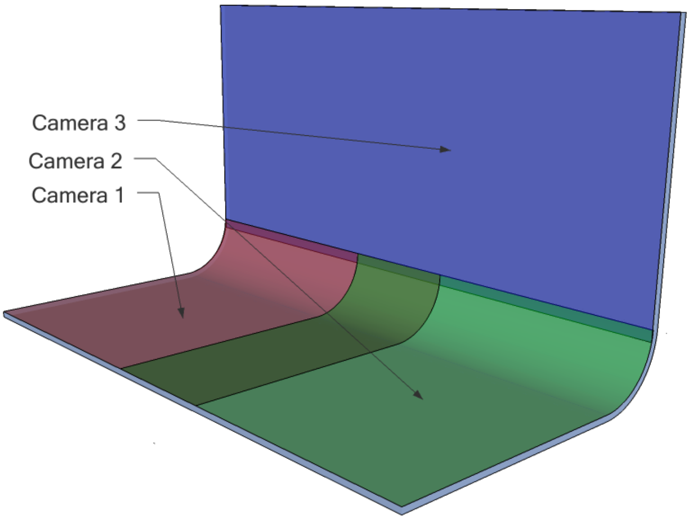


Figure 2.6: The tracking areas of the cameras. Each camera tracks a specific part of the interactive surfaces and detects touches inside this area [Voelker, 2010].

a raised non-interactive strip in front of the board that fixes the acrylic, and also provides an area for the user to rest her hands without creating unintentional input.

Two short-throw projectors behind the surface render the graphical user interface on a diffusor which is placed on top of the acrylic surface. Each projector operates with a resolution of 1024 × 768 pixels. An Optoma EX525ST projector displays the GUI on the vertical surface, while a NEC WT615 projector shows the interface on the curve and the horizontal surface.

Two DLP short-throw projectors are used to display the interface.

To detect touch input we use Frustrated Total Internal Reflection (FTIR) [J. Y. Han, 2005]. The acrylic is surrounded by a closed strip of LEDs that feed infrared (IR) light into the surface. To use FTIR, we apply a thin silicone compliant layer between the acrylic and the diffusor. Three Point Grey FireFly MV cameras with attached IR filters track touches on different parts of the interactive surface as shown in Figure 2.6. Each camera runs at 120 fps and a resolution of 640 × 480 pixels.

Touch points are detect using FTIR.

2.3 Software

To use BendDesk as a desk workspace the software has to do two major tasks: The first is to control how the interface is rendered onto the curved surface. The second is to detect the user's touch input. Both the visual output and the touch tracking system have to be calibrated.

2.3.1 Visual Output

The BendDesk GUI has a resolution of 1024 × 1024 pixels.

The main problem when displaying the interface on a curved surface is that the projectors can only project with a substantial distortion onto this surface. To overcome this problem, we have to compensate for this distortion by distorting and scaling the projection in a way that the GUI displayed on the surface is distortion free. To achieve this, we render the entire GUI into an offscreen buffer. Subsequently, each projector renders a part of this buffer on a bicubic spline patch that compensates for the respective distortion. The resulting GUI has the resolution of 1024 × 1024 pixels, or approximately 26 dots per inch (DPI).

Projector calibration

The projectors need to be calibrated manually.

We employ a manual calibration process to compute the spline patches for each projector. A paper calibration sheet with an imprinted uniform grid of 32 × 32 dots is placed onto the interactive area. Each printed dot with index $(x, y) \in \{0, 1, ..., 31\}^2$ on the sheet maps to a pixel position $P(x, y)$ in the GUI space:

$$P : \{0, 1, ..., 31\}^2 \rightarrow [0, 1024)^2$$

The result of a successful calibration process is a *projected* dot pattern that exactly matches the nodes on the paper grid. That is,

$$D_i(x, y) = P(x, y) \quad \forall (x, y)\text{: frustrum}_i(x, y) = 1$$

where $D_i(x,y)$ is the mapping of projected grid dots to GUI coordinates for each projector $i \in \{1,2\}$, defined analogously to $P(x,y)$. The function $\text{frustrum}_i(x,y)$ indicates whether the paper grid point (x,y) is inside the frustrum of projector i or not:

$$\text{frustrum}_i(x,y) = \begin{cases} 1 & P(x,y) \text{ in frustrum of projector } i \\ 0 & \text{otherwise} \end{cases}$$

Each projector is calibrated separately. When starting the calibration for projector i, it displays a 32×32 uniform grid that covers the entire screen space of the projector. Hence, each projected grid point is shown at a certain *screen coordinate* $S_i(x,y)$ with

A bicubic spline mesh with 32×32 dots is used to calibrate the projectors.

$$S_i : \{0,1,...,31\}^2 \to [0,1024) \times [0,768).$$

Thereafter, the user deselects all grid rows and columns that do not map to rows in the calibration sheet (defines $\text{frustrum}_i(x,y)$). In our case, this means that she deselects the bottom 18 rows for the top projector and the top 16 rows for the bottom projector. Then the user moves the projected grid dots until they fit with the corresponding points on the paper sheet ($D_i(x,y) = P(x,y)$). We implemented a set of transform tools to speed up this manual process.

Finally, when the user confirms the calibration, a sub-grid is extracted that contains all grid dots inside the frustrum of the projector ($\text{frustrum}_i(x,y) = 1$). The corresponding screen coordinates $S_i(x,y)$ then represent the interpolation points of the bicubic spline patch, whereas the values $D_i(x,y)$ are used as texture coordinates to render this part of the interface on the table. This technique easily scales up to setups with more than two projectors, while the process has to be performed only once for each configuration. Figure 2.7 illustrates the manual screen calibration.

Rendering pipeline

To render the GUI on BendDesk the software creates an off screen buffer object and an empty texture with the size of the GUI, which in our case is 1024 px × 1024 px. Then,

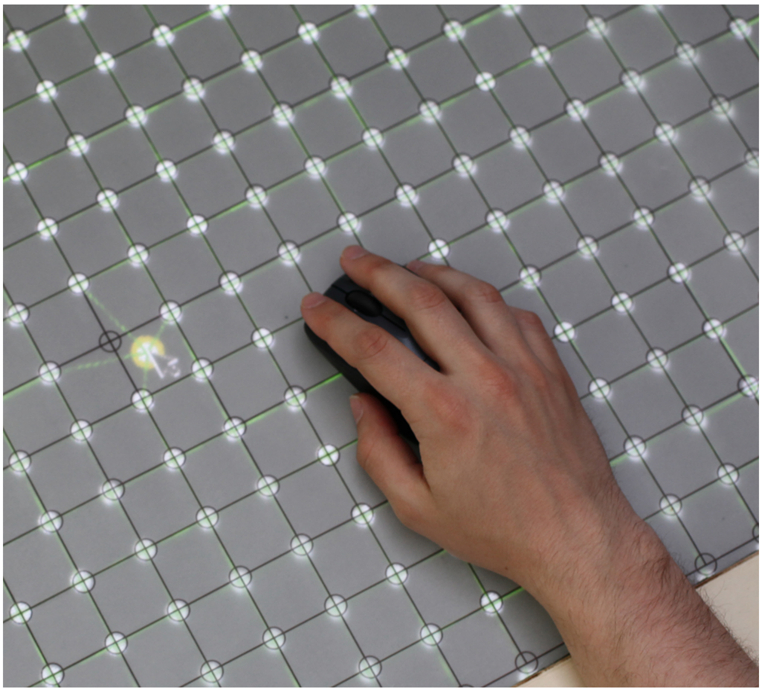


Figure 2.7: To display the interface correctly on to the curved surface, each projector needs to be calibrated manually. For this calibration, a sheet of transparent paper with a grid of dots is placed on the BendDesk. The same pattern is also shown by the system. To calibrate the system, each displayed dot needs to be moved to its corresponding point on the sheet of paper.

The interfaces is rendered into a texture, which is then placed on top of the calibrated spline mesh.

for each projector it reads the spline patch from the calibration process and extracts a high resolution quad patch with texture coordinates. The texture coordinates map the GUI coordinates into the screen coordinates of the projector. The GUI is rendered into the off screen buffer that is bound first to the empty texture and then distributed to the projectors that render the texture on the respective spline patches. This process is hidden by the software so an application is directly rendered into the GUI coordinate space without paying attention to know which projector is displaying which part of the GUI.

2.3.2 Tracking Touches

Our camera setup detects touches on the entire interactive surface, with each camera covering a specific area. Figure 2.6 shows the tracking areas. We used a straightforward algorithm based on a connected component analysis [Dillencourt et al., 1992] after background subtraction. After detecting spots for all cameras, their coordinates are transformed from camera to GUI coordinates. Similar to the screen calibration, we use a bicubic spline patch for this mapping, as described below. Finally, the transformed spots are sent to the application as touch events in GUI coordinates. Note that all camera fields of vision overlap to ensure continuous tracking between the areas. If multiple spots are mapped to nearly the same GUI position, they are merged into a single touch event by averaging their coordinates.

Touches are detected by a connected component analysis.

To ensure the registration of touch events between successive frames, we used a predictive tracking algorithm. That is, for each touch T at position p, we track its velocity and acceleration and extrapolate p to its anticipated position p' in the subsequent frame. If there is a touch close to p' in the next frame, we assume that it is a translated version of T. In practice, the use of predictive tracking dramatically improves the touch registration on our system and reliably nearly eliminates users "losing" dragged or transformed objects.

Quadratic predictive tracking is used to improve the touch detection.

Camera calibration

For each camera $j \in \{1, 2, 3\}$, our calibration process creates a mapping from camera coordinates to global GUI coordinates. When starting the calibration, our software displays an $N \times M$ uniform grid with GUI coordinates $G(x, y)$ that covers the interactive surface. Note that this requires a correct projector calibration.

In the first step, the calibration creates a mapping C_j from

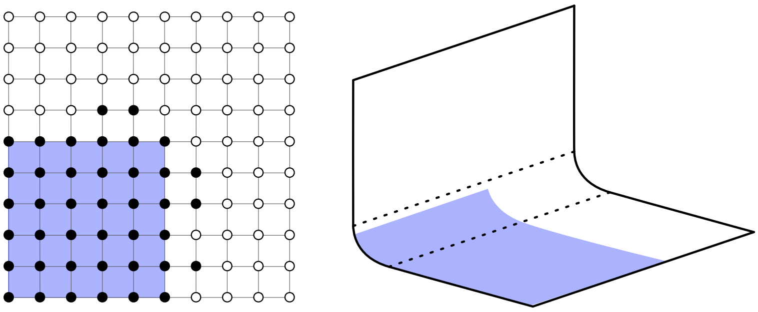


Figure 2.8: Extraction of a spline patch to map from GUI to camera space. Left: Largest rectangle containing visible dots is extracted. Right: Corresponding area on table [Voelker, 2010].

GUI grid point indices to camera pixels:

$$C_j : \{0, 1, ..., N\} \times \{0, 1, ..., M\} \rightarrow [0, 640) \times [0, 480)$$

where N and M denote the grid resolution. The calibration needs to find the camera pixels that match the GUI grid points. Accordingly, we need to find values $C_j(x, y)$ for all (x, y) with $\text{visible}_j(x, y) = 1$, where

$$\text{frustrum}_i(x, y) = \begin{cases} 1 & G(x, y) \text{ visible from camera } j \\ 0 & \text{otherwise} \end{cases}$$

All cameras are calibrated at the same time. The system successively highlights each grid point. For each highlighted dot $(\bar{x}, \bar{y})$, the user touches the surface at that position and then confirms with a button click on a wireless control. Now, our algorithm stores which cameras detected the resulting FTIR spot, i.e., $\text{visible}_j(\bar{x}, \bar{y})$, and at which position, $C_j(\bar{x}, \bar{y})$.

Each camera is calibrated to a subset of the spline mesh of the camera calibration.

As illustrated in Figure 2.8, this manual process yields a visibility map, visible_j, for each camera. We extract the largest rectangle that only contains visible spots by solving the *Maximum Empty Rectangle problem* [Naamad et al., 1984]. Similar to the screen calibration, the extracted point indices together with $G(x, y)$ and $C_j(x, y)$ represent the interpolation points for a bicubic spline patch $\mathcal{P}$ that maps from GUI to camera coordinates for camera j.

However, we need the inverse mapping to detect which positions on the GUI are touched. Our algorithm computes the map C_j^*

$$C_j^* : \{0, 1, ..., 639\} \times \{0, 1, ..., 479\} \rightarrow [0, 1024)^2$$

by uniformly evaluating the patch $\mathcal{P}$ with a high sampling rate. For each sample, the source GUI position is stored at the target camera position in C_j^*. This yields a discrete inverse map for camera j. Afterwards, if a spot is visible in camera j, we can read its GUI position from C_j^*. In order to avoid jitter, we employ bilinear interpolation for this lookup.

2.4 Evaluation

The BendDesk systems differs from other existing touch based desk workspaces because it consists of three seamlessly connected surfaces. Additionally, it introduces a curved area as a touch sensitive interactive surface. Therefore, it is unclear how user can interact with such a system. So we need first to understand how users interact with such a system on a very low level to then be able to develop applications that can be used to solve task in a desk workspace. To explore how users interact with this new system we conducted five user studies. With these studies we want to answer the following questions:

To be able to develop applications for the BendDesk, we first need to understand how people interact with such a system.

- How do users perform simple motoric operations such as dragging on each of the three areas and how does the performance differ between the areas?
- How do users perform dragging operations that involve all three areas?
- How do users perceive information displayed on the surface?
- How do users perform more complex gestures like flicking on such a system and how does the curve influence the user's performance? If there is a performance difference, why is this the case?

2.4.1 General procedure

All five studies where carried out in a dimly lit room, where participants sat in front of the BendDesk. Each task type was introduced by a test trial to familiarize participants with the new task. The task instructions were standardized and emphasized to solve the tasks as fast and accurately as possible.

2.4.2 Study 1: Dragging

The first study investigated dragging performance across the different interactive areas of BendDesk and compared dragging performance across the curve to dragging on the horizontal and vertical areas.

Participants

18 participants (16 males), aged between 24 and 32 years (mean age 27 years) took part in the study. They did not receive any compensation, but we raffled a $25 gift coupon among them. 15 participants were computer scientists, two were school teachers, and one was a mechanical engineer.

Tasks

The users were asked to drag an object into a target object on all three areas of the BendDesk system.

Figure 2.9 shows the experimental task and the conditions. The system displayed the source, a white colored square with a side length of 50 px (4.88 cm), and the target, a white frame of the same size. Both were vertically arranged with a distance a distance of 150 px (14.64 cm) between them. The participant had to drag the source quad onto the target using her index finger. After successfully matching source and target, with a tolerance of 10 px (0.98 cm), the interactive area went blank and the next trial was shown. They appeared in three different areas (on the horizontal surface,

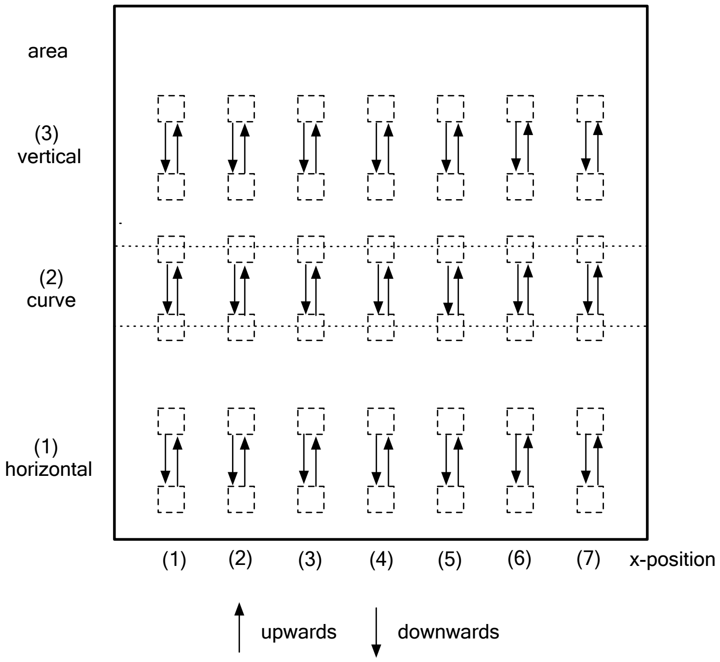


Figure 2.9: Experimental design of the vertical dragging task. The users were asked to drag an object into a target object on all three areas of the BendDesk system.

on the curve, or on the vertical surface), and dragging direction from source to target was either upwards or downwards. This resulted in 3 (area) × 2 (dragging direction) experimental conditions. We also controlled the distribution of trials across the surface by presenting trials on seven different x-positions with two repetitions each. The order of trials was randomized. Participants completed 84 trials with their dominant hand and another 84 trials with their non-dominant hand. This yielded a total number of 168 dragging operations per participant. The dragging duration was defined as the interval from touching the source until correctly releasing it inside the target (given in ms). Dragging trajectory covered the observed length of the finger's movement path, again from touching the source until correctly releasing it inside the target (given in px).

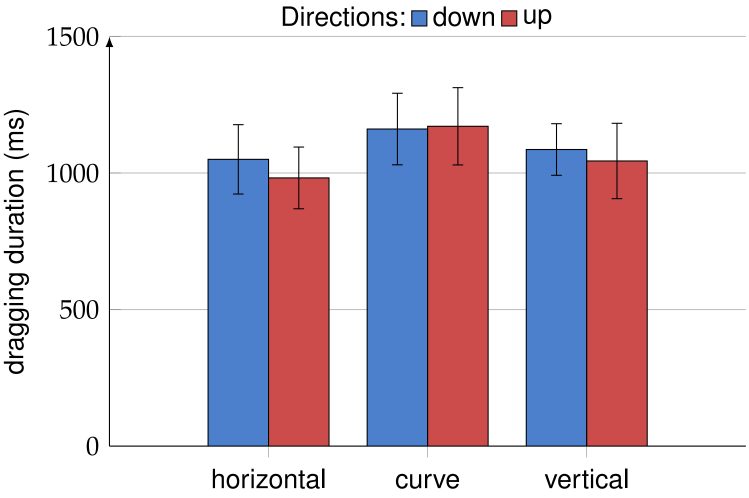


Figure 2.10: Dragging duration depending on area and direction. Dragging on the curve was significantly slower then on the horizontal and vertical areas. Whiskers denote the 95% confidence interval.

For this study we hypothesized the following outcomes:

H1 *(horizontal vs. vertical)*: Dragging (a) duration and (b) trajectory are shorter on the horizontal surface than on the vertical one.

H2 *(planar vs. curve)*: Dragging (a) duration and (b) trajectory are shorter on planar surfaces than on the curved area.

H3 *(down vs. up)*: Dragging (a) duration and (b) trajectory are shorter when moving upwards in GUI coordinates than when moving downwards.

Results

Dragging on the curve was significantly slower than on the horizontal and vertical areas.

The data were analyzed for each of the dependent variables with 3 × 2 repeated measures ANOVA with the within-subject factors *area* and *direction*. Dragging durations are depicted in Figure 2.10. The ANOVA revealed a significant main effect of the factor *area* $(F(2,34) = 14.20; p < 0.01)$. The dragging durations inside the curve

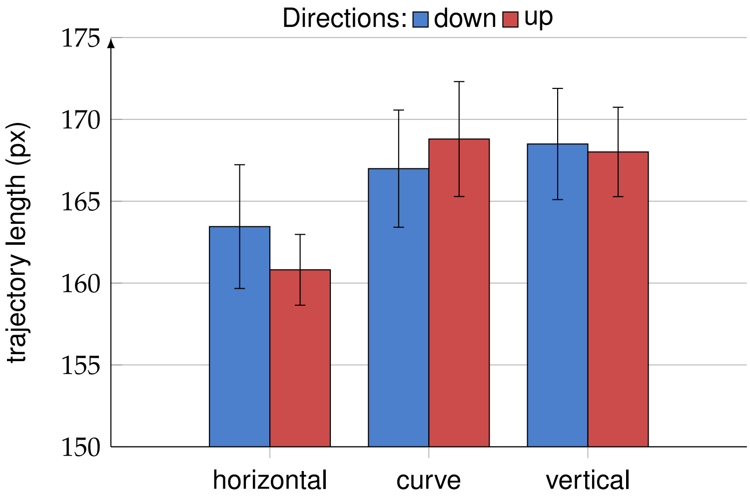


Figure 2.11: Length of dragging trajectory depending on area and direction. The dragging trajectories on the horizontal surface were significant shorter compared to the others. Whiskers denote the 95% confidence interval.

(mean 1166 ms) were 14% (150 ms) longer than the dragging durations on the horizontal area (mean 1016 ms) and 10% (110 ms) longer than the dragging durations on the vertical area (mean 1056 ms). Other main effects and the interaction were not significant.

Figure 2.11 shows the length of dragging trajectories. A repeated measurements ANOVA showed a significant main effect of the factor *area* $(F(2, 34) = 28.84; p < 0.01)$. The dragging trajectories inside the curve (mean 167 px) were 3% (5 px) longer than the dragging trajectories on the horizontal area (mean 162 px). However, dragging through the curve was equally long compared to vertical dragging (mean 168 px). Furthermore, for the horizontal surface, but not for the other areas, upward dragging was significantly shorter than downward dragging. This yielded a significant interaction $(F(2, 34) = 4.73; p < 0.05)$. The main effect of the factor *direction* was not significant.

The dragging trajectories on the horizontal surface were significantly shorter compared to the others.

To sum up, when comparing horizontal and vertical dragging (H1) the results clearly showed shorter trajectories for operations in the horizontal surface. This is in accordance with H1. However, dragging duration was comparable for both planar surfaces. Second, our results support H2: dragging on a planar surface is faster than dragging across the curve. Finally, we hypothesized more efficient upward than downward movements (H3). Although performance data on both planar surfaces slightly hint at an advantage for upward movements, this was only significant for finger trajectories done on the horizontal surface. Thus, overall the data did not confirm H3.

Discussion

Dragging on the horizontal surface is executed by only using the wrist and finger joints and is therefore more accurate.

This first study reveals two main findings: First, dragging on the horizontal surface is more accurate than on the vertical surfaces. This can be explained by how these movements are executed and which joints and muscle groups are involved in the movement. While the dragging operations on the horizontal surface were executed using only the wrist and the fingers, dragging operations on the vertical surface involved all joints of the arm including the shoulder. Previous studies have demonstrated differences in performance between the muscle groups controlling the various upper limb joints. For instance, the hand was proven to be superior to the forearm [Hammerton, Tickner, 1966], and evidence was found that the fingers may possess a higher resolution than the wrist or forearm [Penfield, Rasmussen, 1950]. Langolf et al. [1976] investigated the relationship between Fitts' Index of Difficulty (ID) and movement time for different upper limbs (finger, wrist, and whole arm) over a wide range of movement distances. As predicted by Fitts' Law, the movement time turned out to be a function of increasing ID [Fitts, 1954]. However, the slope of the function differed remarkably between finger, wrist, and whole arm. If the reciprocal of the slope was supposed to infer the information-processing capacity of the motor system, then the fingers showed a much higher information-processing rate (38 bits/sec) than the hand (23 bit/sec) and the arm (10 bit/sec).

Second, dragging on a planar surface is faster and more direct than dragging across the curve, despite the fact that the distances between the source and target were constant for all dragging tasks. This is a rather unexpected finding, as Fitts' Law [Fitts, 1954] would have predicted constant movement durations over all areas. The increased movement durations across the curve went along with a higher curvature in hand paths. From a cognitive point of view, the curved hand path is similar to motor behavior observed when avoiding obstacles. Jax, Rosenbaum [2007] found in their study that the anticipation of obstacles led to more curved hand paths, even when the obstacle was not present. This suggests that our participants perceived the curve as a kind of obstacle, which they tended to avoid. Considering the motor behavior, we assume that the more curved hand paths in the curved portion of the table also results from the more complex motor activity involved in curve dragging: the participants performed dragging on the horizontal surface basically by pushing or pulling the hand backwards or forwards. Analogously, they dragged the target on the vertical surface by lifting or lowering the arm. In contrast, when moving across the curve, users tended to rotate the entire hand while moving it upwards and downwards, which yields a more complex movement. One person stated afterwards that the tendon in his index finger hurt if he did not turn his hand, while another person reported that he was afraid of drilling his index finger into the surface and, thus, turned the hand. Furthermore, four users wanted to change from the index to the middle finger when they unintentionally released an object during the dragging because they considered the middle finger as stronger and more stable.

The curve is perceived as an obstacle and users avoid dragging through it.

2.4.3 Study 2: Cross Dragging

In this second task we analysed dragging performance not within an area as in the first study, but instead from the horizontal surface across the curve to the vertical surface and visa versa. We determined if dragging performance depended on the angle of approach.

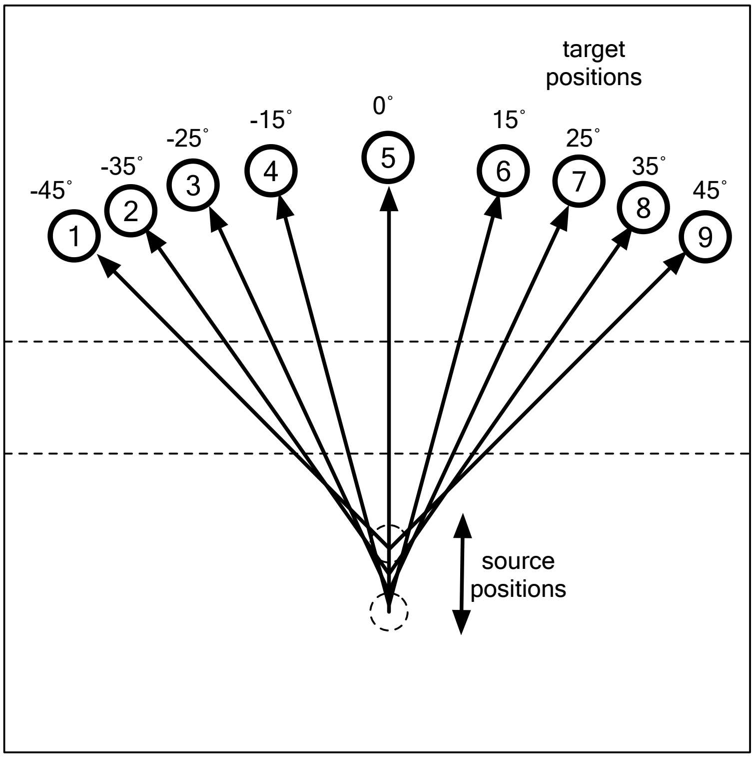


Figure 2.12: The experimental design for the Cross Dragging experiment. The users had to drag an object through the curve into target display a different angles.

Participants

This study was done by the same participants as in the first study.

Tasks

Figure 2.12 shows the experimental task. The system displayed the source, a white colored circle and the target, a black colored circle inside a white ring. Both circles had a diameter of 60 px (5.82 cm) and the width of the target ring was 20 px (1.94 cm). The distance on the surface between source and target was held constant at 600 px (58.20 cm). As in the first task, participants had to drag the source inside the target using their index finger. After successfully moving the source onto the target (within the same tolerance as in the first study) the next trial appeared.

Trials appeared in nine different movement directions (with two repetitions each): (1) 45°, (2) 35°, (3) 25°, (4) 15° to the left, (5) 0° (vertical line), and (6) 15°, (7) 25°, (8) 35°, (9) 45° to the right. The movement started either in the horizontal surface (moving upward to the vertical space) or the vertical surface (moving downward to the horizontal space). The order of trials was randomized. Participants conducted 36 trials with their dominant hand and 36 trials with their non-dominant hand. A total of 72 dragging operations were presented. We measured task completion time and trajectory length in the same way as in the first study.

Users were asked to drag objects across the curve in nine different directions.

For this study we hypothesized the following outcomes:

H4: The dragging (a) duration and (b) trajectory increases with larger dragging angles.

H5: The deviation of trajectories increases with larger dragging angles, thus showing more variance in movement paths.

Results

Data were analyzed for each of the dependent variables with one-factorial repeated measures ANOVA with the within-subject factor *angle*. For dragging durations the ANOVA showed a significant main effect for the factor *angle* ($F(8, 128) = 2.65; p < 0.05$). Dragging durations varied between 1306 and 3806 ms. However, the differences across angles were too small to show statistical significance in post-hoc comparisons.

The mean lengths of dragging trajectories are shown in Figure 2.13. We found a significant main effect of the factor *angle* ($F(8, 128) = 8.94; p < 0.01$). Post-hoc comparison showed that dragging trajectories for targets 45° to the left or to the right of the source (mean 652 px) were significantly

The length of the dragging trajectory is highly dependent on the dragging angle.

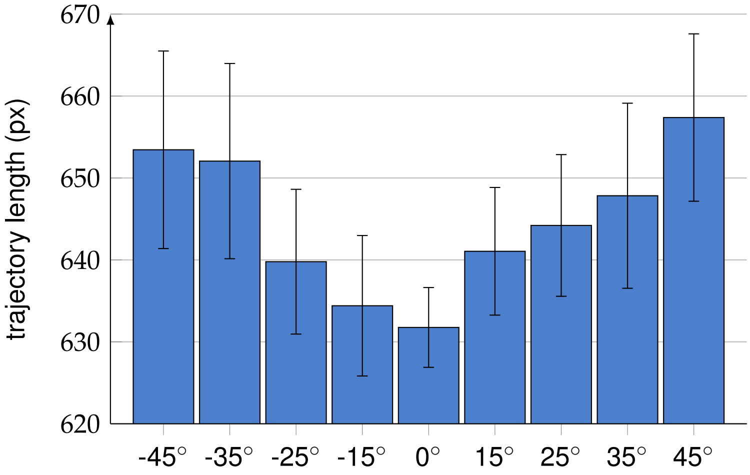


Figure 2.13: Length of dragging trajectory depending on angle. Whiskers denote the 95% confidence interval.

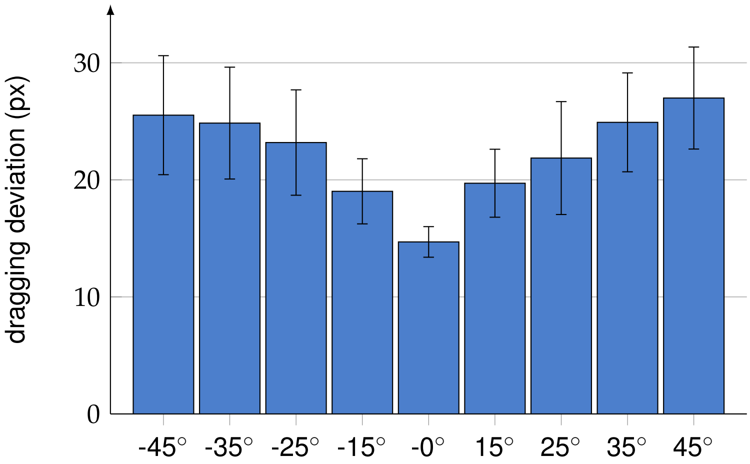
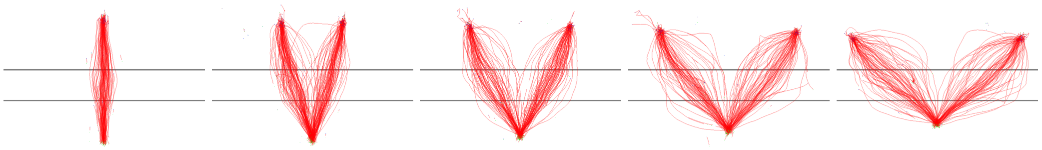


Figure 2.14: Average deviation from the direct line between source and target depending on angle. Whiskers denote the 95% confidence interval.

longer when compared to targets vertically presented to the source (mean 631 px).

In addition, we analysed the deviation of the movement trajectories from the most direct connection between source and target as shown in Figure 2.13 and Figure 2.14. The ANOVA showed significant main effects of the factor *angle* for the maximum $(F(8, 128) = 11.66; p < 0.01)$ as well



(a) 0° (b) ±15° (c) ±25° (d) ±35° (e) ±45°

Figure 2.15: Dragging trajectories for upward dragging across the curve for different angles. The variance significantly increases with higher angles.

as the average $(F(8,128) = 10.51; p < 0.01)$ deviation between presented and observed amplitude (Figure 2.14). The deviation increased by 85% (12 px) for larger angles.

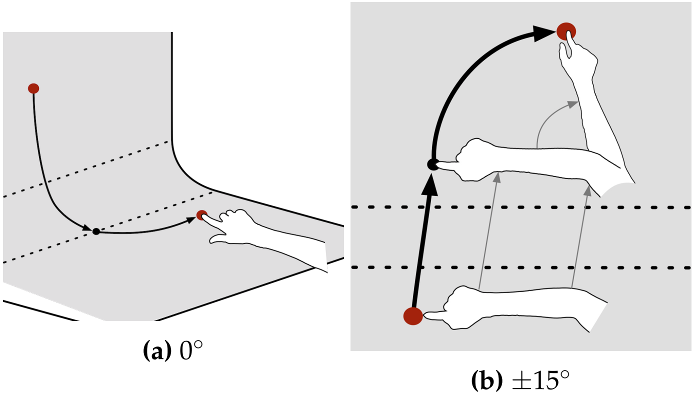
We conclude, in accordance with H4 and H5, that trajectory length and variance in movement paths both increased as the dragging angle moved further from a straight vertical trajectory.

Discussion

The main finding of this study is that the angle had nearly no effect on the duration of dragging operations. However, we observed a significant increase in trajectory length when the angle is increased. In order to gain more insights into the causes for this effect, we plotted out the trajectories for each angle (Figure 2.15). Two effects become apparent: First, at larger angles participants tended to minimize the dragging distance on the curve. Some users even separated the movement into a short path across the curve and a long path for the remaining movement. Second, the higher the angle the higher the angle, the more the observed trajectories deviated from the most direct route. This also matches Figure 2.14, which shows an increased variance in trajectory path for larger angles. Furthermore, our observations revealed that most users optimized their dragging operations to reduce muscle exertion. We noticed that some users dragged downwards by letting the arm quickly fall straight downwards and across the curve before dragging the object to the target, as shown in Figure 2.16.a. Another frequent

Dragging duration is not affected by the dragging angle.

Users tried to minimize the dragging movement through the curve and therefore increase the trajectory length.


(a) 0°

(b) ±15°

Figure 2.16: Observed dragging trajectories that reduce exertion. In both cases, the users minimized the dragging movement through the curve.

movement was an upward dragging, where the user first dragged the object across the curve using a stiff bent arm and finished the dragging by turning the hand and lower arm with the upper arm as rotation axis, as shown in Figure 2.16.b. Moreover, two users reported that approaching the curve at a flat angle feels uncomfortable. In general, we noticed that users tried to create a convenient movement trajectory despite being asked to acquire the target as fast as possible.

