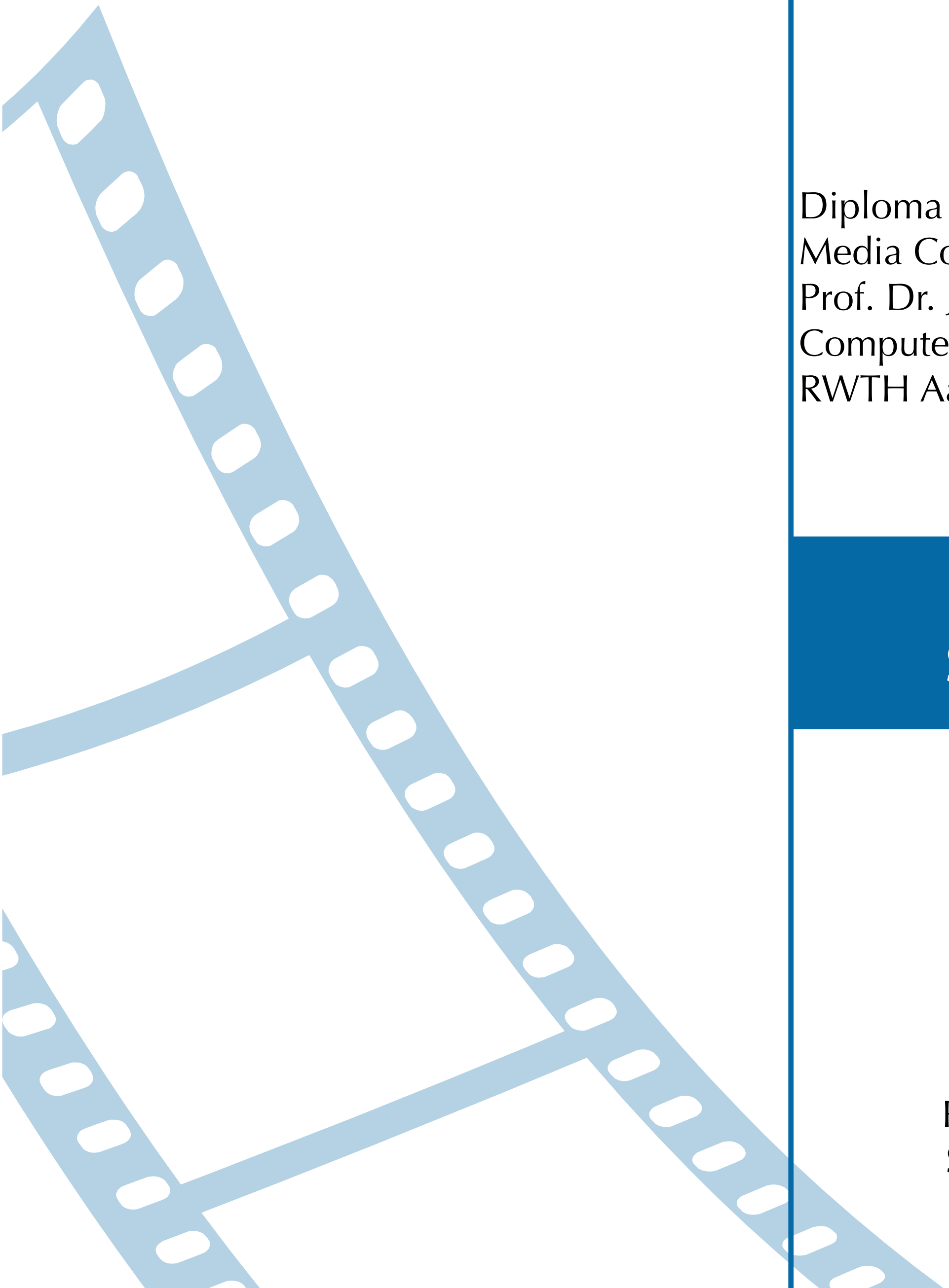
RWTH AACHEN UNIVERSITY

BendDesk:
Seamless Integration of Horizontal and Vertical Multi-Touch Surfaces in Desk Environments



Diploma Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Simon Völker

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Torsten Kuhlen

Registration date:	Jan 12th, 2010
Submission date:	Jun 19th, 2010

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, July2010
Simon Völker

Contents

List of Figures

List of Tables

Abstract

For most people from many different disciplines a desk is the main workspace. They use it for several different tasks that pose different requirements on the desk. Also, a regular desk is not only used as a workspace but also as a piece of furniture that can be used as a storage space for everyday objects. Most desks consist of a horizontal table and one or more vertical displays placed on the table. Both surfaces are used for various tasks but they are clearly separated by the fact that the horizontal table is used for real world objects and tasks while the vertical display is used for digital ones. In recent years a lot of research has been made in the field of interactive touch sensing surfaces and how they can be integrated into work life to improve the efficiency and comfort of working processes. Most of the existing systems combine a horizontal multi-touch surface with a vertical display to a desk environment. Despite all improvements these new technologies bring along, both surfaces are still spatially separated.

To solve this problem we present BendDesk, a multi-touch desk environment which seamlessly combines a vertical and a horizontal surface with a curve into one large interactive workspace. The idea of BendDesk is to offer users the possibility to work at one device that can be used for input and output interactions simultaneously. Due to the seamless combination of both surfaces the user can choose separately which one to use for each task. It is designed to replace a normal desk with all requirements users have such as placing everyday objects on it.
Because of the novel shape of BendDesk it is essential to explore basic interaction techniques on the curved surface, such as performance of dragging gestures and perception of lines. Therefore, we conducted four controlled experiments. The results show that dragging across the curve is significantly shorter than on both other areas. They also show that a flatter angle when crossing the curve leads to a longer average trajectory and a higher variance of trajectories. Additionally, we found that the curve influences the perception of lines crossing it. This result is perhaps the most important one because it affects enhancing operations like flinging.

Überblick

Für die meisten Leute - egal welcher Disziplin - ist der Schreibtisch der Hau tarbeitsplatz. Sie benutzten ihn für alle möglichen Aufgaben und haben mentsprechend alle möglichen Ansprüche an ihren Schreibtisch. Die meis Schreibtische bestehen aus einer waagerechten Arbeitsplatte auf der ein o mehrere Computerbildschirme gestellt werden können. Beide Arbeitsberei werden für alle möglichen Aufgaben genutzt, aber sie sind dadurch eindeu getrennt, dass man auf den Bildschirmen mit digitalen Objekten und auf Arbeitsplatte mit realen hantiert. In den letzten Jahren wurden Touchscre immer populärer. Die Forschung auf diesem Gebiet schreitet immer mehr ran und so wird auch überlegt, wie man diese neuartige Technologie für täglichen Gebrauch nutzen kann. Die meisten existierenden Systeme besteh aus einer waagerechten Multi-Touch-Oberfläche und einem normalen senkrech Bildschirm. Trotz allen Fortschritts in diesem Bereich besteht bei den Systemen i mer noch das Problem, dass beide Arbeitsbereiche räumlich voneinander getre sind.

Um dieses Problem zu lösen haben wir BendDesk entwickelt. Dabei handelt sich um einen Tisch, der aus einer waagerechten und einer senkrechten Mu Touch-Fläche besteht, die durch eine Kurve zu einer einzigen großen Arbeitsflä miteinander verbunden werden. Die Idee hinter BendDesk ist es den Benutz die Möglichkeit zu geben, an einem einzigen Gerät zu arbeiten, das gleichzeitig Eingabe- und als Ausgabegerät fungiert. Durch die nahtlose Verbindung bei Flächen kann der Benutzer selber entscheiden, welche Fläche er für welche A gabe nutzten möchte. Es soll den alten ”normalen” Arbeitsplatz vollständig ers zten und muss somit alle Ansprüche der Benutzter erfüllen. Durch die neuart Form von BendDesk ist es notwendig, grundlegende Interaktionstechniken wie wegung von Objekten und Wahrnehmung gerader Linien in der Kurve neu zu forschen. Dafür haben wir vier Experimente entworfen. Die Ergebnisse zeig dass die Bewegung von Objekten durch die Kurve signifikant langsamer ist als beiden anderen Flächen. Sie zeigen auch, dass ein flacherer Winkel bei der Bew gung von Objekten durch die Kurve zu einer längeren durchschnittlichen Traj torie und einer gröfleren Varianz dieser führt. Auflerdem fanden wir heraus, d die Wahrnehmung gerader Linien, die durch die Kurve verlaufen, durch diese b influsst wird. Das ist vielleicht das wichtigste Ergebnis unserer Tests, weil es w erführende Entwicklungen wie Flinging beeinflusst.

Acknowledgements

First of all, I want to thank Prof. Dr. Jan Borchers for introducing me in the field of HCI, for the opportunity to work on this great project, and to present is on a conference. Special thanks go to my supervisor Malte Weiß, for letting me work on his idea of BendDesk, and for the great support and feedback he gave me during my thesis. I would like to thank Dr. Christine Sutter who gave me very important feedback about my user studies, and who introduced me in how user studies are made. I also like to thank Prof. Dr. Torsten Kuhlen for being my second examiner.

Thanks to everyone who provided assistance of any kinds: Christian Remy, Simon Jakubowski, Florian Schwarz, HC Ebke, Mike Kremer, Jan-Peter Krämer, Thorsten Karrer, David Peters, Stephan Becker, Rene Herpertz, Torsten Palm, the folks form the Media Computing Group, all participants of my two user studies, and all those I forgot.

A special thanks to my brother Tim Hoepner who supported me with some parts of the prototype for free and gave me tips for the hardware setup.

Furthermore, I like to thank my family for supporting me during the whole thesis in several ways.

Last but not least, thanks to my wife Melanie Völker who supported me the whole time.

Conventions

Throughout this thesis we use the following conventions:

- The whole thesis is written in American English.
- Independently of the real gender of our users we will use ″she″ when referring to a single user.
- In **Chapter 6—“Dragging”** we use the following abbreviations:
    - p probability ($p = 0.05$ is called the significance level)

Chapter 1

Introduction

For many people the desk is the daily workspace, they use it in various ways for many different tasks. Most of these desks are separated into two different areas: A vertical area, represented by one or more vertical displays and a horizontal area, the actual table. Both areas offer specific benefits but also have their drawbacks. The vertical area is suitable for reading text, providing an overview of the workspace, or to communicate with remote users. The horizontal area is more appropriate for taking notes, drawing tasks, or just to store everyday objects like sheets of paper or coffee cups.

a normal desk consists of a vertical and a horizontal area

Both areas are separated from each other: the vertical area represents the digital world and the horizontal area is the real world. Therefore it is impossible to exchange documents or objects between both areas. Furthermore, they offer very different possibilities of interaction. While interacting with the vertical area can often only be done indirectly with a mouse or a keyboard, the horizontal area allows many different types of direct manipulation, e.g., using a pen for drawing tasks.

the vertical area represents the digital world, the horizontal one the real world

In recent years, many researchers have developed systems that connect both areas by transforming the horizontal area into the digital world. Ziola et al. [2007] introduced a system that uses the horizontal area as an additional display by projecting images on it. This transforms the horizontal area into a digital output device, but the only way to inter-

connecting both areas by making them digital act with the objects is to use a mouse or a keyboard, which removes the entire benefit of direct manipulation. Morris et al. [2008] placed a tablet PC on the horizontal area which allows to use pen, mouse, and keyboard.

combining a multi-touch surface with a display

Many other systems, like the system introduced by Coldefy and Louis-dit-Picard [2007], have combined a vertical interactive surface and a horizontal multi-touch surface to a desk environment. They benefit from the possibility to use direct manipulations techniques on the multi-touch surfaces, which is comparable to the interaction techniques on a real table.

both areas are spatially separated

All these systems have very large interactive surfaces that are spread over the vertical and the horizontal area. However, they all suffer the same problem: Both areas are not seamlessly connected, there are small gaps between them. According to the Gestalt law of closure by Chang et al. [2002] these gaps induce users to perceive both surfaces as two separated interaction areas. Additionally, other laws like the law of proximity could be violated, because objects that belong together are separated.

most methods not suitable for multi-touch systems

To reduce these problems and to increase the spatial continuity between both surfaces, many methods such as Portals by Everitt et al. [2006] or Repeating Patterns by Wigdor et al. [2006b] have been developed. However, all these methods are not suitable for multi-touch system because they are not designed for direct manipulation interactions. Especially for standard multi-touch interaction techniques such as dragging the gaps between surfaces are problematic because they disrupt these interactions.

BendDesk, a multi-touch desk environment on a bended surface

In this thesis we present BendDesk, a multi-touch desk environment that merges a vertical and horizontal surface into one seamlessly connected, touch-sensitive surface by a curve (Figure 1.1). BendDesk provides the user with a very large but still completely reachable interactive area. The combination of a vertical and horizontal area into one surface allows the users easily to move objects from one area into the other without changing the input techniques. So they can choose for each task which surface they want to use. The form factors of the system allow the users to use BendDesk such as a normal desk.

Figure 1.1: BendDesk seamlessly merges a horizontal and a vertical surface into one large interactive surface by a curve.

developing process and characteristics of BendDesk

This thesis has two main goals: The first is to outline the design and development process of BendDesk, from the first design considerations that had to be made, over the determination of the size of the system by a form factor prototype and the hardware setup of the final system to software algorithms that are needed for the input and output techniques of BendDesk. Furthermore, we describe how applications can be developed for BendDesk and how they can use input and output techniques of the system. The second goal is to explore the characteristics of BendDesk and if and how especially the curve influences basis tasks such as dragging and targeting operations.

In the following, we outline the structure of this thesis and give a short overview of each chapter:

Chapter 2—"Related Work" In this chaper, we present the related work that has influenced this thesis. It consists of four different areas: ″Interactive Desk Environments″ , ″Characteristics of Horizontal and Vertical Interactive Surfaces″, ″Multi Surface Systems″, and ″Non-Flat Interactive Surfaces″.

Chapter 3—"Design Considerations" In this chapter, we explain, which design considerations we had to consider before developing BendDesk.

Chapter 4—"Form Factor Prototype" In this chapter, we describe the hardware setup and the user test we conducted on the form factor prototype.

Chapter 5—"System Overview" In this chapter, we present the actual system from the basic hardware setup to the framework that is used for the visual output and recognizing input. Additionally we present how applications can use this framework to run on BendDesk.

Chapter 6—"Dragging" In this chapter will explore dragging operations on BendDesk and determine if there are any differences in dragging performance between the horizontal, the curved, and the vertical area.

Chapter 7—"Summary and Future Work" In the last chapter, we summaries the content of the presented work in this thesis. Finally, we give an outlook of the next steps: how BendDesk can be used in the future and which applications and studies can be done on BendDesk.

Chapter 2

Related Work

This thesis is influenced by several other publications. The most related ones explore multi-touch interaction on non-flat surfaces. Obviously, systems that combine vertical and horizontal touch displays are also very relevant. Additionally research on interacting with multi-touch system and technical foundations are important as well. In this Chapter we give a short overview over the papers that had a direct impact on this work. The publications are sorted in the following categories: ”Interactive Desk Environments” , ”Characteristics of Horizontal and Vertical Interactive Surfaces”, ”Multi Surface Systems”, and ”Non-Flat Interactive Surfaces”.

2.1 Interactive Desk Environments

Most normal desks consist of a vertical computer display placed on a table. In the past 20 years there where many research projects that created interactive desk environments that go beyond these classical desk environments.

Starfire is only a vision video

BendDesk is mainly inspired by the Sun Starfire vision video Tognazzini [1994] that anticipates a single-user desk working environment that combines a vertical and a horizontal surface. It was only a video prototype, because the main focus of the system was not to create a whole new system, but rather to show how a next-generation desk can look like (Figure 2.1). The whole table consists of one interactive surface with that the user can interact by many different input methods like touch or speech. The vision video shows not only the system itself but also how it supports the daily office work. Starfire became only a video vision because at that time developing such a new system was too expensive. However, it illustrates that an interactive desk environment could be a seamless combination of different surfaces.

Figure 2.1: The *Sun Starfire* system [Tognazzini, 1994].

normal table with digital objects

One of the early real interactive prototypes is the DigitalDesk Calculator by Wellner [1991]. This system projects digital documents by a projector that is positioned above the table onto a normal wooden desk. Users can interact with digital documents by touch and direct manipulation techniques. To recognize these touches they placed a camera above the table that observes the user's hands on the table. The problem of this system configurations is to determine the moment when the user touches an object. From its point of view the camera cannot 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. Furthermore, with this system more complex interactions like dragging or gestures were not possible.

In a journal article by Buxton et al. [2000] they explored the use of large displays in automotive design processes and how those could be improved with , e.g., multi-touch displays. They found out that a traditional drafting table could be replaced by a multi-touch system. Later in this paper they mentioned that each task in the automotive design process poses very different requirements at the system and that one interactive display for all tasks is not suitable. They proposed to use several interactive displays that are only used for highly specialized task.

interactive displays should only be used for highly specialized tasks

In a study by Wigdor et al. [2007a] they observed one user using a horizontal multi-touch table as his workplace over 13 months. He reported that he used the table not only as a computing device but also as furniture for placing everyday objects on it or as a conference table for small meetings. Furthermore, they showed that the use of bimanual interaction is utilized to save time for some tasks. They compared the user′s email written on the multi-touch display with his emails written on a normal PC. They compared, a.o., the length of the text and used words and found no significant differences. 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 insufficient for text entry.

multi-touch system is suitable as desk workspace

All systems and studies in this section show that a desk environment consisting of multi-touch surfaces could be possible. But as it is reported in the study by Ryall et al. [2006] the form factors such as the size, the angle, the height, and the arraignment of the system impacts the usability.

form factors of the system are very important

2.2 Characteristics of Horizontal and Vertical Interactive Surfaces

Horizontal and vertical interactive surfaces have certain benefits and drawbacks. Exploring these characteristics has recently received great interest in the research community.

horizontal surfaces are suitable for annotating tasks

vertical surfaces are suitable for getting an overview

Morris et al. [2007] conducted a study where they analyzed the characteristics of different surfaces for a reading and text annotating task. They compared, u. a., a *vertical* setup that consists of a mouse and a keyboard as input devices and vertical displays as output devices, with a *horizontal* setup consisting of two pen-enabled displays that are positioned horizontally on the table. The results showed that the *horizontal* setup supports an annotation task more than the *vertical* setup. However, they pointed out that using only the *horizontal* setup is very uncomfortable. Therefore, they proposed a system that consists of a vertical space and a horizontal space.

a pen sensing horizontal display combined with a vertical display

problems by interacting with the different devices

In a later paper Morris et al. [2008] added a pen sensing horizontal display to a normal desk environment with a vertical display. They compared this setup with one consisting of two vertical displays. Their results showed that the users had problems to position the horizontal display, a mouse, and a keyboard in a way that they can interact with all three devices in a comfortable manner at the same time. They also pointed out that most of the users had problems to observe both surfaces at the same time. Furthermore, they found that users perceived both surfaces as isolated areas that are not connected. Additionally they proposed that the horizontal screen should afford tilting to increase the comfort of using it.

tilted surfaces have benefits of vertical and horizontal surfaces

A tilted interactive surface is introduced by Muller-Tomfelde et al. [2008]. They described a vertical surface as a "public" space because displayed objects can be seen from a greater distance. In contrast they mentioned a horizontal display as a "private" space, since displayed objects can only be seen by people that are very close to the surface. They pointed out that the tilted surface is a trade off between these two, hence it is not as "public" as a vertical display but also not as private as a horizontal display. Furthermore, they conducted a mock-up study with the result that the tilted surface is preferred by users.

the perception of objects differs between vertical and horizontal surfaces

A study by Wigdor et al. [2007b] analyzed the perception of elementary graphic elements on vertical and horizontal surfaces. They found that the perception of these objects is different for both surfaces. On the vertical surface the perception of an object doesnot change for different posi-

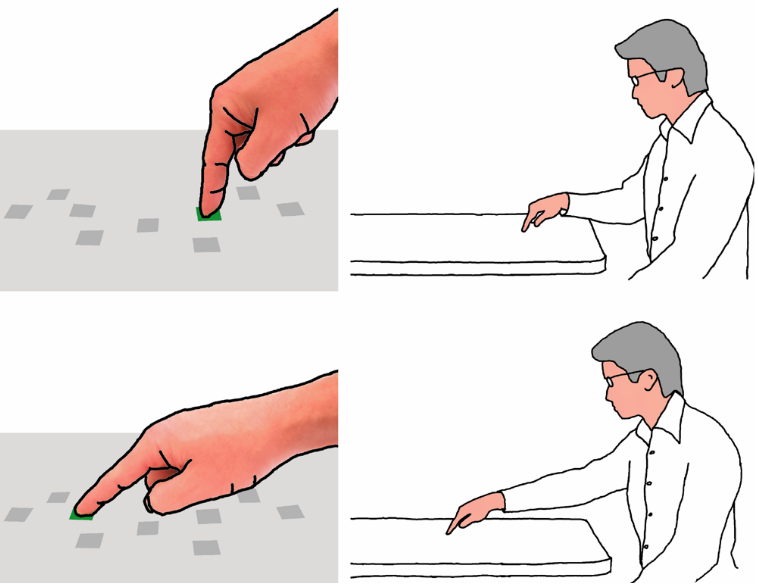


Figure 2.2: The hand pose for touching on the horizontal surface depends on the touch position [Forlines et al., 2007].

tions on the surface. In contrast, the perception of an object on the horizontal surface depends on its position. Because of the flat viewing angle on the horizontal surface the users perceived more distant objects as smaller. They also pointed out that in a mixed-display system objects that are displayed on the vertical surface should not be compared to objects on the horizontal surface.

touch accuracy on horizontal surfaces depends on touch position

Another study by Forlines et al. [2007] pointed out that these differences are not only in the perception of objects but also in the touch interaction with both areas. They mentioned that the hand pose on a vertical display is the same for each position on the display. However when a user touches a more distant position on the horizontal surface the hand pose is different to the pose when she touches an object that is closer, as shown in Figure 2.2. They pointed out that this effect leads to a different touch accuracy depending on the position of the touch. Hence, the size of the touch depends on the hand pose.

2.3 Multi Surface Systems

Creating a hybrid system, that consists of horizontal an vertical surfaces is mostly applied for collaborative and remote collaborative workspaces. While the horizontal surfaces (tabletops) are suitable for side-by-side or face-to-face collaboration, vertical surfaces can provide a better overview over shared data.

system offers data sharing via large displays

One of the first systems introduced by Rekimoto and Saitoh [1999] is a multi user desk environment that consists of a horizontal and a vertical interactive surface. The concept of this system is to allow users easily to share digital data among each other. They assumed that each user has its personal portable computer with personal data. To enable sharing of these data they introduced the *Hyperdragging* concept that allows users to drag objects from the laptop onto the horizontal and vertical surface by using the mouse or the touch pad of her laptop. If users drag an object over the edge of the laptop screen it is automatically displayed on the horizontal surfaces beside the laptop. This system allows to display and share data on large surfaces, but lacks the ability for users to interact with the digital objects directly.

horizontal surface as shared desktop, vertical one as video screen

A more recent system is the Digitable introduced by Coldefy and Louis-dit-Picard [2007] that combines a horizontal multi-touch surface with a vertical display to support remote collaboration. In this system the horizontal surface is the shared desktop and the interaction area. In contrast the vertical surface is only used as a video screen where the remote collaborator is shown.

the systems combine a horizontal multi-touch surface with a vertical display

Another system for face-to-face collaboration is the MultiSpace system by Everitt et al. [2006]. The MultiSpace is designed as a meeting room with a DiamondTouch (Dietz and Leigh [2001]) system placed on a table in the center of the room and a vertical interactive display on the wall. While the tabletop is designed as the central hub of the system that can be used by all collaborators, the wall is only used by one user to present data to other collaborators. The Vicat by Chen et al. [2006] and the WeSpace by Wigdor et al. [2009] are very similar approaches. They differ from the

MulitSpace system by the arrangement of both surfaces. They placed the table directly in front of the vertical display. Additionally in both system the interaction with the vertical display happens via the horizontal table or by remote. All these systems are designed for collaborative or remote collaborative tasks, where group of users interacts with the system at the same time.

Wigdor et al. [2006a] investigated how a vertical and a horizontal interactive surface should be arranged. Their study showed that the most preferred arrangement is the one, where the vertical display is directly behind the horizontal surface, so users can look at both surfaces at the simultaneously.

bazels between displays reduce the connectivity between them

But in all these systems both surfaces are spatially non-aligned which leads to the problem that users pervice these surfaces as two separated systems and not as one connected one. Bi et al. [2010] showed that small non-interactive strips between displays (bazels) already influence the users in their search strategies. The users tend to apply a display-by-display search strategy. Additionally they pointed out that bezels hinder the straight-tunnel steering performance across these bezels. The tunnel is separated into subtunnels and is not perceived as one steering task, but rather as a combination of multiple steering tasks.

To improve the sense of visual and spatial continuity and connectivity among displays that are clearly spatially separated, Wigdor et al. [2006b] proposed several techniques, such as the repeated patterns technique, already mentioned in the introduction. Another one, the *World in Miniature (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.

Other papers from Nacenta et al. [2008] and Baudisch et al. [2004] proposed two different techniques for efficient targeting operations that span over more than one display. But these techniques are only designed for the mouse and cannot be used for direct manipulation.

FLUX is a tiltable mulit-touch table

A completely different approach to combine a vertical and a horizontal area into one system is the FLUX system by Leitner et al. [2009]. It consists of one multi-touch surface 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 mentioned, 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.

2.4 Non-Flat Interactive Surfaces

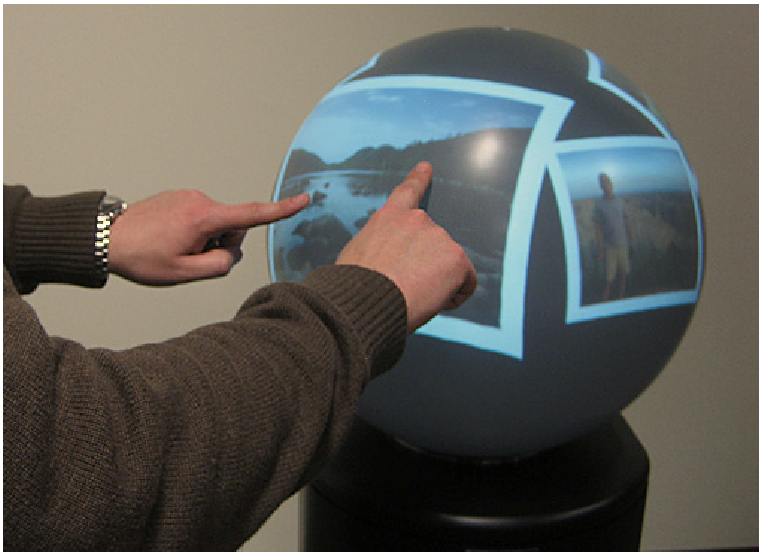


Figure 2.3: The Sphere system [Benko et al., 2008].

the Sphere system is the only existing touch sensitive non-flat surface

Almost all multi-touch systems are limited to planar interactive surfaces. The only existing system that uses a non-flat surface not only as output but also as input device is the Sphere system by Benko et al. [2008]. The Sphere system is a multi-touch sensitive spherical display, shown in Figure 2.3. The Sphere is a diffuse ball with a wide-angle lens at the bottom that allows projection on the spherical surface of the ball. For touch detection they used the Diffused Illumination (DI) technique introduced by Matsushita and Rekimoto [1997]. To enable the touch-sensing through the same lens they used for graphical output, they added an infra red (IR) camera, an IR filter, and an illumination ring as shown in Figure 2.4. This device is not designed as a working environment, it is designed as an *Walk-Up-and-Use* multi-user system. However, it proofs that using touch input on non-flat interactive surfaces is a suitable input method. In a later paper Benko [2009] gives an overview over different interaction techniques for non-flat surfaces. They conceded that flat surfaces have clear benefits over curved or other shaped surfaces. However, they demanded that multi-touch systems should fit into the users real world that is not only flat.

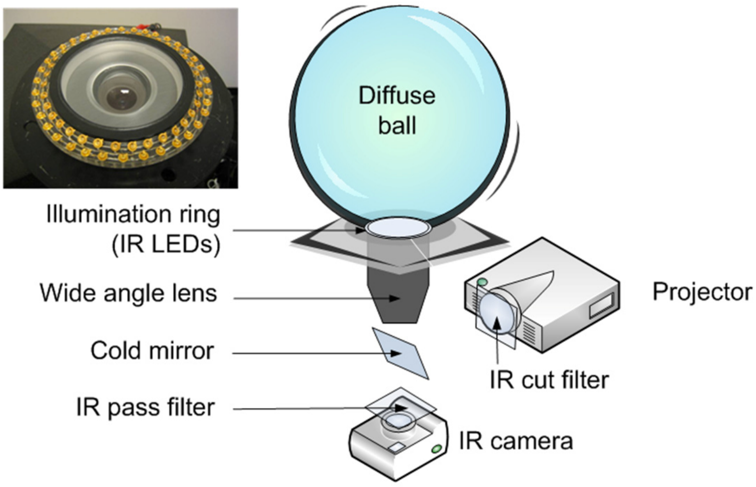


Figure 2.4: Schematic drawing of the Sphere hardware setup [Benko et al., 2008].