2.4.4 Study 3: Virtual Aiming

In the first two studies, we analyzed how users execute simple motoric operations. In this third study we investigated how users perceive digital object displayed on the BendDesk and how users are able to estimate distances and directions on the curved surface. Unlike the previous tasks, the participants now had to perform an open-loop task by aiming at a target. We compared whether virtual aiming was supported with or without a grid displayed on the surface.

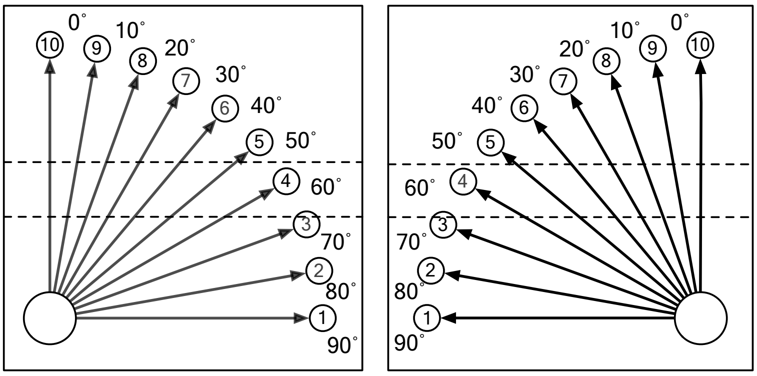


Figure 2.17: Experimental design for the Virtual aiming study. In this study, the users were asked to aim a virtual target by placing 2 fingers inside a starting such that the line that that goes through both fingers hits the target.

Participants

This study was done by the same participants as in first and second study.

Tasks

Users we asked to aim at different targets by creating a line with two fingers that points in the direction of the target.

Figure 2.17 shows the experimental task. The system displayed the source, a gray colored circle with a diameter of 200 px (19.5 cm) and the target, a white colored circle with a diameter of 30 px (2.9 cm). The distance between source and target was 800 px (78.1 cm). Participants had to position their left and right index fingers inside the source area until an imagined line drawn through both finger tips would hit the target area. The system gave visual feedback by rendering circles beneath the touches. When participants felt that they would hit the target they released both fingers and the system displayed a gray line through both touches towards the target area. Then, the next trial appeared.

Trials appeared in ten different movement directions, with two repetitions each. Targets within the horizontal plane:

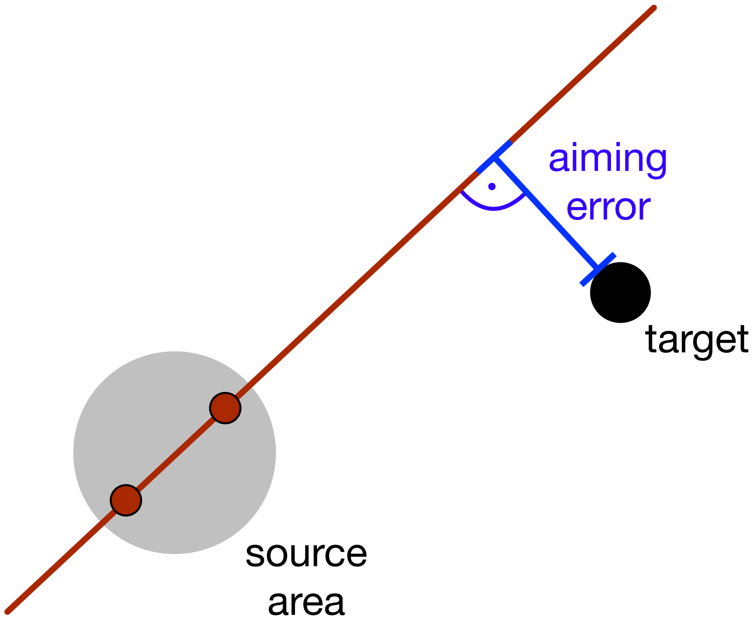


Figure 2.18: Two finger touches in the source area define a straight line towards the target. The aiming error is the deviation between this virtual aiming path and the target area.

(1) 90°, (2) 80°, (3) 70°; targets within the curve: (4) 60°; and targets across the curve and in the vertical plane: (5) 50°, (6) 40°, (7) 30°, (8) 20°, (9) 10°, and (10) 0°. The order of trials was randomized. Participants executed a block with a uniform grid on the system's surface (we displayed a 26 x 26 grid with a cell size of about 40 px × 40 px, or 3.9 cm × 3.9 cm) and another block without a grid but instead with a solid blue-colored surface. This resulted in 10 (angle) × 2 (background) experimental conditions. We further controlled the virtual aiming direction by presenting the source either in the right or left corner of the horizontal area (upward aiming), or in the right or left corner of the vertical area (downward aiming). This resulted in a total of 160 virtual aiming operations. As a dependent variable, we measured the aiming error as shown in Figure 2.18, i.e., the deviation between the virtual aiming path and the target area, or in other words the spatial misjudgment (given in px).

We hypothesized the following outcomes:

H6: The aiming error is smaller for virtual aiming within one surface than across the curve or between different surfaces.

H7: The aiming error is smaller for virtual aiming with a grid displayed on the surface than without a grid.

Results

The aiming errors for orthogonal targets were significantly smaller than for diagonal targets.

The data were analyzed with a 10 × 2 repeated measures ANOVA with the within-subject factors *angle* and *background*. The aiming errors are shown in Figure 2.19. The ANOVA showed a significant main effect of the factor *angle* $(F(9, 158) = 17.24; p < 0.01)$. Aiming errors were smallest when source and target were within the same plane. Especially, virtual aiming at 90° (mean error 9 px) was significantly more accurate than at all other angles (mean error 43 px). Furthermore, aiming at 90° and 0° was supported by the grid displayed on the surface: Aiming errors for the 90° angle were 65% (8 px) smaller with displayed grid (mean 5 px) than without the grid (mean 13 px). For the 0° angle the aiming errors were 57% (19 px) smaller with the grid (mean 14 px) than without the grid (mean 33 px). However, the background did not have any effect on the other angles, yielding a significant interaction $(F(9, 153) = 2.61; p < 0.01)$. The factor *background* alone did not show any significant effect on aiming errors.

In conclusion, the results from the virtual aiming task showed that virtual aiming is most accurate for the orthogonal angles (0°, 90°) when a grid is displayed on the surface. This is only partially in line with our hypotheses H6 and H7.

Discussion

This virtual aiming task revealed the complexity of imagined instead of manual aiming gestures. Participants

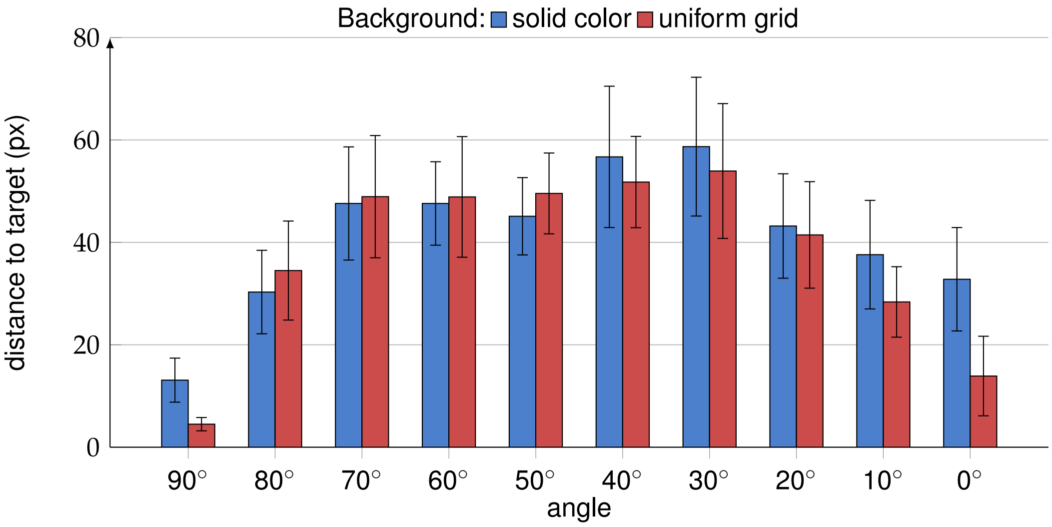


Figure 2.19: Distance from target depending on angle. The results show that the aiming error was the lowest for both orthogonal angles (0°, 90°). Whiskers denote the 95% confidence interval.

The users used the borders of the system as alignment guides.

severely misjudged the spatial relations towards the target. However, at orthogonal angles (0°, 90°) the aiming error was lowest, probably because the table borders provided alignment guides. This is further supported by the observed improvement of virtual aiming at 0° and 90° when a grid was present. In this case, participants could easily touch the grid lines to hit the target. We assume that the complexity of virtual aiming depends on the required cognitive mapping between the 3D and the GUI space. If the user aims at a target on the same surface, she has to compensate for the perspective distortion of the plane, where, according to Wigdor et al. [2007], these distortion effects are stronger on horizontal surfaces. If the target is placed on the opposite side of the bend, the user has to perform a three-dimensional non-linear transformation of the table shape to the rectangular control space.

2.4.5 Study 4: Flicking in Different Areas

In the first two studies, we evaluated how users can execute very basic motoric tasks by analysing dragging operations in the different areas of the BendDesk. In the third study, we evaluated how users can visually plan operations on a curved surface by letting them aim at different targets. In this study, we wanted to evaluate how users perform more complex tasks which involve both visual planning and motoric execution. For this task, we had users execute flicking gestures. The flicking gesture is a very common open-loop control techniques that lets the user define the direction in which the object should move at the beginning of the action. Once the automatic part of the movement is initiated, there is no more opportunity for interaction, so the gesture cannot be corrected or refined. On interactive surfaces, flicking is usually executed with either one finger or the entire hand [Wilson et al., 2008; Wu, Balakrishnan, 2003; Cao et al., 2008]. Hinrichs, Carpendale [2011] showed that even users who had never used an interactive tabletop readily used this gesture to move digital objects. To simulate the flicking of a physical object more realistically, Sato et al. [2008] compared a vision-based approach in which users conducted a flicking gesture above the table to users flicking a real marble. In this study, we compared flicking on the horizontal, vertical, and curved area to investigate if there are differences in flicking accuracy on these different areas of a curved surface.

Flicking is one of the most commonly used interaction techniques on interactive surfaces.

Participants

A total of 16 participants (13 males), age 20–39 (mean=27) volunteered for the study. The group consisted of 12 computer scientists, 2 business administration managers and 2 medical scientists. 14 out of 16 participants were right-handed. All participants had normal or corrected-to-normal vision.

Task design and procedure

Users were asked to flick an object onto several targets.

The goal of this study was to flick a digital source object onto a digital target using the index finger of the dominant hand. To conduct the flicking gesture, participants had to put their index finger onto the digital source object, and slide their finger along the surface in the direction in which they wanted to flick the object. After executing the flick gesture, the source object moved in the computed direction at constant speed until it hit the surface boundaries. Then, the interactive start area went blank and a new trial appeared. Participants were instructed to solve each task as fast and accurately as possible. To measure the direction, we recorded all touch points of the user's sliding movement, and computed a direction vector with a first-degree Least Squares regression. This method to determine the direction was also used by Reetz et al. [2006] for gestures with a pen on a tabletop. They analyzed several approaches to compute the direction, and found that first-degree Least Squares regression was closest to users' expectations. To estimate gesture accuracy, we used the flicking error as a dependent variable. It was defined as the angle in degrees between the direct line along the surface from the source object to the target, and the direction vector defined by the user's actual input (Fig. 2.20).

Figure 2.21 depicts the experimental task and conditions. The system again displayed the source as a blue colored dot (diameter 50 px) and the target as a red colored circle (diameter 60 px). The distance between source and target was fixed to 29 cm (300px) for all conditions. Trials appeared in 3 different areas (horizontal area, curve, vertical area), 2 different source positions (left side or right side) and 7 different flicking angles (with 3 repetitions each): (1) 90°, (2) 75°, (3) 60°, (4) 45°, (5) 30°, (6) 15° and (7) 0°. Participants completed a block of trials with upward movements and another block with downward movements. This resulted in 252 trials total. The order of trials was randomized. We used a 3 (area) × 2 (flicking direction) × 7 (flicking angle) design with repeated measurements. The experiment lasted about 15 minutes. We assumed that a larger flicking angle would yield a lower flicking performance, especially

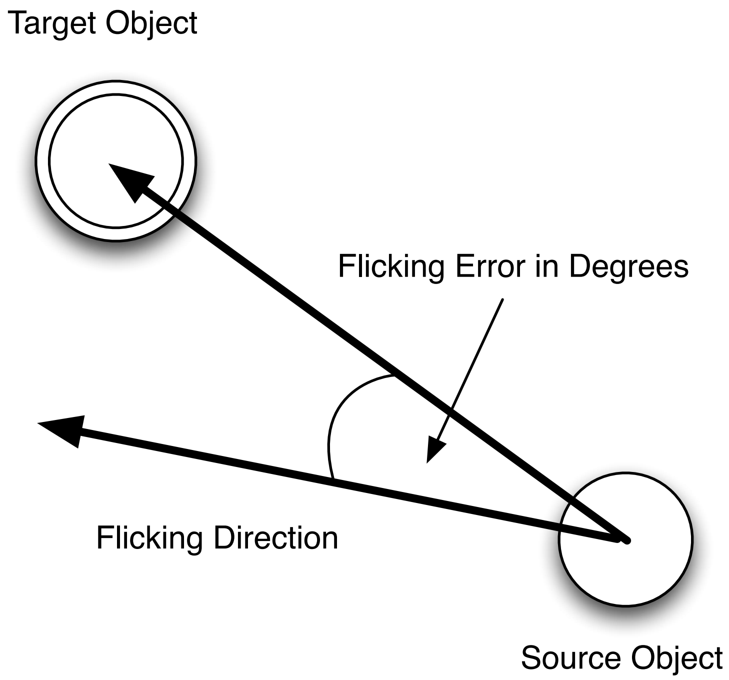


Figure 2.20: Error measurement for flicking gestures. Errors were measured in degrees difference from the target direction to make them independent of distance.

within the curve:

H8: The flicking error is smaller for flicking within the horizontal area than within the vertical area or the curve.

H9: The flicking error increases with larger flicking angles. Flicking within the curve will be more affected by the flicking angle.

H10: Flicking downwards is less accurate than flicking upwards.

Results

A 3 (area) × 2 (flicking direction) × 7 (flicking angle) ANOVA with repeated measurements showed significant main effects for the factors *flicking direction* ($F(1, 15) =$

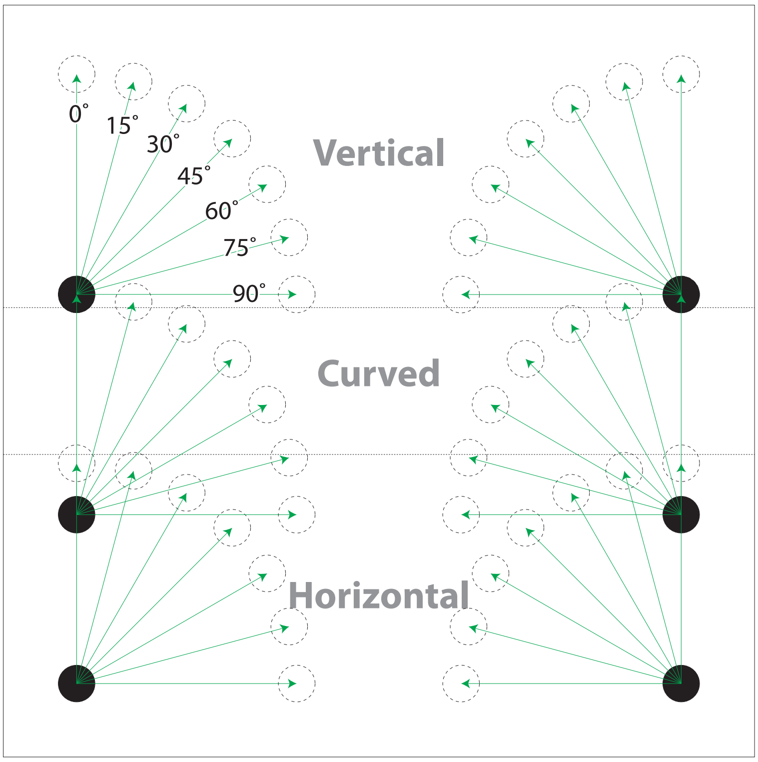
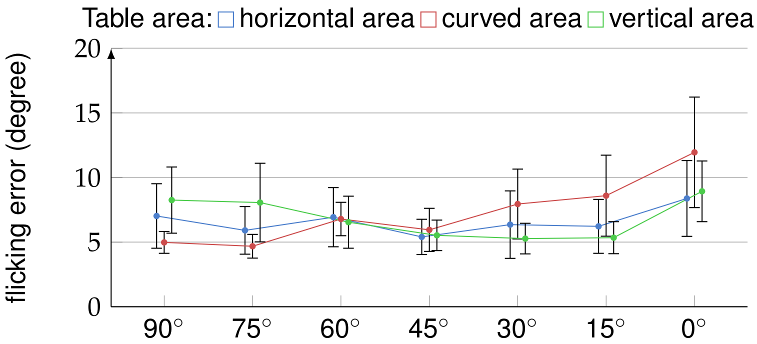


Figure 2.21: Flicking task in the upwards direction. Users were asked to flick an object onto different targets.

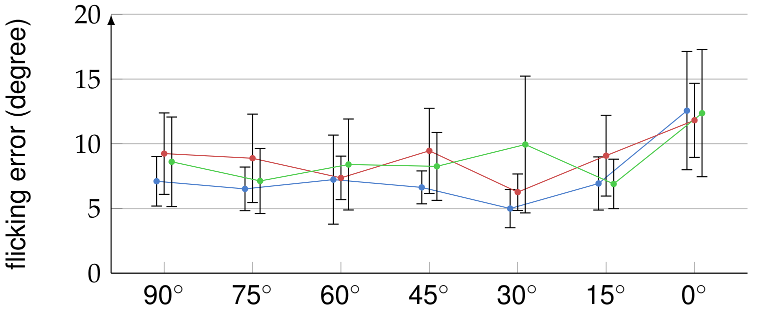
Flicking accuracy is highly dependent on the flicking direction and on the flicking angle.

Flicking upwards is more accurate than flicking downwards.

$18.78; p = 0.001$) and *flicking angle* ($F(6, 90) = 12.81; p < 0.0001$), and a trend for the factor area ($F(2, 30) = 3.05; p = 0.062$). The interaction area × direction × angle was also significant ($F(12, 180) = 2.48; p = 0.005$). The main effect of flicking angle showed a severe inaccuracy for flicking at 0° (error at 10.9°), for all other angles the flicking error ranged from 6.7° to 7.5°. Furthermore, flicking upwards was more accurate than flicking downwards (6.9° vs. 8.3°). There was a trend that flicking within the horizontal area was more accurate than flicking in the curve or within the vertical area (7.0° vs. 8.0° and 7.8°). The three-way interaction revealed similar accuracy for flicking at angles between 15° and 90°. When flicking downwards, actions were most inaccurate at 0°, irrespective of the interaction area. For upward flicking, this inaccuracy was only observed for interactions within the curve, but not for actions within the horizontal or vertical area.



(a) Upwards direction

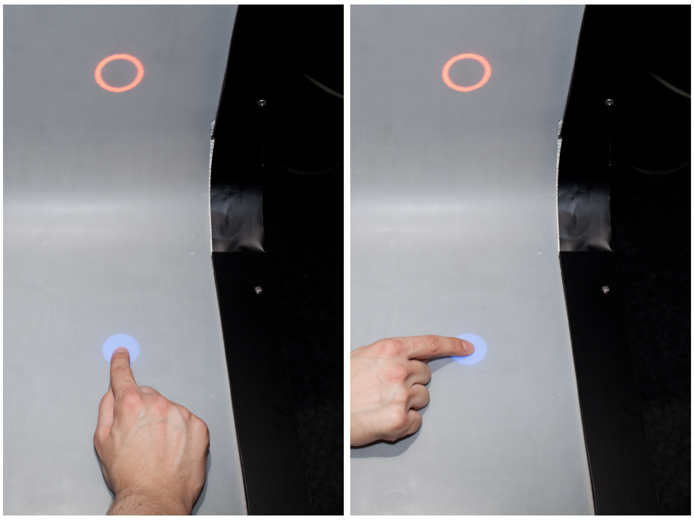


(b) Downwards direction

Figure 2.22: Mean flicking errors dependent on the different areas and angles. In general the differences between the angles are small, except for the 0° condition, in which the flicking error is significantly higher compared to the other angles. Whiskers denote the 95% confidence intervals.

Discussion

There are two main findings from this study: First, flicking upwards was more accurate than flicking downwards (H10). The impact of the interaction area was only partially confirmed. There was a trend for more accurate flicking within the horizontal area, especially for downward flicking at an angle between 15° and 90° (H8). Second, and completely contrary to our hypothesis, we did not find an increase of flicking errors with increasing flicking angle (H9). Accuracy was comparable for flicking angles between 15° and 90°, and interactions were more accurate than for flicking at 0°. This result is quite surprising and in contrast to



(a) hand below **(b)** hand on the side

Figure 2.23: For flicking directly upwards we observed two different hand postures in this task: In the first posture (left), users placed their hand directly above the source object and used their entire arm to conduct the gesture. In the second (right) posture, users placed their hand lateral to the source object, and only moved their finger. In both positions the fingers can only be moved radially therefore flicking directly upwards is difficult.

findings from dragging actions. However, this can be explained by motor constraints of this particular movement.

Flicking can be influenced by the visual planning stage or by the motoric execution.

As described previously, a flicking gesture consists of two different stages: a visual planing stage, in which the user has to determine the spatial relation between the source and the target, and a motoric execution stage in which she executes a sliding gesture in the previously determined direction towards the target. Errors that influence flicking accuracy can appear in both of these stages. To improve flicking on curved surfaces, we first have to understand the location and type of errors that can appear, and how they affect the flicking gesture.

The curve could influence the visual planning phase.

The visual planning stage can be biased through contextual factors so that perception errors occur. Previous psychological research has demonstrated various perceptual illusions that impact visual planning performance. Classic phenomena referring to such perception biases are the Mueller-Lyer illusion of length [Restle, Decker, 1977], the Ponzo illusion of distance and object size [Fineman, 1996], and the Poggendorff illusion of vertical offsets [Green, Hoyle, 1963]. These examples indicate that the shape of an interactive tabletop could be such a contextual factor leading to an error that influences the visual planning phase.

Movements within the horizontal area turned out to be more precise than movements executed in the curve, while the most difficult movements were those in the vertical area. This finding matches the results of Langolf et al. [1976]. The working areas were at different distances to the user's body. When the working area was within the horizontal area, the arm was bent and relaxed in front of the chest, and flicking could be carried out by simply moving the index finger. In contrast, if the working area was in the vertical area, the arm was stretched out straight. In order to accomplish the same action, movement of the whole arm, including all upper limb joints (wrist, elbow, and shoulder) had to be coordinated. Therefore, we assume that the effect of different flicking accuracies on different areas of the curved surface is an error in the motor execution phase.

Flicking on the horizontal surface is more precise since it is executed from a relaxed arm position.

Flicking on the vertical surface involves all arm joints and is therefore inaccurate.

Another effect that we observed was that, independent of the source position, flicking gestures directly upwards or downwards (0°) showed the largest error. This is in direct contrast to the findings that were made in our third study above in which users shot at different targets by defining a line with a two-finger tap gesture. This showed that hitting objects that are placed directly above or below the starting position was highly accurate, in contrast to other target positions. These differences can only be explained by a motor error of the flicking gesture, because according to the study by Langolf et al. [1976], users can identify the correct direction to these targets very well. That most users had significant problems with this flicking direction can be explained by the way they executed this gesture. We primarily observed two different hand postures in this task: In the first posture, users placed their hand directly above the source object and used their entire arm to conduct the gesture (Fig. 2.23.a), which according to Langolf et al. [1976]

Flicking directly upwards is inaccurate due to radial movement of the finger joints.

leads to inaccurate execution of the gesture. In the second posture, users placed their hand lateral to the source object, and only moved their finger.

However, since fingers can only be moved radially in this posture, a sliding gesture pointing directly upwards or downwards is not possible (Fig. 2.23.b). These issues do not occur when simply tapping with two fingers to define a direction, explaining the different results from the third study.

2.4.6 Study 5: Flicking through the Curve

In this study we investigated whether the relative position of the curved area between the source object and the target position has any influence on flicking accuracy. Additionally, we investigated whether the lateral position of the source object has an effect on accuracy.

Participants

This study was done by the same participants from the fourth study.

Task

In this study we used the same flicking task as described in the fourth study. Figure 2.24 depicts the experimental task and conditions. The system again displayed the source as a blue colored dot (diameter 50 px) and the target as a red colored circle (diameter 60 px). The distance between source and target was 368 px (35 cm).

Trials appeared at 2 different distances from the curve (start position closer to (10 cm) or farther away from (20 cm) the curve center), with four different combinations of source position and movement direction (flicking from left to right, from the center to the right, from the center to the

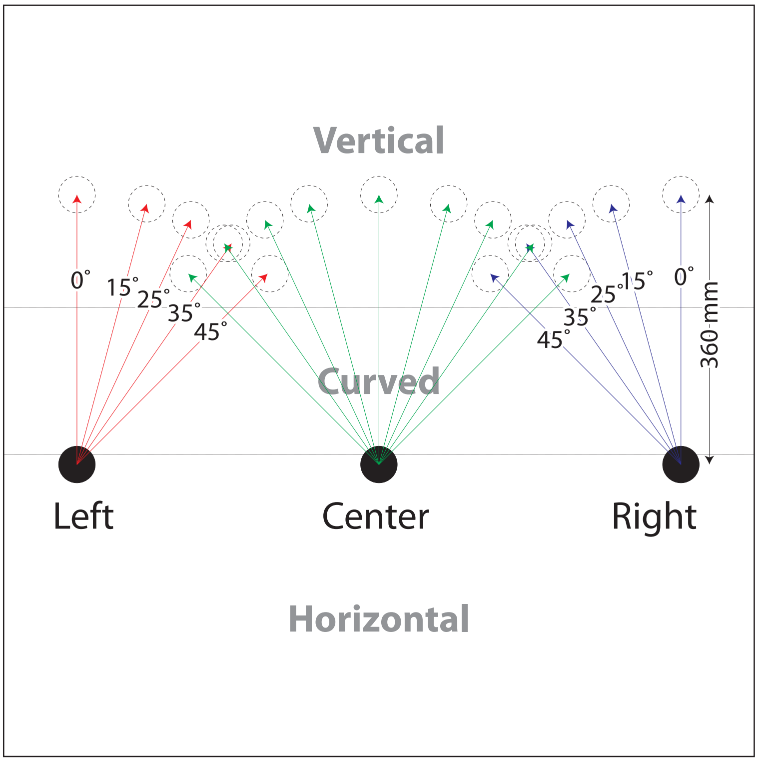


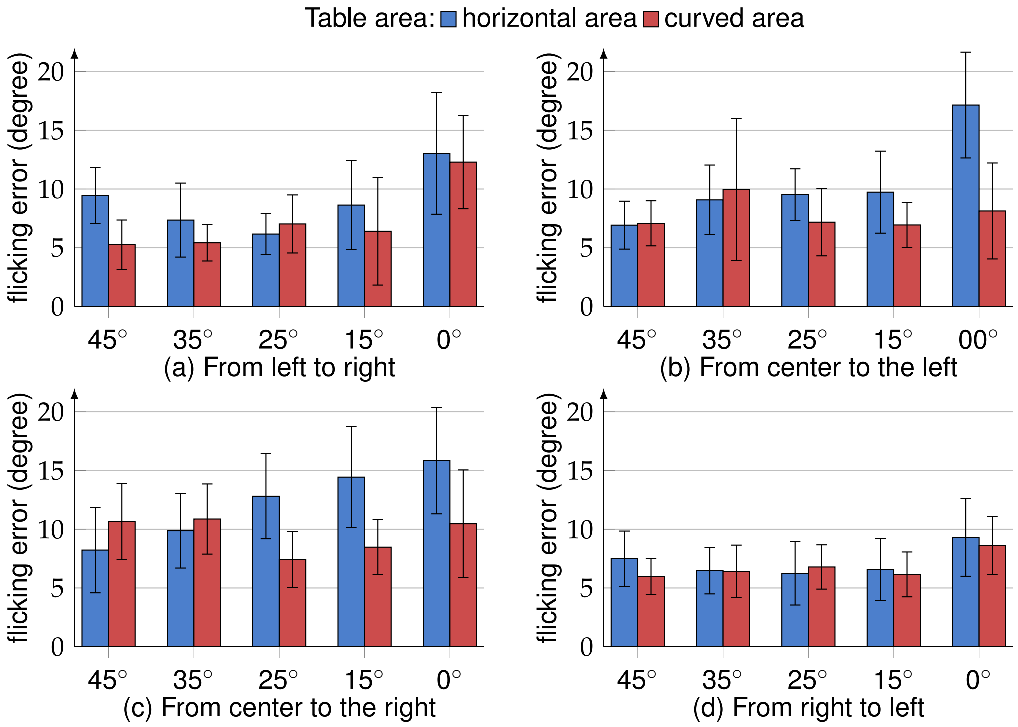
Figure 2.24: Flicking task in the condition close to the curve. In this study the users had to flick from four source positions to 5 different target positions.

left, and from right to left) and 5 different flicking angles, with 3 repetitions each: (1) 45°, (2) 35°, (3) 25°, (4) 15°, (5) 0°. Participants complete a block of trials with the start position near the curve, then through another block with the start position farther away from the curve. This resulted in 120 trials total. The order of trials per source position and order of source positions were randomized. The study is based on a 2 (distance between source and curve center) × 4 (start position × sideward direction) × 5 (flicking angles) design with repeated measurements. The study lasted about 10 minutes.

In this study users were asked to flick through the curve.

We hypothesized the following outcomes:

H11: The flicking error increases if the source position is closer to the curve.



(a) From left to right

(b) From center to the left

(c) From center to the right

(d) From right to left

Figure 2.25: Mean flicking errors dependent on the different start positions and directions, and on the distance between source object and the center of the curve. The flicking error for flicking gestures directly upwards (0°) is larger compared to other conditions. Whiskers denote the 95% confidence intervals.

H12: The flicking error is higher at 0° than when flicking sideways.

Results

A 2 (distance between source and curve center) × 4 (start position × sideward direction) × 5 (flicking angles) ANOVA with repeated measurements revealed significant main effects of the factors *distance from curve* ($F(1,13) = 7.14; p = 0.019$), *start position and sideways direction* ($F(3,39) = 10.63; p = 0.000$) and *flicking angle* ($F(4,52) = 9.21; p = 0.000$). Flicking was more accurate when the source was positioned farther away from the curve center than when it was directly on the curve (7.8° vs. 9.7°).

Flicking accuracy was similar for flicking angles between 15° and 45° (error: 7.6° – 8.4°), and significantly decreased at 0° (error: 11.8°). The impact of flicking angle was more pronounced for start positions near the curve than farther away, yielding a significant interaction $(F(4, 52) = 4.11; p = 0.006)$. The factor *start position and sideways direction* further showed flicking operations were most accurate when the source was positioned to the right side, with flicking to the left (error: 7.0°), and least accurate when the source was centered, with flicking to the right (error: 10.9°). The three-way interaction $(F(12, 156) = 3.04; p = 0.001)$ showed an impact of flicking angle for all start positions and sideways directions, except when the source was on the right side. This was the most accurate condition, and the impact of angle nearly diminished.

The start position of the flicking gesture has more effect than the target angle.

Finally, we found a significant performance decrease for flicking actions that originated near the curved area (H11). This finding supports our assumption that the curved surface introduces a perceptional bias. It seems that its impact decreases the farther away the action takes place. Furthermore, we successfully replicated the impact of flicking angle observed in study 4 (H12). For all source positions we found very inaccurate flicking actions at 0°, except for the source positions on the right side. Here, flicking was most accurate, and flicking angle had almost no effect on performance.

The flicking error is the largest for flicking actions straight upwards (0°).

Discussion

Not surprisingly, the overall results are very similar to the results of the previous study. For understanding the results, please note that in our study all participants were right-handed. Thus, when the source was in the right position flicking performance was most accurate compared to the center or left position of the source. Regarding the distance, the observed errors follow the same pattern described in Langolf et al. [1976]. We conclude that for right-handed users a spatial alignment between dominant hand and source facilitates motor control, while all other source positions required somewhat awkward postures of

A flicking gestures that is executed far away from the user the movement involves more joints. Therefore they are more inaccurate.

the moving limb and therefore restricted motor control. Consequently, left-handed users should feel most comfortable and be most effective when the source is in the left position.

The results of both flicking studies show that the flicking angle has a strong influence on the flicking error. In addition to these observations, we where also interested whether the errors follow a specific pattern and if these errors can be predicated or even corrected. So we had a closer look at the errors made by the users and found out that these error follow a specific pattern that can be modelled. In the follow, we describe a preliminarily model that shows that the flicking error made by the user can be modelled and therefore be corrected.

Compensating the flicking error

The flicking error is systematic and can be corrected by a mathematic model.

To explain the relationship between the flicking angle and the flicking error, we developed a first rough mathematical model from the data acquired at the center position of the third user study. In this model, we redefined the flicking error as a signed angle. The negative values mean error to the right. Therefore, we have nine angles (-45° to -15° to the left of the target, 15° to 45° to the right of the target, and 0° directly above the target). The angular errors are shown in Figure 2.26.

We approximated these results with the following sinusoidal function: $a + b \times sinus(c * Angle)$. From our data, the best-fit parameters are: $a = 1.162$, $b = -8.656$, and $c = 3.444$. Although this function is only a very rough approximation of the results and can only be used to give a hint about the flicking error by flicking angle, it shows that flicking can be described by a mathematical function.

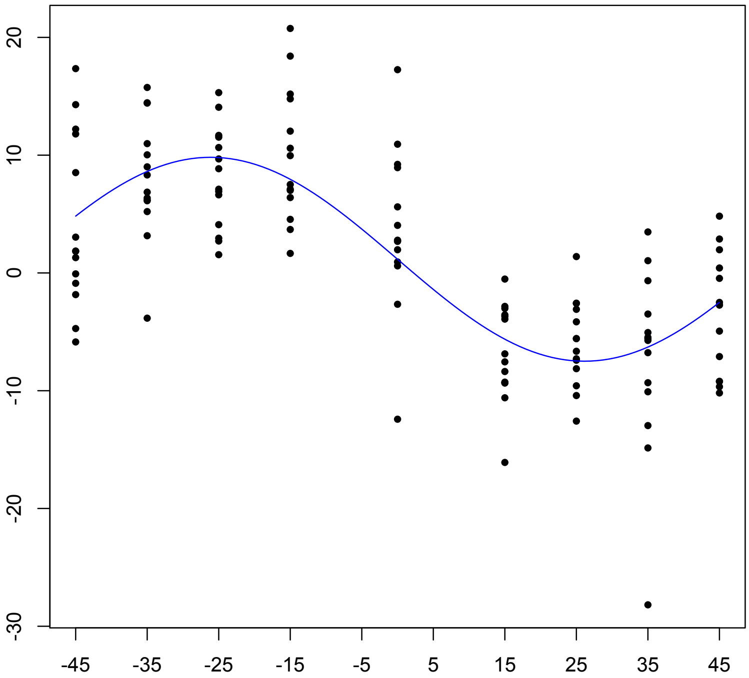


Figure 2.26: Visualization of the first mathematical model to predict the flicking error and the flicking angle for flicking gestures across the curved area. Each dot is a data point, and the blue line is the fitted model.