A lot of research has been done in the field of organic interfaces, where non-flat surfaces like curves or spheres are used as interactive displays. Holman and Vertegaal [2008] explored many different surfaces and freely deformable objects as potential interactive surfaces. One recent example of their visions is Paper Windows by Holman et al. [2005], that combines digital content with the affordances of paper. They projected digital content on an material, freely transformable like paper. Users can interact with the system by deforming the object or by hand gestures that are recognized by a Vicon camera system [Vicon2000].

Chapter 3

Design Considerations

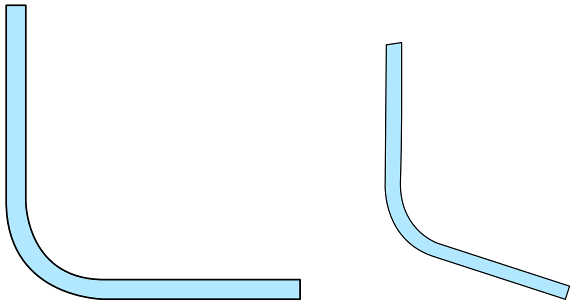
We envisioned a multi-touch desk environment that combines a horizontal and a vertical surface into one seamlessly connected large interactive surface. However, before building BendDesk many considerations about the ergonomic requirements and the hardware setup had to be made.

3.1 Ergonomics

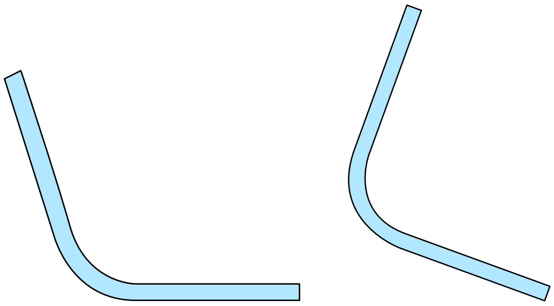
BendDesk is designed as an interactive desk environment where people can do their daily work. The form factors of the table should allow users to work on BendDesk as comfortable as possible. The first consideration that had to be made is about the basic form factors of the system. We decided to build BendDesk in a way the users will have enough space under the table to sit as comfortable as at a normal desk. Therefore, we placed the interactive surface in a height of 72 cm, according to the International Organization for Standardization [1998] norm for non-adjustable tables.

table height like a normal table

After that we had to decide what shape and which size the interactive surface should have. We decided to build the interactive surface out of one curved piece of acrylic. Hence, this is the only way to guarantee that the areas are seamlessly connected without any gaps between them. The



(a) non tilted shape (b) tilted horizontal area



(c) tilted vertical area (d) entire surface tilted

Figure 3.1: Possible shapes of the interactive surface.

one curved acrylic piece as interactive surface

continuous transition between the areas is crucial for basic multi-touch interactions such as dragging. Since there are several ways how the acrylic could be curved, as shown in Figure 3.1, we had to consider which of these shapes is suitable. Although, as mentioned in the related work chapter, there are evidences that tilted surfaces yield high acceptance for specific task; we used a shape without any tilted surfaces. Each of these shapes have their benefits and drawbacks; however, we chose 3.1(a) for many reasons.

non-tilted surface

BendDesk is intended to be a desk workspace. Most desks are not tilted because they are not only used as work spaces. They are also used as storage spaces for everyday objects (e.g. paper sheets, coffee cups, and books). We assumed that if the horizontal surface was tilted as shown in 3.1(b), users would not place any objects on it because they would be afraid that these objects could fall down. Additionally, a non tilted horizontal surface offers a more comfortable area to rest the arms on. In our case, we have an additional problem: If we tilted the horizontal surface, the position of the vertical area would be raised. That makes it impossible to reach this area in a comfortable manner. To solve this problem we could mount the entire surface in a lower height, but this would result in the problem that users cannot sit at BendDesk.

vertical tilting reduces the reachability

Tilting the vertical area (3.1(c)) would maybe increase the readability but also reduce the reachability in this area. A possible solution for this problem would be to position the vertical area closer to the front table edge, but this would reduce the size of the horizontal area resulting in a way too small workspace for proper use.

The fourth shape (3.1(d)) leads to the same problem as 3.1(b) where the user cannot place any objects onto the surface without worrying that they could slide down. Another problem is that the upper parts of the vertical area would be in a height that is over the user's head. That would make the interaction with these parts very exhaustive. Additionally in this shape the user's focus point would lie inside the curved area.

form factor prototype to determine the size of the interactive surface

To determine the size of the interactive area we conducted a test with a form factor prototype, described in Chapter 4. Furthermore, we assumed that the users would need a non-interactive area where they could rest the arms on without touching the interactive surface. This area should be a small strip at the front edge of the table. How useful it is and which size this area should have, we also investigated with the form factor prototype.

users should be able to sit at BendDesk

But, one of the most important form factors that we had to consider is that people like to sit very close at a table to increase the reachable workspace as much as possible. Therefore, they need a lot of space for their legs under the table. Since the horizontal area should be an interactive area where digital objects are displayed we had to solve the problem how we display these objects onto the surface without using the space under the table so that people can sit at it. How we solved this problem is described in the following section.

3.2 Hardware Considerations

As previously mentioned we had to decided how we display the interface on the interactive surface. Since we intended to use a curved surface as display we cannot use any type of LCD or plasma display, because such displays do not exist. Therefore, we had to use projectors that project on the surface. Due to the form factors of this system we had to use two projectors because one projector cannot cover the entire surface. We decided to place the projectors behind the acrylic surface (Figure 5.8) for three reasons: The first is that we used a special projector that has a very flat frustum which allows us to use rear projection on the horizontal surface and offers enough space for the user's legs. The second is that this solution reduces the size of the entire construction. If we had to use front projection we had to place the projectors over the surface, that would have increased the size of the system. The third reason is, if we had used front projection the user would occlude large areas of the horizontal area every time she interacts with the vertical area.

rear projection more suitable for BendDesk

For the table frame we decided to use plywood instead of other materials such as aluminum frames, because wood is much cheaper and it is simpler to customize, especially in the curved area.

3.2.1 Tracking Techniques

There are several multi-touch tracking techniques but just five techniques are suitable for such a large surface. The only possible tracking techniques are: Frustrated Total Internal Reflection (FTIR) introduced by Han [2005], Diffused Illumination (DI) introduced by Matsushita and Rekimoto [1997], Laser Light Plane (LLP), Diffused Surface Illumination (DSI) (Hilliges et al. [2009]), and Inverted FTIR introduced by Echtler et al. [2009].

LLP and DSI are not suitable because of the curved shape of our surface. The laser used for the LLP technique can only be used on planar surfaces. For using DSI we would need a special type of acrylic called EndLighten [Evonik2004]. But this acrylic type is only available in planar shape.

For inverted FTIR we would have to place one or more cameras in front of the table. To position these cameras that they cover the entire surface without any occlusion by the user is not possible. Therefore, inverted FTIR cannot be used for BendDesk.

DI could be a suitable technique but we decided to use FTIR instead. We assumed that DI would be too inaccurate for the touch detection, especially in the curve.

Chapter 4

Form Factor Prototype

BendDesk is designed to be a desk where people can work at. As for normal tables, the size and the form factors of BendDesk are very important, because the users should be able to use the system in a position as comfortable as possible. To guarantee that we had to make design decisions about the height of the table, the size of the vertical surface, the size of the curved surface, and the size of the horizontal surface. To create a real desk working environment we had to make sure that the users can sit at BendDesk like at a normal table. Since BendDesk is a multi-touch system where reachability is crucial, we also had to design the system such that the whole surface is reachable from the sitting position of the user.

form factors are very important for a desk workspace

Therefore, we decided to build a form factor prototype to determine these factors with a basic user test.

4.1 Requirements

The requirements for the form factor prototype are that the whole system is very flexible but also robust. The adjustabllity of this prototype is crucial to allow as much testing configurations as possible. The configuration changes have to be done very fast since the user has to conduct tasks in different setup configurations. The most important re-

the prototype should be freely configurable quirement is that the prototype has to simulate multi-touch interaction. It has to support direct interaction techniques on many different positions simultaneously.

4.2 Hardware Setup

The form factor prototype consists of a normal table with a height of 72 cm, which equates to the International Organization for Standardization [1998] norm for non-adjustable tables. The table has a width of 110 cm and a length of 80 cm.

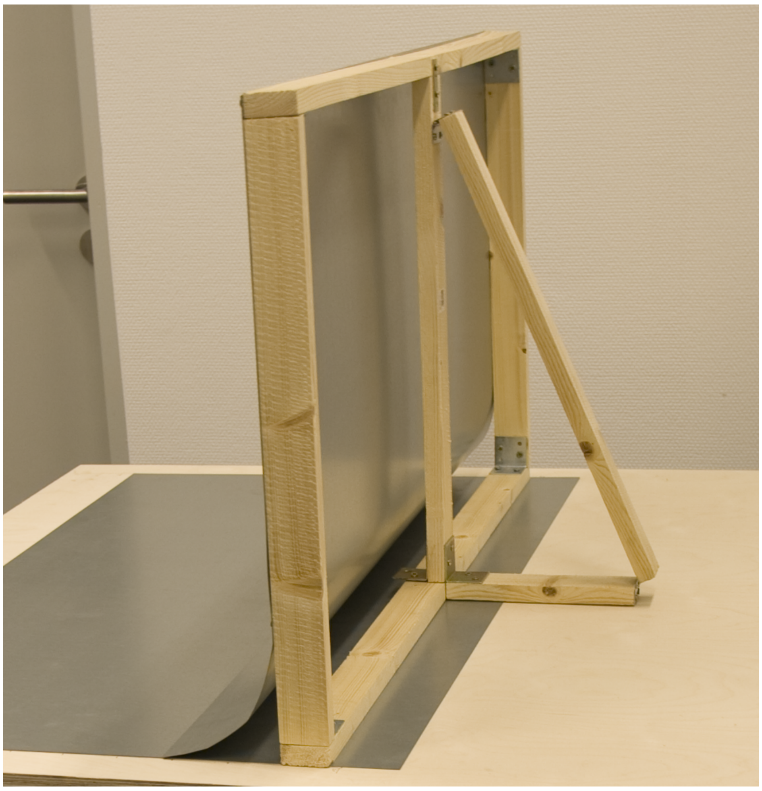


Figure 4.1: The vertical surface mounted onto a wooden frame.

A stainless steel plate with a size of 100 cm × 60 cm, is placed directly on the table. The position of this plate is freely adjustable. Another stainless steel plate with the same size is mounted on a wooden frame in vertical position on the table. The lower 10 cm of this plate are bended to a curve of 90°, as shown in Figure 4.1. The height of the vertical wooden frame is 45 cm, which is a little bit higher than the average eye level of a person sitting at a table. The position of this frame is also freely adjustable on the whole table. These two plates simulate the interactive surface of the system. Due to that fact that both plates are freely adjustable the size of the interaction area is variable. Additionally, the distance between the horizontal surface and the front edge of the table is freely adjustable. This space simulat the non-interactive area in front of the interactive area represented by the metal plates.

stainless steel plates as interactive prototype interace

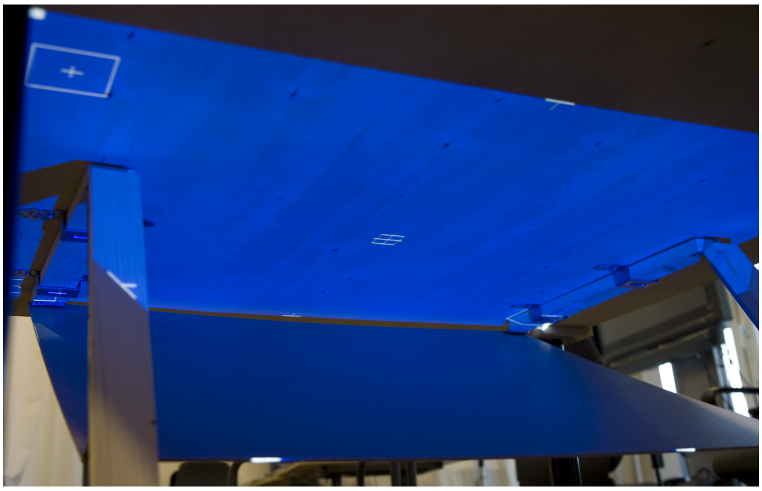


Figure 4.2: Substructure that simulates the projector cone.

Beneath the table a wooden construction is shaped accordingly to the projection cone of the projector that will display the interface onto the horizontal surface (Figure 4.2). This construction is adjustable in its angle and its position under the table. We set the angle of this construction to 25° because this is an angle where a user has enough space for her legs and this is the minimal angle that the projector of the final prototype can project onto the horizontal surface. The whole prototype is shown in Figure 4.3.

substructure simulates the projection cone

Figure 4.3: The form factor prototype.

4.3 Qualitative Evaluation

testing various positions of the vertical surface

The purpose of the form factor prototype is to determine the dimensions for the actual system. To get these data we had to create a user test that uses the entire surface and is easy to learn but also takes some time so the users can experience the system dimensions and decide if the system is comfortable, or not. With this test we wanted to determine the following questions:

1. Which size should the horizontal (a), the curved (b), and the vertical (c) area have, so that the whole system is easy to reach?

2. Which size should the hand rest area in front of the interactive area have?
3. At which angle and position of the substructure is there still enough space to sit at the table just like at a real table?

4.4 Experimental Design

creating a workflow diagram

The experiment consists of two very similar tasks. In both tasks the user has to create a workflow diagram about a specific activity described in a text. The first task is to create a diagram about finding and using an ATM. In the second task, the user has to create a diagram about arriving at an airport. Each user has to conduct both tasks which have nearly the same number of states and transitions. Each test has to be conducted in a different testing configuration. The testing configurations differ by the following two variables:

1. Distance of the vertical surface to the edge of the table in cm. The distance can be freely adjustable between 35 cm and 55 cm.
2. Size of the non-interactive area in front of the horizontal surface. The size can be in between 5 cm and 15 cm.

questionnaire for evaluation

After each task the user has to fill out a questionnaire about the form factors of the current testing configuration (see Appendix A).

4.4.1 Participants

A total of eight volunteers participated, six male and two females, between the ages of 22 and 30 with an average age of 25.75. All six male participants were from the field of computer science, while both women were from different disciplines. The body height of the participants was between 1.65 cm and 1.92 cm with an average of 1.79 cm.

4.4.2 Methods

using magnetic post-its as objects

At the beginning of the test the experimenter explained the system and the tasks. A text given to the users described which states and transitions the workflow diagram should have. The participants were provided with a pen and a lot of magnetic post-its of two different types: a quad shaped post-it with a size of 5 cm × 5 cm and a rectangle shaped post-it with a size of 7 cm × 2 cm (Figure 4.4). The quad-shaped post-its were for the diagram states and the rectangle shaped ones were for the transitions. The user's task was to create a state diagram as described in the text. First, the user had to label the post-its with a state name or an arrow for a transition and place it onto the table.

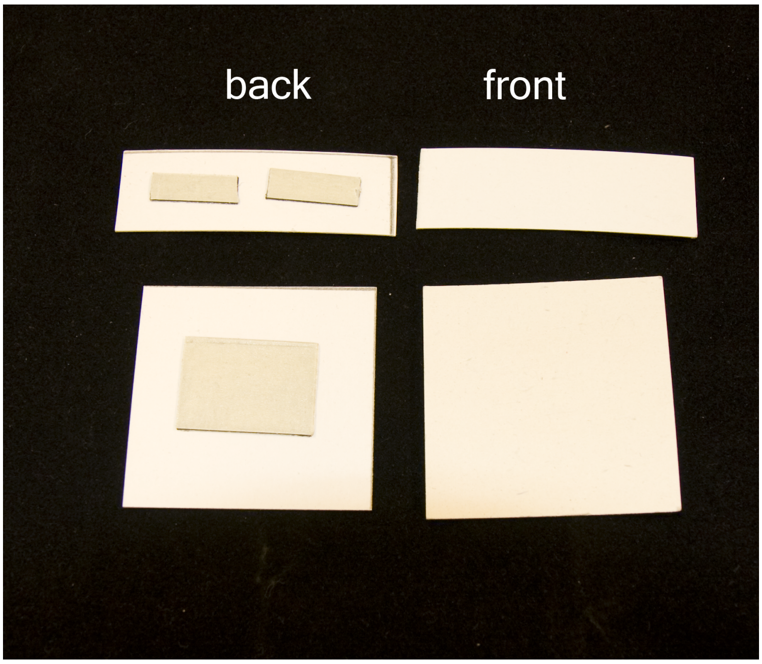


Figure 4.4: Magnetic Post-its. Front side shown on the right, back side with magnetic strips on the left.

Magnetic Paper Prototyping for Multi-Touch Systems

The most obvious reason why we used magnetic post-its is that they stick on the curved and vertical surfaces. Additionally, magnetic post-its are very suitable to simulate a multi-touch interaction experience. Similar to a basic multi-touch environment, users can translate or rotate the objects. Only interaction techniques like filling or scaling cannot be done. However, they could be simulated by a "Wizard of Oz" environment. The benefit of the magnetic post-its over normal post-its is that the user can really push or pull them across the entire surface. A normal post-it has to be picked up and put back on the new location. The whole dragging operation that has to be done on a multi-touch system cannot be done with normal post-its. With the magnetic post-its users can use bimanual interaction and are able to interact with several post-its at a time.

magnetic post-its are suitable to simulate a mulit-touch environment

4.4.3 Results

Most of the participants (six of eight) felt that the setup configuration where the distance to the vertical surface was 50 cm and the size of the non-interactive area was 8 cm was the most comfortable configuration. Seven of them strongly agreed that in this configuration all three different areas where completely reachable. Only one participant pointed out that the interaction with the vertical surface was a bit exhausting. In this configuration three participants said that the horizontal surface was almost too small, four mentioned that is was almost too big.

vertical surface in a distance of 50 cm is preferred

In the test configuration where the distance of the vertical surface to the edge of the table was 45 cm, 40 cm, and 30 cm, four of eight participants pointed out that the horizontal surface was too small. Three of the users even felt that it was much too small. Two commented to this configuration as very "claustrophobic" because they had this huge metal wall very close in front of them.

In the case where the distance of the vertical surface was 55 cm from the edge of the table all four participants, who vertical surface in 55 cm distance is not reachable

had to conduct one task in this test configuration, had a problem to reach the entire vertical surface. They also pointed out that the interaction with the vertical surface in that distance was much more exhausting than in all the other test configurations.

The table width was suitable for most of the participants. Three users pointed out that if the width was larger, the vertical surface would not be entirely reachable. Only one participant mentioned that the width of the interactive area could be larger.

hand rest area of 8 cm is preferred

Seven of the participants preferred the non-interactive area with a size of 5 to 10 cm. Four participants felt that it is very uncomfortable to sit at the table when the size of the non-interactive area is smaller than 8 cm. In the testing configurations all participants strongly agreed that they had enough space for their knees.

In the utilization of the areas the participants can be divided into two groups. One group wrote onto the post-its on the horizontal surface and dragged them upon the vertical surface. They distinguished the areas into a writing and a displaying area. The other group wrote onto the post-its most of the time on the horizontal surface but later on they used all three areas for writing.

4.4.4 Conclusions

vertical surface in 50 cm distance

The form factor prototype clearly showed that the most comfortable and usable position for the vertical surface is about 50 cm away from the table edge, in contrast to the results of the study by Toney and Thomas [2007], where the maximum reaching area was only 42cm for males and 38 cm for females. Almost all participants said that the test configuration was the best where the horizontal surface had a suitable size and the entire vertical surface was comfortable to reach. The results of this user test gave us the following answers to our initial questions:

- 1 (a): The size of the horizontal area should be 40 cm deep and 100 cm wide.
- 1 (b): The curved area should have an angle of 90° with a radius of 10 cm.
- 1 (c): The vertical area should be 45 cm high and 100 cm wide.
- 2: The best size for the non-interactive area in front of the horizontal area is 8 cm.

Additionally, the results show that it is possible to sit very comfortable at the table with a table substruction with an angle of 25°. This answers our third initial question.

Chapter 5

System Overview

The developing process of creating such a system like BendDesk consists of many different steps and tasks that have to be done. Starting system designs over hardware design and actual hardware construction to the software design and software implementation each of these steps has its own specific requirements and for each of these steps several decisions have to be made.

In this Chapter we present the entire developing process of the final system. It begins with the hardware setup followed by the software algorithms. In the final part of this chapter, we present how applications can be developed for the BendDesk.

5.1 Hardware Setup

BendDesk consists of two components

The hardware of BendDesk can be divided into two components: The front component, which is the interactive table and the back component which contains most of the electronic hardware like the projectors and the cameras. We decided to divide the system into these two components because of the size of the system. If we had to build the entire system in only one component we were unable to move the system because it is too heavy to carry and too large to fit through a normal door. Both components are connected

Figure 5.1: Side view of BendDesk.

by two hinges on both sides of the table. The hinges hold the components very tightly together so that they cannot shift against each other. Figure 5.1 shows a side view of both components connected by the hinges.

5.1.1 Front Component

the front component consists of an acrylic surface placed on a wooden frame

The front component consists basically of one acrylic surface that is mounted on a wooden frame. The size of the this acrylic surface is 104 cm × 104 cm and it is 1 cm thick. It is bended with a radius of 10 cm by an angle of 90°. As shown in Figure 5.2 this curve divides the surface into three seamless connected areas:

- **The horizontal area** with a size of 104 cm × 42 cm
- **The curve** with a size of 104 cm × 16 cm bended with a radius of 10 cm by an angle of 90°
- **The vertical area** with a size of 104 cm × 45 cm

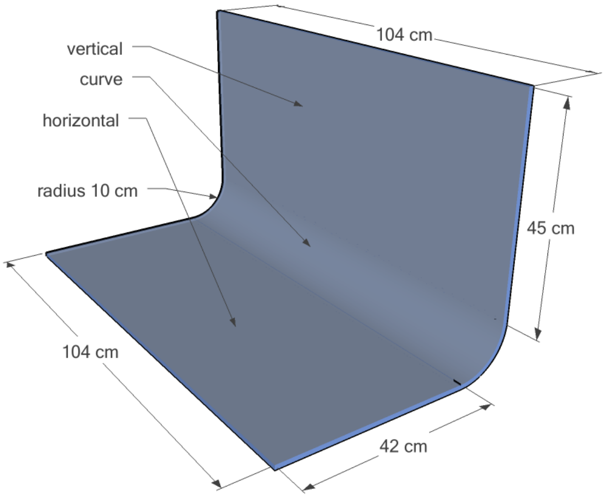


Figure 5.2: Acrylic surface.

The wooden frame consists of two side parts that are connected by three wooden bars as illustrated in Figure 5.3. The acrylic surface is mounted onto this wooden frame in a height of 72 cm. The overall size of the wooden frame is 112 cm wide, 130 cm high, and 63.5 cm deep.

side parts consist of two wooden plates

A side part of the wooden frame consists of an inner and an outer component. Both parts are custom made of 3 cm thick multiplex plywood plates, that have nearly the same shape as the acrylic surface. The inner component is used as the locating surface for the acrylic surface. The outer component is 1 cm higher than the inner one so it has the same height as the upper side of the acrylic surface. Figure 5.4 illustrates both components with their measurements.

plywood does not expand or shrink in different humidities

We use plywood for the front components because this type of wood does not expand or shrink very much exposed to different humidities. This is import because if the wood shrinks to much it will destroy the acrylic surface. The relative position of the acrylic surface to the camera or the projector could be changed as well, so that every time the air humidity changes the system has to be newly calibrated.

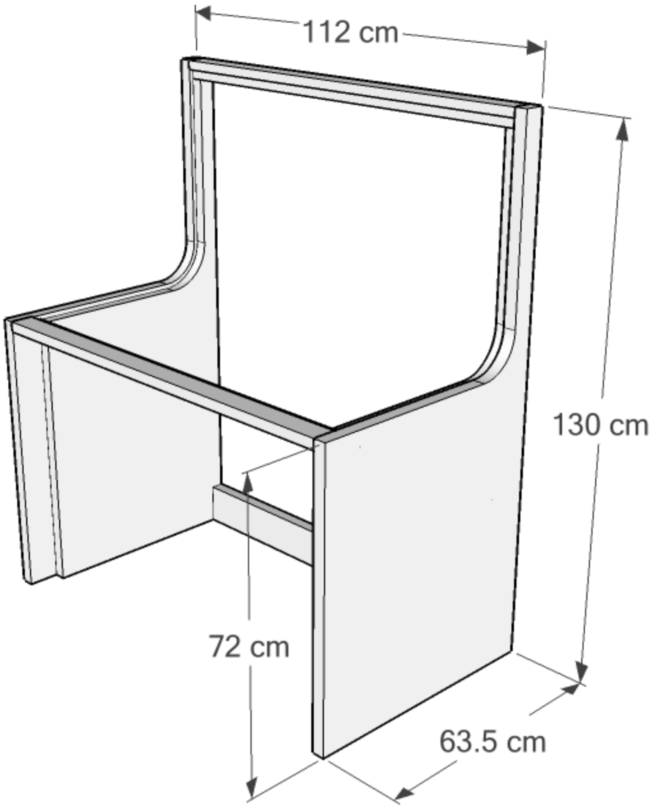


Figure 5.3: Wooden front frame.

The front side of the wooden frame is just one bar that connects both side parts in a height of 72 cm. It consists of a locating surface for the acrylic surface and an outer part that lines up precisely with the upper front side of the acrylic plate.

On the backside of the frame there are two wooden bars at the top and the bottom that connect the side parts with each other and stabilize the construction. Like the bar at the front of the frame the top bar has also a locating surface and an outer part that lines up precisely with the upper front side of the acrylic plate. The acrylic surface is placed on the locating surfaces so that on each of these location surfaces there is 1.1 cm space between the acrylic surfaces

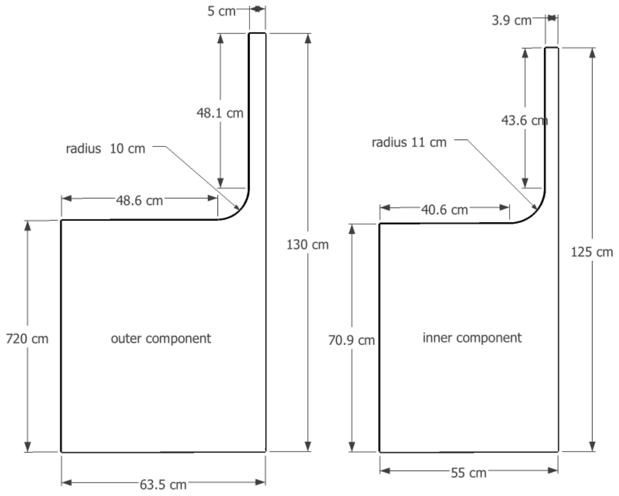


Figure 5.4: Side components of the wooden frame.

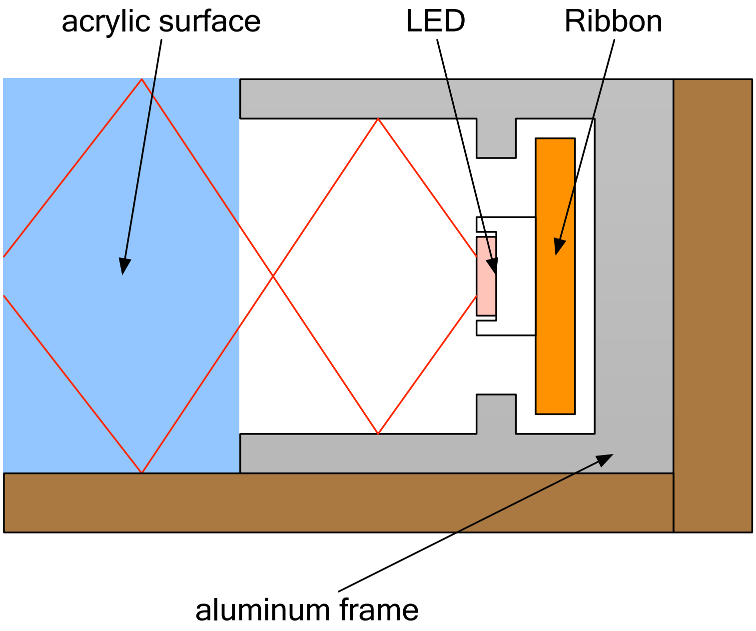


Figure 5.5: LED setup at the edges of the acrylic surface.