2.5 Conclusion

In this chapter we presented an interactive desk system that merges a vertical and a horizontal display into a spatially cohesive surface by using a curve. This provides a large interactive surface that users can reach in a comfortable sitting position. We introduced a technique to project on the curved surface, as well as algorithms for multi-touch detection under strong distortions (C 1.1).

However, our BendDesk system is only a research prototype that was used to explore how an interactive workspace could look like and to understand how user could interact with such a system. Therefore, it as several limitations that should be address in the next version of this system.

The main limitations of BendDesk are its low resolution and the touch tracking method.

The two major limitations are the resolution of the display and the tracking method that we used to track touch points. As mentioned above the resolution is only 25 DPI (in comparison a normal display has at least 90 DPI and modern smartphone has about 300 DPI). As mentioned above, to track touch points, we use a vision-based approach that makes the tracking extremely sensitive to external lighting conditions. However, this could be solved in the future by using capacitive touch tracking on top of a bendable OLED.

Despite the hardware limitations, we were able to explore this kind of interactive desk workspace and analyse how user could use such a system by conducting a series of user studies.

Dragging on the horizontal surface is easier and more accurate than on the other areas.

In the first two user studies, we investigated the question how users perform simple motoric operations (C 1.2) such as dragging on a curved system. We showed that users perform these simple operation better on the horizontal surface compared to the vertical and curved surfaces. Furthermore, these user studies also suggest that the curve represents a slight but noticeable physical barrier. It leads to longer interaction times when crossing it and some users tend to minimize the dragging distances in that area when approaching it with a flat angle.

In the third study, we analysed how such a curved surface influences the user's spatial perception (C 1.3). We showed that the curve impairs the user's perception such that they have problems estimating angles and distances to a target that is placed on the other side of the curve.

Flicking gestures are influenced by the curve but can be corrected by a mathematic model.

In the last two studies, we analysed how users perform more complex gestures by exploring flicking gestures (C 1.4). In these studies, we showed that the curve influences the visual planning phase of these gestures but also that the motoric execution has more impact on the accuracy of these gestures. However, we were able to show that users make systematic errors that can described by a mathematic model and therefore can be compensated.

These results suggest that the three interaction areas should not be considered as a single interactive surface. Users will likely reduce the number of interactions across the curve, and the user interface should not require cross-dragging with angles more parallel to the curve. Furthermore, since interacting with the vertical surface is cumbersome and inaccurate, interaction with it should be limited. The characteristics of the curve must be taken into account and can even be exploited to divide the surface into logical units. For example, an application could use the horizontal display to create content that is stored in the curve before it is assembled at the vertical area. That is, the three areas would represent steps in a workflow. Another scenario is remote collaboration as shown by Hennecke et al. [2013b], where the vertical space represents a public space showing content visible to all co-workers, while the horizontal area is a private space for individual content. In these cases, the curve could act as an intermediate storage, or as a "dock" or "taskbar".

Interaction with the vertical surface should be limited.

The curve could be used as a dock to store object.

Overall, having a interactive desk workspace that consist of horizontal and vertical surfaces has a lot of advantages for a large variety of tasks. However, directly interacting with the vertical surface is problematic and not a practical solution and should be replaced by a more comfortable interaction concept.

Chapter 3

Improving Interaction on Interactive Desk Workspaces

Direct interactions with a vertical surface are cumbersome.

In the previous chapter, we explored interaction with interactive desk workspace and their construction in depth. Even so we showed that touch based interaction with desk workspaces has it advantages, the general ergonomic limitations of touch interactions with vertical surfaces apply [Boring et al., 2009; Hincapié-Ramos et al., 2014].

To address these issues, Schmidt et al. [2009] proposed an indirect multi-touch system. In such a system, the users create input on a horizontal touchscreen that is then mapped

---

Publications: The work in this chapter is a collaboration with Chat Wacharamanotham, Andrii Matviienko, and Johannes Schöning. The author is the main author of both publications. He is also responsible for writing parts of the software, designing the experiments, and analyzing data from the experiments. Part of this work was first published as a paper at the CHI 2013 conference [Voelker et al., 2013d] and as a paper at the SUI 2015 conference [Voelker et al., 2015b]. Several sections of this chapter are taken from these publications. One of the authors of 2013 CHI Paper [Voelker et al., 2013d], Chat Wacharamanotham, reanalyzed the data from this paper and published them in his dissertation [Wacharamanotham, 2016]. With permission of the author, we present the updated results of his analysis. Furthermore, parts of this work is also published in the master thesis from Matviienko [2014] who worked on combining gaze and indirect touch.Indirect touch input allows user to

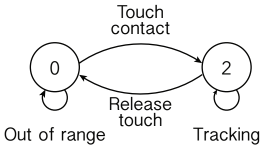


Figure 3.1: The two-state model for direct touch input. In this interaction concept the user's finger is either not on the touch-surface (*out of reach*) or placed on the touch-screen (*engaged*). In the *engaged* state, users directly manipulate the digital objects below their fingers.

Indirect touch input allows users to interact with a vertical surface in a comfortable way.

to a vertical surface that is only used as output. The benefit of this setup is that users can now use touch input on the horizontal surface while looking at the output displayed on the vertical surface. This prevents both neck pain and the gorilla arm effect [Schmidt et al., 2009; Moscovich, Hughes, 2008].

Im comparison to direct touch, in a indirect touch system the users cannot visually align their finger to the target before touching it. Therefore, the two-state touch interaction model [Buxton, 1990] (Fig. 3.1) needs to be extended by a *tracking* state that allows users to aim for a specific object on the vertical surface without being afraid of accidentally manipulating another object. Figure 3.2 shows the extend interaction model [Buxton, 1990].

Adding a tracking state by hovering the hand above the surface is not a practical solution.

Schmidt et al. [2009] added this *tracking* state by displayed cursors when the users' fingers hover near the horizontal surface. They compared their system with a direct multi-touch system in an aiming task. The results indicate that users perform slower in the indirect system. The authors surmised that the user experience degraded due to the fatigue of hovering fingers to track the cursors. Therefore, resting the hand on the screen should be used for cursor tracking. This leads us to the questions: How can users switch into the engaged state that allows them to manipulate the object below the cursor?

In addition to the tracking state problem, indirect touch systems also have the problems that the horizontal surface is only used as an input area. Should the horizontal input

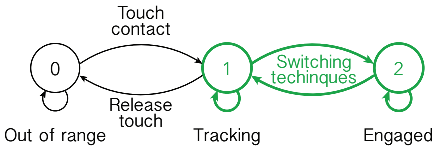


Figure 3.2: A state model for indirect touch systems. The highlighted state transitions between the *tracking* and the *engaged* are transitions that need to be investigated.

surface also become an output surface, a mode switching problem arises: Users somehow need to specify if a touch was meant for a direct control of the horizontal display they physically touched, or if it should provide an indirect control over the vertical display in front of them.

In indirect touch systems the horizontal surface is input only.

In this chapter, we will address questions of how users can switch between the *tracking* and the *engaged* state by empirically evaluate four different switching interaction techniques (C 2.1, C 2.2). In the second part of this chapter, we propose to use the users gaze as switching mechanism to determine whether the user wants to interact with the vertical surface or with the horizontal one (C 2.3). We present two different gaze-based mode switching techniques and compare them to the direct touch system with horizontal and vertical touchscreens.

3.1 Related Work

However, we are not the first who use indirect touch or combining the users gaze with touch input. Therefore, in the following section, we will present the related work in the fields of indirect touch input, three-state touch input, and gaze supported touch interaction.

3.1.1 Indirect Touch Input

For standard pointing tasks with a stylus, indirect input performed very similar to direct input, in terms of target acquisition and error rates. However, indirect input benefits from less occlusion for difficult targets [Forlines, Balakrishnan, 2008]. While direct touch was faster, indirect touch had a lower error rate and was more precise for 2D/3D rotation, scale, and translation [Knoedel, Hachet, 2011].

In a mixed direct and indirect touch environment, indirect input can be used as a high-precision mode because it reduces content occlusion and allows the user to interact with faraway objects near the body [Kosara, 2011; Benko et al., 2006; Moscovich, Hughes, 2008].

Aspect ratio between input and output surface should be the same.

Gilliot et al. [2014] analyzed the impact of size and aspect ratio differences between the input and the output surfaces on indirect pointing tasks. Their studies showed that especially a different aspect ratio between input and output surface has a strong negative effect on the user's performance. In contrast to the related work, we show how to switch between the two modes (direct and indirect touch) and thus, we overcome the problem of indirect touch systems where the horizontal touch-surface is degraded to merely an input surface.

3.1.2 Three-state touch input

While the Tracking state is necessary for indirect touch input, other projects introduced the Tracking state to direct touch input because of benefits of preventing errors or increasing the richness of the input [Buxton, 1990]. Therefore, the state switching methods from direct touch systems could inform the design of those in indirect touch systems.

Pressure could be used to switch between tracking and engaged state.

In pressure-based methods, every lightweight touch is recognized as an input in the Tracking state; the system switches to the Engaged state only when the pressure is increased [Buxton et al., 1985]. Due to friction between the finger and the surface, retaining the pressure while moving the touch on the surface could be uncomfortable. Forlines et al. [2005] used the light touches to preview a sequence of actions, and pressured touch to confirm the action.

Surface gesturing, e.g., lifting the finger while it is positioned over the target (take-off) [Potter et al., 1988] or rubbing the target [Olwal et al., 2008] are alternatives. In comparative studies, pressure technique and rubbing performs take-off. Multiple finger gestures [Matejka et al., 2009] or a bimanual gesture [Olwal et al., 2008] could also be used for state switching.

The Lift-and-tap technique has been shown as promising on a trackpad.

MacKenzie, Oniszczak [1998] compared three switching techniques on trackpads: depressing a physical button, increasing the finger pressure (and receiving tactile feedback), and lift-and-tap. To execute the lift-and-tap technique, the user starts with the finger on the trackpad (tracking the cursor), then lifting the finger, then placing the finger on the same position within a very short duration, and then lifting the finger again. They showed that the most accurate but also slowest condition was the physical button. The lift-and-tap technique was a good trade-off between error rate and speed. Thus, the lift-and-tap technique could be used for state switching.

While the above-mentioned methods allow single-touch direct-input state switching, there is no comparison of the methods for multi-touch systems. In this chapter, we will identify design criteria for state switching methods for multi-touch indirect-input systems. These criteria lead to a selection of four representative methods which were tested empirically.

3.1.3 Touch and Gaze Input

Gaze input is inaccurate and error prone to unexpected events.

Results by Stellmach et al. [2011] indicate that gaze input may be used as a natural input channel as long as certain design considerations are taken into account. Other researchers conclude that due to the inaccuracy such as the *double role of eye gaze* and the *Midas Touch* problem [Stellmach, Dachselt, 2013], it is ineffective to use eye gaze to directly manipulate digital content or control cursors. The study by Turner [2013] shows that manual input conditions outperform gaze in transferring objects from a personal device to a public display and vice versa. Also, it is shown that a dwell time method is slower in comparison to techniques that allow users to confirm their actions using touch [Turner et al., 2011].

Eye focus selection as an independent channel of input is used in several research projects due to its high speed, familiarity and naturalness [Smith, Graham, 2006; Stellmach, Dachselt, 2012; Vertegaal et al., 2005]. The eyes typically acquire a target well before manual pointing is initiated, following the principle "what you look at is what you get" [Zhai, 2003]. Users tend to look at a target before issuing a command, starting an interaction, and look at the screen of interest, which makes gaze tracking a good interaction technique for window targeting [Shell et al., 2004; Vertegaal et al., 2005].

Gaze can be used for rough selections but not for precise movement.

Using gaze as an additional input, modality was also studied with a variety of user modalities other than touch. A study by Ashdown et al. [2005] shows that combining head tracking with mouse input for a multi-monitor system is preferred by the users due to reduced mouse movement. The MAGIC (Manual And Gaze Input Cascade) technique proposed by Zhai et al. [1999] is a combination of mouse and gaze input for fast target selection. Fono, Vertegaal [2005] present an attentive windowing technique that uses eye tracking for focus selection.They evaluated four focus selection techniques and conclude that eye-controlled zooming windows with key activation provides an efficient and effective alternative to current focus windows selection techniques. Eye tracking with key activation is, on average, about as twice as fast as mouse or hotkeys. Fono's and Vertegaal's results also show that despite the difference in speed between automatic activation and key activation for eye input, the eye input with key activation is a more effective method overall for focus window selection (about 72% faster than manual conditions), and was also preferred by most of the participants.

Nancel et al. [2013] investigated high precision pointing techniques for remotely acquiring targets and concluded that using head orientation for coarse control of the cursor and touch for precise selection was the most favorable and successful technique.

There are several approaches to combine gaze with manual interaction. Turner et al. [Turner, 2013; Turner et al., 2011] combine gaze with mobile input modalities in order to transfer data between public and close proximity personal displays. The techniques for interaction in such environments are already outlined Shell et al. [2004] and Turner et al. [2011]. Turner et al.'s study shows that manual input conditions outperform gaze positioning, which gives an advantage to the usage of manual input and leaves gaze a supporting role as a switching technique between the screens of interest.

Gaze and touch are often used for public display settings.

As shown by Pfeuffer et al. [2014], the user's gaze can be used to perform this mode switch. They use gaze in combination with a single tabletop to place the user's touch points at the point of the display where the user is looking at by following the principle "gaze selects & touch manipulates". Their qualitative user study confirms the benefits of combining gaze and touch such as reachability, no occlusion, speed, less fatigue and less physical movement. They provide a design space that compares the properties of combining gaze and touch versus direct touch, and present several applications that explore how gaze-touch can be used alongside direct touch.

Use gaze of selecting an object and touch for manipulating the object.

3.2 State-Switching Techniques

Since we want to allow users to perform the same kind of interaction with indirect touch as they can use in direct touch, our main design consideration is that each finger can switch the input state independently from other fingers. Therefore, we are only interested in interaction techniques that users can perform with each finger individually. Wang, Ren [2009] characterized four types of finger input properties that modern touch-screens can detect:

We want to have a single finger state switching method.

1. Position properties: coordinate (x, y)
2. Movement properties: velocity and acceleration

3. Physical properties: contact area (size, shape, and orientation) and touch pressure
4. Event properties: such as tap and flick

The position and movement properties are already used for basic operations such as selecting or moving object. Because of that, we need to focus on the physical and event properties in order to find possible state-switching methods.

As shown by Wang, Ren [2009] controlling the orientation of the contact area of one finger is hard to change without influencing the orientation of the other fingers. Also, due to the softness of the finger controlling the finger pressure independently from the shape and size of the contact area is almost impossible [Pawluk, Howe, 1999]. In fact, Benko et al. [2006] showed that the size and the shape can be used to estimate the pressure. Thus, in the following, we use the term pressure to refer to these properties together.

For the event property, flicking cannot be used because it influences the position and movement properties. The tap event (same as lift-and-tap above) and the hold event (dwelling on the same position longer than a duration threshold) are possible state switching methods.

From the combination of the physical and event properties, we derived four techniques from the exisiting literature: *lift-and-tap*, *hold*, *pressure hold*, and *pressure switch*. In the following, we describe each technique and how it is detected by the touch-screen. To determine the thresholds for these techniques, we conducted an informal pre-study with five participants [Voelker et al., 2013d].

3.2.1 Lift-and-Tap

The Lift-and-Tap technique is based on the Lift-and-Tap technique by MacKenzie and Oniszczak [1998] as explained above. Using the Lift-and-Tap technique, the user lifts the finger off the screen and quickly lands it back at the same position. Lifting the finger may slightly change the contact point due to the softness of the fingertip and the drifting of it. Therefore, we use for the position of the tap event the last position of the finger before the finger is lifted. Each tap switches from the Tracking state to Engaged state, or vice versa.

Lift-and-Tap is an established interaction technique on trackpads.

To distinguish a lift-and-tap from a finger repositioning, we used a radius threshold r_{max} in which the fingers has to land to be recognized as a lift-and-tap event. For the timing of the lift-and-tap we used two duration thresholds: t_{min} and t_{max}. We found that the lower case t_{min} is needed since moving the finger quickly over the touchscreen may cause discontinuous touch. The upper threshold t_{max} allows us to distinguish a lift-and-tap from an intentional lifting and landing.

Our pre-study [Voelker et al., 2013d] revealed that an r_{max} of 4.14 millimeters, $t_{min} = 0.09$ seconds, and $t_{max} = 1.18$ seconds allows the system to reliably detect a lift-and-tap event.

3.2.2 Hold

The hold technique is a common touchscreen gesture that is already used in most mobile operating system such as iOS or Android. To switch from the Tracking state to Engaged state, the user places the finger on the screen and hold this position for specific duration t_{max}. Until the finger is lifted from the surface, the cursor stays in the Engaged state. The hold technique does not allow transitioning from the Engaged state back to the Tracking state. We used the t_{max} of 0.5 seconds, based on the threshold used in/by iOS. To determine how still the finger needs to be, we asked the users of our pre-study to place each finger still on the screen and capture the position of this finger for 0.5 seconds. We used the radius threshold $r_{max} = 1.94$ millimeters which is the 75% percentile of the captured data.

Hold is a common interaction technique on mobile devices.

3.2.3 Pressure Hold

In 1990 Buxton already proposed the use of pressure as an state switching method.

The pressure hold technique is based on an interaction technique introduced by Buxton [1990]. If a user is applying small pressure on the surface the touch is in the Tracking state. Strong pressure is used to switch from the Tracking state to Engaged state. To stay in the Engaged state, the pressure must be maintained. Going back to the Tracking state is done by reducing the pressure.

As shown by Ohtsuki [1981] each finger applies different pressure levels, which also changes when pressing multiple fingers at the same time. Thus, using absolute pressure as threshold to determine the state switch should be avoided. Therefore, we used the rate of pressure change over time as switching method ($\frac{dP}{dT}$).

Pressure can be estimated by the size and shape of the contact area of the finger.

Due to the fact, that capacitive touchscreens cannot measure pressure, we need to derive the pressure from the other properties. However, as mentioned above, Benko et al. [2006] showed that pressure can be estimate by the size and shape of the contact area of the finger on the surface. For this reason, we used the length of the major axis of the touch contact area to determine the pressure. From now on, we will use the term pressure to refer to this approximation, and the pressure thresholds below refers to the length of the major axis in millimeters.

The contact size of a finger changes while moving the finger.

However, there are serval problems while determining the pressure using the touch contact area. The main problem is that if users apply pressure, the center of this contact area shifts, so while applying pressure the courser is also shifting. We use a radius threshold r_{max} to address this problem. In addition, when the user moves the finger on the touchscreen, the shape of the touch ellipse also changes quickly according to the contact angle of the fingertip. Thus, we only detect a pressure hold event when the pressure change occurs within a short duration t_{max}.

In summary, to switch from the Tracking state to Engaged state, users increase the pressure quickly ($\frac{dP}{dT} > \delta_{engage}$ and $dT < t_{max}$), while the center of the contact area only changes within the radius r_{max}. To go back to the tracking state, users can either release pressure in a short timeframe ($\frac{dP}{dT} > \delta_{disengage}$), or reduce pressure below the absolute threshold P_{min}.

To determine these thresholds, users have use the pressure hold technique while dragging an object from a starting point into a goal on the screen. We used the 75th percentiles of the tested thresholds: $t_{max} = 0.70$ seconds, $r_{max} = 7.07$ millimeters, $\delta_{engage} = 1.30$ millimeters/second, $\delta_{disengage} = 1.14$ millimeters/second, and $P_{min} = 0.55$ millimeters.

3.2.4 Pressure Switch

In this interaction technique we used a short impulse of pressure to switch between the Tracking and the Engaged state. This technique is similar to the technique to access the quick-actions-menu on modern iPhones.

To detect the pressure switch, we only use the rate of pressure change ($\frac{dP}{dT}$). We determined this threshold by asking our users to perform this switch technique on 12 positions, evenly spaced in a grid across the screen. We used the $\frac{dP}{dT}$ = 1.30 mm/s, which is the same for the pressure hold technique.

3.3 Evaluation of the State-Switching Techniques

To find out how user can perform these techniques, we conducted three experiments: single-finger, two-fingers, and two-hands. These experiments were designed to cover the most common use case of multi-touch input. In the following, we describe the tasks, the experimental setups, and the results of all three experiments.

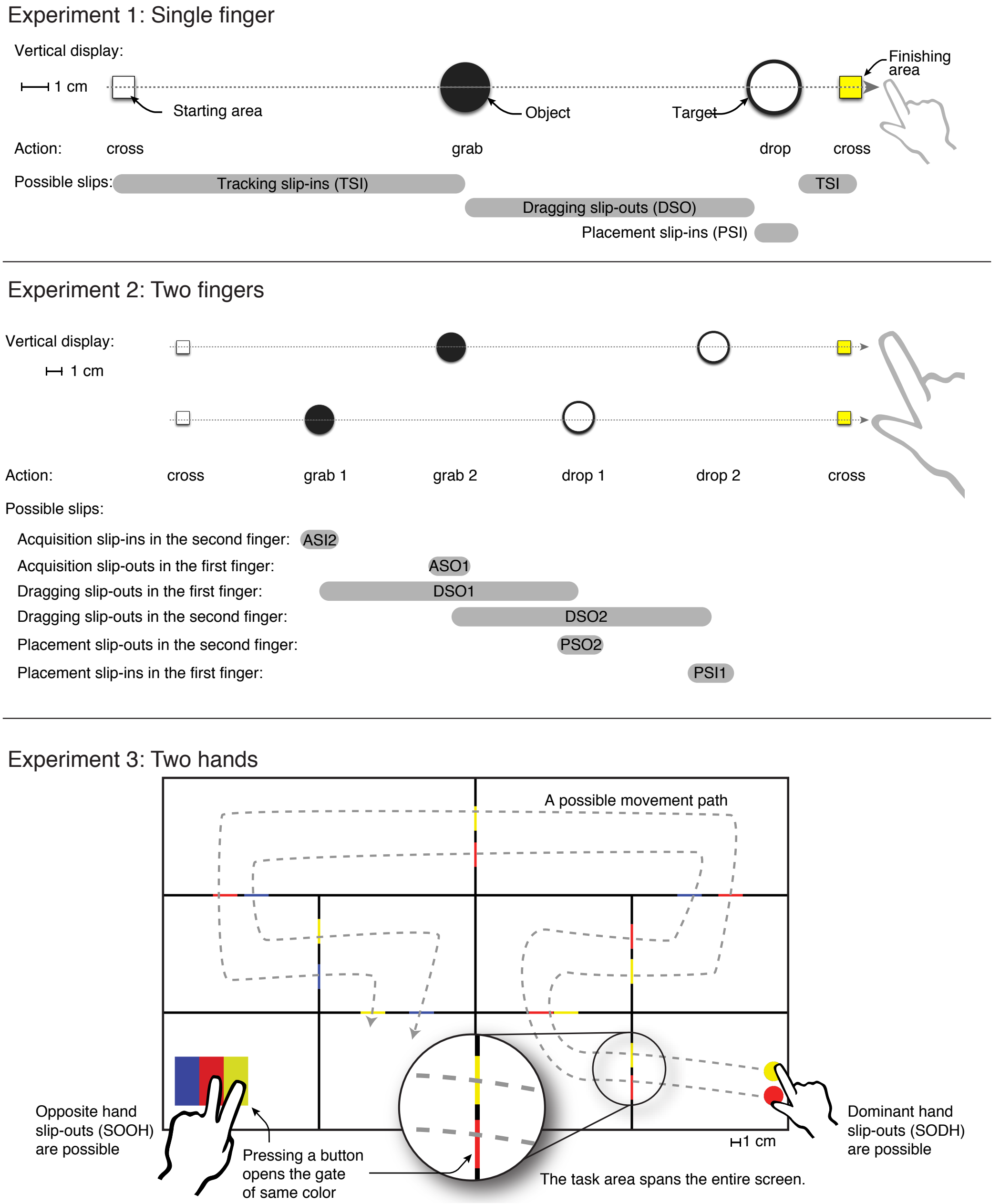


Figure 3.3: The task configuration for the three experiments and associated slips. (Originally from Voelker et al. [2013d], modified version from Wacharamanotham [2016].)

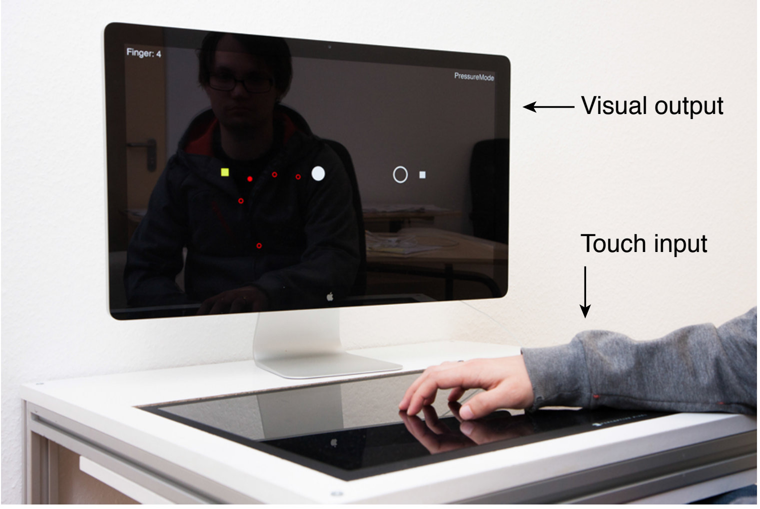


Figure 3.4: The indirect multi-touch system used in our experiments [Voelker et al., 2013d].

3.3.1 Apparatus

The basic study setup for all three studies is show in Figure 3.4. We used two displays of the same size and output resolution (27”, 2560 × 1440 pixels).

Touches were detected by a capacitive touch-screen from Perceptive Pixel (touch frame rate: 205 Hz). During the experiments nothing was displayed on this screen. The task was shown on a vertical Apple cinema display.

The participants controlled multiple cursors on the vertical screen by touching the horizontal screen. For each contact point a circle cursor with a diameter of 7 mm (30 px) was displayed. The cursors were outlined in the Tracking state and were filled in the Engaged state.

To reduce the cognitive load of mapping the cursers to the fingers, we chose a very simple 1:1 absolute mapping without curser acceleration.

3.3.2 Tasks

Users were asked to interact with an object using the four proposed interaction techniques.

To cover all state switching conditions, we created the following tasks based on Forlines, Balakrishnan [2008]. Figure 3.3 shows all three tasks.At the beginning of each task, the users had to move the cursor into a starting area (blank square). After that they move the cursers while being in the tracking state onto the objects (filled circles). On the objects they had to switch to the Engaged state and drag the objects onto the targets (blank circles) and then move the cursors to cross the finishing area (filled square). In the third experiment, the users had to drag two objects through a mace with colored gates. After using each technique, we asked the participants to comment on speed, accuracy, fatigue and their preferred technique.

3.3.3 Experimental Design

The focus of these experiments was to analyze whether the users are able to switch between the Tracking and Engaged state without unintentionally *slip in* or *slip out* of the desired state. For example, while dragging the object the users may unintentionally switch from the Engaged state to the Tracking state. To be able to analyze these *slip in* and *slip out* errors we measured the following errors:

Experiment 1: Single Finger:

- **Tracking slip-ins (TSI):** The number of slip-ins between the starting area and the object.
- **Dragging slip-outs (DSO):** The number of slip-outs that occur while dragging the object towards the target.
- **Placement slip-ins (PSI):** The number of slip-ins after the object is dropped onto the target.

Experiment 2: Two Fingers:

- **Acquisition slip-ins in the second finger (ASI2):** While the first finger is trying toacquire the object, ASI is the number of slip-ins in the second finger.

- **Acquisition slip-outs in the first finger (ASO1):** While the second finger is trying to acquire the object, ASO is the number of slip-outs in the first finger.

- **Dragging slip-outs in the first finger (DSO1)**, and **Dragging slip-outs in the second finger (DSO2):** The number of slips-outs during dragging from each of the fingers.

- **Placement slip-outs in the second finger (PSO2):** During the placement of the first object, PSOI is the number of slip-outs of the second finger.

- **Placement slip-ins in first finger (PSI1):** During the placement of the second object, PSI1 is the number of slip-ins of the first finger.

Experiment 3: Two Hands:

- **Dominant hand slip-outs (SODH)**, and **Opposite hand slip-outs (SOOH)**:The number of slip-outs of the respective hand during bi-manual interaction.

The terms are the modified terms defined by Wacharamanotham [2016], originally defined by Voelker et al. [2013d]. In addition, figure 3.3 shows where these slips can occur. Additionally, we also measured the trial completion time in the single-finger experiment. Each trial comprised of the movement in all directions for each finger as indicated below.

In all three studies we used a within-subject design. Table 3.1 summarizes the experimental design and demographics of the users.

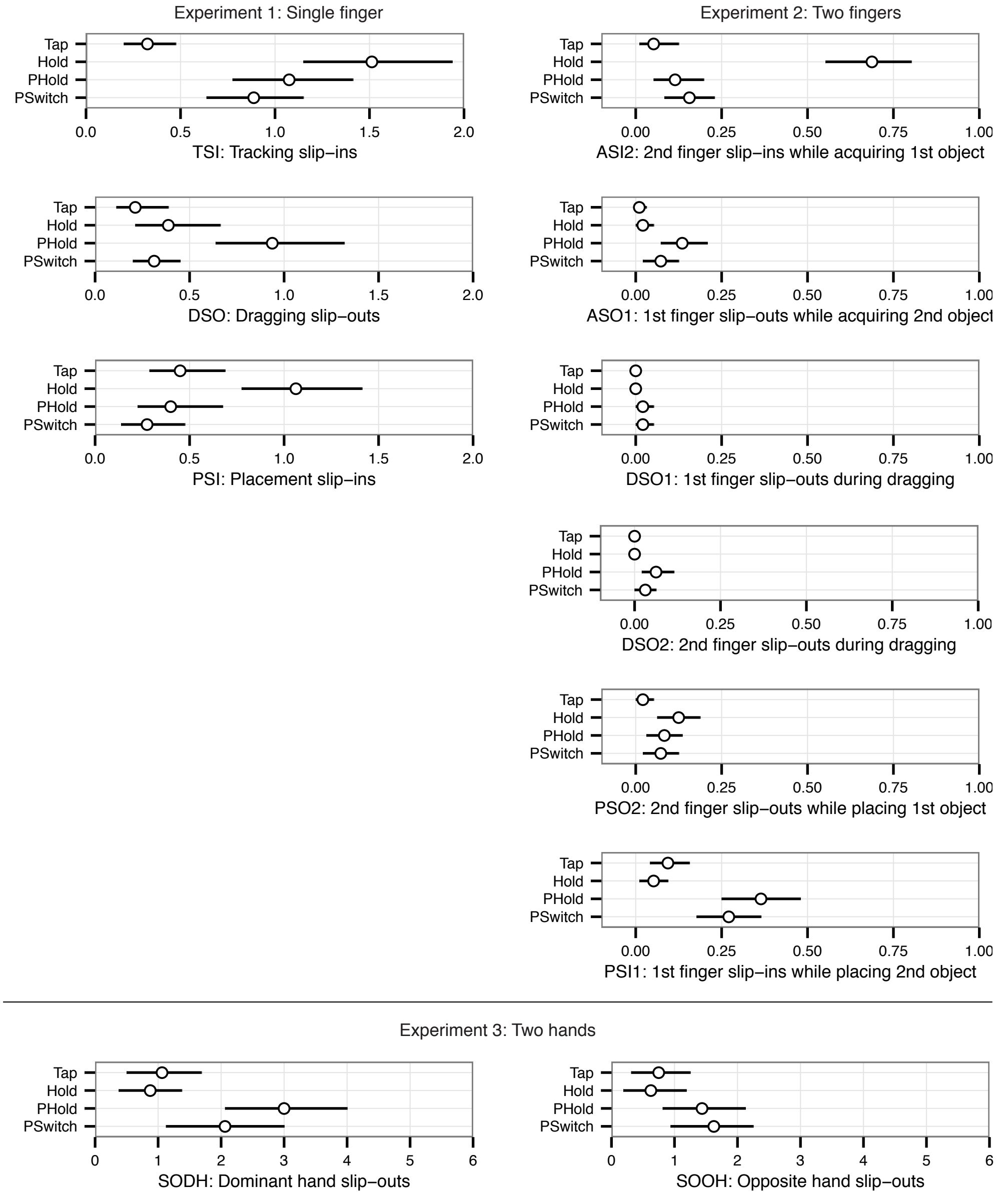


Figure 3.5: Slips per trial from the three experiments. (Mean and 95% CI without within-subjects adjustment) The length of each abscissa differs between experiments [Wacharamanotham, 2016].

	Experiment 1: Single finger	Experiment 2: Two fingers	Experiment 3: Two hands
Age	24–34	24–38	24–30
Gender	All males	One female	Two females
Handedness	All right-handed	All right-handed	All right-handed
Movement directions	←, →, ↑, ↓	←, →	←, →
Fingers used	10 fingers	6 combinations of thumb, index, middle finger of each hand	Thumb and index finger
Per participant	160 trials 30 minutes	48 trials 35 minutes	8 trials 15 minutes

Table 3.1: Demographic information and experimental designs

3.3.4 Results

For all 12 slip types, we averaged the number of occurrences per technique per user. To determine the means, we used ordinary non-parametric bootstrapping (10,000 replicates). CIs were calculated with the bias-corrected and accelerated method (BCa) as described by Wacharamanotham [2016]. All error bars are 95% CIs.

For the task completion time in the first experiment, we log-transformed the data prior of all calculations. The plots presented are anti-logged to the original scale. Such that in the task completion time, the means are geometric, and the differences between means are ratios.

The results are shown in figure 3.5. The main finding is that the lift-and-tap techniques creates fewer slips then the other techniques. In the following, we highlight the main differences between these techniques.