312 LEDs around the acrylic induce IR light

and the outer part of the frame. Due to the fact that we use Frustrated Total Internal Reflection (FTIR) to detect touches on the surfaces (introduced by Han [2005]), this space is used for the LEDs that feet infrared (IR) light into the surface. We use LEDs that are mounted on a 5 meter strip [H:EnvironmentalLights2006]. 312 LEDs with a distance of 1.2 cm between them are equally surrounding the acrylic surfaces. As illustrated in Figure 5.5 we place them into an aluminum frame to protect them from getting crushed between the acrylic and the wooden frame. Additionally the aluminum frame dissipates heat from the LEDs.

laths hold the acrylic surface in place

To use and to project onto the acrylic surface additional layers have to be placed over the acrylic surface. Therewith the acrylic surface and the layers cannot be shifted they are pressed onto the locating surface by laths that are mounted on top of the wooden frame as shown in Figure 5.6. Furthermore, the lath placed on the front edge of the table is the hand rest area. The laths reduces the visible size of the acrylic surface by 2 cm on each side that leads to an interactive surface with a size of 100 cm × 100 cm.

Figure 5.6: Laths on top of the acrylic surface that hold all layers in place.

Acrylic Surface Layers

silicone and Dura-Lar on top of the acrylic

Since the acrylic surface is transparent we need a defusing layer on the acrylic where the interface can be projected on. We use a sheet of Dura-Lar as diffusor; it is a very thin material that defuses just enough light to create a very sharp and colorful image on it. Regarding to its very fine surface structure it is suitable for touch interactions. However, to use FTIR we need an additional layer between the acrylic and the Dura-Lar layer (diffusor). Like most research groups we use a custom made silicone sheet. We tested over 30 different kinds of silicone. The final silicone is made by a technique called *Tinkerman's method* [nuigroup.com2006]. In this technique, standard silicone sealing material from a hardware store is spread with a fine pored paint roller onto the Dura-Lar sheet. This results in a very thin silicone layer (< 0.1 mm) that sticks on the Dura-Lar.

5.1.2 Back Component

projectors and cameras placed in the back component

The back component of BendDesk contains all electronic devices such as the projectors, the cameras, and the computer that runs the whole system. It consists of a wooden frame with an overall size of 114 cm wide, 130 cm high, and 113 cm deep. Its main function is to hold the projectors and the cameras in their exact positions. Furthermore, it hides the electronic device from the users. Figure 5.7 shows the entire back component. To reduce the incidence of IR light into the cameras the entire back component is covered with wooden boards.

The computer is a Mac Pro and it is placed behind the projectors as shown in Figure 5.1. It contains two extra Firewire bus cards to allow us to use the cameras with their maximum frame rates.

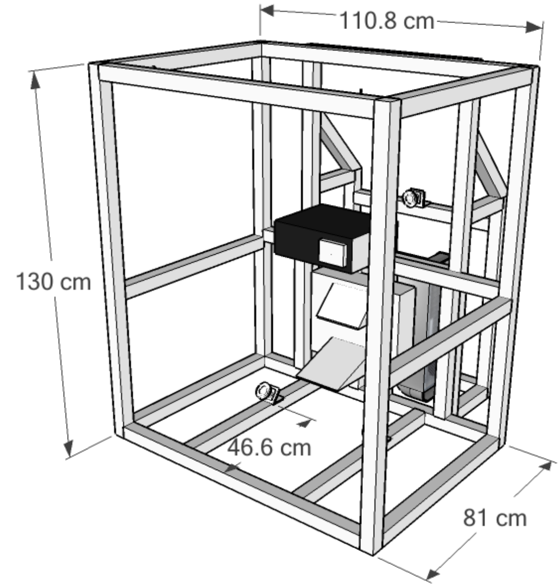


Figure 5.7: Back components.

Projectors

two short throw projectors

The BendDesk system renders the Graphical User Interface (GUI) with two short throw projectors as shown in Figure 5.8. The upper projector that projects on the vertical surface is an Optoma EX525ST Digital Light Processing (DLP) short throw projector. The projector is placed with a distance of 60 cm in a height of 55 cm behind the acrylic surface.

lower projector has a very flat frustum

The lower projector that renders the GUI onto the horizontal and the curved area is a NEC WT615 DLP short throw projector. According to its two aspheric mirrors this projector can project the GUI on the acrylic surface with a very flat frustum. The projector is placed with a distance of 62 cm in a height of 25 cm behind the acrylic surface.

Both projectors have a resolution of 1024 px × 768 px and they overlap on the vertical surface, that results in a total GUI resolution of 1024 px × 1024 px.

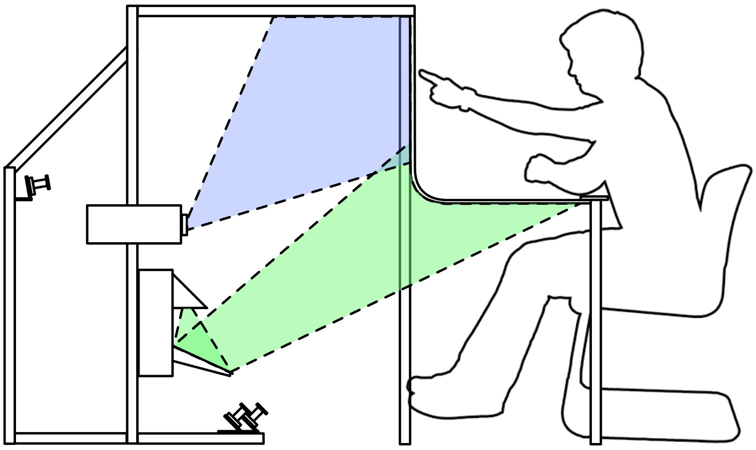


Figure 5.8: Placing of projectors and cameras.

Cameras

The used cameras are three FireFly MV cameras with attached IR filter, that filters all lights except IR light. All cameras are located in the back component. Each camera runs at 60 fps with a resolution of 640 px × 480 px. All three cameras observe different areas of the acrylic surface (Figure 5.9). Two cameras are located 46 cm behind the acrylic surface at the bottom of the back component. The third camera is positioned behind the upper projector in a height of 65 cm.

three cameras observe the acrylic surface

5.2 Software Algorithms

To use BendDesk as a desk environment the basic software has to do two major tasks: The first is to control how the GUI is rendered onto the surface. The second is to recognize the user's touches and to allow applications to react on these touches. However, both systems, the visual output and the touch tracking system have to be calibrated. This calibration process is done with an Agent application called *MultiScreen Agent* (Figure 5.10). In this section we present how both systems are calibrated, how they work, and how the *MultiScreen Agent* supports the use and the calibration

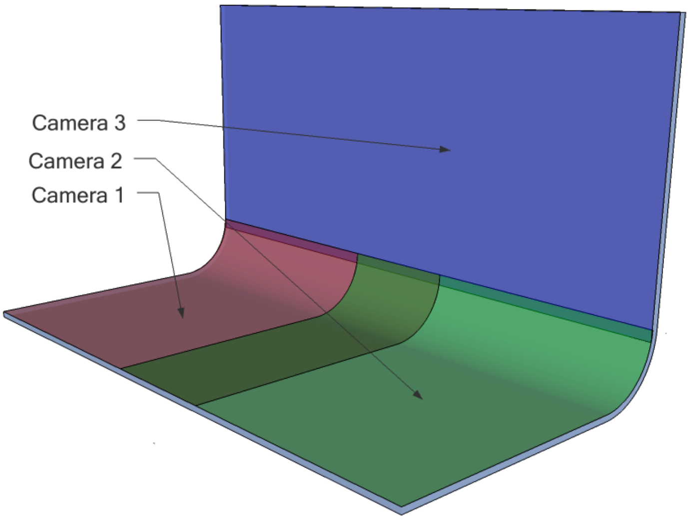


Figure 5.9: The acrylic surface and the tracking areas for each camera.

of BendDesk.

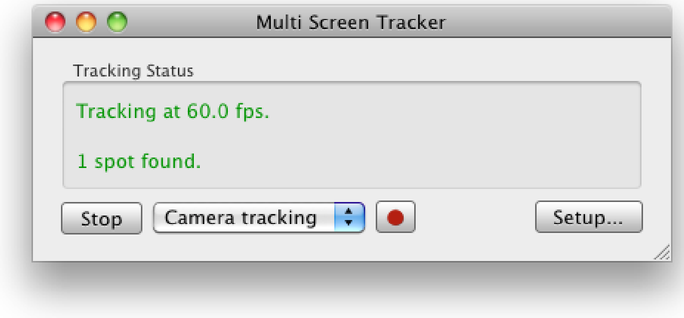
5.2.1 Visual Output

software has to compensate distorted projector images

Hence, our interactive surface is a curved non-planar surface the projectors can only project with a substantial distortion onto the surface. We have to compensate this distortion by perspectively distorting and scaling the projection in a way that the GUI displayed on the surface is distortion free. Therefore, we render the entire GUI into an offscreen buffer, in our case an *OpenGL Framebuffer object* [openGL.org2005]. Each of the projectors renders a part of the buffer object onto a Bicubic spline patch that compensates the respective distortion.

As illustrated in Figure 5.11 the *MultiScreen Agent* offers to select the display or the projector which should be calibrated. It also provides the option to change the size of the GUI and the calibration grid size, which is explained in the next section.

The size of the BendDesk GUI that is mapped isomorphi-



Tracking Status

Tracking at 60.0 fps.

1 spot found.

Figure 5.10: The MultiScreen Agent.

cally to the interactive area which is 1024 pixels (px) × 1024 pixels (px). The origin of this coordinate-system (pixel (0, 0)) is in the front left corner of the horizontal area. The pixel (1023, 1023) is mapped on the top right corner of the vertical area. A detailed mapping is illustrated in Figure 5.12. As shown in Figure 5.12 we define the *upwards* direction as a vector with a positive y-coordinate in GUI coordinates. The downward *direction* is defined analogously.

GUI resolution is 1024 px × 1024 px

Projector Calibration

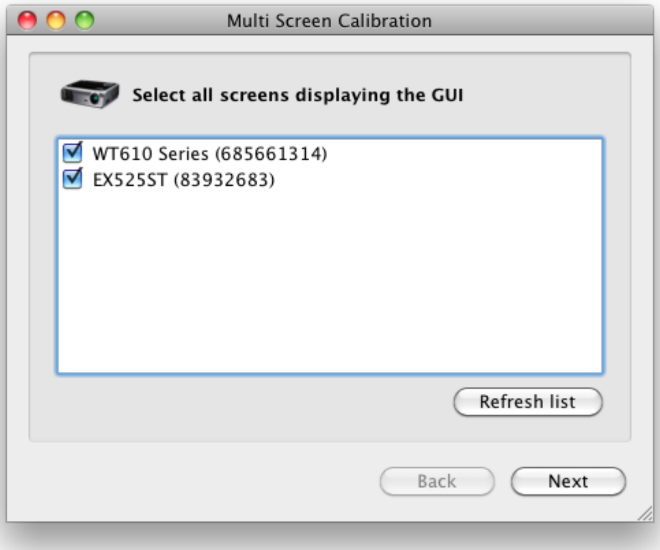
The spline patch for each projector is computed individually by a manual calibration process. We use almost transparent paper with an imprinted uniform grid of the size of 32 × 32 dots (Figure 5.13) that we place on the interactive area. Each printed dot with an index $(x, y) \in \{0, 1, ..., 31\}^2$, starting from the front left corner of the table, maps to a pixel position $P(x, y)$ in the GUI coordinate system:

a printed grid is used for the calibration process

$$P : \{0, 1, ..., 31\}^2 \rightarrow [0, 1024)^2 \qquad (5.1)$$

The calibration process results in a *projected* grid that exactly aligns with the nodes on the paper grid. That is,

$$D_i(x, y) = P(x, y) \quad \forall (x, y)\text{: frustum}_i(x, y) = 1 \qquad (5.2)$$



Select all screens displaying the GUI

☑ WT610 Series (685661314)
☑ EX525ST (83932683)

Figure 5.11: The MultiScreen Agent allows the users to select the screens they want to use to display the BendDesk GUI.

where $D_i(x, y)$ is the mapping of the *projected* grid dots to GUI coordinate system for each projector for $i \in \{1, 2\}$, defined analogously to $P(x, y)$. Since not all points of the paper grid are in the frustum of projector i we define the function $frustum_i(x, y)$, that indicates whether the point (x, y) is inside the frustum or not:

$$\text{frustum}_i(x, y) = \begin{cases} 1 & P(x, y) \text{ in frustum of projector } i \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

each projector is calibrated separately

Each of the projectors is calibrated separately. For the screen calibration process of a projector the *MultiScreen Agent* displays a grid with the same size as the paper grid (32 × 32) on the current projector. Each of this grid points is shown at a certain *screen coordinate* $S_i(x, y)$ with:

$$S_i : \{0, 1, ..., 31\}^2 \rightarrow [0, 1024) \times [0, 768). \tag{5.4}$$

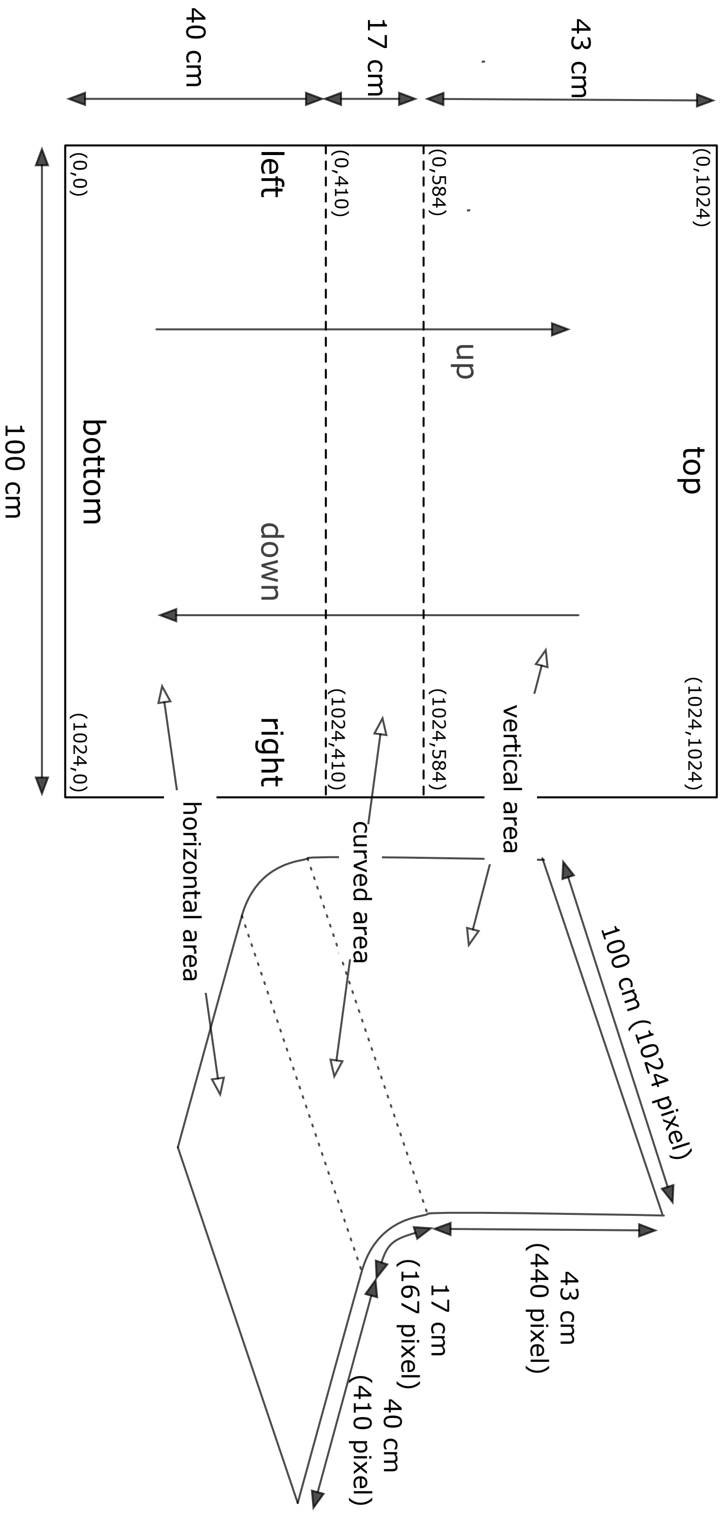


Figure 5.12: Mapping between the 3D interactive surface and the 2D interface.

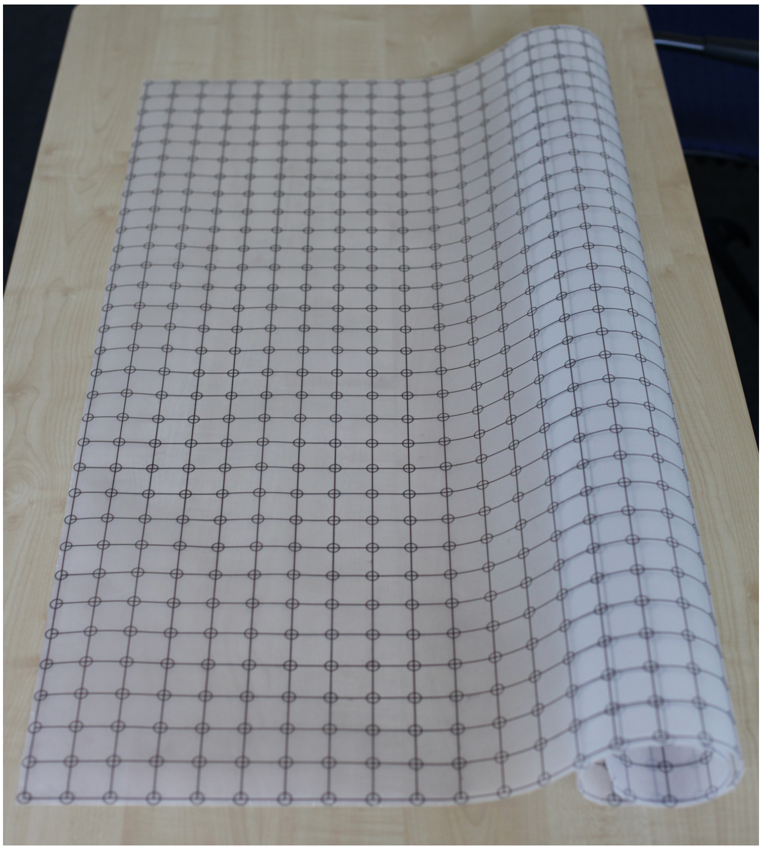


Figure 5.13: Screen calibration paper sheet with inprinted 32×32 grid.

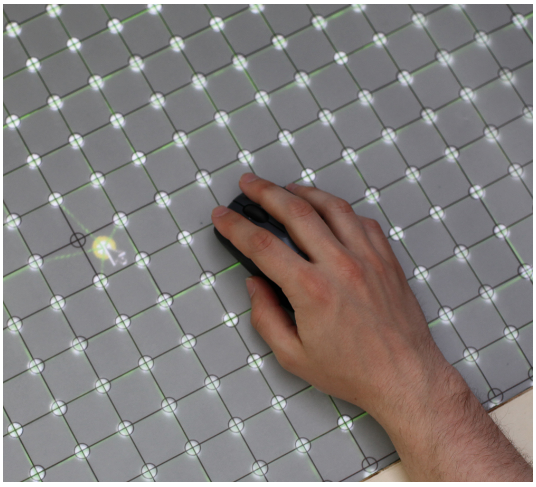


Figure 5.14: Calibration process for each screen.

To calibrate the screen the user deselects all grid rows and columns that do not map to the rows and columns of the paper-grid (defined by frustum$_i(x, y)$), which is in our case for the bottom projector the upper 16 rows and for the upper the first 18 rows. Thereafter, the user moves the *projected* grid points onto the paper grid points until they are at the exact same position ($P(x, y) = D_i(x, y)$) (Figure 5.14). Moving is done with the mouse for rough positioning and with the keyboard for exact positioning. To speed up this process the *MultiScreen Agent* offers additional functions:

calibration is done by moving the digital dots onto the imprinted dots

- moving the entire grid ,
- moving single rows or columns, and
- scaling the grid in both dimensions separately.

In this case scaling does not change the number of grid points, it only increases the distance between the points.

After a successful screen calibration a sub-grid contains all *projected* grid points inside the frustum of the projector i (frustum$_i(x, y) = 1$). The corresponding screen coordinates $S_i(x, y)$ are used as interpolation points of the bicubic spline patch, whereas the $D_i(x, y)$ are used as texture coordinates to render the specific part of the interface on the interactive surface. With this technique the *MulitScreen Agent* does not only support two projectors, it is scalable to many more projectors. Additionally it can be used for projecting distortion free interfaces on other non-planar or organic surfaces.

calibrated points used as interpolation points in the bicubic spline patch

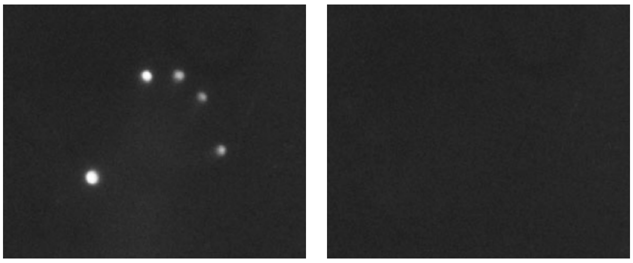
Rendering Pipeline

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

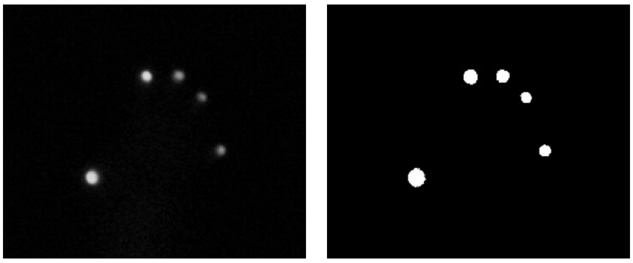
Then it reads for each projector or display, that is used to display the GUI on BendDesk, the spline patch from the calibration process and extracts a high resolution quad patch

GUI is rendered into an off screen buffer

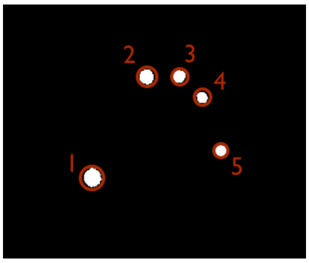
with texture coordinates from it. The texture coordinates map the GUI coordinates into the screen coordinates of the projector. The GUI is rendered into the off screen buffer that is bound first to the empty texture and then distributed to the projectors that outputs the texture on the respective spline patches. This process is hidden by the framework so an application is directly rendered into the GUI coordinate space without paying attention which projector is displaying which part of the GUI.



(a) raw image (b) background



(c) background substraction (d) threshold



(e) detected spots

Figure 5.15: Spot detection algorithm.

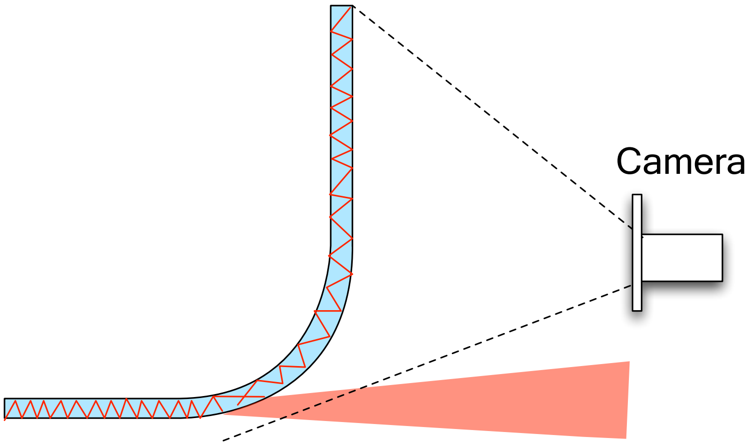


Figure 5.16: At the transitions of the curve the reflection angle is too flat so IR light leaves the acrylic surface.

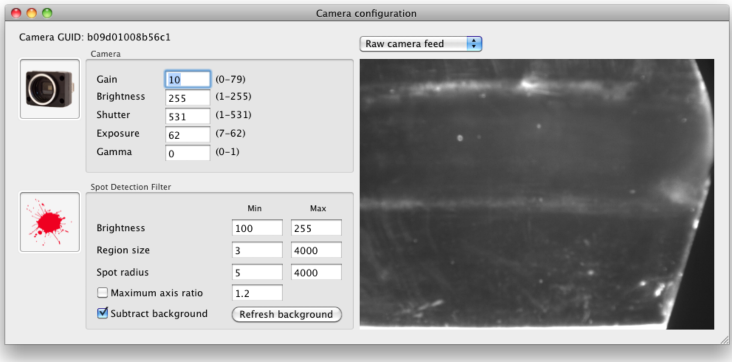
5.2.2 Tracking

As mentioned previously we are using FTIR as touch detecting technique. With the FTIR technique a finger touch on the surface results in a bright IR *spot* at the same position shown in Figure 5.15(a). To track this *spot* with the cameras we use a nearly straightforward algorithm: For each frame each camera records its visible area into an 8 Bit gray scale image. From this raw image we subtract a previously recorded background image (Figure 5.15(b)) to remove static *spots* and to compensate different lighting conditions in the acrylic surface. This is especially important for the camera that observes the vertical area. As shown in Figure 5.16 the IR light that is induced into the horizontal surface leaves the surface at the beginning of the curved area. Because at this particular point the angle by which the light hits the surface border is not steep enough to reflect the entire light back into the acrylic. The resulting image of the background subtraction is shown in Figure 5.15(c).

background subtraction

To determine if a pixel is bright enough to be a part of a *spot* we compute for each pixel if it is below or above a specific threshold. The results of this comparison are stored in a bitmap shown in Figure 5.15(d). By a frame rate of 60 frames per second (fps) and a camera resolution of 640 px × 480 px the system tests 55,296,000 pixel per second. To

brightness thresholding for each pixel



Camera GUID: b09d01008b56c1

Camera

Gain	10	(0–79)
Brightness	255	(1–255)
Shutter	531	(1–531)
Exposure	62	(7–62)
Gamma	0	(0–1)

Spot Detection Filter

	Min	Max
Brightness	100	255
Region size	3	4000
Spot radius	5	4000
☐ Maximum axis ratio	1.2	
☑ Subtract background		

Figure 5.17: The MutliScreen Agent camera preview offers many options for each camera.

detect larger groups of pixels that could be *spots* on this bitmap we use a simple detection algorithm based on a connected component analysis. For each of this component we compute the following values:

- **Identifier:** A unique ID to identify the spot.
- **Center:** The center of the *spot* is used as the exact position of the s pot.
- **Radius:** The radius is used to determine the size of the *spot*.
- **Main axis:** The main axis is used to determine the orientation of the *spot*.
- **Axis ration:** The axis ration is used to distinguish an elliptically shaped *spot* from other shapes.

In the final step of this *spot* detection algorithm we filter these components by size: We remove all components that are too small or too large to be a *spot* resulting of a finger touch on the acrylic surface. The remaining components are the detected *spots* illustrated in Figure 5.15(e). This detection algorithm is done for each camera independently. The

MultiScreen Agent provides the user with a preview mode for each step shown in Figure 5.15 for each camera. As shown in Figure 5.17 it offers the user additional options to configure each camera individually; not only values as the camera brightness or the shutter; furthermore, values for the threshold comparison and spot size for the filter.

transforming camera coordinates into GUI coordinates

After all *spots* for all cameras are detected we transform their coordinates from the camera coordinate system into the GUI coordinates system. Similar to the screen calibration mapping we use a Bicubic spline patch for this transformation. The exact description of this transformation is presented in the following Camera Calibration section.

Hence, we use three cameras to track the whole surface. The visible areas of the camera overlap to ensure a continuous tracking between these areas. If multiple *spots* are tracked by multiple cameras in these areas and their distance is smaller as a specific threshold, they will merge into one spot by averaging their coordinates. The threshold is configurable in the *MultiScreen Agent*.

predictive tracking algorithm to decide if the spot is a new touch or if it updates an old one

The next step in the touch detection process is to transform the *spots* into *touches*, that can be used by multi-touch applications. Furthermore, it decides if the *spot* is a new *touch* or the *spot* belongs to a moving *touch* from the previous frames. To ensure the registration of *touches* between following frames we use a predictive tracking algorithm which stores all touches from the previous frame in a list. The algorithm computes for each touch in the list if there is a spot in the current frame that updates this touch. For simplicity, we assume that the time difference between each frame is the same. We compute for each touch T at position p_0 two velocitys:

$$V_0 = p_0 - p_1 \tag{5.5}$$

$$V_1 = p_1 - p_2 \tag{5.6}$$

were p_1 is the previous position and p_2 is the position before that. With this both velocitys we compute the current acceleration A and the velocity V for T:

$$A = \frac{V_0 - V_1}{2} \tag{5.7}$$

$$V = V_0 + A \tag{5.8}$$

With the velocity we can now compute the predictive touch position p':

$$p' = p_0 + V \tag{5.9}$$

If there is a touch that is in a specific radius r around p' in the next frame we assume that this *spot* is the translated touch T and the data of this touch are updated. If there is a touch that has no updated spot the touch is removed from the list and assumed as ended. All spots that do not update an existing touch are transformed into a new touch.