The biggest difference between the lift-and-tap technique and the other techniques can be seen in the tracking slip-ins (TSI). These results show that while using the lift-and-tap technique users almost never unintentionally slip-in into the Engaged state. In comparison, using the other techniques the error rate is significantly higher, especially while using the hold technique.

Using lift-and-tap, users almost never make a slip-in error.

In general, compared to the other techniques, users tend to do more slip-in errors while using the hold technique. This is especially noticeable for the ASI2, PSI and the PSI1 slip-ins.

Comparing both pressure techniques, the pressure hold technique creates more slip-outs then the pressure switch technique. This can been seen for the DSO, DSO2 and SODH slip-outs.

Users preferred the lift-and-tap method.

The qualitative results agree with presented results above. Most of the participants (single-finger: 4/8; two-fingers: 7/8; Two-hands: 5/8) chose the tap technique as their preferred technique. Several participants commented "I would use this if there were no instructions." and "It's the closest to the mouse."

3.3.5 Discussion

The lift-and-tap method is the most explicit gesture and therefore can be controlled very easily.

The results show that the lift-and-tap techniques outperform the other techniques. It was also preferred by the participants in all three experiments. Therefore, we recommend lift-and-tap as the default state switching method (C 2.1). While using this technique, the coupling between fingers does not seem to influence the user's performance (low ASI2, ASO1, PSO2, and PSI1). Also a lack of tactile feedback did not influence the lift-and-tap technique, even when the focus of attention is away from the fingers (low SOOH). However, designers should also consider the influence of the form factor, the UI widgets, and the task. For example, since lift-and-tap loses the touch temporarily, two nearby touches may trade their places, especially in a small device.The nature of the UI widget and the task should also be considered when choosing a method. For example, hold and pressure switch may be more suitable than lift-and-tap for an on screen quasi-mode. Interaction designers may allow an alternative switching method on these UI widgets in addition to lift-and-tap. In some scenarios, a combination of lift-and-tap and hold can be beneficial in bi-manual interactions. As the third experiment demonstrated, hold yielded almost no errors—comparable to lift-and-tap—for

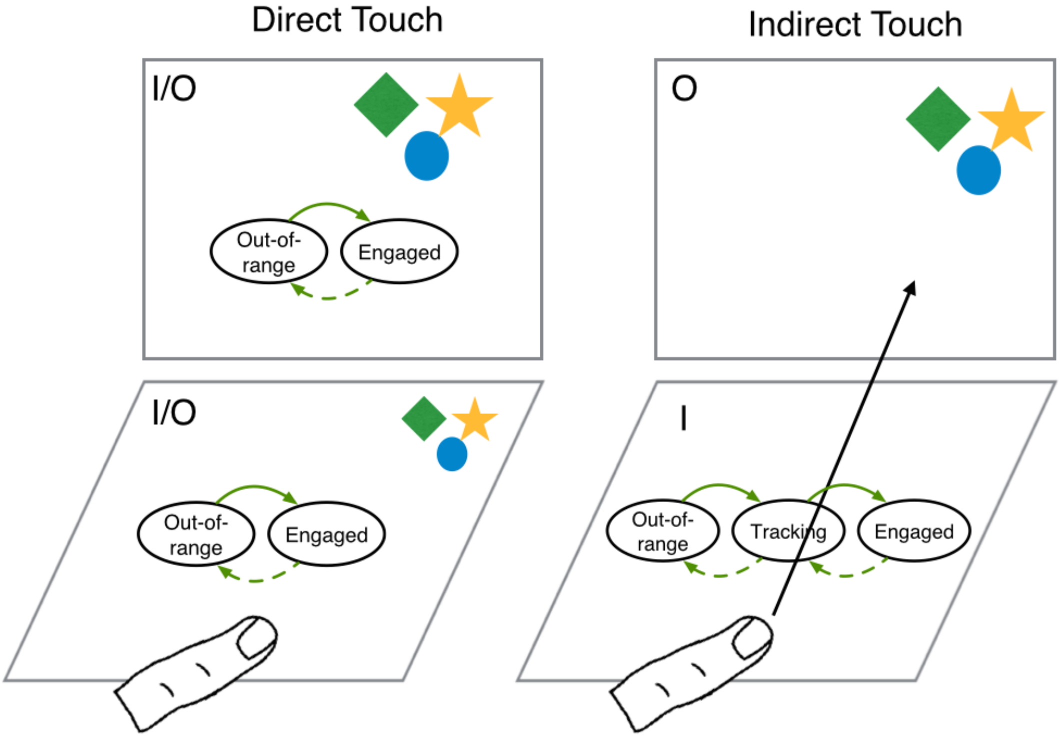


Figure 3.6: Direct and indirect touch interaction models for interactive workspaces. In a direct touch system (left) both surfaces are used as input and output at the same time. The users interact with both surfaces using common two-state touch input. In an indirect touch system (right) the horizontal surface is used for input and the vertical surface is used for output. The users are interacting with the system by creating three-state touch input that is mapped to the vertical surface [Voelker et al., 2015b].

targets with fixed-position, e.g., a button. For example, in a 3D scene construction application [Kin et al., 2011], the user could use the non-dominant hand to select a virtual object while placing it on the scene with the dominant hand.

However, for the following parts, we will use the lift-and-tap technique as the default mode switching technique for indirect touch input.

3.4 Combining direct and indirect touch

In an indirect touch system, the horizontal surface is used only as input.

In the first part of this chapter we introduced the indirect touch concept allowing users to interact with digital content on the vertical surface by creating the input on the horizontal surface. We addressed the issues of how user can approach objects using indirect touch without unintentionally modifying other objects by introducing a tracking state that allows users to aim a specific target without manipulating other objects. We introduced four interaction techniques that enable users to use this tracking state and in three user-studies we examined which of the four techniques would be most suitable as a switching technique.

We want to use gaze to decide whether the input should be direct or indirect.

In the following we will address the problem that in an indirect touch system the horizontal surface is only used as an input devices since in an indirect touch system all touches are directly mapped to the vertical surface. We solved this problem by allowing the users to choose using their gaze for each touch point whether it should be used for direct or indirect touch input. That gaze and touch input can be easily combined as it is shown in the related work section above. In the following, we will introduce two gaze and touch techniques that enable users to combine direct and indirect touch using gaze input. Afterwards, we will compare both techniques with a pure direct touch setup in order to evaluate how users perform using both gaze and touch techniques compared to this base line setup.

3.4.1 Indirect Touch Surface Selection (ITSS)

The first interaction technique, named ITSS, combines absolute direct touch (DT) and absolute indirect touch (IT) as such: If the gaze is directed towards the horizontal touch surface, the system maps the touch point to the horizontal screen, allowing the user to interact with the object using the two-state DT interaction model (Fig. **??** left). If the user is looking at the vertical surface, the touch is translated using an absolute mapping to this particular vertical surface.

For example, if both surfaces have the same size and resolution, when a user touches at point $P_H(10,10)$ on the horizontal screen, the touch point is mapped to point $P_V(10,10)$ on the surface currently in view. Now, instead of using the two state interaction model, the three state indirect touch model, as described above, is used. Each new touch point that is mapped to a vertical surface is in a *tracking* state, which permits the user to move the finger over the surfaces without manipulating any object. To change the touch to the *engaged* state, which allows object manipulation, the user has to execute a *lift-and-tap* gesture. This process is illustrated in Figure 3.7.

With the ITSS technique, the users are selecting the surface to which the touches should be mapped using their gaze.

In order to provide feedback, a cursor represents the touch point, displayed on the surface to which it is mapped. In both cases, if the touch is mapped to the horizontal or to the vertical surface, the touch stays on the surface until the user releases the finger from the input surface. However, this is not the focus of this paper.

For the ITSS interaction technique we used an absolute mapping to prevent confusion when multiple cursors are presented on the screen. Especially the usage scenarios which involve multi-touch and bi-manual multi-touch input, the cognitive load from mapping multiple touches and multiple cursors would be overwhelming.

3.4.2 Indirect Touch Object Selection (ITOS)

The first step of the ITOS interaction technique is the same as the one of ITSS. Again, if the gaze is directed towards the horizontal touch surface, the system maps the touch point to the horizontal screen, allowing the user to interact with the object using the two-state DT interaction model.

For the second step, we initially planned to use the user's gaze not only to select the surface as in ITSS, but instead we transfer the initial touch point to the position of the surface where the user is currently looking at. Anyway, to the constant movement of the eyes, it is complicated to determine the exact position where the user is looking at as shown by

Using ITOS, the users are selecting the object to which the touches should be mapped using their gaze.

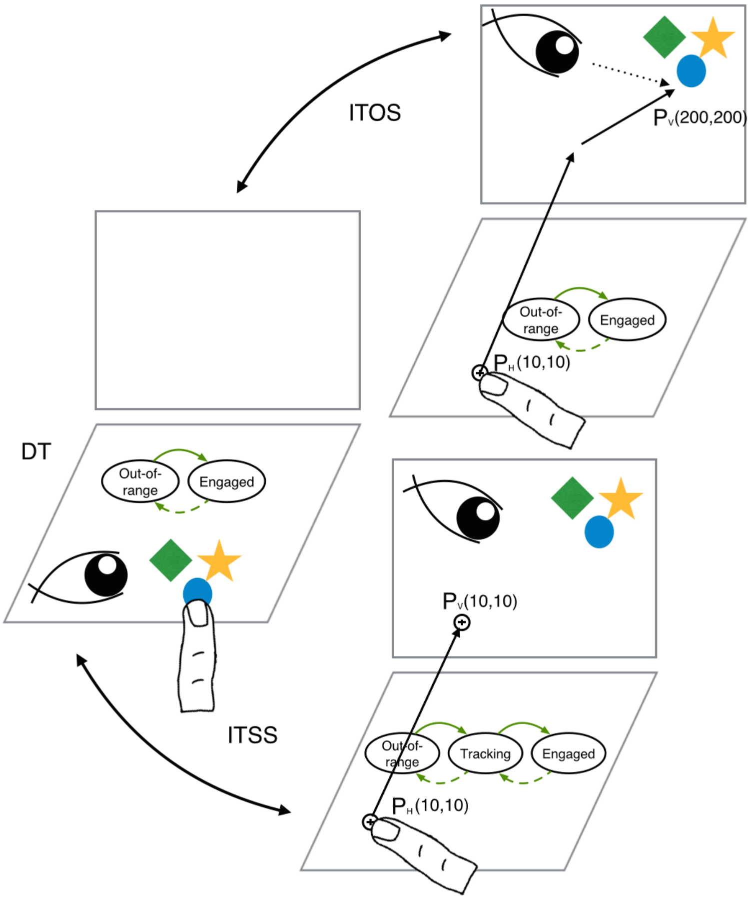


Figure 3.7: DT: traditional direct touch interaction. While using ITSS, the user's touch from the horizontal screen is absolutely mapped to the vertical screen, if the user is looking at it. While using ITOS, the user's touch is directly mapped to the object on the vertical screen that is in the user's focus [Voelker et al., 2015b].

Stellmach, Dachselt [2012]. Figure 3.7 illustrates the interaction concept.

In a plot study, we also encountered the problem that users actually looked at a point to which the touch point should be mapped while they started the finger movement to create a touch point. But at the same time when the finger touched the surface, the user's gaze was already directed at a different location on the screen. This leads to the problem to determine at which point in time exactly (before the touch is recognized by the system) the user did look at the location where he or she wants to place the cursor.

Therefore, we choose to use a similar snapping mechanism for the ITOS technique as proposed by Pfeuffer et al. [2014]. If the user is looking at the vertical surface or outside the horizontal screen, the touch is now translated to the object, on which the users gaze is concentrated. In this case, the user's gaze selects an object by highlighting it and a touch confirms this selection. To determine which object the user is currently looking at, we use an approach similar to the Bubble Cursor technique introduced by Grossman, Balakrishnan [2005]. If a user touches the surface while he is looking at the vertical screen, the system calculates the area at which the user was looking in the last 50 ms. If this area contains only one object this object is selected.

To determine which object is selected we used the Bubble Cursor approach.

If multiple objects are located in this area, the system calculates the center of the area and then selects the object which is closest to this center. For example, if a user touches point $P_H(10,10)$ on the horizontal screen and looks at the object on the vertical screen that is located at $P_V(200,200)$, the touch point $P_H(10,10)$ is mapped to point $P_V(200,200)$. In contrast to the ITSS technique, using ITOS requires no *Tracking* state, since the system highlights an object to which a touch is mapped before the user touches the screen. This allows the user to be certain that he or she is only interacting with one specific object without manipulating other objects.

3.5 Evaluation

We designed three different experiments to compare the user's interaction with different interaction techniques. We compared ITSS and ITOS against a DT baseline condition for a *tapping*, *dragging* (dragging an object on the vertical or the horizontal surface) and *cross dragging* (dragging an object from the vertical to the horizontal surface and vice versa) task.

In this chapter, we focus only on single-touch tasks in order to get a principal idea of how users perform utilizing our two proposed interaction techniques in basic tasks. Our experiments aim to answer the following questions:

1. Which technique is preferred in the indirect touch setup?
2. Which technique allows users to complete tasks faster and more accurate?

All three experiments were within subject experiments and we used the same setup and general procedure.

3.5.1 Participants

We recruited 14 participants (five females and nine male) aged between 23 and 36 (mean age 27.0). Twelve of the participants were right-handed and two were left-handed. All three experiments were conducted with the dominant hand of the user. On average, it took the participants about 1.2 h to complete all three experiments.

3.5.2 Apparatus

Participants sat at a desk with two touch displays, as shown in Figure 3.8. As a horizontal screen we used a capacitive touch-sensing 27″ Acer Touch display embedded in a custom made table at a height of 72 cm following ISO9241-5. For the vertical screen, we used a 27″ Perceptive Pixel display which was placed 55 cm from the edge of the table. Both displays had the same resolution of 2560 x 1440 pixels and the size of 597 x 336 mm. Both displays were connected to a Mac Pro running the software for the experiments. The effective touch frame rate for both displays was set to 60 Hz.

To determine the gaze of the users, we used the *Dikab-*

Figure 3.8: The Gaze + Touch system setup. With help of the users gaze, one can interact on the horizontal surface using direct touch and on the vertical surface via indirect touch.

lis Glasses by Ergoneers[1]. The *Dikablis Glasses* are a head-mounted eye-tracking system that is able to detect the position of the user's gaze in a visual marker coordinate system. Two markers were placed around the vertical display, as shown in Figure 3.8. By doing so, we can convert the gaze coordinates into the pixel coordinate system of the vertical screen with an accuracy of about 1.5 cm (63 px). The effective frame rate of the eye-tracker was also set to 60 Hz. The eye-tracker was calibrated with a standard routine that comes with the eye tracker for each user before conducting the user study. This calibration process took about 30 seconds.

3.5.3 General Procedure

The participants conducted each experiment with all three interactions techniques:DT, ITSS and ITOS. The experiments were executed by each participant in a random order. No learning effects were observed by the experimenter or appeared in the data. Before the experiments, the users

---

[1] `http://www.ergoneers.com/`

could run a ten-minute test trial to familiarize themselves with the new interaction techniques. It was emphasized to solve a task as fast and as accurately as possible.

3.5.4 Experiment 1: Tapping

In the first experiment, we investigated the effect of the three different interaction techniques on the users' performance by running the tapping task on both the horizontal and vertical surface.

Task

Users were asked to tap objects on the horizontal and on the vertical surface using all three interaction techniques.

Participants were asked to touch blue circles, which were displayed alternately on the horizontal and vertical surface. As soon as the user touched the circle he had to hold his finger for 0.5 seconds on the circle before the circle disappeared and a new target circle on the other surface appeared. The task time was measured from the moment the target circle was visible till the moment it was successfully touched by the user. In order to complete one trial, users had to repeat this task for 50 targets, 25 on the vertical and 25 on the horizontal surface. The exact position of these targets was predefined and was the same for all the users. Furthermore, during a trial the circle size was fixed. The users had to conduct one trial for three different circle radii—63 px (1.5 cm), 126 px (3 cm), 252 px (6 cm). The 1.5 cm circle represents the smallest touchable button on a mobile device such as the Apple iPhone, the 3 cm circle a control element, and the 6 cm circle a picture or a document. The experimental design was a 3 (interaction technique) × 3 (target size) × 2 (target surface) mixed design with repeated measurements, which summarizes to a total of 450 tapping tasks per user. Since the required arm movement in the ITSS condition is expected to be smaller compared to ITTS and DT, we hypothesized the following outcome:

H1: Touching a target displayed on the vertical surface using indirect touch object selection is faster than using indirect touch surface selection and direct touch.

Results

ITOS was the fastest method.

The measured values were logarithmically transformed, according to the logarithmic distribution of the data. The data was analyzed for all dependent variables *interaction technique*, *target size* and *target surface* using a repeated measures ANOVA. We saw a significant main effect for the factor *interaction technique* in the ANOVA results $(F(2, 221) = 438.8255; p = 0.0001)$.

The post-hoc Tukey HSD test comparison showed that overall tapping durations using ITOS (mean 0.61 sec) were 32% shorter while using DT (mean 0.9 sec) and 60% shorter than ITSS (mean 1.54 sec). The ANOVA showed a significant main effect of the factor *target size* $(F(2, 221) = 78.7119; p = 0.0001)$.

The post-hoc Tukey HSD comparison showed that the tapping time on objects with a size of 63 px (1.5 cm) (mean 1.14 sec) was 19% longer than on objects with a size of 126 px (3 cm) (mean 0.92 sec) and 29% longer than on objects of size 252 px (6 cm) (mean 0.8 sec) for all three techniques. The main effect of the factor *target surface* was not significant.

For DT and ITSS, the target size had a strong influence on the tapping time.

The ANOVA showed a significant interaction effect between the factors *interaction technique, target size* and *target surface* $(F(4, 221) = 4.301; p = 0.0001)$. The post-hoc Tukey HSD comparison revealed (among other results) the following: Using ITSS on the vertical screen the tapping times for all three target sizes was significantly slow compared to both other interaction techniques. On average the users need 2.15 sec to tap the small circles, 1.87 sec to tap the medium circles, and 1.68 sec to tap the large circles. Compared to both other interaction techniques, the users were faster using ITOS (H1). On average the users need 0.52 sec to tap the small circles, 0.5 sec to tap the medium circles, and 0.49 sec to tap the large circles.

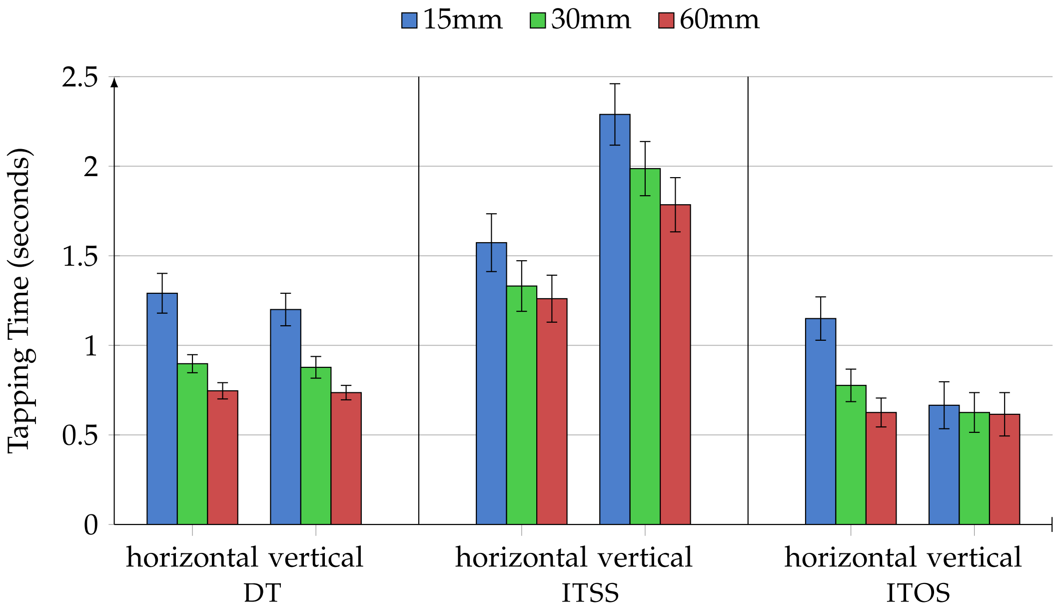


Figure 3.9: The users tapping times using all three interaction techniques in the Tapping experiment. The results show that the tapping time on the vertical surface is significantly shorter in the ITOS condition compared to the other conditions. Whiskers denote the 95% confidence interval.

Discussion

ITOS is the best method for tapping tasks since users can select the object with their gaze.

As expected, the ITOS technique was altogether the fastest tapping technique in comparison to DT and ITSS (Fig. 3.9). This can be explained by observing how users executed these tapping tasks. At the moment the new target was displayed, the users already touched the horizontal surface, since they previously touched a target object on the horizontal surface. So they only had to find and look at the new target to trigger the selection process. As soon as the target was highlighted, they only had to lift and tap anywhere on the horizontal surface again. Both of these actions can be executed very fast, especially finding and looking at an object on a nearly empty display. But also executing a lift and tap gesture on the horizontal surface can be done very fast: the users did not have to hit the same position on the display which they touched before releasing the finger.

ITSS requires to execute the lift-and-tap gesture on top of the objects.

In comparison to other interaction techniques, ITSS required a longer interaction sequence. In the direct touch condition, the users had to move their entire arm to touch an object on the vertical surface, which required more time, since not only the arm muscles are involved in the movement but also the shoulder. In the ITSS condition, the users also had to find and look at the new target, but instead of lifting and tapping anywhere on the surface, the users had to execute a more complex sequence of actions. First, the users had to estimate to which surface their touch is currently mapped. Secondly, the users had to move their arms to the estimated area and touch the horizontal surface with their fingers. Next, the users had to identify whether the cursor on the vertical surface was actually on the target. If so, the user had to execute a lift-and-tap gesture in order to successfully hit the target. If not, the users had to move their fingers until the cursor was on the target and then execute the gesture. In contrast to the lift-and-tap gesture in the ITOS condition, the users had to make sure that they tapped on the target.

Considering the object size, the objects with a bigger size were selected faster than smaller ones. However, this is not true in the ITOS condition for targets that where displayed on the vertical surface. For these targets, no significant differences were observed. This can be explained by the fact that finding and looking at an object on another empty screen is very fast and is not influenced by the size of the object, at least in this experiment. In other use cases where in a small area of the surface a lot of objects are displayed (e.g. menu with multiple buttons), the user would require more time to find the desired object. Also, due to the constant eye movement, it took the system longer time to decide at which of the object the user is currently looking.

3.5.5 Experiment 2: Parallel Dragging

After we analyzed how three interaction techniques influenced the users' performance on tapping the objects, we wanted to explore how the users' performance is influenced by our interaction techniques while dragging the objects on the horizontal and vertical surfaces. Furthermore, for the direct touch condition, we wanted to check whether the vertical dragging introduces a fatigue effect that influences the users' performance.

Task

Users were asked to drag objects on the horizontal and on the vertical surface using all three interaction techniques.

Users were asked to drag blue circles (160 px) to yellow rings (160 px) within the same display on the horizontal and vertical screens one after another. The initial scene displayed two circle ring pairs with a fixed distance of 1300 px (30 cm). Users were instructed to first start with the horizontal screen. The object is accounted as being at the destination if the position of the circle matches the destination ring within a range of 20 px. When circle and ring match and the user releases the hand, both objects disappear from the scene. This task is then repeated on the vertical surface. The next trial started from the screen where the previous one was finished. To complete this task, users had to drag 25 objects into its targets on each screen. As depended variables we measured the dragging times on vertical and horizontal screens. Time was measured from the moment the circle was touched by the user until it was successfully released in its target ring on the same surface. Furthermore, the length of dragging trajectories was recorded.

The experimental design was a 3 (interaction technique) × 2 (surface) mixed design with repeated measurements, which summates a total of 300 dragging tasks per user. Based on the results presented in the second chapter that investigated the use of DT for an interactive workspace, and based on the fact that the user cannot rest their arms while interacting directly with the vertical surface, we hypothesized the following outcomes for the second experiment:

H2: Direct dragging is faster than indirect.

H3: Direct dragging an object on the vertical surface is less accurate than direct dragging on the horizontal and indirect dragging on the vertical surface.

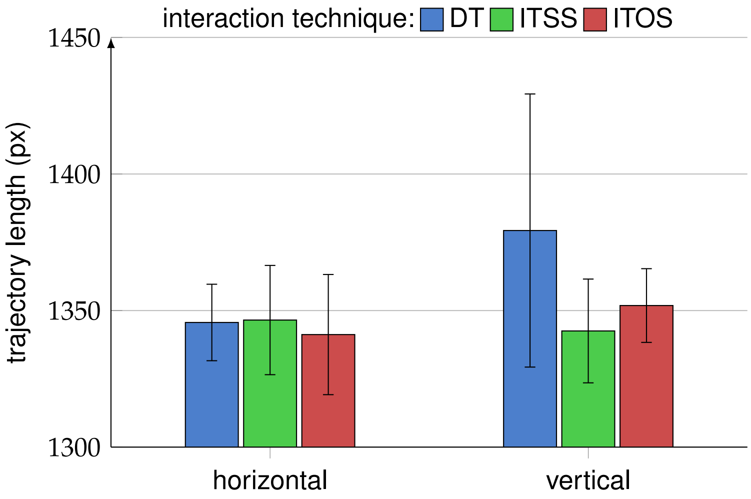


Figure 3.10: User's dragging trajectory length using all three interaction techniques in the Dragging experiment. Scale starts at 1300 px which was the minimal distance the user had to drag. Whiskers denote the 95% confidence intervals.

H4: The dragging trajectory length increases over time while dragging objects on the vertical surface using DT.

Results

Directly dragging on the horizontal surface was the fastest.

Due to the logarithmic distribution of the measured values for both depended variables, dragging time and trajectory length were logarithmically transformed. We analyzed both depended variables using a repeated measures ANOVA. For the *trajectory length*, the ANOVA reported a significant main effect of the factors *interaction technique* ($F(2,65) = 13.4972; p = 0.0001$) and *surface* ($F(2,65) = 18.5804; p = 0.0001$). The interaction also showed a significant effect ($F(2,65) = 12.5804; p = 0.0001$).

The post-hoc Tukey HSD showed that the dragging trajectory for the DT (mean 1368 px) was significant longer then the dragging trajectories for ITSS (mean 1347 px) and ITOS (mean 1346 px). It also showed that the trajectory length on the vertical surface (mean 1362 px) was significant longer than the trajectory length on the horizontal surface (mean 1345 px).

The post-hoc Tukey HSD for the interaction showed that the trajectory length for the DT condition on the vertical surface (mean 1391 px) was significant longer than the other conditions (mean 1344–1350 px), as shown in Figure 3.10.

For the variable *time*, the ANOVA reported a significant main effect of the factors *interaction technique* $(F(2,65) = 27.6531; p = 0.0001)$ and *surface* $(F(2,65) = 19.2332; p = 0.0001)$. The interaction also showed a significant effect $(F(2,65) = 6.9140; p = 0.0001)$.

The post-hoc Tukey HSD showed that the dragging time for the DT (mean 1.583 sec) was significant shorter than the dragging trajectories for ITSS (mean 1.901 sec) and ITOS (mean 1.8729 sec). It also showed that the dragging time on the horizontal surface (mean 1.869 sec) was significantly shorter than the dragging time on the vertical surface (mean 1.695 sec).

The post-hoc Tukey HSD for the interaction (Fig. **??**) showed that the dragging time for the ITOS and the ITSS condition on the vertical surface (mean 1.994 sec; 2.08 sec) was significantly longer than the other conditions (mean 1.571–1.75 sec).

Discussion

Dragging on the horizontal surface only involves the forearm and the wrist and is therefore fast.

As shown in Figure 3.10, the user's dragging trajectory is longer while dragging an object directly on the vertical surface in comparison to dragging it directly on the horizontal or indirectly on the vertical surface. As this could be expected, it can be explained by the understanding of the user's dragging operation execution. Movement of the fingers on the horizontal surface involves mostly the movement of the forearm and the wrist. However, users are able to rest their hands on the table during the horizontal drag-

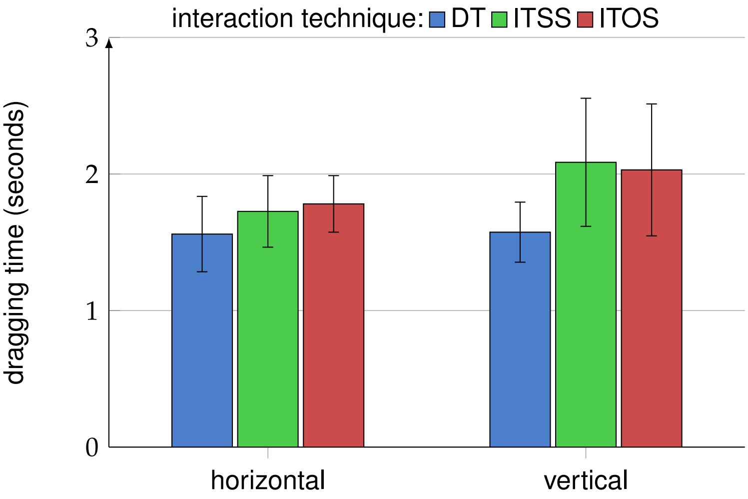


Figure 3.11: User's dragging times using all three interaction techniques for both subtask in the Dragging experiment. Whiskers denote the 95% confidence interval.

ging operation. When users are directly touching the vertical surface, the dragging movement mostly involve the upper arm and shoulder joints, which is more inaccurate as shown by Hammerton, Tickner [1966]. Interestingly, as shown in Figure 3.11, the DT technique was predominantly the fastest dragging technique on the vertical screen in comparison to the other two.

The shorter task-completion time in the DT condition might be caused by a fatigue users experienced after some time of interaction. Therefore, physical exhaustion decreases the time users want to spend holding their hands in the air. Furthermore, this could also indicate that dragging an object using indirect touch is cognitively more challenging than dragging it directly. Both points need to be taken into consideration when designing interaction workspace for all-day use.

Dragging directly on the vertical surface requires the users to move their entire arms and is therefore slow.

Considering the distance an object traveled on the vertical screen, it was longer for the DT technique than for the other two techniques, which could be explained by the loss of accuracy after a long-term DT interaction on the vertical screen. Physical movement of the hand while using ITSS and ITOS was always performed on the horizontal surface, which was causing less fatigue over time, because users were resting their hands on the surface while interacting. This shows an interesting interplay between the vertical and the horizontal surface. After this dragging experiment, mostly all users (twelfth) stated that especially dragging objects in the DT condition was extremely exhaustive. However, H2 was rejected since our recorded data did not show that this had any effect on the dragging time or trajectory length. But since the experiment took only about 3–5 minutes maybe it was too short to show a fatigue effect using direct touch on the vertical surface.

3.5.6 Experiment 3: Cross Dragging

After analyzing dragging operations that were only involving one of the surfaces, we wanted to explore how the interaction techniques affect the user performance in dragging tasks that involve switching from one to another surface. Additionally, we also wanted to explore if the effect that we have seen in the cross surface dragging experiment from the second chapter can be observed using our interaction techniques. They showed that in diagonal dragging operations, which involved a horizontal and a vertical surface, the user dragging trajectories are significant longer than in dragging operations that go straight up or downwards.

Task

Users were asked to drag objects from one surface onto the other with all three interaction techniques.

The task setup is similar to the cross dragging experiment presented in the second chapter. Users were asked to drag a blue circle (160 px) placed on the one surface into a white ring (160 px) placed on another surface. To execute this task, users had to drag the blue circle to the edge of the surface such that it is visible on the other surface. Then they had to switch to the other surface to continue dragging the circle. The initial scene displays a circle ring pair on the fixed distance of 1631 px (37 cm). Trials appeared in seven different movement angles: 45°, 30°, 15° to the left, 0° (which is straight up or downwards) and 15°, 30°, 45° to the right.

Dragging had to start either on the horizontal (upwards) or vertical (downwards) display. The object is accounted as being at the destination if the position of the circle matches the destination ring within a range of 20 px. When the circle and ring match and the users release their hand, both objects disappear and a new pair appears. Participants worked through 35 upwards and 35 downwards trials for each of the three interaction technique, which results in a total of 210 dragging operations per user. The system automatically stores horizontal/vertical distance, and vertical, horizontal, and switch time.

The experimental design was a 3 (interaction technique) × 2 (vertical direction) × 7 (dragging angle) mixed design with repeated measurements. With five repetitions per target, each user had to perform 210 cross surface dragging operations. Again, by extrapolating the results of the study in the second chapter (H7) and based on the assumption that users can glance at an object faster than touching the object (H5, H6), we hypothesized the following outcomes:

H5: In total, users complete the dragging operations faster using ITOS than using the other interaction techniques.

H6: Using ITOS, the time in which the user switches from interacting with one surface to interacting with the other surface is the shortest.

H7: The overall dragging trajectory is longer for larger dragging angles.

Results

Due to the logarithmic distribution of the measured values for all dependent variables, such as *overall time* (overall task completion time), *vertical time* (time needed to move an object on the vertical screen), *switching time* (time needed to switch from the horizontal to the vertical screen and vice versa), *overall trajectory* (overall physical distance user's finger traveled on both screens), *vertical trajectory length*

(physical distance user's finger traveled on the vertical screen), *horizontal trajectory length* (the physical distance user's finger traveled on the horizontal screen), all of them were logarithmically transformed.

A repeated measures ANOVA was conducted to compare the effect of interaction technique, dragging direction, and dragging angle as well as their interactions on the overall, horizontal and vertical dragging trajectory length, time, and switching time. The significant results are shown in Figure 3.2. For the post-hoc test the student's t-test was used for the dragging direction variable. For the other variables we used the Tukey HSD test.

	df		F	p
Overall time				
Interaction technique	2	276	57.1986	<.0001
Dragging direction	2	276	12.7149	<.0001
Vertical time				
Interaction technique	2	276	10.9089	<.0001
Dragging direction	2	276	146.1459	<.0001
Switching time				
Interaction technique	2	276	33.1058	<.0001
Dragging direction	2	276	15.3929	<.0001
Overall trajectory				
Interaction technique	2	276	65.3526	<.0001
Dragging angle	2	276	29.1385	<.0001
Vertical trajectory length				
Interaction technique	2	276	3.4964	.0316
Dragging angle	2	276	9.5764	<.0001
Dragging direction	2	276	26.3364	<.0001
Horizontal trajectory length				
Interaction technique	2	276	6.7045	.0014
Dragging angle	2	276	9.5764	<.0001
Dragging direction	2	276	12.0905	<.0001

Table 3.2: Significant main effects and interactions for the dependent variables in the Cross Dragging experiment.

ITSS was the slowest technique.