This radius r is configurable in the *MulitScreen Agent*. The value of r is a trade between two situations: If the radius is very large it is possible to track a very fast finger movement but the minimal distance where the system can distinguish two touches from each other is also very large. If the value of r is very small, the minimal distance where the system can distinguish two touches from each other is very small, but tracking of a fast finger movement is not possible.

In practice we use a varied version of equation (5.9):

$$p' = p_0 + V/2 \tag{5.10}$$

This version of the equation strongly improves tracking of very fast translations and especially translations with very fast direction changes.

After all touches are detected for the current frame the *MultiScreen Agent* distributes all touches via the *NSNotificationCenter* [H:Apple2010], a build-in system that allows applications to send and receive data. We used for a *touch* the data structure created by Hafeneger et al. [2008]. In this data structure a touch has among other things the following important informations:

- **Start timestamp:** The timestamp of the first spot of this touch.

- **Timestamp:** The timestamp of the latest spot.
- **Touch phase:** The phase of the touch:
  - TouchBegin: The touch was created in the current frame.
  - TouchMoved: The touch was updated in the current frame.
  - TouchEnd: The touch was not updated in the current frame.
- **Start location:** The position of the first spot of this touch.
- **Location:** The position of the latest spot.
- **Previous location:** The position of the spot from the previous frame.
- **Touch information:** The spot informations as mentioned earlier.

Camera Calibration

The camera calibration process creates for each camera $i \in \{1, 2, 3\}$ a mapping from camera coordinates to the global GUI coordinates. When starting the camera calibration the *MultiScreen Agent* uses all calibrated screens to display an N × M uniform gird (in our case a 20 × 20 grid) with the coordinates $G(x, y)$, that covers the entire interactive surface.

camera calibration is done with a grid

In the first step of the calibration process the algorithm creates a mapping C_j from the GUI grid points to the camera pixel coordinates:

$$C_j : \{0, 1, ..., N\} \times \{0, 1, ..., M\} \rightarrow [0, 640) \times [0, 480) \quad (5.11)$$

where N and M represent the grid resolution for each dimension. The algorithm searches the camera pixel that matches with the GUI grid point. Therefore, it needs to compute $C_j(x, y)$ for each (x, y) that are visible for the camera. A point is visible for a camera if $\text{visible}_j(x, y) = 1$, where

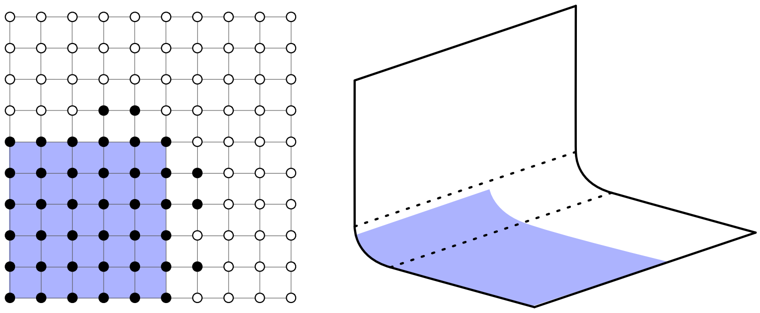


Figure 5.18: Extraction of spline patch to map from GUI to camera space. Left: largest rectangle containing visible points is extracted. Right: corresponding area on table.

$$\text{visible}_j(x, y) = \begin{cases} 1 & G(x, y) \text{ is visible from camera } j \\ 0 & \text{otherwise.} \end{cases} \tag{5.12}$$

In the camera calibration process all three cameras a calibrated simultaneously. Each of the grid points is successively highlighted, starting in the front left corner of the horizontal surface. The user has to touch each of this highlighted grid points $(\bar{x}, \bar{y})$ and then press a button on the keyboard to confirm that she touches the highlighted point. The algorithm detects this point and stores its position $C_j(\bar{x}, \bar{y})$ and which camera can see this spot $visible_j(\bar{x}, \bar{y})$.

the algorithm creates a visibility map for each camera

The result of this process is a visibility map for each camera that consists of all grid points (x, y) where $\text{visible}_j(x, y) = 1$. From this visibility map we compute the largest rectangle that consists of only visible grid points by solving the *Maximum rectangle problem* with the algorithm presented by Naamad and Lee [1984]. The figure 5.18 illustrates the maximum rectangle and its position on the acrylic surface. The extracted points and their values for $C_j(x, y)$ and $G(x, y)$ represent the interpolation points of the bicubic spline Patch $\mathcal{P}$ that maps the GUI coordinates to the camera coordinates for camera j.

But we need a mapping that allows us to determine the GUI coordinate to a given camera position. Therefore, we need to inverse the mapping C_j. The algorithm computes the mapping C_j^*:

$$C_j^* : \{0, 1, ..., 639\} \times \{0, 1, ..., 479\} \rightarrow [0, 1024)^2 \qquad (5.13)$$

by evaluating the bicubic spline patch $\mathcal{P}$ with high sampling rate. Each sample GUI coordinate is stored at the target camera position in C_j^*. This provides the system with a discrete inverse mapping for each camera j, that allows the system to read the GUI coordinate for each camera coordinate. The evaluation of the patch $\mathcal{P}$ is not done with the real GUI coordinates but between 0 and 1. The algorithm has to transform the resulting points of C_j^* into the GUI coordinates with respect to the position of the patch $\mathcal{P}$ in the GUI coordinate system. Hence, both coordinate systems do not have the same size, this mapping can result in jittering. To avoid that we use bi-linear interpolation for this lookup.

All calibration data of both calibration processes are stored by the *MultiScreen Agent*. Therefore, both calibration processes have only to be done once, if the camera position or the projector position is not changed.

5.3 Software Architecture

In the previous section we presented how both systems, the visual output and the multi-touch tracking, work and how they are implemented. In this section we present how both systems can be used to create applications for BendDesk.

5.3.1 Rendering

As already mentioned the rendering process can be hidden so that an application can directly render into the GUI space. Thereto the application just like the framework can load all information from a file provided by the *MulitScreen*

applications can read the texture coordinates from a MultiScreen Agent file

Agent. It contains the entire GUI size, the displays or projectors that are used for rendering the GUI and for each of the screens the spline patch with the specific texture coordinates. For each of these screens the application creates an OpenGL full screen view with a shared context so that all views can share their textures and other buffer objects. So, the texture that contains the GUI has only to be rendered by one view for each frame. All other views have only to map this texture onto their specific spline patch. A off screen buffer can be used to render into the texture. We used a *OpenGL Framebuffer object* but other buffers such as a vertex buffer can also be used. The render mechanism into this buffer is basically the same as rendering into a normal front or back buffer.

5.3.2 Touch Detection

applications receive touches through the NSNotificationCenter

As already mentioned the *MultiScreen Agent* distributes all touches to *NSNotificationCenter* [H:Apple2010] with a specific name. Each application that wants to use this touches can receive this touches by sending a request with the specific name to the *NSNotificationCenter*. From that point on if the *NSNotificationCenter* will receive touches from the *MultiScreen Agent* they are forwarded to the application. With this technique many applications can receive the touches at the same time without hinder each other.

5.3.3 SLAP Framework

To create applications we modified an OpenGL version of the SLAP Framework that is based on the work of Wagner [2009]. Conceptually the OpenGL version of the SLAP Framework differs only in the fact that it uses OpenGL as rendering framework. To use the BendDesk rendering pipeline we had to subclass the *SLAPView* class which does all rendering operations in the framework. We added the *Framebuffer object* and the texture as described above. To enable the framework to receive the *touch event* from the *MultiScreen Agent* we had to add a class that lists to the *NSNotificationCenter* and forwards the *touches* to the *SLA-*

PUITK class. Since we used the same data structure for the touch events we did not change the data structure of the *touches*.

With this changes the Framework can be used on BendDesk just like on normal multi-touch systems, except the widget detection. Since BendDesk does not use DI the tracking of widgets is not possible. We only used the framework GUI objects and its touch mechanisms. The only GUI objects that we could not use was the *SLAPMovie* object because we had to use the 64 bit version of the Framework and the *Apple quicktime framework* does not support this.

Chapter 6

Dragging

BendDesk differs from other multi-touch systems because it consists of three seamlessly connected areas. Additionally, BendDesk introduces a curved area as a touch sensitive interactive surface. After designing and building the system we wanted to explore how users can interact with it. The central questions we wanted to answer were:

- Are there any differences in the way users interact with the three different areas?
- How does the curved area perform compared to the two planar areas?
- Are there any design implications that have to be considered when developing applications for the BendDesk system?

focus on dragging operations

However, before applications can be developed for the BendDesk system, basic interaction techniques like dragging have to be analyzed. We decided to focus on basic dragging operations because they are some of the most frequently used gestures on a table. Although pointing is the most basic task for multi-touch systems, we decided to analyze dragging since it is most severely affected by the form factors of the system. However, pointing needs to be investigated which is beyond the scope of this thesis. We will point out this issue in Section 7.2.

four different user tests

In this chapter, we present four different user tests where the users have to perform basic interaction tasks on BendDesk. The purpose of the first user test was to analyze dragging operations on each of the three areas of the system and how they differ from each other. In the second test we explored dragging operations that involved all areas. In the third study we analyzed how the form factors influence the aiming accuracy on the system. At last, the fourth study was a qualitative test where the participants had to conduct basic dragging operations.

6.1 Participants

18 participants conducted all tests

We conducted all four user tests with 18 voluntary participants: two females and 16 males between the ages of 24 and 32 with an average age of 26.9. 15 of them were computer scientists, two were schoolteachers, and one was a mechanical engineer. After the tests, we raffled off a 20 Euro gift coupon among all users despite their specific performance.

6.2 Test Conditions

same user test order for each user

All participants conducted the test under the same lighting conditions in the same room guided by the same experimenter. The room is shown in Figure 6.1. Each participant had to conduct the user tests in the same order. They had to run each of the first three tests four times. Prior to each new test they had the chance to explore the test within a training trial. The order was as follows:

- **User test 1:** One training phase followed by two actual runs.
- **User test 2:** One training phase followed by two actual runs.
- **User test 3:** One training phase followed by two actual runs.

Figure 6.1: Experimental room.

- 2 minutes break
- **User test 1:** Two runs.
- **User test 2:** Two runs.
- **User test 3:** Two runs.
- **User test 4:** One run.

To increase the touch tracking, which is done via infrared cameras, we darkened the room with black cloth in front of the windows, such that no infrared light form the outside is reflected into the cameras. For the implementation of all user tests we used the modified version of the *SLAP Framework* as described in Section 5.3.3

user tests conducted in a darkened room to increase tracking performance

6.3 User Test 1

In this test we investigated the dragging performance on each of the different interactive areas of BendDesk individually. We compared the performance of dragging operations across the curve to dragging on the vertical and

differs the dragging performance between the areas?

horizontal areas. Performance measurements included the dragging duration, the length of the movement trajectory, and the deviation of the movement trajectory from the direct line between the objects. We hypothesized the following outcomes:

- H1 *(horizontal vs. vertical)*: Dragging (a) duration and (b) trajectory are shorter on the horizontal area than on the vertical one.
- H2 *(planar vs. curve)*: Dragging operations on planar areas are shorter in (a) duration and (b) length than operations crossing the curved area.
- H3 *(down vs. up)*: The trajectory is (a) temporally and (b) spatially shorter when moving upwards in GUI coordinates than when moving downwards.
- H4: The (a) average and (b) maximum deviation of the movement trajectory from the direct line on the planar areas are shorter than on the curved area.

6.3.1 Experimental Design

dragging operations in each area

An object and a target were shown on BendDesk. Both of them were vertically aligned with a distance of 150 pixels (about 14.64 cm). The user's task was to move the object into the target as fast as possible by dragging it into the target. This task had to be performed on each of the three interactive areas which was the first independent variable named *area* with three conditions:

(1) Dragging on the vertical area,

(2) dragging across the curve, and

(3) dragging on the horizontal area.

The second independent variable was the direction in which the user had to drag the object. The variable *direction* consists of two conditions:

(1) Dragging downwards, and

(2) dragging upwards.

This made a total of six conditions. For each of these conditions, the participants had to conduct the dragging operation for seven *x-positions*. The *x-positions* were uniformly distributed over the entire horizontal-axis of the BendDesk system. The order in which the six conditions and the *x-positions* had to be processed was randomized. The randomization was the same for all participants. This resulted in a total of 42 dragging operations for each *trial*. Each participant had to conduct four trials, two for the dominant and two for the non-dominant hand. This yielded a total of 168 dragging operations for each participant. Figure 6.2 illustrates the experimental design.

seven dragging operations per area

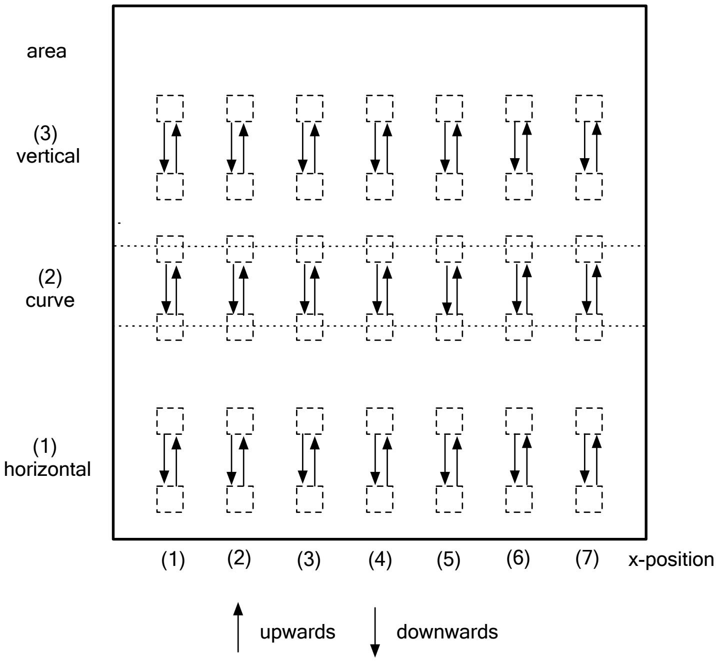


Figure 6.2: Experimental design of the vertical dragging test.

To verify our hypotheses we recorded the following dependent variables:

Duration: The time interval from touching the object to releasing it at the target position.

Distance: The length of the dragging trajectory, measured by the sum of the distances of each pair of successive sample points in the movement trajectory.