The post-hoc test for the *interaction technique* showed that the overall time using ITSS (4.99 sec) was significantly longer than ITOS (3.183 sec) and DT (2.87 sec). Further-

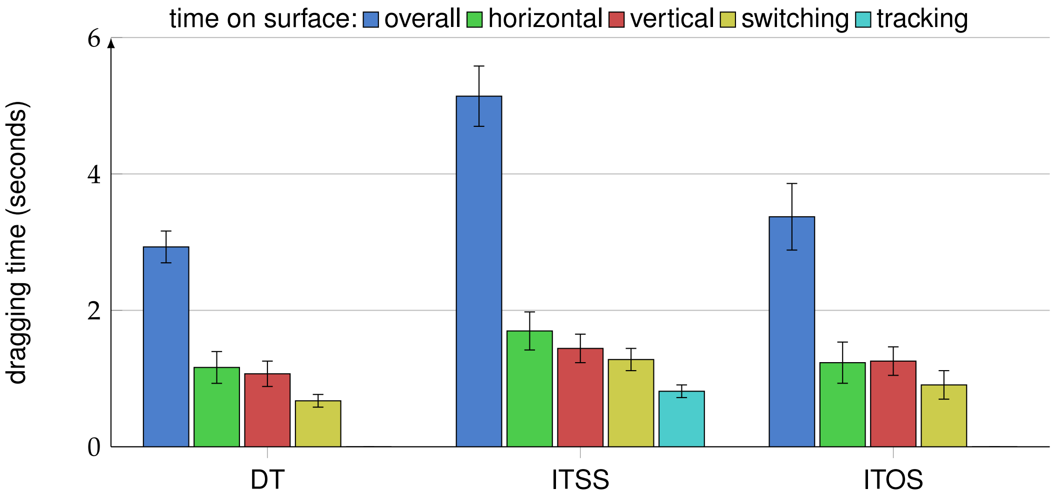


Figure 3.12: User's dragging times for the different subtask in the CrossDragging experiment. Whiskers denote the 95% confidence interval.

more, upwards dragging (3.87 sec) was significantly slower than the downwards dragging (3.29 sec).

The post-hoc test for the *vertical time* revealed the same results as for the overall time. Using ITSS (1.54 sec) was significantly slower than ITOS (1.06 sec) and DT (0.98 sec). Also, dragging upwards on the vertical surface (1.78 sec) took significantly longer than dragging downwards (0.77 sec).

Similarly, for the horizontal dragging time, the post-hoc test revealed for the *interaction technique* the same results as for the overall time. Using ITSS (1.29 sec) was significantly slower than ITOS (1.08 sec) and DT (0.95 sec). Dragging upwards (0.94 sec) on the horizontal surface was significantly faster than dragging downwards (1.53 sec).

Switching using DT (0.59 sec) was significantly faster than ITOS (0.83 sec), which was significantly faster than ITSS (1.32 sec). Switching from the vertical to the horizontal surface (0.74 sec) was faster than switching from horizontal to the vertical one (1.01 sec).

The post-hoc test for the *interaction technique* showed that

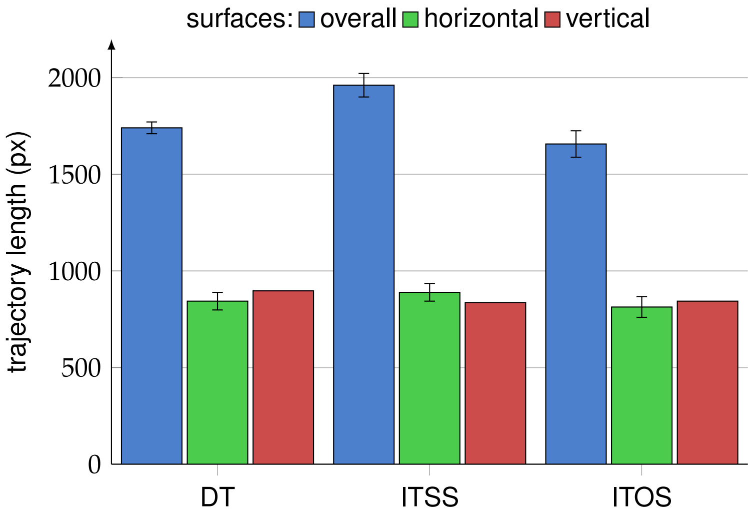


Figure 3.13: User's dragging trajectory length in the Cross-Dragging experiment. Whiskers denote the 95% confidence interval.

ITOS was the most accurate technique.

the *overall length* using ITOS (1568 px) was significantly shorter than for DT (1689 px) and ITSS (1879 px). For the *deltaAngle* factor the post-hoc test showed that *overall length* for 0° angle (1707 px) was significantly shorter than for 15° (1756 px), 30° (1809 px) and 45° (1851 px). The same tendency was shown for the factor *horizontal length*: for 0° angle *horizontal length* (802 px) was significantly shorter than for 15° (831 px), 30° (857 px) and 45° (879 px).

Discussion

While using DT, users could easily regrab the object on the other surface.

As expected, the overall time and time on the vertical surface were longer for the ITSS technique than for the other two (Fig. 3.12). The primary reason is the existence of an additional *Tracking* state in the interaction model. Users spend a lot of time on moving the cursor to the object they want to select, whereas for DT, they could directly physically reach the target or for ITOS just look at the object of interest, which in total requires a much smaller amount of time. However, since no difference between the users' performance between using ITOS and DT was found, H5 was rejected. A not obvious factor is the physical distance the user's arm had to travel in the air while switching between the vertical and horizontal surface.

For DT this distance is fixed and the smallest and equals to the distance between the lower edge of the vertical screen and the upper edge of the horizontal one. For ITOS this distance is not fixed and depends on the strategy the user has chosen to use. As far as the user is not restricted by the touch area, after reaching the border between the horizontal and vertical screen, the distance depends on where the user touched the horizontal surface. Therefore, it equals the distance between the lower edge of the vertical screen and the point on the horizontal surface the user touched, which lays between the higher and lower edges of the horizontal screen.

While using ITSS, the users had to move their arms from the lower part of the horizontal surface to regrab the obejct.

In the case of ITSS, traveling distance is always equal to the maximum—the distance between the lower edge of the vertical screen and the lower edge of the horizontal. Moreover, for both the upward and downward moving direction those distances were the same. For this reason, the switch time as shown in Figure 3.12 is the longest for ITSS and comparably shorter for ITOS and DT. In addition, since in this case DT outperformed ITOS in terms of switching time, H6 also does not hold.

Overall, ITOS is faster and more accurate then the ITSS technique.

Considering the overall time duration on the horizontal screen, it was the longest for ITSS in comparison to the other two techniques. The horizontal time for the three techniques for the upward direction is the same; because they repeat the same sequence of actions, the most influential part lays on the downward direction. As mentioned above, for the DT technique the physical movement distance in the air is static for both upwards and downwards directions. However, for the ITOS technique, users could overcome the border between the screens without regrabbing an object and move it for some time on the horizontal screen without reaching the target. Therefore, the time needed for the movement on the horizontal screen was comparably lower than for ITSS, where users always had to move an arm from the lower edge to the upper edge of the horizontal screen.

Users tend to prefer direct touch over indirect touch.

H7 was confirmed by our results and it seems that the dragging trajectories are longer for more diagonal dragging operations. These results show, that this effect is not only true on curved surfaces, as shown in the previous chapter, but also for systems that combine horizontal and vertical surfaces that are not connected by a curved surface. Interestingly, this effect is also true when the users are not directly interacting with the vertical surface. If we look at the results for the horizontal and vertical dragging trajectories, this effect is only visible for the horizontal surface. This leads to the assumption that users try to minimize their movement on the vertical surface even when they are not directly interacting with it. In general, these results indicate that even if interaction using indirect touch and direct touch are executed on the same surface, user tend to prefer direct touch over indirect touch in basic operations.

3.6 Conclusion

In the previous chapter we found out that directly interacting vertical surface is not only cumbersome but also inaccurate. In this chapter, we address this issue by bringing the indirect touch concept into the interactive desk workspace (C 2.1). Using indirect touch users can comfortably interact with the vertical surface by creating the touch input on the horizontal surfaces. As explained above, using indirect touch introduces two main drawbacks:

1. Users cannot aim for a specific object on the vertical surface without being afraid of accidentally manipulating another object; and
2. the horizontal surface is reduced to only an input device and cannot be used to display digital information.

Lift-and-tap is the most suitable state switching method.

The first drawback is due to the fact that touch input lacks a tracking state that allows to aim at a specific target. In the first part of this chapter, we address this problem by adding this tracking state and identifying four possible interaction techniques that allow the users to switch between a tracking and an engaged state for each individual finger (C 2.2). We showed that our *lift-and-tap* technique was not only the most preferred interaction technique of the users but also the most suitable technique with the lowest error rate.

One of the limitations of this study is that we tested two pressure based switching techniques and were only able to indirectly measure pressure by measuring the contact size of the finger, as suggested by Benko et al. [2006]. However, repeating the study with a modern touch devices such as the latest iPhone[2], which are able to detect pressure directly, could reveal other results.

ITOS is a suitable interaction technique to combine direct and indirect touch.

Using indirect touch and the user's gaze seems to be a suitable interaction concept for interactive desk workspaces.

The second drawback of indirect touch input is that the horizontal surfaces are only an input device and cannot be used to display digital information. We address this issue by creating an interactive desk workspace in which the users use their gaze to choose whether they want to directly interact with the horizontal surface or indirectly with the vertical surface (C 2.3). We propose two novel gaze-based interaction techniques, namely ITSS and ITOS, for easy touch interaction for interactive workspaces. With the help of these gaze supported interaction techniques, we showed that it is possible to enrich the interaction with interactive workspaces as first envisioned by Tognazzini's Starfire [Tognazzini, 1992] concept. By introducing gaze as an additional modality, we are able to reduce the time that is needed to reach targets on the vertical screen as well as reduce effort that is needed to interact with the system. This enables users to comfortably interact with interactive workspaces for a longer time (e.g. a full working day). Nevertheless, further studies are needed to investigate long-term effects.

---

[2] www.apple.com

Chapter 4

Bringing Haptics to Interactive Desk Workspaces

In the previous chapters, we presented two possible solutions that are able to overcome the ergonomic challenges of large interactive surfaces in desk workspaces as described in the introduction of this thesis. In this chapter, we will address the second issue of large touch-based surfaces: their limited haptic feedback.

As mentioned before, touch-screens have a lot of advantages over classical input devices such as mouse and key-

---

Publications: The work in this chapter is a collaboration with Kjell Ivar Øvergård, Christian Thoresen, Chat Wacharamanotham, Kosuke Nakajima, Jan Thar and Christian Cherek. The author is the main author of most of the papers; he was also responsible for developing parts of the hardware, writing parts of the software, designing the experiments, and analyzing data from the experiments. Part of this work was first published as a short paper at the ITS 2013 conference [Voelker et al., 2013c], as a paper at the IPSJ Interaction conference 2014 Nakajima et al. [2014], as demos at the ITS conferences 2013 [Voelker et al., 2013a], 2015 [Cherek et al., 2015] and at the UIST conference, 2013 [Voelker et al., 2013b], as a paper at the UIST 2015 conference [Voelker et al., 2015a], and, finally, as a short paper at the ITS 2015 conference [Voelker et al., 2015c]. Several sections of this chapter are taken from these publications. Furthermore, parts of this work were also published as master thesis from Linden [2015] who developed the basic software framework and Thar [2015] who developed the electronics of PERC tangibles.

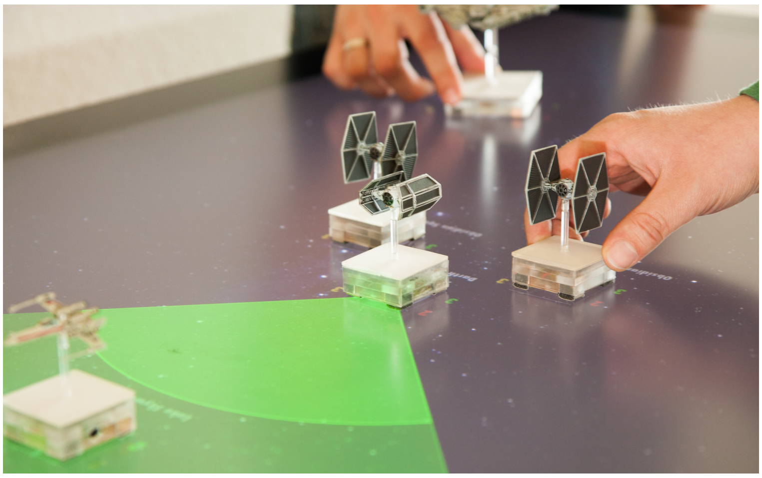


Figure 4.1: Tangibles are physical objects that allow users to interact with the digital content using physical objects on top of the screen. In this example, the user can directly interact with the tangible space ships, and the screen displays digital information according to the position of the tangibles on the touch-screen [Voelker et al., 2015a].

Touch-screens only provide very limited haptic feedback.

board. As also mentioned in the introduction, one of their main drawbacks is the lack of haptic feedback. Users cannot feel the shape of on-screen objects, and they do not receive tactile feedback when triggering actions. This leads to the problem that users cannot rely on tactile feedback and they always have to visually confirm whether their fingers are interacting with the correct digital object or not. For small mobile devices, where touch is the dominate input method, this is not a big problem since the devices are usually that small that the user can always see the entire interface. However, for larger systems, where input and output is not always displayed at the same position or even on the same surface, users have to constantly switch between their focus between the area where the digital content is displayed and the area where they create their input [Weiss et al., 2009].

One solution for bringing back tactile feedback to interactive surfaces is to use tangibles on top of these surfaces [Weiss, 2012]. As shown in Figure 4.1, tangibles are physical objects that allow the users to interact with the digital content using physical objects. It has been shown that they are useful in a large variety of application scenarios [Terrenghi et al., 2007], from music creation [Jordà et al., 2007], to collaborative search [Jetter et al., 2011], and to medical teaching simulations [Zadow et al., 2013]. Most of these tangibles have been designed for older vision-based multi-touch systems [Weiss et al., 2009; Jordà et al., 2007]. These visual systems are often bulky and sensitive to external light conditions [Schöning et al., 2010].

Tangibles are physical objects that provide haptic feedback.

Modern capacitive touch displays are more suitable in a workspace. They are usually smaller and are not affected by lighting conditions. These displays detect touches by creating an electric field above their surface. When an object with high capacitance, such as a human finger, comes close to the surface, this electric field changes. The touchscreen measures this change and reports a touch. However, detecting tangibles on these displays is complicated. Existing tangibles, such as Capstones [Chan et al., 2012] or TUIC [Yu et al., 2011], normally use electrically conductive material on their bottom and sides, in that way that a user touching them increases their capacitance enough to register it as a touch. However, this means that for the tangible to be detected, the user has to continue touching it. As soon as the user releases the tangible, the capacitance drops, and the system fails to detect the tangible—even if it remains on the surface. This leads to the problem that the display cannot reliably decide if the tangibles were removed from the surface or if a user stopped touching it.

Capacitive touch-screens can only detect tangibles while they are touched by a user.

In this chapter, we will give an overview of existing tangible detection approaches and, furthermore, their problems. Afterwards, we will present PERC tangibles, our solution for the tangible detection problem on capacitive touch displays (C 4.1). In a large-scale technical evaluation we are going to show that these PERC tangibles can be reliably detected on a number of unmodified, commercially available capacitive touch-screens (C 4.2). Finally, in addition to the technical evaluation, we will present a user study that demonstrates that PERC tangibles do not only provide haptic feedback but that they also can improve the user's performance in some tasks (C 4.3).

4.0.1 Tangible Detection Methods

Over the last 20 years several methods have been proposed which allow detecting tangibles on interactive surfaces. To provide a better overview of the benefits and drawbacks of these tangible detection methods, we defined a set of basic requirements that we think should be fulfilled by a system in order to allow the user a seamless tangible interaction:

1. At any time, the system has to be able to determine which tangibles are currently placed on the interactive surface, whether they are being touched or not.
2. Each tangible has to be uniquely identifiable.
3. The system needs to be able to detect the exact position and orientation of each tangible.
4. Position and orientation updates of fast-moving tangibles should be detected without noticeable delays.

In the following, we describe the existing literature about detecting tangibles on interactive surfaces and we will evaluate them according to these requirements.

Most tangible systems use vision-based tracking methods.

One of the popular approaches to detect tangibles on interactive surfaces is to employ vision-based tracking using cameras above or below the surface. As one of the first tangible systems, URP [Underkoffler, Ishii, 1999] used a camera above the surface to detects a specific dot pattern on top of each tangible. In a more modern approach SLAP [Weiss et al., 2009] (Fig. 4.2 uses diffuse illumination (DI) [Matsushita, Rekimoto, 1997] in order to detect reflective markers attached to the bottom of each tangible. In this system, LEDs below the table are emitting infrared light which are reflected by markers attached to the tangibles. This reflected light is captured by an infrared camera below the surface. Similarly, ReacTable [Jordà et al., 2007] and Bullseye [Klokmose et al., 2014] detect tangibles using fiducial markers. All of these systems – except for the URP system – fulfil all four requirements. In the URP system, the system cannot detect whether the tangible is placed on

Figure 4.2: SLAP knobs allow controlling digital contents with physical objects. In this example, the SLAP knob is used to change properties of an image displayed on the touch-screen. Image courtesy of Weiss [2012].

the surface or hovers above the surface. However, vision-based interactive surfaces suffer from impaired reliability under many lighting conditions, and are mostly rather voluminous[Schöning et al., 2010].

Because of this, several projects have explored alternative tracking technologies: Audiopad [Patten et al., 2002] attached two radio frequency tags to each tangible to determine its position and orientation. Bricks [Fitzmaurice et al., 1995] uses an existing input device as a tangible. Sensetable [Patten et al., 2001] uses electromagnetic sensing to track tangibles. All of these systems fulfil the first two requirements, but not requirements 3 and 4, since they cannot detect the exact position and orientation of tangibles. Also Bricks and Sensetable are limited in the number of tangibles that can be detected at the same time. Gausstones [Liang et al., 2014] track magnetic tangibles using a hall sensor grid below the touch-screen. Since the small magnetic tangibles can only be detected over a very short range, this technique only works in combination with thin touch-screens.

Often additional hardware is used to track the tangibles.

Figure 4.3: Capstone blocks on an iPad allow stacking but are only detected if touched by a user. Image courtesy of Chan et al. [2012]

Capacitive touch-screens can only detect tangibles while they are touched by a user.

Tangibles on capacitive screens can usually only be detected while the user is touching them. SmartSkin [Rekimoto, 2002] showed how tangibles can be tracked on custom-made capacitive touch-screens, by simulating touch-points that mimic a human touch. However, these touch-points can only be simulated while a user is touching the tangible. CapWidgets [Kratz et al., 2011] applied this concept to commercially available capacitive touch-screens such as the Apple iPad. Capstones [Chan et al., 2012] (Fig. 4.3) extended this concept by allowing the user to stack tangibles onto each other. To distinguish a large number of tangibles, Yu et al. [2011] created active tangibles that can be uniquely identified by enabling and disabling the touch-points with a specific time-based pattern. With this approach, tangibles are identified by their own unique marker-frequency. All these capacitive systems violate requirement 1, since the system cannot tell if a tangible is still present on the screen or not, when the user stops touching it.

Therefor, our goal is to overcome these problems by developing tangibles that can be detected by a modern capacitive touch-screen while not being touched by a user.

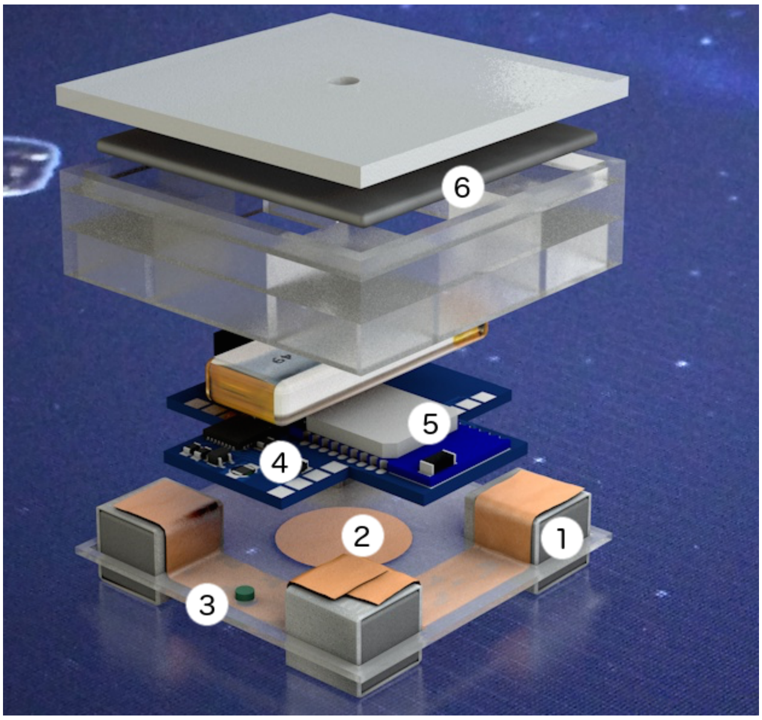


Figure 4.4: The six main components of a PERC tangible: (1) marker pattern, (2) field sensor, (3) light sensor, (4) micro controller, (5) Bluetooth element, and (6) lead plate.

4.1 PERCs: Persistently Trackable Tangibles

PERC tangibles have three main components: marker, field sensor, and light sensor.

Persistently Trackable Tangibles on capacitive multi-touch displays (PERCs) comply with all four requirements mentioned above. They archive this by combining three main components: A passive *marker pattern* that is detected by the capacitive touch-screen and is used to determine the position and orientation of the tangible on the display. A *field sensor* that is capable of detecting if the tangible is placed on a capacitive touch display, even when the marker pattern is not detected. And a*light sensor* that can detect the color of the display below the tangible. Figure 4.4 shows the main components of a basic PERC tangible.

In addition to these three main components each PERC tangibles also includes a micro controller, a Bluetooth 4.0 chip, a battery, and a lead plate on top of the tangible to increase the tangible's weight for a better touch detection.

In the following sections we will explain how capacitive touch-screen detect touches, followed by the description of the components of a PERC tangible.

4.1.1 Capacitive Touch Tracking

Capacitive touch displays sense the presence of a grounded electrical conductor, typically a human finger, in close proximity to the screen, using transparent electrodes located above the display panel. Barrett, Omote [2010] distinguishes two main sensing techniques, self capacitance and mutual capacitance. However, most modern capacitive touch-screens use mostly mutual capacitance, therefore, we will only focus on the mutual capacitance.

Mutual Capacitance

Capacitive touch-screens consist of a grid of electrodes on top of a LCD.

The electrode configuration of a mutual capacitance display consists of a set of rows and a set of columns (Fig 4.5). One set acts as transmitters (Tx) and the other as receivers (Rx) [Rekimoto, 2002]. When a signal is applied to one of the Tx electrodes, the capacitance between this Tx electrode and an intersecting Rx electrode couples the signal to the Rx electrode [Silicon Labs, 2011]. By measuring the signal from each of the Rx electrodes, the touch controller determines the capacitance between the active Tx electrode and each of the Rx electrodes. The controller activates one Tx electrode at a time and measures the capacitance for each Rx. Using this time multiplexing approach, the controller is able to measure this capacitance at all the $Tx - Rx$ electrode intersections on the display.

When a grounded conductor like a finger gets close to one of these $Tx - Rx$ electrode intersections, capacitance between the two electrodes is reduced as the electric field between them is disturbed by the conductor [Zimmerman et

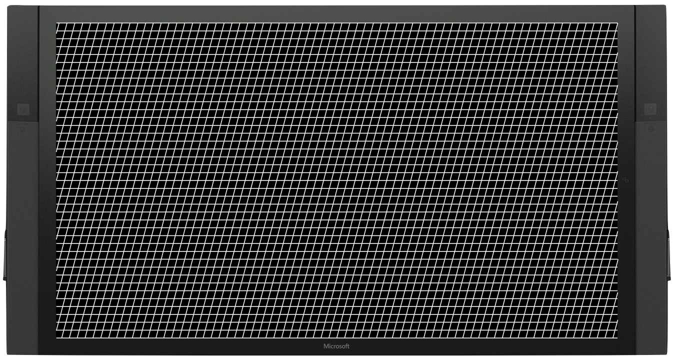


Figure 4.5: The capacitive sensing grid of a Microsoft Surface Hub consists of transmitter electrodes (rows) that are used to emit a signal and receiver electrodes (columns) that are used to measure the change of this signal. By time multiplexing the screen can measure the capacitance for each intersection individually.

al., 1995]. With a typical electrode pitch between 3.5 - 5 mm, a finger touching the display will affect more than one intersection. Using interpolation, the controller is able to accurately determine the center of the touched area and reports this as a touch event. Since controllers are designed to detect finger touches, they search for elliptical shapes about the size of a fingertip. Other touch shapes and sizes are either ignored or may cause unpredictable touch events to be reported.

In summary, to make the controller report a touch event,

1. the $Tx - Rx$ electrode capacitance needs to be reduced below a certain threshold, and
2. this needs to happen over an elliptic area about the size of a fingertip.

4.1.2 Passive Marker Pattern

The marker pattern is used to detect the position and orientation of the tangible.

The marker pattern is detected by the capacitive touch-screen as a set of touch points and is used to determine the position and orientation of the tangible. A marker pattern consists of two or more conductive pads which are detected as touch points. These pads are electrically connected by a conductive material (e.g. copper foil or cables).

A tangible needs to be grounded in order to be detected.

To be detected by the display, each pad has to conform the technical two requirements mentioned in the previews section. The first requirement can be fulfilled by grounding the marker pattern. The second requirement can be fulfilled by shaping the marker pad as a round pad of a particular size. However, the size of the pad depends on the electrode grid resolution of the display. We will explain this in detail in the next section.

The straightforward way to ground a pad is to use the body capacitance of a user as proposed by Rekimoto [2002]. This requires that a user touches the tangible, that the pads of the widget are conductive, and that they are electrically connected to the part of the tangible touched by the user. In this case, the widget simply functions as an electrical conductor between user and touch-surface. In this approach, the display cannot detect the tangible anymore, as soon as the user lets go of it.

One approach to replace the user as electrical ground is to use a conductive wire that permanently connects the widget to a relatively grounded object, for example, the battery ground connector of a tablet computer. However, permanently wired widgets are not a very practical setup for experiments, user studies or interaction design prototypes.

PERC tangibles ground themselfs to capacitive coupling to the capacitive display.

Therefore, the marker pattern of a PERC tangible uses a different technique that allows them to be detected without the need to be grounded or touched by the user. They utilize the capacitive coupling to a second area on the display as a ground. Through several pads on each marker pattern that are electrically connected to each other, currently active intersections on the touch-screen are coupled to each

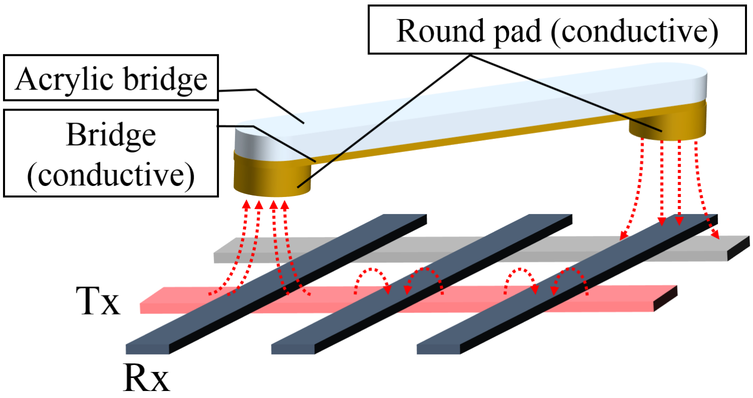


Figure 4.6: The basic concept of a Bridge marker. Red connections indicate capacitive coupling between marker and electrodes. While the red Tx electrode is active the gray Tx electrode is connected to the ground of the display. The pad on top of the gray Tx electrode creates a capacitive coupling to the ground Tx electrode and therefore grounds the entire tangible [Voelker et al., 2013c].

other, currently inactive intersections serve as a ground.

The simplest example of this principle is a “Bridge” PERC marker pattern that creates two touch points (Fig. 4.6). Its marker consists of two round pads that are used to achieve the first technical requirement. The pads are connected to each other using a conductive material. When a Tx electrode under one pad is active and the Tx electrodes under the other pad are inactive (at ground level), then this second pad has a capacitive coupling to the ground. This ground coupling is sufficient enough to reduce the $Tx - Rx$ intersection capacitance under the first pad below the threshold for touch detection. Similarly, when the Tx electrodes are active under the second pad (when the touch-screen scanning algorithm reaches that area), the Tx electrodes under the first pad are no longer active, and thus couple to the ground. This lets the Bridge PUC generates one touch event for each of the two pads, without the aid of external grounding.

PERC tangibles ground themselfs through inactive parts of the display.

Yet, if both pads are aligned with the Tx electrodes, both will couple to the same Tx electrode, and the marker will

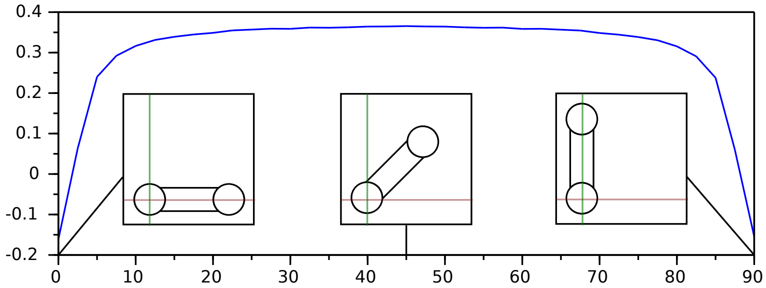


Figure 4.7: Simulated decrease in intersection capacitance below a pad for different orientations of a Bridge marker. Scale: 0 is base capacitance, 1 is capacitance for a grounded conductor in contact with adisplay [Voelker et al., 2013c].

If both pads are aligned with the same electrode, the tangible cannot be grounded.

no longer have a sufficient coupling to the ground to bring the $Tx - Rx$ capacitance down. Likewise, if both pads are aligned with the Rx electrodes, the Bridge provides an additional coupling from Tx to Rx. This time the coupling goes through the Bridge from the active Tx electrode to a second point on the Rx electrode. In both cases, the $Tx - Rx$ capacitance at each pad will not drop below the detection threshold and may, in fact, increase, as it does for a single unconnected pad. Since electrodes in touch-screens are mostly laid out in a horizontal-vertical grid, in practice, a two-pad marker like the Bridge PERC marker pattern will disappear when its position on the screen is horizontally or vertically aligned.

To support our approach, we modelled the capacitances between two crossing electrodes in a touch-screen and between electrodes and conductive pads contacting or hovering above the screen. This was done using 2D electrostatics models in the FEMM[1] software tool for finite element method simulation of electromagnetics. The resulting capacitances were used to calculate intersection capacitances for different marker geometries as seen by the touch-screen controller.

Figure 4.7 shows how the simulated decrease in capacitance below one end of a Bridge marker is changing as the marker is rotated. This corresponds well with our observation that

---

[1] www.femm.info

Figure 4.8: With the three pads we ensure that at least one area of the marker is always capacitively coupled with several inactive intersections, independent of its orientation.

the marker was undetectable when aligned with either set of electrodes, as the graph shows an increase in capacitance above the base level for this condition.

To address the horizontal and vertical alignment problem we created a three pad marker (shown in Figure 4.8 that most of the time can be detected continuously independent of its orientation. This marker consists of three pads that are connected with conductive material that hovers very closely above the display surface. Since the electrical field reaches out of the display, this material also creates a capacitive coupling with the intersections under it. This setup ensures that at least one area of the marker is always capacitively coupled with several inactive intersections, independent of its orientation. The hovering material does not fulfil the area requirement, so it does not create false touch points.

A three marker pattern solves the alignment problem.

In addition to these alignment problem, these marker pattern suffer from a filtering problem. Many capacitive touch

Filtering mechanism eventually remove stationary tangibles.

systems adapt their filtering algorithms to changing electrical background noise over time. Since the marker pattern pushes the limits of touch detection on these systems, they are likely to fall under this adaptation. This problem, fortunately, only occurs, if the tangible is stationary for at least several seconds. As soon as a tangible is moved across the surface, all touch-points are immediately detected again. However, this problem prohibiting us from determining if a stationary tangible has been removed from the touch-screen or if its touches have just been filtered out. For our PERC tangibles, we solve these problems by adding a field sensor to each tangible that detects if it is placed on a capacitive touch-surface or not, even if the marker pattern is filtered out by the capacitive touch display.

4.1.3 Field Sensor

The field sensor detects the probing signal of a capacitive touch-screen.

The field sensor is the part of a PERC tangible that actively determines if the tangible is placed on a touch-surface at any given time or not. For this purpose, an antenna at the bottom of the tangible picks up the signature of the electric field above the surface, which is created by every capacitive touch-screen. We have measured the fields of several commercially available devices (Figure 4.9) and found that all signals exhibit a regular pattern of strong peaks at a fixed frequency, which can be easily distinguished from the noise component of the signal. The field sensor consists of this antenna, additionally, of a comparator integrated circuit that detects peaks above a certain voltage threshold, and a micro-controller to trigger a timeout if the next peak was not detected within a specific period of time, determined by the touch-screen's pulse frequency. In our current implementation, the threshold of the field sensor is set to detect the capacitive touch-screen if the distance between the touch-surface and the tangible falls below 1 mm. However, other configurations that allow the tangible to detect the signal of a capacitive screen from a different distance is possible (e. g. 20 mm above the screen).

Whenever the field sensor detects the presence of a capacitive touch-surface, the tangible sends a *set* event via Blue-

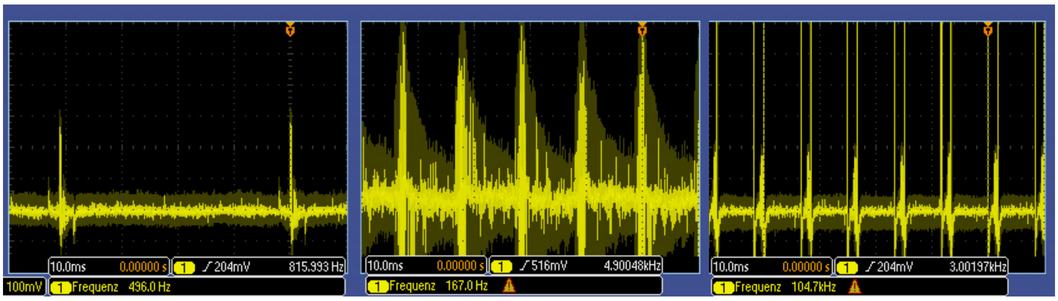


Figure 4.9: Touch detection signals by (left to right) iPad 4, 3M screen, and Microsoft 55″ capacitive screen [Voelker et al., 2015a].

tooth 4.0 (BLE) to the system. The system correlates this message temporally to newly appearing touches in order to link the UUID to the tangible's position. As we will show in the tangible evaluation section, the *set* event and the touches arrive at the system within 144 ms in over 99% of the cases. This allows us to set the time window for grouping UUID and location to 150 ms, which eliminates most false negatives. False positive groupings that could result from multiple PERCs being set down on the table within this time window are resolved using a light sensor, as described below. Until the tangible recognizes the absence of the electric field and sends a corresponding *lift* event, the system considers it as being on the table even if its touches are filtered out. As soon as the timeout mechanism of the field sensor is triggered, the PERC tangible sends a *lift* event via BLE to the system to indicate that the tangible was removed from the table. If the tangibles' touches disappear between these two events, the system ignores this change, because the touches disappearing may have been caused by the filtering algorithm.

As soon as the field sensor detects the presence of a capacitive touch-surface, the tangible sends a *set* event via BLE to the system.

The field sensor is a very simple circuit that theoretically could be trigged by other electrical devices that emit a signal with strong peaks at a similar frequency as a touchscreen. However, this approach is relatively robust against stray electric fields for two reasons: (1) A tangible is only detected if the *set* event from the field sensor and the touch points of the marker pattern occur in a short time frame. (2) Since electric fields are strongly attenuated over distances, it would require a very strong electric field to trigger

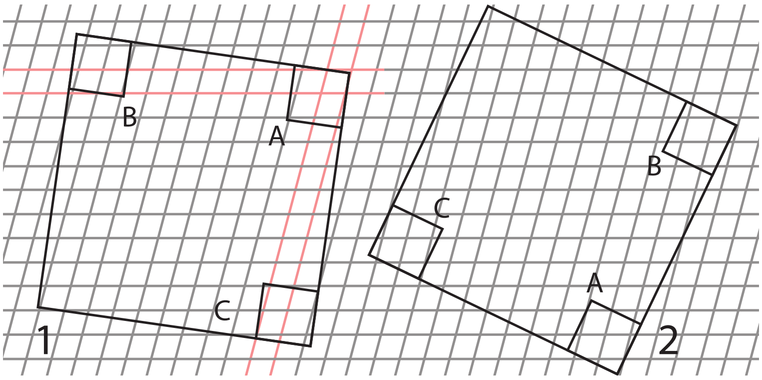


Figure 4.10: PERCs tangibles on the transmitter and receiver electrodes of the Microsoft 55″ capacitive screen. For (1) marker A is not detectable due to the alignment of the electrodes. In (2) all markers are detected reliably. [Voelker et al., 2015a]

the field sensor from a distance. In short, electric fields of this strength usually do not exist in an environment where touch-screens are used.

At specific orientations of the tangibles only two markers are detected.

While the combination of our marker pattern and field sensor lets the system reliably detect which tangibles are on the surface, we found that if a PERC tangible is placed on the touch-surface at certain angles, only two of the three marker pads are detected. The reason for this is a combination of the marker pattern and the arrangement of the transmitter as well as the receiver electrodes of the capacitive surface. Whenever a tangible is oriented as shown in Figure 4.10, the corner pad A is located at the crossing of two electrodes that are also covered by the outer pads B and C. In this case, the capacitive touch-surface will not detect a touch at pad A. Because pads B and C are still detected reliably, the system is able to determine the location and the angle of the tangible, but only modulo a 180° orientation ambiguity. Similar to the situation with multiple tangibles being placed on the capacitive touch-surface at the same time, these ambiguities can be resolved using the PERCs' light sensor.

The exact angles at which pad A is not detected depend on the geometric configuration of the electrodes in the touch-surface. On many common capacitive touch-screens the electrodes are aligned orthogonally to each other. Therefore, for our marker setup, shown in Figure 4.10 angles around full 90° rotations are critical. On other devices, such as our main test-screen, one set of the electrodes is rotated by 15°, so the critical angles for our marker setup are shifted according to this amount (see Fig. 4.12).

4.1.4 Light Sensor

The light sensor is responsible for resolving the two ambiguities—UUID assignment for multiple PERCs that are set on the surface simultaneously and when only two of the three touches of a PERC are registered by the touch-surface—that can occur during normal operation. The sensor is mounted underneath the tangible, offset to one side from the diagonal line between the pads B and C.

The light sensor detects color changes below the tangible.

The off-axis position ensures two different possible locations of the light sensor when both the position and the angle of the tangible are known but the orientation is unknown. Whenever the system receives a *set* event accompanied by only two touches that have the expected distance between two touch points as B and C, the system sends a "visual ping" to one of these possible light sensor locations by locally changing the brightness of the display for a moment. This approach is similar to Touchbugs [Nowacka et al., 2013]. If this change in brightness is detected by the light sensor, the tangible communicates this via BLE to the system. Consequently, if the light sensor does not detect the visual ping, it must be located on the other side of the diagonal between the pads B and C. Either way, the system can recover the orientation of the tangible.

The light sensor is only used if only two markers are detected.

Note that this process is only necessary immediately after setting a tangible down on the capacitive touch-surface at one of the critical angles, where pad A is not detected. As soon as the tangible is moved, all three pads are detected dependably, and the exact orientation of the tangible can be determined without the overhead of additional communication to the tangible. In the cases where this disambiguation procedure is necessary, the whole process increases detection duration by roughly 100 ms, as we have shown in the evaluation section (see Fig. 4.13).

Apart from resolving the orientational ambiguity, the light sensor also serves to tell apart multiple tangibles if they where placed on the capacitive touch-surface within the 150 ms time frame between receiving the *set* event via BLE and detecting the touches of the tangible. In this case, a sequence of visual pings is sent—one ping to the location of the light sensor of each tangible in question—and the sequence of BLE answers resolve the UUID assignment ambiguities.

4.1.5 Components and Power Consumption

Our PERC tangibles are built from cheap, off-the-shelf components. For the current implementation, we used an MSP430G2553 micro-controller, a BLE112 Bluetooth module, a TEMD6200FX01 light sensor, and a Renata LIPO battery (3.7 V, 175 mAh). The total cost of all parts including the acrylic frame and the marker pads is less than US$25.

The tangibles operate in BLE-master-/slave-mode; the limiting factor for the maximum number of tangibles is the number of touches that can be detected at the same time, rather than the maximal Bluetooth connection. We successfully connected all our 27 prototype tangibles to the PC-based system and 10 tangibles to an iPad.

PERC tangibles hold up to 60 hours with one battery charge.

PERCs have a very low energy consumption: one battery charge yields approximately 60 hours of continuous use. Making the tangibles issue a recharging-warning via bluetooth would be a straightforward extension to our current prototype. Similarly, the recharging mechanism could easily be changed from cable-based to inductive power transfer, allowing the tangibles to recharge in their storage tray or box. The energy consumption could be further improved by reducing the communication frequency or the use of the light sensor.

4.2 Technical Evaluation

In this section we evaluate our PERC tangible design by conducting a set of technical experiments. In the first experiment we determine the minimal size of the marker pattern that is still detectable on a capacitive display by determining the minimal marker pad size and the minimal distance between the pads. After finding the minimal marker size, we performed a series of automated experiments in order to technically evaluate the capabilities of the PERCs tangibles and the degree to which they meet the four requirements identified above. In these experiments we measured: detection, detection time, position offset, and angular offset.

4.2.1 Pad Size and Distance

In this experiment, we varied pad sizes and distances between the pads.We tested the detection rate on six different multi-touch devices (iPad 1 & 3, iPhone 4 & 4S, and Perceptive Pixel 27", Microsoft 55"). Our first informal observations indicate that square pads are better detected than round pads. Thus, we used square pads for our experiment.

We tested the size and the distances between the pads.

As we tested the pad size, we tested pads from 4x4 mm to 10x10 mm in steps of 2 mm. Considering the distance, we tested distances between 10 mm to 50 mm in steps of 10 mm. The pads were connected with conductive copper foil that hovered 1 mm above the display surface. Each marker was placed onto the display ten times, and we counted how often the display was able to detect all pads for at least 5 seconds without the user touching the marker.

The marker pattern had a detection rate of 100% for pads of 8–10 mm with a minimal distance of 20 mm between the pads. Below 8 mm, detection dropped to 0%. Reducing distance between the pads to 10 mm shifted the detection rate to 50% for pads with a size of 8 mm and above.

From these results we conclude that the minimal marker pattern that is reliably detected on most capacitive touch displays should conform to the following two requirements:

1. The pad should be squared and should have a size of 8 x 8 mm.
2. The distance between the pads should be not shorter than 20 mm.

With these requirements a minimal 3-pad marker has the size of 38 x 38 mm. With a small casing around the marker pattern we choose to use 40 x 40 mm as the minimal size of tangible with a 3-pad marker. Figure 4.4 show the two 40 x 40mm tangibles. Tangible a is designed for a capacitive touch-screen on which the sensing grid is orthogonal (e.g. iPad). Tangible b is designed for a capacitive touch-screen on which the sensing grid is shifted by 15 degrees (e.g. Mircosoft 55″ display).

4.2.2 Techniqual Evaluation

A robot that continuously placed and removed a tangible on one of the tested capacitive displays.

In the second series of experiments, we evaluate the capabilities of the PERCs tangibles, additionally. the degree to which they fulfil the four requirements identified above. For this purpose, we built a robot Figure 4.11 that performed a large number of test cycles on three different capacitive touch-screens: a Microsoft 55″ capacitive touch-screen (MS display), a 27″ Perceptive Pixel display (PPI display), and an iPad 4 (iPad). Each cycle consists of setting down a small PERC tangible (40 mm by 40 mm) at a specified location and angle, waiting for one second, and then lifting up the tangible before changing the angle for the next cycle. This resulted in testing 73 distinct angles at nine different positions on each touch-screen.

For each cycle, we measured and logged the positions and time stamps of all touches reported by the touch-screen as well as the time stamps and event types for all incoming BLE communication. Whenever the tangible was detected,

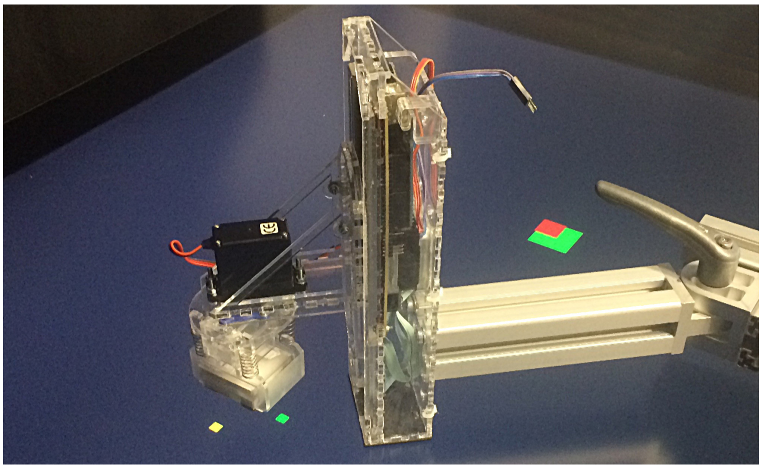


Figure 4.11: The testing robot places and lifts a PERC tangible on the touch-screen. This setup allows to evaluate the detection accuracy of our system by comparing the detect position with the actual position of the tangible.

the system calculated the position and angle, compared both to the expected values for the cycle, and logged the positional and angular detection errors.

The reasons for performing the experiment with a robot are twofold: First, using a robot allowed us to gather a much larger sample size of measurements and granted exact repeatability of each individual placement cycle. Secondly, when setting down the tangible manually, the parasitic capacitance of the experimenter's hand actually results in much more accurate touch locations (even though there is no conductive connection to the pads). Therefore, the experimental setup we used allowed us to give a very conservative "worst case" estimate for the detection accuracy of the system.

For this evaluation, we ran 65700 cycles on the MS display (900 per angle) at nine different positions on the screen. In addition to that, we performed 2190 cycles on both the iPad and the PPI display (30 trials per angle). This adds up to a total amount of over 70000 trials; given an average number of 64 detection reports from the touch-screens over each cycle, we recorded about 4.4 million data points.

4.2.3 Results

The detection rate of the field sensor was at 100 %.

The detection rate of the field sensor was at 100 % through all trials and all touch-screens. The field sensor was always able to detect if a tangible was placed on the surface and if it was lifted from the surface. The average time difference between the *set* event and the *lift* event is 1.3 s with a standard deviation of 0.038 s.

As expected, the detection rate of the marker pattern points depends on the angle of the tangible. As shown in Figure 4.12, around 75°, 165°, 255° and 345°, sometimes only touch points for pad B and C are detected. On the iPad and the PPI we found similar results at 0°, 90°, 180° and 270°. As explained earlier, these angles reflect the alignment of transmitter and receiver electrodes of the capacitive surface.

The average detection duration mirrors the use of the light sensor and the additional communication overhead in these cases. Figure 4.13 shows the detection duration for successfully detected tangibles; cases where the light sensor had to be used are highlighted in red.

Display	Detection time	Detection time with light sensor	Position error	Angle error	Single touch detection
MS 55″	50 ms	190 ms	1.5 mm	-0.78°	2.2%
PPI 27″	65 ms	176 ms	2.1 mm	-1.84°	2.5%
iPad 4	55 ms	167 ms	2.5 mm	-1.94°	3.5%

Table 4.1: Average measurements of the technical evaluation of the PERC tangibles on three modern capacitive touch-screens.

Table 4.1 shows the average detection time when all three markers were detected, the detection time when the light sensor was used, the displacement and angular errors, and the percentage of trials in which only one marker was detected for all three tested touch-screens. More detailed results are shown in Figure 4.14 and Figure 4.15. These results suggest that the displacement and the angular error of a tangible depends on the orientation of the tangible.

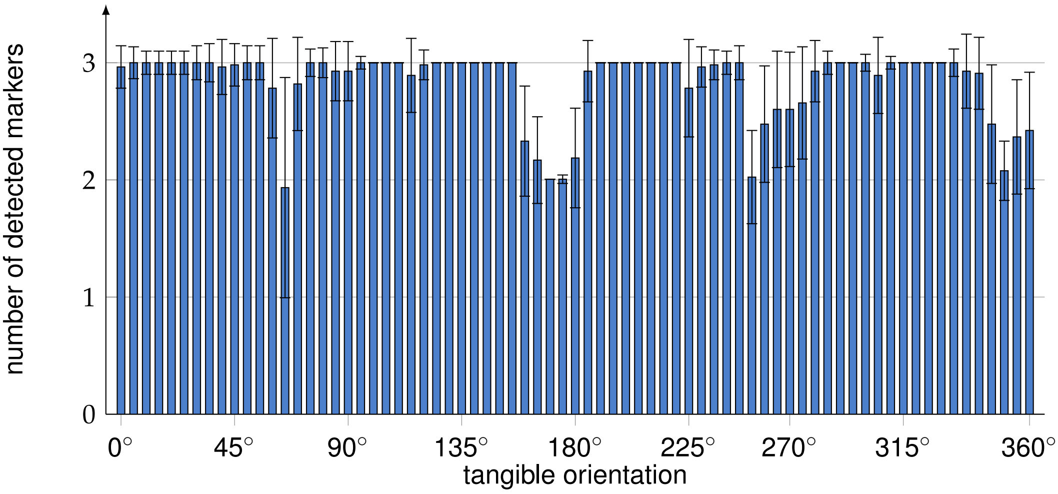


Figure 4.12: Number of marker points found, depending on the orientation of the tangible. The result shows that there are four areas in which only two markers are detected. The whiskers denote the standard deviation. Results were measured on the MS display [Voelker et al., 2015a].

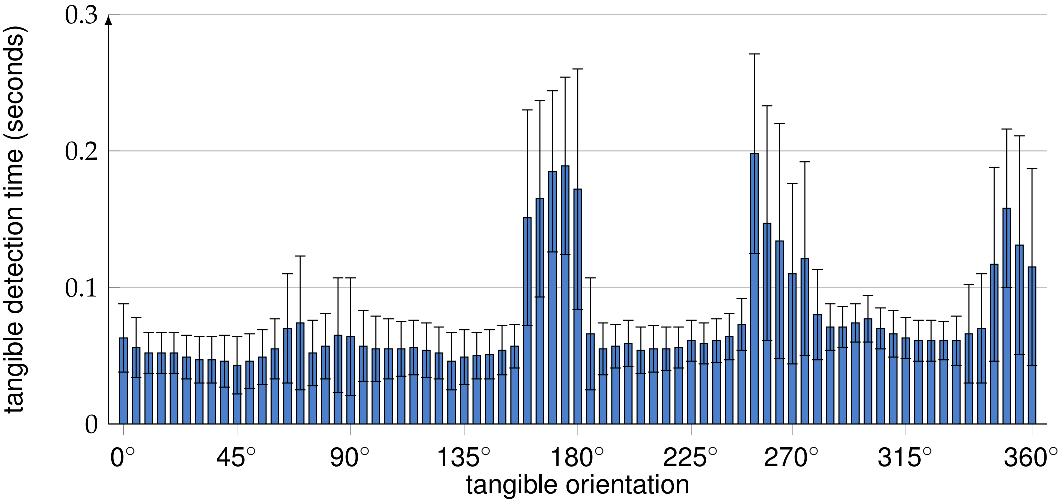


Figure 4.13: Average duration of receiving the information that the tangible is close to a capacitive screen via BLE until the tangible is correctly detected. The whiskers denote the standard deviation. Results were measured on the MS display [Voelker et al., 2015a].

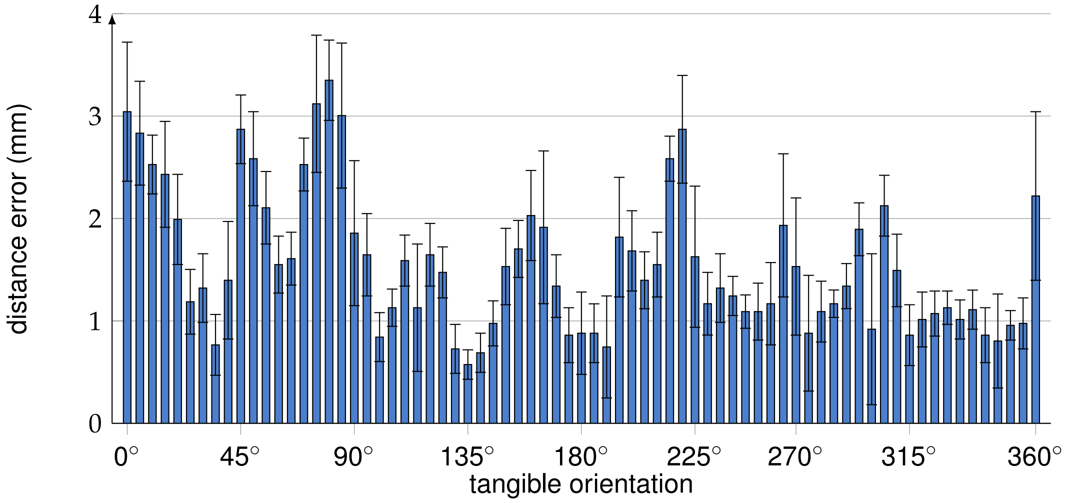


Figure 4.14: Average tangible position error depending on the orientation of the tangible. The whiskers denote the standard deviation. Results were measured on the MS display [Voelker et al., 2015a].

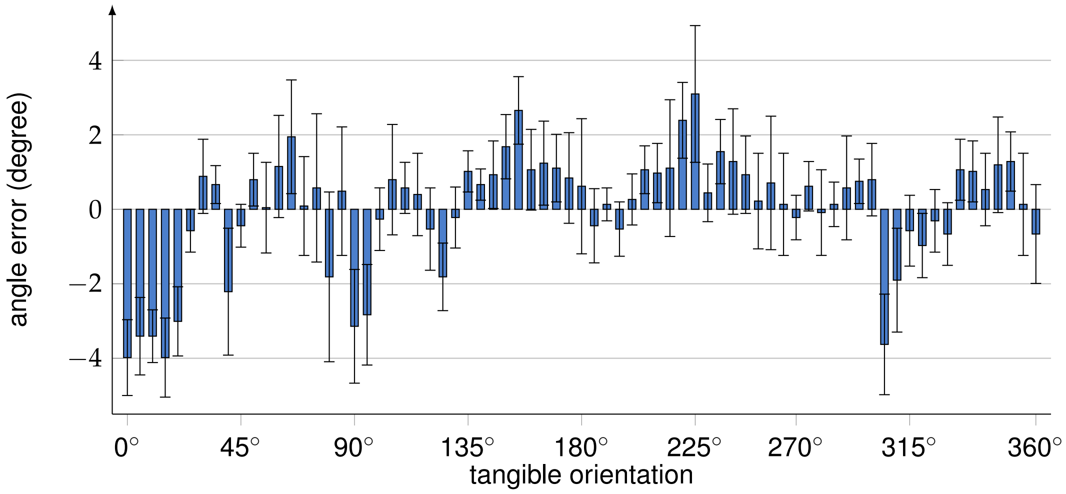


Figure 4.15: Average tangible angular error depending on the orientation of the tangible. The whiskers denote the standard deviation. Results were measured on the MS display [Voelker et al., 2015a].

4.2.4 Discussion

From these experiments, the following observations regarding the four requirements mentioned above can be recorded:

Regarding requirement 1, PERCs can reliably be detected (with 100 % accuracy) if they are located on a capacitive touch-surface. Through the application of the marker pattern concept, about 98 % of the tangibles are correctly detected on a MS display. This detection is independent of a user's touch since we use the the passive marker pattern concept to create the touch points. The field sensor counters the problem of stationary touch points being filtered out over time. Therefore, our system can reliably decide if a tangible was removed from the surface or was just filtered out.

PERCs also fulfil requirement 2, since every tangible has its own Bluetooth UUID. If two *set* events occur at very close temporal proximity, the light sensor acts as a fallback mechanism for disambiguation. Consequently, our system is able to uniquely identify each tangible at any given point in time.

Position and angle of PERCs can be detected with high fidelity: the mean position error on the MS display is 1.5 mm, the mean angular error is -0.78°. Hence, PERCs meet requirement 3. Both, position and angular accuracy can be further improved by employing machine learning algorithms to the collected results.

Requirement 4 is fulfilled as well, since all three of a PERC's touch points are steadily detected as soon as the tangible has moved over the capacitive surface. At this point, positional and angular information are obtained directly from the touches, which are updated with the capacitive touch-surface's scan rate.

4.3 User Evaluation

In the previous sections, we explained the PERC tangible design and conducted a set ofexperiments that showed how dependably these tangibles are detected by modern capacitive touch-screens.

Related works are divided in whether tangibles or touch rotary knobs are better.

In the introduction of this chapter we presented several related works that showed that tangibles provide benefits for the users and also improve their performance for several tasks. However, for rotary knobs, the literature does not provide a clear answer to the question if tangibles are better then touch inputs. The results of a study of SLAP Widgets favor tangible knobs [Weiss et al., 2009], but the results from a study of CapWidgets suggest the opposite [Kratz et al., 2011]. Although the tasks in both studies are similar (using the rotary knobs to navigate a video and mark outstanding frames), several other (possible) factors differ: SLAP Widgets were placed out of the users' sight (eyes-free input), CapWidgets were placed on top of the video (eyes-on input). They used different knob diameters (5 cm vs. 2 cm, respectively), and different tracking technologies (FTIR + DI vs. capacitive). Because of these different factors, it is difficult to identify why these two studies are contradicting each other.

Tangible rotary knobs are one of the most commonly used tangible input device.

Rotary *tangible knobs* are one of the most commonly used tangible input device. They have been proposed for video navigation [Hilliges et al., 2007; Kratz et al., 2011; Weiss et al., 2009], menu selection [Hilliges et al., 2007] as well as interactive data exploration tasks [Jansen et al., 2012; Ma et al., 2015]. Therefore, it is crucial to understand in which situations they provide benefits from touch input and in which situations touch input is better. In the following section, we will answer this question by conducting an experiment that compares both types of rotational input.

We hypothesize that tangible knobs outperform touch-based input in eyes-free tasks.

We hypothesize that the main factor of the contradicting results of both studies was the difference between eyes-free vs eyes-on input. In an eyes-free task, the tangible knob provides haptic feedback that guides the user's input. However, in an eyes-on task the tangible is blocking parts of the interface which makes it complicated for the user to see the target area. To verify this hypothesis, we present the results of an experiment in which we compared two tangible knobs with two virtual knobs in terms of speed and accuracy in eyes-on and eyes-free tasks. To ensure that we only test these two factors, we kept all other factors the same.

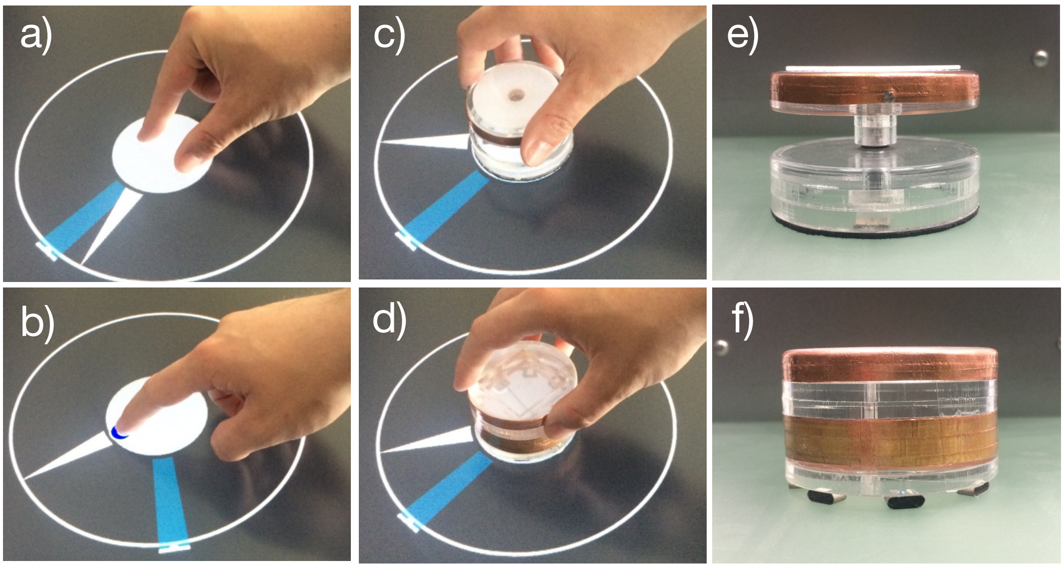


Figure 4.16: Four types of rotary widgets compared: (a) one- and (b) two-touch virtual knobs, (c, e) a tangible knob, of which rotor is independent of the base, and (d, f) a tangible puck: the entire widget is a rotor [Voelker et al., 2015c].

For this purpose, we define two main independent variables: *widget* type and *output* area. All widgets are 7 cm in diameter. All input was done on a horizontal touch-screen. We compared four types of rotary widgets (Figure 4.16):

One-touch virtual knob (a) Similar to an analog telephone dial, this knob can be rotated by dragging a small circle inside the knob to a desired angle.

Two-touch virtual knob (b) This is similar to pinch-to-rotate gesture in touch-screen phones. To rotate this knob, the user touches with two fingers inside the knob area. The orientation of the line connecting the two fingers determines the knob orientation.

Tangible knob (c, e) based on the SLAP knob design [Weiss et al., 2009]. The rotor is attached to the base of the knob by a spindle. This allows rotational input without translating the base of the knob.

Tangible puck (d, f) based on a design in form of a Photohelix [Hilliges et al., 2007]. This puck is a single rigid body. To rotate, the user manipulates the entire tangible, resulting in a coupling between translation and rotation.

For all four widget types, the user had to start each trial inside the output area. While turning touch knobs the users were able to drag outside of the input area, just like typical touch-screen widgets.

We compared three output areas, representing the distance of the tangible from the locus of attention:

Eyes-on: Visual feedback was displayed around the knob, providing single locus of attention. Nevertheless, the hand of the user may occlude the display during rotation.

Eyes-free: Visual feedback was shown on a separate vertical display in front of the user, hence, no hand occlusion. However, the user cannot see the widget while looking at the visual output.

Peripheral: Under this condition, the output was displayed on the horizontal screen with 20 cm offset in front of the user. Although the widgets are outside of the locus of attention, they are still in the peripheral vision.

The size of the visual feedback was 7 cm diameter in all output conditions.

4.3.1 Apparatus

During the experiment participants stand in front of a horizontal 55″ Microsoft capacitive touch-screen. The table surface is 90 cm above the floor. The effective touch frame rate of our setup was 60 Hz.

A 46″ vertical display was placed directly behind the horizontal surface. Both displays had the resolution 1920 x 1080

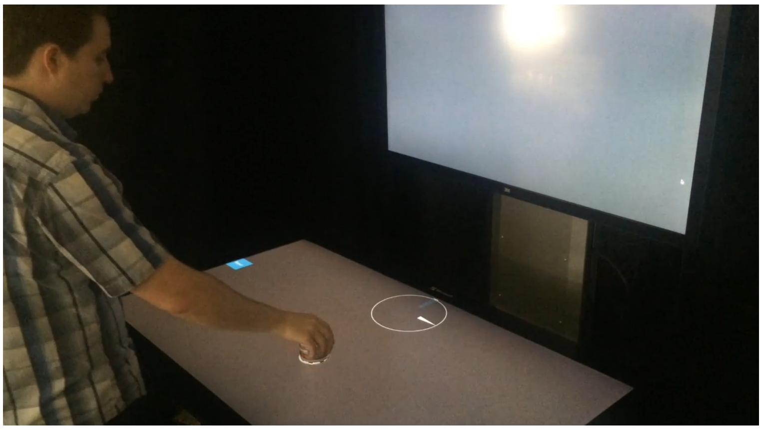


Figure 4.17: The experimental setup of the rotary knob study. In this study the users create rotational input (tangible or touch) to rotate the white indicator to a specific position.

pixels and were connected to a Mac Pro running the software for the experiments. The experiment setup is shown in 4.17.

4.3.2 Task

We used the task from a previous knobology study [Øvergård et al., 2007]. In each condition, the system displays a gray circle (input area), a white ring with a blue target area and a white orientation indicator (visual output).

Users were asked to rotate the input device such that a indicator points to the target position.

The widgets were already placed on the input area prior to each trial. Before each trial, the user turned the widget such that the white orientation indicator was inside a starting area, which is always at the 12:00 position (= 0°). After keeping the indicator one second inside the starting area, a blue static target area was displayed. The participants were asked to turn the knob until the indicator was inside the target area. The target area was randomly displayed at one of six predefined target areas, which are roughly equally-spaced around the knob, ($-150°$, $-100°$, $-50°$, $50°$, $100°$, and $150°$ from the starting area. To complete the trial, the user had to keep the indicator one second inside the target area.

4.3.3 Dependent Variables

Task completion time and number of overshoots were measured.

We measured the *movement time*, that is,the moment the indicator left the starting area until the trial was completed. For data analysis, we deducted the seconds that the users needed to wait inside the target area to complete the trial. For the sake of accuracy, we count the *number of overshoots*, when the cursor exits the target area.

4.3.4 Study Design

A 4 × 3 × 6 × 5 (input type × output area × target area × iteration) repeated measures experiment was used. All factors were within-subject independent factors, leading to a total of 360 trials per participants (on average, 30 minutes per participant). The experiment was divided into four blocks, one for each knob. All iterations in one block were done with the same knob. The order of blocks (hence also input devices) were determined by a Latin square that ensured counterbalancing.

The hypotheses tested in our experiment was as follows:

- H1: We expect that both tangibles will have faster movement times than the two virtual knobs.
- H2: The domination of tangible knobs will be particularly pronounced for the eyes-free condition.
- H3: We expect tangible knobs to have less errors associated with usage (e.g. less overshooting) than virtual knobs.

4.3.5 Participants

20 participants (18 males, 2 females) aged 20 to 36 years ($M = 27$) volunteered for the study. All participants had normal or corrected-to-normal vision. 17 out of 20 participants were right-handed.

4.3.6 Results

Results on iterations and different goal areas were aggregated into mean scores for movement and overshooting. For each multivariate test, we used 4 x 3 (widget × output) GLM repeated measures analysis. [2] Upon significant effects, post-hoc Bonferroni-corrected paired *t*-tests were performed (Familywise error rate $\alpha = .05$). Table 4.2 shows descriptive statistics with 95% CIs (unadjusted). Below, the CIs of mean differences were adjusted for paired design and were Bonferroni-corrected. We provide extensive statistical results in the supplement of this thesis.

Condition	Movement time	Number of overshoots
One-touch	1.47 [1.39, 1.54]	1.06 [0.91, 1.22]
Two-touch	1.83 [1.78, 1.88]	0.58 [0.49, 0.67]
Tangible Knob	1.21 [1.15, 1.28]	0.60 [0.50, 0.69]
Tangible Puck	1.14 [1.10, 1.18]	0.43 [0.37, 0.49]
Eyes-on	1.40 [1.32, 1.48]	0.66 [0.57, 0.75]
Peripheral	1.38 [1.31, 1.45]	0.67 [0.56, 0.78]
Eyes-free	1.45 [1.37, 1.53]	0.67 [0.55, 0.78]

Table 4.2: Overall descriptive statistics by condition (mean [95% CI].

---

[2] When Mauchly's sphericity test was significant, the Greenhouse-Geisser correction was used, resulting in degrees of freedom with decimal points.

Movement Time

We found a very large main effect of widgets on movement time ($F_{1.68,31.91} = 43.81$, $p < .001$, $\eta_p^2 = .70$). As shown in Figure 4.18, the two-touch widget was slower than the one-touch widget ($M_{\text{two}-\text{one}} = 0.36$ s, $CI[0.22, 0.50]$). One-touch widget was again slower than puck and knob ($M_{\text{one}-\text{puck}} = 0.33$ s, $[0.07, 0.59]$; $M_{\text{one}-\text{knob}} = 0.26$ s, $[0.02, 0.49]$). The difference between the tangibles was negligible ($M_{\text{knob}-\text{puck}} = 0.07$ s, $[-0.06, 0.21]$).

A large main effect of the output was also observed on movement time ($F_{1.18,22.50} = 6.26$, $p = .016$, $\eta_p^2 = .25$). The movements under the peripheral condition were faster than under the eyes-free condition ($M_{\text{free}-\text{perip.}} = 0.07$ s, $[0.05, 010]$). Other differences were negligible ($M_{\text{free}-\text{on}} = 0.05$, $[-0.01, 0.11]$), $M_{\text{on}-\text{perip.}} = 0.02$, $[-0.05, 0.10]$)).

A large interaction effect between widget and output was also evident ($F_{2.78,52.86} = 9.05$, $p < .001$, $\eta_p^2 = .32$). Figure 4.18 shows the overview of the interaction effect. In the eyes-free condition (H2), the simple effect of the widgets agreed with the main effect analysis above. Under the eyes-on condition, we found that the tangible puck yielded faster movement time than one-touch widgets ($M_{\text{one}-\text{puck}} = 0.29$ s, $[0.26, 0.54]$) (cf. Kratz et al., 2011). In the one-touch widget, eyes-free output was slower than others ($M_{\text{free}-\text{on}} = 0.22$ s, $[0.10, 0.34]$, $M_{\text{free}-\text{perip.}} = 0.24$ s, $[0.20, 0.28]$).

Number of Overshoots

The results are shown in Figure 4.19. There was a very large main effect of input devices ($F_{1.68,30.78} = 24.17$, $p < .001$, $\eta_p^2 = .560$). The one-finger widget was significantly more error-prone than the others ($M_{\text{one}-\text{two}} = 0.49$ s, $[0.20, 0.77]$; $M_{\text{one}-\text{knob}} = 0.47$ s, $[0.22, 0.47]$; $M_{\text{one}-\text{puck}} = 0.64$ s, $[0.30, 0.97]$). However, the two-finger widget was comparable to both tangibles ($M_{\text{two}-\text{knob}} = -0.02$ s, $[-0.18, 0.14]$; $M_{\text{two}-\text{puck}} = 0.15$ s, $[-0.00, 0.30]$). The tangible knob was slightly more error-prone than the tangible

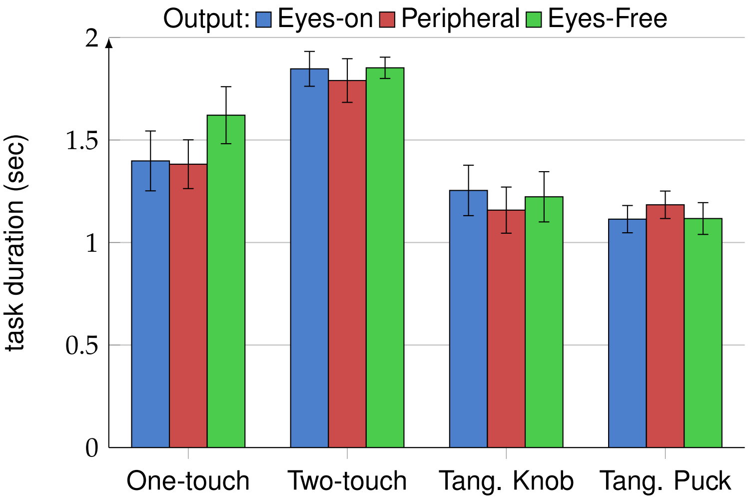


Figure 4.18: The mean movement time for each condition (mean and 95% CI). The results show that the two-touch input method was the slowest, and both tangibles were faster then the one-touch method.

puck ($M_{\text{knob}-\text{puck}} = 0.17$ s, $[0.02, 0.31]$).

No statistical or practical main effect of the location of information was found ($F_{2,38} = 0.22$, $p = .979$, $\eta_p^2 = .001$) and the mean number of overshooting was almost identical for all three output areas.

We found a large interaction effect between widget and output $F_{2.84,54,02} = 4.30$, $p = .01$, $\eta_p^2 = .18$). Figure 4.19 shows an overview of the interaction effect. When focusing on the eyes-free condition, however, the two-touch widget yielded more overshoots than the tangible puck ($M_{\text{two}-\text{puck}} = 0.13$ s, $[0.05, 0.21]$).

4.3.7 Discussion

The results indicate that the tangibles (knob and puck) outperformed both touch widgets (one-touch and two-touch). Specifically, tangibles were faster across the board (supporting H1). In terms of overshooting, the tangibles yielded fewer overshoots than the one-touch widget, but were comparable to the two-touch widget (partially sup-

Tangibles outperform touch input in all conditions.

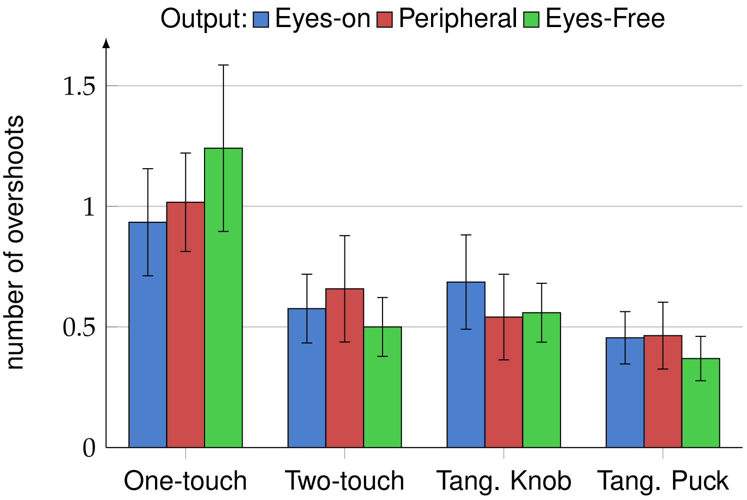


Figure 4.19: The number of overshoots for all conditions (mean and 95% CI). The results reveal that the most overshoots were done under the one-touch condition.

porting H3).

The friction of the second finger reduces the speed of the two-touch condition but makes it more accurate.

Under the eyes-free condition, the performance of one-touch widgets degraded more than others. The two-touch widget retained its performance under the eyes-free condition. We surmise that the additional friction of the second finger slowed down the input, allowing users to better control the virtual knob. The accuracy of a two-touch widget rivalled the tangible knob, but the two-touch widget was still worse than the tangible puck. Both tangibles were superior than touch widgets in terms of movement time (supporting H2). The tangible knob performed slightly worse than the tangible puck, probably because of the friction from the rotary mechanics inside the tangible knob. This can be improved by a better manufacturing process.

The peripheral condition was faster than the eyes-free condition. We speculate that the awareness of hand movement allows the users to be more confident when manipulating the widgets. Since the speed was comparable to the peripheral and the eyes-on condition, we speculate that occlusion does not influence the users' confidence in the rotation movements. For one-touch widgets, both speed and accuracy were improved by placing the widget in peripheral vision instead of using it eyes-free.

Our results further support previous researches: tangible controls are faster and less error-prone than touch-based controls [Lucchi et al., n.d.; Terrenghi et al., 2007; Tuddenham et al., 2010; Weiss et al., 2009]. However, our data is contrary to the findings of Kratz et al. [2011] as the tangible puck outperforms the one-touch widget in both movement time and overshoots. Of practical interest is the finding that differences between tangibles and touch-based interaction techniques are influenced by the locus of attention. This is particularlyinteresting for evaluating the suitability for the use of touch-interfaces in operative contexts where the human operator's visual attention must be directed towards an outside world, away from the controller [Bjelland et al., 2007].

Touch-based input is especially problematic in eyes-free tasks.

4.4 Conclusion

In this chapter, we addressed the issue of the limited tactile feedback of touch-screens by bringing tangibles to modern capacitive multi-touch displays (C 4.1). Our PERC tangibles were detected even when they were not touched by a user and, unlike previous designs, they did not get filtered out over time by the adaptive signal filters of the touch-screen. This was achieved by adding a field sensor that detects the electric field of the touch-surface. PERCs can be easily and affordably constructed from off-the-shelf components, and they work on a variety of commercially available touch-screen models.

PERC tangibles can be detect by a capacitive touch-screen when they are not touched by a user.

While we were able to show that PERC tangibles can be reliably detected on a number of common capacitive touch-screens (C 4.2), there is still potential to improve their accuracy and detection rate. For example, we plan to apply machine learning algorithms to minimize the displacement and angular error during the tangible detection. A more detailed investigation into how the pattern of detected markers is geometrically skewed on different touch-surfaces and at different angles could yield more precise estimations for the tangibles' positions.

The detection accuracy of PERC tangibles can be improved by using smarter tracking algorithms.

Since the different touch-screen models exhibit specific characteristics in how they pulse their electric field, the field sensor could be used to detect on which touch-screen model a tangible is currently placed. This would allow us to adapt thresholds, timeouts and error correction functions, specifically tuned for each screen model.

Tangibles rotary knobs outperform virtual knobs.

In addition to the technical contribution, we presented an experiment comparing tangible rotary knobs with virtual rotary knobs in eyes-on and eyes-free tasks (C 4.3). We found that tangible knobs are on average 20% faster than virtual knobs. In contrast to virtual knobs, tangibles did not perform significantly worse in eyes-free tasks compared to eyes-on tasks. The users were slower with the two-finger knob than with the one-finger knob, but they were more accurate using the two-finger knob.

Therefore, we draw the following design implications for rotary knobs:

1. For the best performance in rotation tasks, we recommend using tangible pucks over tangible knobs and over virtual touch widgets.
2. If it is not possible to use tangibles, use two-touch widgets for the tasks that require accuracy, and one-touch widgets for the tasks that require speed.
3. Design the user interface such that rotary widgets stay in peripheral vision of the user in order to increase manipulation confidence.
4. In eyes-free tasks, avoid using one-touch widgets. We hope that these design implications help designers to make an informed decision on using either tangibles or virtual knobs depending on which kind of tasks and applications.

Chapter 5

Conclusion

The aim of this thesis was to bring touch input to desk workspaces. Touch input allows the user to manipulate content directly in an intuitive way. This directness has several benefits: the interaction is easy to learn, and users can immediately see if their actions are furthering their goals, and if not, simply change the direction of their activity. In contrast to its success on mobile platforms, where touching is the dominant input method these days, touching on desktop workspaces has not fully exploited its potential. In the introduction of this thesis, we hypothesized that two main reasons prevented this development:

Two main challenges prevent the development of interactive desk workspaces.

1. The ergonomic challenges of touch interaction on large surfaces, and
2. the limited haptic feedback in touch-based systems.

In this thesis we addressed these two challenges by developing and analysing interactive desk workspaces that use touch input as their main input method. In the following, we will summarize our contributions and additionally give an outlook on future research.

5.1 Contributions

In the second chapter we explored the ergonomic challenges by developing the BendDesk system, an interactive desk workspace that merges a horizontal and a vertical surface into one large interactive workspace. We demonstrated how to display digital content and how to detect touch points on a curved interactive desk workspace (C 1.1).

Directly interacting with the vertical surface is inaccurate.

The curve influences the visual planning and motoric execution of gestures.

In addition to the technical contribution, we conducted an in depth analysis of how users interact with such a curved interactive desk workspace. Our studies showed that users perform simple dragging operations better on the horizontal surface compared to the vertical and curved surfaces. Therefore it should be used as main interaction area (C 1.2). The results also suggest that the curve represents a slight but noticeable physical barrier and should be rather used as a dock or transition area between both planar surfaces and not as the main interaction area. Furthermore, we proved that the curve impairs the user's perception such that they have problems estimating angles and distances to a target that is placed on the other side of the curve (C 1.3). Finally, our studies revealed that curved surfaces influence the visual planning phase of flicking gestures (C 1.4). However, we were able to show that users make systematic errors that can described by a mathematic model and therefore can be compensated successfully.

Indirect touch allows users to interact with the vertical surface comfortably.

In the third chapter, we addressed the issues of direct interaction by bringing the indirect touch-concept into the interactive desk workspace (C 2.1). Using indirect touch users can easily interact with the vertical surface by creating the touch input on the horizontal surface. We addressed the targeting issue of indirect touch by extending the basic two-state touch input model with an additional *tracking* state, which allows the user to aim at a target without unintentionally manipulating other objects. In an in-depth analysis we found that the *lift-and-tap* gesture is the most suitable switching method which allows users to switch between the new *tracking* state and the existing *engaged* state (C 2.2).

Also, in the third chapter, we explored the problem that in an indirect touch-system the horizontal surface is only an input device and cannot be used to display digital information. We solved this issue by creating an interactive desk workspace in which the users utilize their gaze to choose whether they want to interact with the horizontal surface directly or with the vertical surface indirectly (C 2.3). We propose two novel gaze-based interaction techniques, namely ITSS and ITOS, for easy touch interaction in interactive workspaces. Our studies revealed, that by introducing gaze as an additional modality, we are able to reduce the time that is needed to reach targets on the vertical screen as well as reduce effort that is needed to interact with the system. This enables users to comfortably interact with interactive workspaces for a longer period of time (e.g. a full working day). Nevertheless, further studies are needed to investigate long-term effects.

The user's gaze can be used to switch between direct and indirect touch.

In the fourth chapter, we introduced PERC tangibles to investigate the issue of the limited haptic feedback in touch-based systems (C 4.1). PERC are tangibles for modern capacitive touch-screens that can be detected even when they are not touched by a user and, unlike previous designs, they do not get filtered out over time by the adaptive signal filters of the touch-screen. This is achieved by adding a field sensor that detects the electric field of the touch-surface. PERCs can be easily and affordably constructed from off-the-shelf components, and they work on a variety of touch-screen models which are commercially available. In extensive studies, we were able to show that PERC tangibles can be reliably detected on a number of common capacitive touch-screens (C 4.2). In addition to the technical contribution, we presented an experiment comparing tangible rotary knobs with virtual rotary knobs in eyes-on and eyes-free tasks (C 4.3). We found that tangible knobs are on average 20% faster than virtual knobs. From these findings we derived a set of guidelines to help decide in which use case tangible knobs should be used.

PERC tangibles provide haptic feedback on capacitive touch-screens.

5.2 Future Work

Each of the chapters presented in this thesis offers interesting opportunities for follow-up research. However, at this point we will concentrate on the high-level questions.

The BendDesk system can be improved by using modern technologies.

In chapter 2 we described how do develop a curved desk workspace using a vision-based approach. We pointed out that this approach has several drawbacks and could be replaced by modern technologies in the near future. In Box 1 we speculate how such a modern curved desk workspace could be developed in the near future. In terms of user interactions, we focused only on basic interaction concepts and did not investigate what applications for such a system should look like and what kind of tasks are suitable for these systems. Hennecke et al. [2013c] presented several interesting concepts. However, an in-depth analysis of what kind of task benefits from such a desk workspace is still missing.

BOX 1: MODERN INTERACTIVE DESK WORKSPACES TECHNOLOGIES

A modern version of our BendDesk could probably use two key technologies: OLED displays [T.-H. Han et al., 2012] for displaying the digital content and a graphene layer on top of the display to detect touch-input [Jurewicz et al., 2014].

OLEDs are organic light-emitting diodes that emit light in response to an electric current. These displays work without a backlight which makes them thinner and lighter than a liquid crystal display (LCD). They are also flexible which makes them suitable for non-planar displays.

Graphene is an allotrope of carbon. It is about 100 times stronger than the strongest steel. It conducts heat and electricity efficiently and is almost transparent. This makes it a very good candidate to replace the Indium tin oxide (ITO) layer that is used to detect touch input in modern capacitive touchscreens. Jurewicz et al. [2014] showed that a graphene layer can even be used as bendable capacitive touch detection layer.

However, both technologies are not producible in large sizes and quantise yet. Therefore, it will probably take until 2020 before large touch-surfaces will use these technologies. In comparison to our BendDesk prototype, a modern version would be small, would have a high resolution in both output and input, and would possibly be bendable.

Figure 5.1: Concept of how our gaze supported interaction-techniques can be applied to multi-screen desk workspaces. In this setup users are able to interact with multiple vertical screens while creating the input on the horizontal surface [Voelker et al., 2015b].

The full potential of the combination of gaze and touch still needs to be explored.

In chapter 3 we introduced indirect touch to interactive desk workspaces. We also showed how the users' gaze can be used for switching between interacting with the horizontal surface directly and interacting with the vertical surface indirectly. That the combination of gaze and touch has several benefits is already shown by Pfeuffer et al. [2014]. Whether this concept is suitable in a desk workspace environment is still unclear. Furthermore, the gaze and touch concept could also be extended to system setups that include multiple vertical surfaces (Fig. 5.1). This would allow users to interact with several displays at the same time even if the user cannot reach these displays. However, in order to understand the benefits and limitations of such a system a more detailed analysis is required.

In which tasks tangibles improve the interactions still needs0 to be explored.

In the last chapter we introduced PERC tangibles and demonstrated that they can be detected reliably on modern capacitive touch-screens. We also showed that there is still potential to improve the detection in terms of accuracy and detection rate. A possible solution could be to apply machine-learning algorithms to minimize the displacement and angular error during the tangible detection. We also did not investigate for which tasks these tangibles could be used to improve the interactions on interactive desk workspaces. In our user study we proved that tangible knobs outperform virtual knobs. Nonetheless, we only conducted this study in a very controlled lab environment. Further studies in a real desk workspace with realistic tasks would be needed.

5.3 Closing Remarks

In this thesis we aimed to exploit interactive desk workspaces by overcoming both issues mentioned above. We exhibited how to develop an interactive desk workspace that addresses the ergonomic needs of a desk workspace and how to bring back haptic feedback to touchscreens using tangibles. During the development of all systems, we faced many engineering challenges and there is great potential for further iterations and improvements. We hope that concepts like BendDesk and PERCs can soon be realized with improved technologies.

Our studies were focused on basic interaction concepts

In our studies we analyzed how users interact with interactive desk workspaces that combine horizontal and vertical interactive surfaces. We mostly focused on basic interaction concepts such as tapping, dragging, and how users can use the extended interaction model of indirect touch. However, for fully exploring the potential of touch input in desk workspaces, future work needs to focus on high-level interaction concepts and on interfaces for tasks that are done in a desk workspace.

First products incorporate the presents concepts.

First products such as the HP Sprout already incorporate concepts that are very similar to our findings. For example, the main interaction area is the horizontal surface and the interaction with the vertical surface is very limited. However, we believe there are still a lot of open questions that need to be answered before interactive desk workspaces can replace classic desk workspaces. We hope that the results of this thesis can be used as a foundation and inspiration not only for further research but also for developers and designers who intend to develop interactive desk workspaces.

Bibliography

Ashdown, Mark, Oka, Kenji, Sato, Yoichi. “Combining Head Tracking and Mouse Input for a GUI on Multiple Monitors”. *CHI '05 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '05. Portland, OR, USA: ACM, 2005, pp. 1188–1191. URL: `http://doi.acm.org/10.1145/1056808.1056873`.

Bachynskyi, Myroslav, Palmas, Gregorio, Oulasvirta, Antti, Steimle, Jürgen, Weinkauf, Tino. “Performance and Ergonomics of Touch Surfaces: A Comparative Study Using Biomechanical Simulation”. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: ACM, 2015, pp. 1817–1826. URL: `http://doi.acm.org/10.1145/2702123.2702607`.

Banham, Joanna. *Encyclopedia of Interior Design*. Routledge, 1997.

Barrett, Gary, Omote, Ryomei. “Projected-Capacitive Touch Technology”. *Information Display* 26.3 (2010), pp. 16–21.

Benko, Hrvoje, Jota, Ricardo, Wilson, Andrew. “MirageTable: Freehand Interaction on a Projected Augmented Reality Tabletop”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 199–208. URL: `http://doi.acm.org/10.1145/2207676.2207704`.

Benko, Hrvoje, Wilson, Andrew D., Baudisch, Patrick. “Precise Selection Techniques for Multi-touch Screens”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montreal, Québec, Canada: ACM, 2006, pp. 1263–1272. URL: `http://doi.acm.org/10.1145/1124772.1124963`.

Bi, Xiaojun, Bae, Seok-Hyung, Balakrishnan, Ravin. “Effects of Interior Bezels of Tiled-monitor Large Displays on Visual Search, Tunnel Steering, and Target Selection”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Sys-*

tems. CHI '10. Atlanta, Georgia, USA: ACM, 2010, pp. 65–74. URL: `http://doi.acm.org/10.1145/1753326.1753337`.

Bjelland, Hans Vanhauwaert, Hoff Thomas Øvergård, Kjell Ivar, Bjørkli, Cato. "A Case Study of a Touch-based Interface for In-car Audio Systems". *Design Journal* 10.1 (2007), pp. 24–34. URL: `http://dx.doi.org/10.2752/146069207789318054`.

Boring, Sebastian, Jurmu, Marko, Butz, Andreas. "Scroll, Tilt or Move It: Using Mobile Phones to Continuously Control Pointers on Large Public Displays". *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*. OZCHI '09. Melbourne, Australia: ACM, 2009, pp. 161–168. URL: `http://doi.acm.org/10.1145/1738826.1738853`.

Bridge, Mark. *An encyclopedia of desks*. Wellfleet, 1989.

Buxton, William. "A Three-state Model of Graphical Input". *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction*. INTERACT '90. Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., 1990, pp. 449–456. URL: `http://dl.acm.org/citation.cfm?id=647402.725582`.

Buxton, William, Fitzmaurice, George, Balakrishnan, Ravin, Kurtenbach, Gordon. "Large displays in automotive design". *IEEE Computer Graphics and Applications* 20.4 (July 2000), pp. 68–75. ISSN: 0272-1716. URL: `http://dx.doi.org/10.1109/38.851753`.

Buxton, William, Hill, Ralph, Rowley, Peter. "Issues and techniques in touch-sensitive tablet input". *SIGGRAPH Comput. Graph.* 19.3 (1985), pp. 215–224. URL: `http://doi.acm.org/10.1145/325165.325239`.

Cao, Xiang, Wilson, A.D., Balakrishnan, R., Hinckley, K., Hudson, S.E. "ShapeTouch: Leveraging contact shape on interactive surfaces". *3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems '08.* 2008, pp. 129–136. URL: `http://dx.doi.org/10.1109/TABLETOP.2008.4660195`.

Card, Stuart K., Mackinlay, Jock D., Robertson, George G. "The Design Space of Input Devices". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '90. Seattle, Washington, USA: ACM, 1990, pp. 117–124. URL: `http://doi.acm.org/10.1145/97243.97263`.

Chan, Liwei, Müller, Stefanie, Roudaut, Anne, Baudisch, Patrick. "CapStones and ZebraWidgets: Sensing Stacks of Building Blocks, Dials and Sliders on Capacitive Touch Screens". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 2189–2192. URL: `http://doi.acm.org/10.1145/2207676.2208371`.

Chang, Dempsey, Dooley, Laurence, Tuovinen, Juhani E. "Gestalt Theory in Visual Screen Design: A New Look at an Old Subject". *Proceedings of the Seventh World Conference on Computers in Education Conference on Computers in Education: Australian Topics - Volume 8*. CRPIT '02. Copenhagen, Denmark: Australian Computer Society, Inc., 2002, pp. 5–12. URL: `http://dl.acm.org/citation.cfm?id=820060.820062`.

Cherek, Christian, Voelker, Simon, Thar, Jan, Linden, Rene, Busch, Florian, Borchers, Jan. "PERCs Demo: Persistently Trackable Tangibles on Capacitive Multi-Touch Displays". *Proceedings of the 2015 International Conference on Interactive Tabletops and Surfaces*. Maderia, Portugal: ACM, Nov. 2015, pp. 389–392. URL: `http://dx.doi.org/10.1145/2817721.2823474`.

Coldefy, François, Louis-dit-Picard, Stphane. "DigiTable: an Interactive Multiuser Table for Collocated and Remote Collaboration Enabling Remote Gesture Visualization". *2007 IEEE Conference on Computer Vision and Pattern Recognition*. June 2007, pp. 1–8. URL: `http://dx.doi.org/10.1109/CVPR.2007.383471`.

Dillencourt, Michael B., Samet, Hanan, Tamminen, Markku. "A General Approach to Connected-component Labeling for Arbitrary Image Representations". *J. ACM* 39.2 (Apr. 1992), pp. 253–280. URL: `http://doi.acm.org/10.1145/128749.128750`.

Everitt, Katherine, Shen, Chia, Ryall, Kathy, Forlines, Clifton. "MultiSpace: enabling electronic document micro-mobility in table-centric, multi-device environments". *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP '06)*. Jan. 2006, URL: `http://dx.doi.org/10.1109/TABLETOP.2006.23`.

Fineman, M. *The Nature of Visual Illusion*. New York: Dover, 1996.

Fitts, Paul M. "The information capacity of the human motor system in controlling the amplitude of movement." *Journal of experimental psychology* 47.6 (1954), p. 381.

Fitzmaurice, George W., Ishii, Hiroshi, Buxton, William A. S. "Bricks: Laying the Foundations for Graspable User Interfaces". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '95. Denver, Colorado, USA: ACM

Press/Addison-Wesley Publishing Co., 1995, pp. 442–449. URL: `http://dx.doi.org/10.1145/223904.223964`.

Fono, David, Vertegaal, Roel. "EyeWindows: Evaluation of Eye-controlled Zooming Windows for Focus Selection". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '05. Portland, Oregon, USA: ACM, 2005, pp. 151–160. URL: `http://doi.acm.org/10.1145/1054972.1054994`.

Forlines, Clifton, Balakrishnan, Ravin. "Evaluating tactile feedback and direct vs. indirect stylus input in pointing and crossing selection tasks". *Proc. CHI '08*. ACM, 2008, pp. 1563–1572. URL: `http://doi.acm.org/10.1145/1357054.1357299`.

Forlines, Clifton, Shen, Chia, Buxton, Bill. "Glimpse: A Novel Input Model for Multi-level Devices". *CHI '05 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '05. Portland, OR, USA: ACM, 2005, pp. 1375–1378. URL: `http://doi.acm.org/10.1145/1056808.1056920`.

Forlines, Clifton, Wigdor, Daniel, Shen, Chia, Balakrishnan, Ravin. "Direct-touch vs. Mouse Input for Tabletop Displays". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: ACM, 2007, pp. 647–656. URL: `http://doi.acm.org/10.1145/1240624.1240726`.

Fostervold, Knut Inge, Aarås, Arne, Lie, Ivar. "Work with visual display units: long-term health effects of high and downward line-of-sight in ordinary office environments". *International journal of industrial ergonomics* 36.4 (2006), pp. 331–343.

Gilliot, Jérémie, Casiez, Géry, Roussel, Nicolas. "Impact of Form Factors and Input Conditions on Absolute Indirect-touch Pointing Tasks". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Toronto, Ontario, Canada: ACM, 2014, pp. 723–732. URL: `http://doi.acm.org/10.1145/2556288.2556997`.

Gindrat, Anne-Dominique, Chytiris, Magali, Balerna, Myriam, Rouiller, Eric M., Ghosh, Arko. "Use-Dependent Cortical Processing from Fingertips in Touchscreen Phone Users". *Current Biology* 25.1 (2015), pp. 109–116. URL: `http://www.sciencedirect.com/science/article/pii/S0960982214014870`.

Green, R. T., Hoyle, E. M. "The Poggendorff Illusion as a Constancy Phenomeno". *Nature* 200 (1963), pp. 611–612.

Grossman, Tovi, Balakrishnan, Ravin. "The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '05. Portland, Oregon, USA: ACM, 2005, pp. 281–290. URL: `http://doi.acm.org/10.1145/1054972.1055012`.

Hammerton, M, Tickner, AH. "An investigation into the comparative suitability of forearm, hand and thumb controls in aquisition tasks". *Ergonomics* 9.2 (1966), pp. 125–130.

Han, Jefferson Y. "Low-cost Multi-touch Sensing Through Frustrated Total Internal Reflection". *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. UIST '05. Seattle, WA, USA: ACM, 2005, pp. 115–118. URL: `http://doi.acm.org/10.1145/1095034.1095054`.

Han, Tae-Hee, Lee, Youngbin, Choi, Mi-Ri, Woo, Seong-Hoon, Bae, Sang-Hoon, Hong, Byung Hee, Ahn, Jong-Hyun, Lee, Tae-Woo. "Extremely efficient flexible organic light-emitting diodes with modified graphene anode". *Nature Photonics* 6.2 (2012), pp. 105–110. URL: `https://doi.org/10.1038/nphoton.2011.318`.

Hardy, John. "Experiences: A Year in the Life of an Interactive Desk". *Proceedings of the Designing Interactive Systems Conference*. DIS '12. Newcastle Upon Tyne, United Kingdom: ACM, 2012, pp. 679–688. URL: `http://doi.acm.org/10.1145/2317956.2318058`.

Hennecke, Fabian, De Luca, Alexander, Nguyen, Ngo Dieu Huong, Boring, Sebastian, Butz, Andreas. "Investigating Pointing Tasks across Angularly Coupled Display Areas". *Human-Computer Interaction–INTERACT 2013*. Springer, 2013, pp. 720–727.

Hennecke, Fabian, Matzke, Wolfgang, Butz, Andreas. "How Screen Transitions Influence Touch and Pointer Interaction Across Angled Display Arrangements". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 209–212. URL: `http://doi.acm.org/10.1145/2207676.2207705`.

Hennecke, Fabian, Voelker, Simon, Schenk, Maximilian, Schaper, Hauke, Borchers, Jan, Butz, Andreas. "PerspectiveTable: Blending Physical and Virtual Collaborative Workspaces". *Workshop on Blending Interaction*. Paris, France, Apr. 2013. URL: `http://hci.uni-konstanz.de/blendedinteraction2013/`.

Hennecke, Fabian, Voelker, Simon, Schenk, Maximilian, Schaper, Hauke, Borchers, Jan, Butz, Andreas. “Simplifying Remote Collaboration Through Spatial Mirroring”. *INTERACT ’13: Proceedings of the 14th IFIP TC13 Conference on Human-Computer Interaction*. INTERACT ’13. Cape Town, South Africa: Springer Berlin Heidelberg, 2013, pp. 624–631. URL: `http://dx.doi.org/10.1007/978-3-642-40477-1_41`.

Hilliges, Otmar, Baur, Dominikus, Butz, Andreas. “Photohelix: Browsing, Sorting and Sharing Digital Photo Collections”. *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP ’07. Second Annual IEEE International Workshop on*. Oct. 2007, pp. 87–94. URL: `http://dx.doi.org/10.1109/TABLETOP.2007.20`.

Hincapié-Ramos, Juan David, Guo, Xiang, Moghadasian, Paymahn, Irani, Pourang. “Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-air Interactions”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’14. Toronto, Ontario, Canada: ACM, 2014, pp. 1063–1072. URL: `http://doi.acm.org/10.1145/2556288.2557130`.

Hinrichs, Uta, Carpendale, Sheelagh. “Gestures in the Wild: Studying Multi-touch Gesture Sequences on Interactive Tabletop Exhibits”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’11. Vancouver, BC, Canada: ACM, 2011, pp. 3023–3032. URL: `http://doi.acm.org/10.1145/1978942.1979391`.

Hopfengaertner, Christoph. *Christoph Hopfengaertner, Meisterstueck, 1790*. 1790. URL: `https://commons.wikimedia.org/wiki/File:Hopfeng%C3%83%C2%A4rtner_Meisterst%C3%83%C2%BCck.jpg`.

Hutchins, Edwin L, Hollan, James D, Norman, Donald A. “Direct manipulation interfaces”. *Human–Computer Interaction* 1.4 (1985), pp. 311–338. URL: `http://dx.doi.org/10.1207/s15327051hci0104_2`.

IDC. *International Data Corporation*. 2016. URL: `www.idc.com`.

ITU. *United Nations specialized agency for information and communication technologies*. 2016. URL: `www.itu.int`.

Izadi, Shahram, Brignull, Harry, Rodden, Tom, Rogers, Yvonne, Underwood, Mia. “Dynamo: A Public Interactive Surface Supporting the Cooperative Sharing and Exchange of Media”. *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*. UIST ’03. Vancouver, Canada: ACM, 2003, pp. 159–168. URL: `http://doi.acm.org/10.1145/964696.964714`.

Jansen, Yvonne, Dragicevic, Pierre, Fekete, Jean-Daniel. "Tangible Remote Controllers for Wall-size Displays". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 2865–2874. URL: `http://doi.acm.org/10.1145/2207676.2208691`.

Jax, Steven A, Rosenbaum, David A. "Hand path priming in manual obstacle avoidance: evidence that the dorsal stream does not only control visually guided actions in real time." *Journal of Experimental Psychology: Human Perception and Performance* 33.2 (2007), p. 425.

Jetter, Hans-Christian, Gerken, Jens, Zöllner, Michael, Reiterer, Harald, Milic-Frayling, Natasa. "Materializing the Query with Facet-streams: A Hybrid Surface for Collaborative Search on Tabletops". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 3013–3022. URL: `http://doi.acm.org/10.1145/1978942.1979390`.

Jordà, Sergi, Geiger, Günter, Alonso, Marcos, Kaltenbrunner, Martin. "The reacTable: Exploring the Synergy Between Live Music Performance and Tabletop Tangible Interfaces". *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*. TEI '07. Baton Rouge, Louisiana: ACM, 2007, pp. 139–146. URL: `http://doi.acm.org/10.1145/1226969.1226998`.

Jurewicz, Izabela, Fahimi, Azin, Lyons, Phillip E, Smith, Ronan J, Cann, Maria, Large, Matthew L, Tian, Mingwen, Coleman, Jonathan N, Dalton, Alan B. "Insulator-Conductor Type Transitions in Graphene-Modified Silver Nanowire Networks: A Route to Inexpensive Transparent Conductors". *Advanced Functional Materials* 24.48 (2014), pp. 7580–7587. URL: `https://doi.org/10.1002/adfm.201402547`.

Kin, Kenrick, Miller, Tom, Bollensdorff, Björn, DeRose, Tony, Hartmann, Björn, Agrawala, Maneesh. "Eden: A Professional Multitouch Tool for Constructing Virtual Organic Environments". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 1343–1352. URL: `http://doi.acm.org/10.1145/1978942.1979141`.

Klokmose, Clemens Nylandsted, Kristensen, Janus Bager, Bagge, Rolf, Halskov, Kim. "BullsEye: High-Precision Fiducial Tracking for Table-based Tangible Interaction". *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ITS '14. Dresden, Germany: ACM, 2014, pp. 269–278. URL: `http://doi.acm.org/10.1145/2669485.2669503`.

Knoedel, Sebastian, Hachet, Martin. “Multi-touch RST in 2D and 3D spaces: Studying the impact of directness on user performance”. IEEE, 2011, pp. 75–78. URL: `http://dx.doi.org/0.1109/3DUI.2011.5759220`.

Kosara, R. “Indirect multi-touch interaction for brushing in parallel coordinates”. *Proc. VDA ’11*. 2011, pp. 786–809.

Kratz, Sven, Westermann, Tilo, Rohs, Michael, Essl, Georg. “CapWidgets: Tangile Widgets Versus Multi-touch Controls on Mobile Devices”. *CHI ’11 Extended Abstracts on Human Factors in Computing Systems*. CHI EA ’11. Vancouver, BC, Canada: ACM, 2011, pp. 1351–1356. URL: `http://doi.acm.org/10.1145/1979742.1979773`.

Lange, Wolfgang, Windel, Armin. *Kleine Ergonomische Datensammlung*. TÜV Media GmbH, 2008.

Langolf, G. D., Chaffin, D. B., Foulke, J. A. “An investigation of Fitts’ law using a wide range of movement amplitudes”. *Journal of Motor Behaviour* 8 (1976), pp. 113–128.

Leitner, Jakob, Powell, James, Brandl, Peter, Seifried, Thomas, Haller, Michael, Dorray, Bernard, To, Paul. “Flux: a tilting multi-touch and pen based surface”. *CHI’09 Extended Abstracts on Human Factors in Computing Systems*. ACM. 2009, pp. 3211–3216.

Liang, Rong-Hao, Chan, Liwei, Tseng, Hung-Yu, Kuo, Han-Chih, Huang, Da-Yuan, Yang, De-Nian, Chen, Bing-Yu. “GaussBricks: Magnetic Building Blocks for Constructive Tangible Interactions on Portable Displays”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’14. Toronto, Ontario, Canada: ACM, 2014, pp. 3153–3162. URL: `http://doi.acm.org/10.1145/2556288.2557105`.

Lin, John, Wu, Ying, Huang, T. S. “Modeling the constraints of human hand motion”. *Human Motion, 2000. Proceedings. Workshop on Human Motion*. 2000, pp. 121–126. URL: `http://dx.doi.org/10.1109/HUMO.2000.897381`.

Linden, Rene. “Multitouchkit: A Software Framework for Touch Input and Tangibles on Tabletops and”. Englisch (eng). Master’s Thesis. Aachen: RWTH Aachen University, Sept. 2015.

Lucchi, Aurélien, Jermann, Patrick, Zufferey, Guillaume, Dillenbourg, Pierre. “An Empirical Evaluation of Touch and Tangible Interfaces for Tabletop Displays”.

Proc. TEI '10. Cambridge, Massachusetts, USA: ACM, ACM, 177–184. ISBN: 978-1-60558-841-4. URL: `http://doi.acm.org/10.1145/1709886.1709917`.

Luff, Paul, Heath, Christian, Kuzuoka, Hideaki, Yamazaki, Keiichi, Yamashita, Jun. "Handling Documents and Discriminating Objects in Hybrid Spaces". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montreal, Quebec, Canada: ACM, 2006, pp. 561–570. URL: `http://doi.acm.org/10.1145/1124772.1124858`.

Ma, Joyce, Sindorf, Lisa, Liao, Isaac, Frazier, Jennifer. "Using a Tangible Versus a Multi-touch Graphical User Interface to Support Data Exploration at a Museum Exhibit". *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '15. Stanford, California, USA: ACM, 2015, pp. 33–40. URL: `http://doi.acm.org/10.1145/2677199.2680555`.

MacKenzie, I. Scott, Oniszczak, Aleks. "A Comparison of Three Selection Techniques for Touchpads". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '98. Los Angeles, California, USA: ACM Press/Addison-Wesley Publishing Co., 1998, pp. 336–343. URL: `http://dx.doi.org/10.1145/274644.274691`.

Matejka, Justin, Grossman, Tovi, Lo, Jessica, Fitzmaurice, George. "The Design and Evaluation of Multi-finger Mouse Emulation Techniques". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: ACM, 2009, pp. 1073–1082. URL: `http://doi.acm.org/10.1145/1518701.1518865`.

Matsushita, Nobuyuki, Rekimoto, Jun. "HoloWall: Designing a Finger, Hand, Body, and Object Sensitive Wall". *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*. UIST '97. Banff, Alberta, Canada: ACM, 1997, pp. 209–210. URL: `http://doi.acm.org/10.1145/263407.263549`.

Matviienko, Andrii. "GazeTouch: Using Gaze Tracking to Select Indirect Touch Targets". Englisch (eng). Master's Thesis. Aachen: RWTH Aachen University, Dec. 2014.

Morris, Meredith Ringel, Brush, AJ Bernheim, Meyers, Brian R. "A field study of knowledge workers' use of interactive horizontal displays". *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*. IEEE. 2008, pp. 105–112.

Morris, Meredith Ringel, Brush, AJ Bernheim, Meyers, Brian R. "Reading revisited: Evaluating the usability of digital display surfaces for active reading tasks". *Hor-*

izontal Interactive Human-Computer Systems, 2007. TABLETOP'07. Second Annual IEEE International Workshop on. IEEE. 2007, pp. 79–86.

Moscovich, Tomer, Hughes, John F. "Indirect Mappings of Multi-touch Input Using One and Two Hands". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy: ACM, 2008, pp. 1275–1284. URL: `http://doi.acm.org/10.1145/1357054.1357254`.

Müller-Tomfelde, Christian, Wessels, Anja, Schremmer, Claudia. "Tilted tabletops: In between horizontal and vertical workspaces". *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*. IEEE. 2008, pp. 49–56.

Naamad, A, Lee, DT, Hsu, W-L. "On the maximum empty rectangle problem". *Discrete Applied Mathematics* 8.3 (1984), pp. 267–277.

Nakajima, Kosuke, Itoh, Yuichi, Voelker, Simon, Thoresen, Christian, Øvergård, Kjell Ivar, Borchers, Jan. "PUCs: Detecting Transparent, Passive Untouched Capacitive Widgets on Unmodified Multi-touch Displays". *Proc. IPSJ Interaction '14: Proceedings of the Conference on Information Processing Society of Japan*. Information Processing Society of Japan, 2014, pp. 108–113. URL: `http://ci.nii.ac.jp/naid/110009867115`.

Nancel, Mathieu, Chapuis, Olivier, Pietriga, Emmanuel, Yang, Xing-Dong, Irani, Pourang P., Beaudouin-Lafon, Michel. "High-precision Pointing on Large Wall Displays Using Small Handheld Devices". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: ACM, 2013, pp. 831–840. URL: `http://doi.acm.org/10.1145/2470654.2470773`.

Nowacka, Diana, Ladha, Karim, Hammerla, Nils Y., Jackson, Daniel, Ladha, Cassim, Rukzio, Enrico, Olivier, Patrick. "Touchbugs: Actuated Tangibles on Multi-touch Tables". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: ACM, 2013, pp. 759–762. URL: `http://doi.acm.org/10.1145/2470654.2470761`.

Ohtsuki, Tatsuyuki. "Inhibition of individual fingers during grip strength exertion". *Ergonomics* 24.1 (1981), pp. 21–36.

Olwal, Alex, Feiner, Steven, Heyman, Susanna. "Rubbing and Tapping for Precise and Rapid Selection on Touch-screen Displays". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy: ACM, 2008, pp. 295–304. URL: `http://doi.acm.org/10.1145/1357054.1357105`.

Øvergård, Kjell Ivar, Fostervold, Knut Inge, Bjelland, Hans Vanhauwaert, Hoff, Thomas. "Knobology in use: an experimental evaluation of ergonomics recommendations". *Ergonomics* 50.5 (2007), Taylor & Francis, 694–705. URL: `http://dx.doi.org/10.1080/00140130601168046`.

Patten, James, Ishii, Hiroshi, Hines, Jim, Pangaro, Gian. "Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '01. Seattle, Washington, USA: ACM, 2001, pp. 253–260. URL: `http://doi.acm.org/10.1145/365024.365112`.

Patten, James, Recht, Ben, Ishii, Hiroshi. "Audiopad: A Tag-based Interface for Musical Performance". *Proceedings of the 2002 Conference on New Interfaces for Musical Expression*. NIME '02. Dublin, Ireland: National University of Singapore, 2002, pp. 1–6. URL: `http://dl.acm.org/citation.cfm?id=1085171.1085175`.

Pawluk, Dianne TV, Howe, Robert D. "Dynamic contact of the human fingerpad against a flat surface". *Journal of biomechanical engineering* 121.6 (1999), pp. 605–611.

Penfield, W., Rasmussen, T.. *The cerebral cortex of man*. New York: Macmillan, 1950.

Pfeuffer, Ken, Alexander, Jason, Chong, Ming Ki, Gellersen, Hans. "Gaze-touch: Combining Gaze with Multi-touch for Interaction on the Same Surface". *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: ACM, 2014, pp. 509–518. URL: `http://doi.acm.org/10.1145/2642918.2647397`.

Potter, R. L., Weldon, L. J., Shneiderman, B. "Improving the Accuracy of Touch Screens: An Experimental Evaluation of Three Strategies". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '88. Washington, D.C., USA: ACM, 1988, pp. 27–32. URL: `http://doi.acm.org/10.1145/57167.57171`.

Reetz, Adrian, Gutwin, Carl, Stach, Tadeusz, Nacenta, Miguel, Subramanian, Sriram. "Superflick: a natural and efficient technique for long-distance object placement on digital tables". *Proc. Graphics Interface '06*. Canadian Information Processing Society, 2006, pp. 163–170. URL: `http://dl.acm.org/citation.cfm?id=1143079.1143106`.

Rekimoto, Jun. "SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces". *Proceedings of the SIGCHI Conference on Human Factors in Com-*

puting Systems. CHI '02. Minneapolis, Minnesota, USA: ACM, 2002, pp. 113–120. URL: `http://doi.acm.org/10.1145/503376.503397`.

Rekimoto, Jun, Saitoh, Masanori. "Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '99. Pittsburgh, Pennsylvania, USA: ACM, 1999, pp. 378–385. URL: `http://doi.acm.org/10.1145/302979.303113`.

Restle, F., Decker, J. "Size of the Mueller-Lyer illusion as a function of ist dimensions: Theory and data". *Theory and data. Perception and Psychophysics* 21 (1977), pp. 489–503.

Ryall, Kathy, Morris, Meredith Ringel, Everitt, Katherine, Forlines, Clifton, Shen, Chia. "Experiences with and observations of direct-touch tabletops". *Horizontal Interactive Human-Computer Systems, 2006. TableTop 2006. First IEEE International Workshop on*. IEEE. 2006, 8–pp.

Sato, T., Fukuchi, K., Koike, H. "Implementation and evaluations of vision-based finger flicking gesture recognition for tabletops". *3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems '08*. 2008, pp. 137–144. URL: `http://dx.doi.org/10.1109/TABLETOP.2008.4660196`.

Schaechter, Moselio. *Desk encyclopedia of microbiology*. Academic Press, 2010.

Schmidt, Dominik, Block, Florian, Gellersen, Hans. "A Comparison of Direct and Indirect Multi-touch Input for Large Surfaces". *Proc.INTERACT*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 582–594. URL: `http://www.springerlink.com/content/c1689n68104100p1/`.

Schöning, Johannes, Hook, Jonathan, Bartindale, Tom, Schmidt, Dominik, Oliver, Patrick, Echtler, Florian, Motamedi, Nima, Brandl, Peter, Zadow, Ulrich von. "Building Interactive Multi-touch Surfaces". English. *Tabletops - Horizontal Interactive Displays*. Ed. by Christian Müller-Tomfelde. Human-Computer Interaction Series. Springer London, 2010, pp. 27–49. URL: `http://dx.doi.org/10.1007/978-1-84996-113-4_2`.

Schwarz, Tobias, Hennecke, Fabian, Lauber, Felix, Reiterer, Harald. "Perspective+detail: a visualization technique for vertically curved displays". *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM. 2012, pp. 485–488.

Sears, Andrew. "Improving touchscreen keyboards: design issues and a comparison with other devices". *Interacting with computers* 3.3 (1991), pp. 253–269.

Shell, Jeffrey S., Vertegaal, Roel, Cheng, Daniel, Skaburskis, Alexander W., Sohn, Changuk, Stewart, A. James, Aoudeh, Omar, Dickie, Connor. "ECSGlasses and EyePliances: Using Attention to Open Sociable Windows of Interaction". *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*. ETRA '04. San Antonio, Texas: ACM, 2004, pp. 93–100. URL: `http://doi.acm.org/10.1145/968363.968384`.

Shneiderman, Ben. "The future of interactive systems and the emergence of direct manipulation". *Behaviour & Information Technology* 1.3 (1982), pp. 237–256. URL: `http://dx.doi.org/10.1080/01449298208914450`.

Silicon Labs. *How to Minimize Touchscreen Electromagnetic Interference*. Tech. rep. 2011.

Smith, J. David, Graham, T. C. Nicholas. "Use of Eye Movements for Video Game Control". *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. ACE '06. Hollywood, California, USA: ACM, 2006. URL: `http://doi.acm.org/10.1145/1178823.1178847`.

Sommerich, Carolyn M, Joines, Sharon MB, Psihogios, Jennie P. "Effects of computer monitor viewing angle and related factors on strain, performance, and preference outcomes". *Human Factors: The Journal of the Human Factors and Ergonomics Society* 43.1 (2001), pp. 39–55.

Stellmach, Sophie, Dachselt, Raimund. "Look & Touch: Gaze-supported Target Acquisition". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 2981–2990. URL: `http://doi.acm.org/10.1145/2207676.2208709`.

Stellmach, Sophie, Dachselt, Raimund. "Still Looking: Investigating Seamless Gaze-supported Selection, Positioning, and Manipulation of Distant Targets". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: ACM, 2013, pp. 285–294. URL: `http://doi.acm.org/10.1145/2470654.2470695`.

Stellmach, Sophie, Stober, Sebastian, Nürnberger, Andreas, Dachselt, Raimund. "Designing Gaze-supported Multimodal Interactions for the Exploration of Large Image Collections". *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*. NGCA '11. Karlskrona, Sweden: ACM, 2011, 1:1–1:8. URL: `http://doi.acm.org/10.1145/1983302.1983303`.

Tabard, Aurélien, Hincapié-Ramos, Juan-David, Esbensen, Morten, Bardram, Jakob E. “The eLabBench: An Interactive Tabletop System for the Biology Laboratory”. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’11. Kobe, Japan: ACM, 2011, pp. 202–211. URL: `http://doi.acm.org/10.1145/2076354.2076391`.

Terrenghi, Lucia, Kirk, David, Sellen, Abigail, Izadi, Shahram. “Affordances for Manipulation of Physical Versus Digital Media on Interactive Surfaces”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’07. San Jose, California, USA: ACM, 2007, pp. 1157–1166. URL: `http://doi.acm.org/10.1145/1240624.1240799`.

Thar, Jan. “APUCS: Active Tangible Input Devices for Capacitive Multitouch Surface”. Englisch (eng). Master’s thesis. Aachen: RWTH Aachen University, June 2015.

Tognazzini, Bruce. *TOG on Interface*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1992. ISBN: 0201608421.

Tuddenham, Philip, Kirk, David, Izadi, Shahram. “Graspables Revisited: Multi-touch vs. Tangible Input for Tabletop Displays in Acquisition and Manipulation Tasks”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’10. Atlanta, Georgia, USA: ACM, 2010, pp. 2223–2232. URL: `http://doi.acm.org/10.1145/1753326.1753662`.

Turner, Jayson. “Cross-device Eye-based Interaction”. *Proceedings of the Adjunct Publication of the 26th Annual ACM Symposium on User Interface Software and Technology*. UIST ’13 Adjunct. St. Andrews, Scotland, United Kingdom: ACM, 2013, pp. 37–40. URL: `http://doi.acm.org/10.1145/2508468.2508471`.

Turner, Jayson, Bulling, Andreas, Gellersen, Hans. “Combining Gaze with Manual Interaction to Extend Physical Reach”. *Proceedings of the 1st International Workshop on Pervasive Eye Tracking & Mobile Eye-based Interaction*. PETMEI ’11. Beijing, China: ACM, 2011, pp. 33–36. URL: `http://doi.acm.org/10.1145/2029956.2029966`.

Underkoffler, John, Ishii, Hiroshi. “Urp: A Luminous-tangible Workbench for Urban Planning and Design”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’99. Pittsburgh, Pennsylvania, USA: ACM, 1999, pp. 386–393. URL: `http://doi.acm.org/10.1145/302979.303114`.

Vertegaal, Roel, Mamuji, Aadil, Sohn, Changuk, Cheng, Daniel. “Media Eyepliances: Using Eye Tracking for Remote Control Focus Selection of Appliances”.

CHI '05 Extended Abstracts on Human Factors in Computing Systems. CHI EA '05. Portland, OR, USA: ACM, 2005, pp. 1861–1864. URL: `http://doi.acm.org/10.1145/1056808.1057041`.

Voelker, Simon. "BendDesk: Seamless Integration of Horizontal and Vertical Multi-Touch Surfaces in Desk Environments". Englisch (eng). Diploma Thesis. Aachen: RWTH Aachen University, June 2010. URL: `http://hci.rwth-aachen.de/materials/publications/voelker2010a.pdf`.

Voelker, Simon, Cherek, Christian, Thar, Jan, Karrer, Thorsten, Thoresen, Christian, Øvergård, Kjell Ivar, Borchers, Jan. "PERCs: Persistently Trackable Tangibles on Capacitive Multi-Touch Displays". Englisch (eng). *UIST '15: Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. UIST '15. Charlotte, NC, USA: ACM, Nov. 2015, pp. 351–356. URL: `http://dx.doi.org/10.1145/2807442.2807466`.

Voelker, Simon, Matviienko, Andrii, Schöning, Johannes, Borchers, Jan. "Combining Direct and Indirect Touch Input for Interactive Desktop Workspaces using Gaze Input". *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*. SUI '15. Los Angeles, CA, USA: ACM, Aug. 2015, pp. 79–88. URL: `http://dx.doi.org/10.1145/2788940.2788949`.

Voelker, Simon, Nakajima, Kosuke, Thoresen, Christian, Itoh, Yuichi, Øvergård, Kjell Ivar, Borchers, Jan. "PUCs Demo: Detecting Transparent, Passive Untouched Capacitive Widgets". *ITS EA '13: Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*. St. Andrews, Scotland, United Kingdom: ACM, 2013, pp. 325–328. URL: `http://dx.doi.org/10.1145/2512349.2514595`.

Voelker, Simon, Nakajima, Kosuke, Thoresen, Christian, Itoh, Yuichi, Øvergård, Kjell Ivar, Borchers, Jan. "PUCs Demo: Detecting Transparent, Passive Untouched Capacitive Widgets on Unmodified Multi-touch Displays". *UIST EA '13: Extended Abstracts of the ACM Symposium on User Interface Software and Technology*. St. Andrews, Scotland, United Kingdom: ACM, 2013, pp. 1–2. URL: `http://dx.doi.org/10.1145/2508468.2514926`.

Voelker, Simon, Nakajima, Kosuke, Thoresen, Christian, Itoh, Yuichi, Øvergård, Kjell Ivar, Borchers, Jan. "PUCs: Detecting Transparent, Passive Untouched Capacitive Widgets on Unmodified Multi-touch Displays". *ITS '13: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. St. Andrews, Scotland,United Kingdom: ACM, 2013, pp. 101–104. URL: `http://dx.doi.org/10.1145/2508468.2514926`.

Voelker, Simon, Øvergård, Kjell Ivar, Wacharamanotham, Chat, Borchers, Jan. "Knobology Revisited: A Comparison of User Performance between Tangible and Virtual Rotary Knobs". *ITS '15: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. Maderia, Portugal: ACM, Nov. 2015, pp. 35–38. URL: `http://dx.doi.org/10.1145/2817721.2817725`.

Voelker, Simon, Sutter, Christine, Wang, Lei, Borchers, Jan. "Understanding Flicking on Curved Surfaces". *CHI '12: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 189–198. URL: `http://dx.doi.org/10.1145/2207676.2207703`.

Voelker, Simon, Wacharamanotham, Chat, Borchers, Jan. "An Evaluation of State Switching Methods for Indirect Touch Systems". *CHI '13: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Paris, France: ACM, Apr. 2013, pp. 745–754. URL: `http://dx.doi.org/10.1145/2470654.2470759`.

Wacharamanotham, Chatchavan. "Drifts, Slips, and Misses: Input Accuracy for Touch Surfaces". Dr. Aachen, Germany: RWTH Aachen University, Feb. 2016. URL: `https://publications.rwth-aachen.de/record/567774`.

Wang, Feng, Ren, Xiangshi. "Empirical Evaluation for Finger Input Properties in Multi-touch Interaction". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: ACM, 2009, pp. 1063–1072. URL: `http://doi.acm.org/10.1145/1518701.1518864`.

Weiss, Malte. "Bringing Haptic General-Purpose Controls to Interactive Tabletops". PhD thesis. RWTH Aachen University, 2012.

Weiss, Malte, Voelker, Simon, Borchers, Jan. "The BendDesk Demo - Multi-touch on a Curved Display". *ITS '10: Adjunct Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. Saarbrücken, Germany: ACM, Nov. 2010, p. 317. URL: `http://dx.doi.org/10.1145/1936652.1936745`.

Weiss, Malte, Voelker, Simon, Sutter, Christine, Borchers, Jan. "BendDesk: Dragging Across the Curve". *ITS '10: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. Saarbrücken, Germany: ACM, Nov. 2010, pp. 1–10. URL: `http://dx.doi.org/10.1145/1936652.1936654`.

Weiss, Malte, Wacharamanotham, Chat, Voelker, Simon, Borchers, Jan. "FingerFlux: Near-surface Haptic Feedback on Tabletops". *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST '11. Santa Barbara, California, USA: ACM, 2011, pp. 615–620. URL: `http://doi.acm.org/10.1145/2047196.2047277`.

Weiss, Malte, Wagner, Julie, Jansen, Yvonne, Jennings, Roger, Khoshabeh, Ramsin, Hollan, James D., Borchers, Jan. "SLAP Widgets: Bridging the Gap Between Virtual and Physical Controls on Tabletops". *Proc. CHI '09*. Boston, MA, USA: ACM, 2009, pp. , 481–490. URL: `http://doi.acm.org/10.1145/1518701.1518779`.

Wellner, Pierre. "The DigitalDesk Calculator: Tangible Manipulation on a Desk Top Display". *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology*. UIST '91. Hilton Head, South Carolina, USA: ACM, 1991, pp. 27–33. URL: `http://doi.acm.org/10.1145/120782.120785`.

Wertheimer, Max. "Laws of organization in perceptual forms". *A source book of Gestalt Psychology* (1923).

Wigdor, Daniel, Jiang, Hao, Forlines, Clifton, Borkin, Michelle, Shen, Chia. "WeSpace: The Design Development and Deployment of a Walk-up and Share Multi-surface Visual Collaboration System". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: ACM, 2009, pp. 1237–1246. URL: `http://doi.acm.org/10.1145/1518701.1518886`.

Wigdor, Daniel, Shen, Chia, Forlines, Clifton, Balakrishnan, Ravin. "Perception of Elementary Graphical Elements in Tabletop and Multi-surface Environments". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: ACM, 2007, pp. 473–482. URL: `http://doi.acm.org/10.1145/1240624.1240701`.

Wigdor, Daniel, Shen, Chia, Forlines, Clifton, Balakrishnan, Ravin. "Table-centric Interactive Spaces for Real-time Collaboration". *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI '06. Venezia, Italy: ACM, 2006, pp. 103–107. URL: `http://doi.acm.org/10.1145/1133265.1133286`.

Wikimedia. *Henry VIII's writing desk, about 1525 V and A Museum no. W.29:1 to 9-1932*. 2008. URL: `https://commons.wikimedia.org/wiki/File:Henry_VIII's_writing_box.jpg`.

Wilson, Andrew D., Izadi, Shahram, Hilliges, Otmar, Garcia-Mendoza, Armando, Kirk, David. "Bringing physics to the surface". *Proc. Symposium on User Interface Software and Technology '08*. ACM, 2008, pp. 67–76. URL: `http://doi.acm.org/10.1145/1449715.1449728`.

Wimmer, Raphael, Hennecke, Fabian, Schulz, Florian, Boring, Sebastian, Butz, Andreas, Hussmann, Heinrich. “Curve: revisiting the digital desk”. *Proc. Nordic Conference on Human-Computer Interaction ’10*. ACM, 2010, pp. 561–570. URL: `http://doi.acm.org/10.1145/1868914.1868977`.

Wu, Mike, Balakrishnan, Ravin. “Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays”. *Proc. Symposium on User Interface Software and Technology ’03*. ACM, 2003, pp. 193–202. URL: `http://doi.acm.org/10.1145/964696.964718`.

Yu, Neng-Hao, Chan, Li-Wei, Lau, Seng Yong, Tsai, Sung-Sheng, Hsiao, I-Chun, Tsai, Dian-Je, Hsiao, Fang-I, Cheng, Lung-Pan, Chen, Mike, Huang, Polly, Hung, Yi-Ping. “TUIC: Enabling Tangible Interaction on Capacitive Multi-touch Displays”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’11. Vancouver, BC, Canada: ACM, 2011, pp. 2995–3004. URL: `http://doi.acm.org/10.1145/1978942.1979386`.

Zadow, Ulrich von, Buron, Sandra, Harms, Tina, Behringer, Florian, Sostmann, Kai, Dachselt, Raimund. “SimMed: Combining Simulation and Interactive Tabletops for Medical Education”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’13. Paris, France: ACM, 2013, pp. 1469–1478. URL: `http://doi.acm.org/10.1145/2470654.2466196`.

Zhai, Shumin. “What’s in the Eyes for Attentive Input”. *Commun. ACM* 46.3 (Mar. 2003), pp. 34–39. ISSN: 0001-0782. URL: `http://doi.acm.org/10.1145/636772.636795`.

Zhai, Shumin, Morimoto, Carlos, Ihde, Steven. “Manual and Gaze Input Cascaded (MAGIC) Pointing”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’99. Pittsburgh, Pennsylvania, USA: ACM, 1999, pp. 246–253. URL: `http://doi.acm.org/10.1145/302979.303053`.

Zimmerman, Thomas G., Smith, Joshua R., Paradiso, Joseph A., Allport, David, Gershenfeld, Neil. “Applying Electric Field Sensing to Human-computer Interfaces”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’95. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 280–287. URL: `http://dx.doi.org/10.1145/223904.223940`.

Index

Own Publications

Papers (Peer-reviewed, archival)

Hennecke, Fabian, **Voelker, Simon**, Schenk, Maximilian, Schaper, Hauke, Borchers, Jan, Butz, Andreas. “Simplifying Remote Collaboration Through Spatial Mirroring”. *INTERACT ’13: Proceedings of the 14th IFIP TC13 Conference on Human-Computer Interaction*. INTERACT ’13. Cape Town, South Africa: Springer Berlin Heidelberg, 2013, pp. 624–631. URL: `http://dx.doi.org/10.1007/978-3-642-40477-1_41`.

Nakajima, Kosuke, Itoh, Yuichi, **Voelker, Simon**, Thoresen, Christian, Øvergård, Kjell Ivar, Borchers, Jan. “PUCs: Detecting Transparent, Passive Untouched Capacitive Widgets on Unmodified Multi-touch Displays”. *Proc. IPSJ Interaction ’14: Proceedings of the Conference on Information Processing Society of Japan*. Information Processing Society of Japan, 2014, pp. 108–113. URL: `http://ci.nii.ac.jp/naid/110009867115`.

Voelker, Simon, Cherek, Christian, Thar, Jan, Karrer, Thorsten, Thoresen, Christian, Øvergård, Kjell Ivar, Borchers, Jan. “PERCs: Persistently Trackable Tangibles on Capacitive Multi-Touch Displays”. Englisch (eng). *UIST ’15: Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. UIST ’15. Charlotte, NC, USA: ACM, Nov. 2015, pp. 351–356. URL: `http://dx.doi.org/10.1145/2807442.2807466`.

Voelker, Simon, Corsten, Christian, Hamdan, Nur Al-huda, Øvergård, Kjell Ivar, Borchers, Jan. “An Interaction Model for Grasp-Aware Tangibles on Interactive Surfaces”. *ITS ’14: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. Dresden, Germany: ACM, Nov. 2014, pp. 279–282. URL: `http://dx.doi.org/10.1145/2669485.2669494`.

Voelker, Simon, Matviienko, Andrii, Schöning, Johannes, Borchers, Jan. “Combining Direct and Indirect Touch Input for Interactive Desktop Workspaces using Gaze Input”. *Proceedings of the 3rd ACM Symposium on Spatial User Interaction.*

SUI '15. Los Angeles, CA, USA: ACM, Aug. 2015, pp. 79–88. URL: `http://dx.doi.org/10.1145/2788940.2788949`.

Voelker, Simon, Nakajima, Kosuke, Thoresen, Christian, Itoh, Yuichi, Øvergård, Kjell Ivar, Borchers, Jan. "PUCs: Detecting Transparent, Passive Untouched Capacitive Widgets on Unmodified Multi-touch Displays". *ITS '13: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. St. Andrews, Scotland,United Kingdom: ACM, 2013, pp. 101–104. URL: `http://dx.doi.org/10.1145/2508468.2514926`.

Voelker, Simon, Øvergård, Kjell Ivar, Wacharamanotham, Chat, Borchers, Jan. "Knobology Revisited: A Comparison of User Performance between Tangible and Virtual Rotary Knobs". *ITS '15: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. Maderia, Portugal: ACM, Nov. 2015, pp. 35–38. URL: `http://dx.doi.org/10.1145/2817721.2817725`.

Voelker, Simon, Sutter, Christine, Wang, Lei, Borchers, Jan. "Understanding Flicking on Curved Surfaces". *CHI '12: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 189–198. URL: `http://dx.doi.org/10.1145/2207676.2207703`.

Voelker, Simon, Wacharamanotham, Chat, Borchers, Jan. "An Evaluation of State Switching Methods for Indirect Touch Systems". *CHI '13: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Paris, France: ACM, Apr. 2013, pp. 745–754. URL: `http://dx.doi.org/10.1145/2470654.2470759`.

Voelker, Simon, Weiss, Malte, Wacharamanotham, Chat, Borchers, Jan. "Dynamic Portals: A Lightweight Metaphor for Fast Object Transfer on Interactive Surfaces". *ITS '11: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. Kobe, Japan: ACM Press, 2011, pp. 158–161. URL: `http://dx.doi.org/10.1145/2076354.2076384`.

Weiss, Malte, **Voelker, Simon**, Sutter, Christine, Borchers, Jan. "BendDesk: Dragging Across the Curve". *ITS '10: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. Saarbrücken, Germany: ACM, Nov. 2010, pp. 1–10. URL: `http://dx.doi.org/10.1145/1936652.1936654`.

Weiss, Malte, Wacharamanotham, Chat, **Voelker, Simon**, Borchers, Jan. "FingerFlux: Near-surface Haptic Feedback on Tabletops". *UIST '11: Proceedings of the 24th annual ACM symposium on User interface software and technology*. Santa Barbara, CA, USA: ACM, 2011, pp. 615–620. URL: `http://dx.doi.org/10.1145/2047196.2047277`.

Book chapters (Edited, archival)

Borchers, Jan, Corsten, Christian, Möllers, Max, **Voelker, Simon**. "Von der Fläche zur Kurve: Multi-Touch auf beliebigen Oberflächen und Objekten". *Multi-Touch: Interaktion durch Berührung*. Xpert.press. Springer Verlag, Feb. 2014. Chap. 17, pp. 369–392. ISBN: 978-3-642-36113-5.

Posters (Peer-reviewed, non-archival)

Hamdan, Nur Al-huda, **Voelker, Simon**, Borchers, Jan. "Conceptual Framework for Surface Manager on Interactive Tabletops". *CHI '13 EA: Extended Abstracts of 2013 ACM SIGCHI Conference on Human Factors in Computing*. Paris, France: ACM, 2013, pp. 1527–1532. URL: `http://dx.doi.org/10.1145/2468356.2468629`.

Heller, Florian, **Voelker, Simon**, Wacharamanotham, Chat, Borchers, Jan. "Transporters: Vision & Touch Transitive Widgets for Capacitive Screens". *CHI '15: Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems*. Seoul, Republic of Korea: ACM, Apr. 2015, pp. 1603–1608. URL: `http://dx.doi.org/10.1145/2702613.2732707`.

Voelker, Simon, Corsten, Christian, Hamdan, Nur Al-huda, Øvergård, Kjell Ivar, Borchers, Jan. "An Interaction Model for Touch-Aware Tangibles on Interactive Surfaces". *CHI '14: Extended Abstracts of the 2014 ACM Conference on Human Factors in Computing Systems*. ACM, Apr. 2014. URL: `http://dx.doi.org//10.1145/2559206.2581273`.

Weiss, Malte, **Voelker, Simon**, Borchers, Jan. "BendDesk: Seamless Integration of Horizontal and Vertical Multi-Touch Surfaces in Desk Environments". *Adjunct Proceedings ITS '09. ACM*. Banff, Alberta, Canada: ACM, 2009. URL: `http://dx.doi.org/10.1.1.405.2239`.

Demos (Juried, non-archival)

Cherek, Christian, **Voelker, Simon**, Thar, Jan, Linden, Rene, Busch, Florian, Borchers, Jan. "PERCs Demo: Persistently Trackable Tangibles on Capacitive Multi-Touch Displays". *Proceedings of the 2015 International Conference on Interactive Tabletops and Surfaces*. Maderia, Portugal: ACM, Nov. 2015, pp. 389–392. URL: `http://dx.doi.org/10.1145/2817721.2823474`.

Voelker, Simon, Nakajima, Kosuke, Thoresen, Christian, Itoh, Yuichi, Øvergård, Kjell Ivar, Borchers, Jan. "PUCs Demo: Detecting Transparent, Passive Untouched Capacitive Widgets". *ITS EA '13: Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*. St. Andrews, Scotland, United Kingdom: ACM, 2013, pp. 325–328. URL: `http://dx.doi.org/10.1145/2512349.2514595`.

Voelker, Simon, Nakajima, Kosuke, Thoresen, Christian, Itoh, Yuichi, Øvergård, Kjell Ivar, Borchers, Jan. "PUCs Demo: Detecting Transparent, Passive Untouched Capacitive Widgets on Unmodified Multi-touch Displays". *UIST EA '13: Extended Abstracts of the ACM Symposium on User Interface Software and Technology*. St. Andrews, Scotland, United Kingdom: ACM, 2013, pp. 1–2. URL: `http://dx.doi.org/10.1145/2508468.2514926`.

Weiss, Malte, **Voelker, Simon**, Borchers, Jan. "The BendDesk Demo - Multi-touch on a Curved Display". *ITS '10: Adjunct Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. Saarbrücken, Germany: ACM, Nov. 2010, p. 317. URL: `http://dx.doi.org/10.1145/1936652.1936745`.

Workshop papers (Juried, non-archival)

Hennecke, Fabian, **Voelker, Simon**, Schenk, Maximilian, Schaper, Hauke, Borchers, Jan, Butz, Andreas. "PerspectiveTable: Blending Physical and Virtual Collaborative Workspaces". *Workshop on Blending Interaction*. Paris, France, Apr. 2013. URL: `http://hci.uni-konstanz.de/blendedinteraction2013/`.

Theses

Voelker, Simon. "BendDesk: Seamless Integration of Horizontal and Vertical Multi-Touch Surfaces in Desk Environments". Englisch (eng). Diploma Thesis. Aachen: RWTH Aachen University, June 2010. URL: `http://hci.rwth-aachen.de/materials/publications/voelker2010a.pdf`.

Typeset October 4, 2016