Overshoots: The number of times where the target area has been reached and left again without releasing the object.

Erros: The number of times where the user has lost the object and has to touch it again.

Maximum deviation: The maximum distance of the object to the direct line.

Average deviation: The average distance of the object to the direct line.

Variance of deviation: The variance of all distances of the object to the direct line.

6.3.2 Methods

the task is to drag an object into a target

For each dragging operation a white quad (the object) with a red border and a white empty frame (the target) were displayed. Both, the target and the object had a size of 50 px × 50 px (4.88 cm × 4.88 cm). The user's task was to drag the center of the object into the center area of the target using her index finger. The center area of the target was an invisible circle with a radius of 10 px. If the center of the object was dragged inside this center area the border of the object changed to green. The dragging task was complete when the user released the object inside this target area. After the task was completed object and target disappeared and a new pair of target and object of the same conditions were presented. When the user had conducted all seven dragging operations in the condition, the next condition was randomly selected. The test is shown in Figure 6.3.

Before the first trial, each participant got a handout in which the task was described. A demo version of the distributed handout is included in Appendix B.1. Additionally, each user had to conduct a training trial to familiarize herself with the task. In this training trial we asked the user to conduct seven upwards dragging operations across the curve.

6.3.3 Results

For each dependent variable in this test we conducted a 2 (*direction*) x 3 (*area*) repeated measures Analysis of Variance (ANOVA) and a pairwise comparison between the different conditions for each independent variable.

The results of the ANOVA on the mean dragging duration are illustrated in Figure 6.4. They show a significant main effect for the *area* condition $(F(2, 34) = 14.203; p < 0.001)$. Hypotheses H3 (a) does not hold because the *direction* condition is not significant. The results for the *area* condition show that the dragging duration inside the curve is significantly longer then dragging durations on the planar areas. The ANOVA shows no significant differences in the duration of the dragging operations between the horizontal and the vertical *area* conditions can be found. This disproves hypothesis H1 (a). The dragging duration inside the curve (mean 1166 ms) is 14% (145 ms) longer than the dragging duration on the horizontal area (mean 1016 ms) and 10%

dragging inside the curve is significantly longer as in both other areas

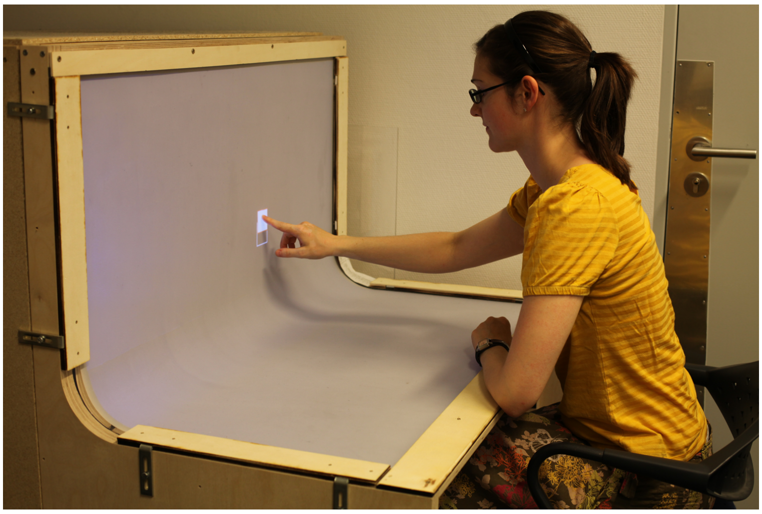


Figure 6.3: User conducting User Test 1.

area	sig.	up (px)	down (px)
horizontal	–	5.702	5.666
curve	$F(1,17) = 9.54; p < 0.01$	5.688	6.115
vertical	$F(1,17) = 9.079; p < 0.01$	5.854	5.642

Table 6.1: Comparison between the variance of the deviations between both *direction* conditions for each *area* condition

(110 ms) longer than the dragging duration on the vertical area (mean 1056 ms). This confirms the hypothesis H2 (a).

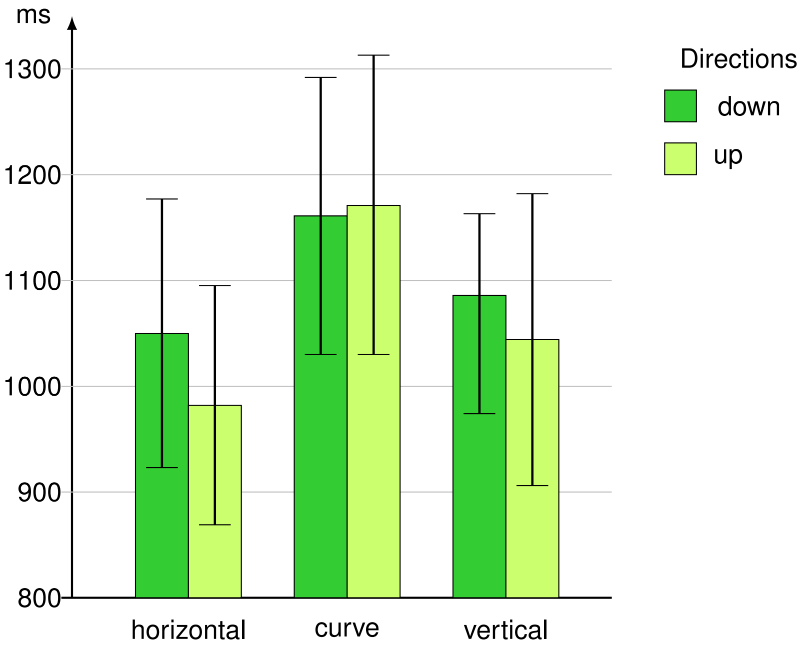


Figure 6.4: Average dragging duration depending on area and direction. Whiskers denote the 95% confidence interval.

Figure 6.5 illustrates the results of the ANOVA on the mean trajectory length. It found significant effects for the *area* condition $(F(2,34) = 28.846; p < 0.001)$ and for interaction $(F(2,34) = 4.737; p < 0.05)$. Only for the horizontal area the upwards trajectory length is significantly shorter than the downwards trajectory length $(F(1,17) = 5.537; p < 0.05)$. This approves hypothesis H3 (b) but only for the horizontal *area* condition.

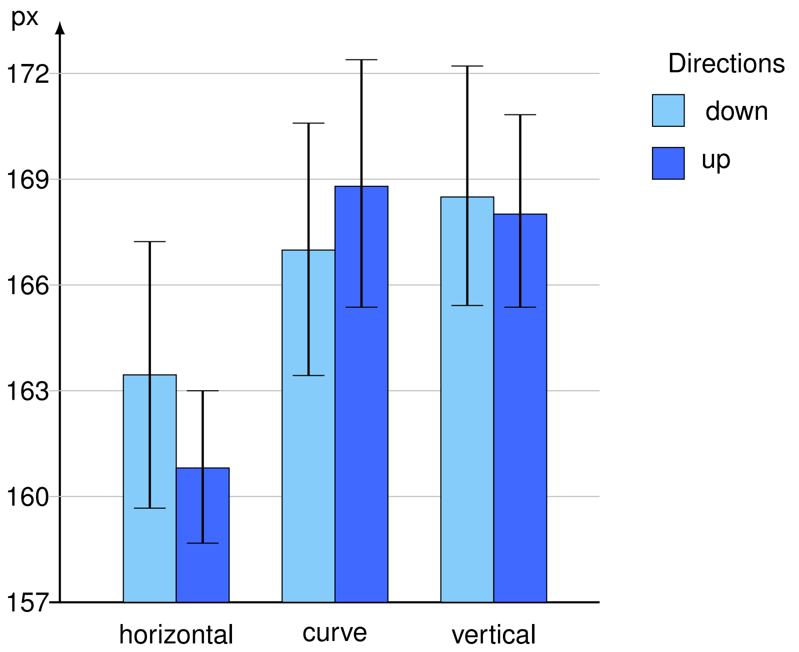


Figure 6.5: Average length of dragging trajectory depending on area and direction. Whiskers denote the 95% confidence interval.

The trajectory length on the horizontal area is significantly shorter than in the other conditions ($p < 0.001$), as the pairwise comparison between the different *area* conditions show. No significant differences can be detected between the curved *area* condition and the vertical *area* condition. Therefore, hypothesis H1 (b) is confirmed but hypothesis H2 (b) does not hold for the vertical *area* condition.

trajectory length is significantly shorter on the horizontal area than on the other ones

Significant effects were found by the ANOVAs on the maximum deviation of the object center to the direct line $(F(2,34) = 12.686; p < 0.001)$ and the average deviation $(F(2,34) = 15.980; p < 0.001)$. The pairwise comparisons of the maximum deviations in the different *area* conditions shows that the maximum deviation on the horizontal area is significantly smaller than on both other areas. In contrast, the pairwise comparison for the average deviation confirms that the average deviation for the curved area is significantly larger than on both other areas (Figure 6.6). This validates hypotheses H4 (a), but also rejects H4 (b).

deviation on the horizontal area is smaller than on both other areas

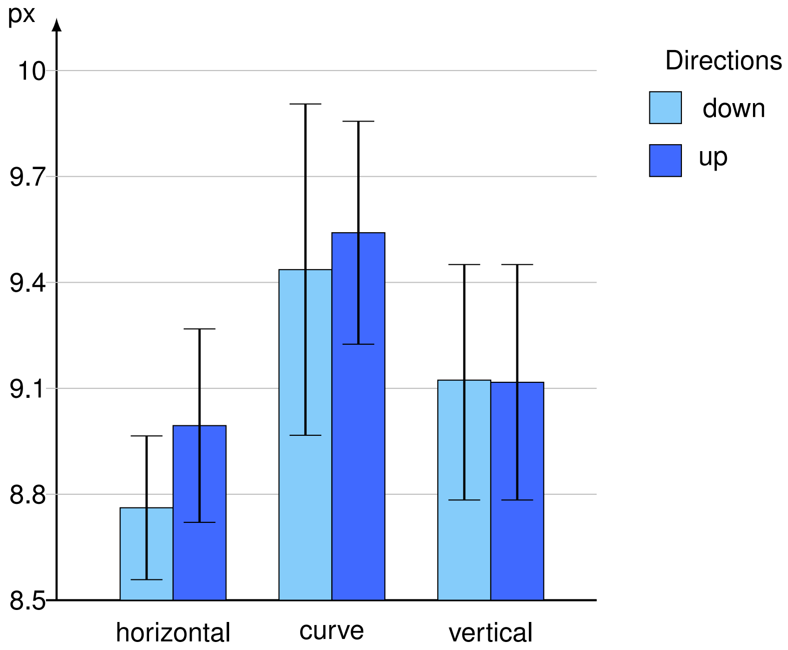


Figure 6.6: Average deviation to the direct line depending on area and direction. Whiskers denote the 95% confidence interval.

For the variance of the deviations of the object to the direct line, significant effects can be found for the interaction of both independent variables ($F(2, 34) = 12.628; p < 0.001$). For each *area* condition the comparison of both *direction* conditions are illustrated in Table **??**.

The ANOVAs for the number of errors and the number of overshoots did not find any significant effects.

6.3.4 Discussion

more complex motoric activity in the curve

The results show that dragging across the curve is significantly slower than on both planar areas, even though the distance was constant for all three conditions. We assume that a more complex motoric activity in the curve leads to this effect. On the horizontal area, the participants basically dragged the object by pushing or pulling the hand backwards or forwards, on the vertical area by lifting or lowering the arm. In contrast to both planar conditions, in the curve the participants had to move their arms in two dimensions. In the upwards direction they also tended to turn in the entire hand

in upward directions most users tended to turn their hand

After performing the user test, one person stated that he was afraid of overstretching his index finger if he does not turn the hand. Another person pointed out that he had the feeling of drilling his index finger into the surface and thus turned his hand. Four users wanted to use the middle finger instead of the index finger when unintentionally releasing an object because they considered the middle finger as stronger and more stable.

The length of the trajectory on the horizontal area is significantly shorter than on both other areas. Probably this results from the way the participants performed the dragging operations. On the horizontal area they used only the hand; the dragging operation can be conducted with a higher precision as in the curve and vertical area, where the entire arm is involved. This is also confirmed by the fact that the dragging operation on the horizontal area is performed more accurately than on both other areas.

operations which only involve the hand are more accurate then those involving the whole arm

Additionally, the results show that the deviation of the movement trajectory to the direct line between the object and the target is higher than on both planar areas. This indicates that the accuracy on the curve is less accurate. We assume that this effect is not only sourced in the more complex motoric activity required. Furthermore, we hypothesize that users' perception of the curve also affects the accuracy, because the exact shape of the curve is hard to identify and the planing on the arm movement is difficult.

6.4 User Test 2

In the second test we intended to explore, whether the dragging performance on the curve is influenced by the dragging angle. We performed a dragging test with a constant distance between object and target. We varied the an-

dragging across the curve with different angles

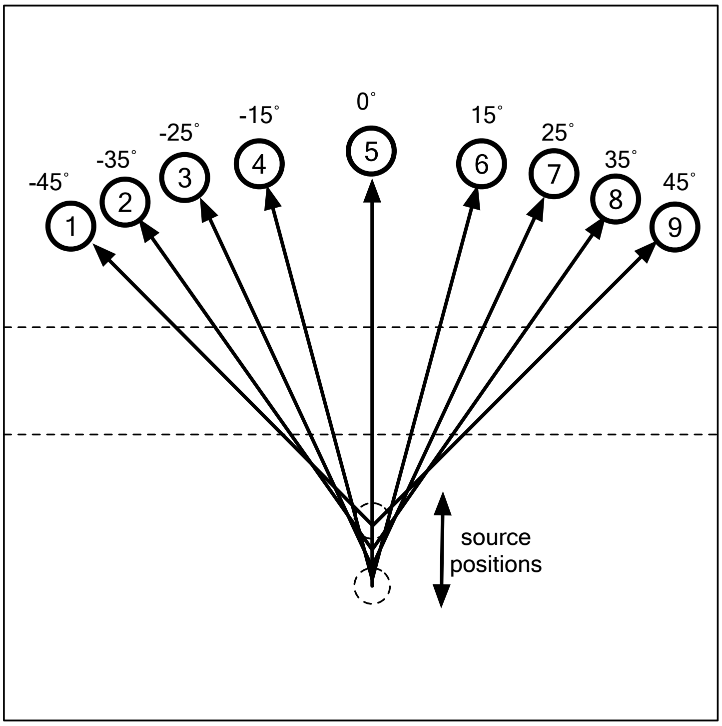


Figure 6.7: Experimental design for cross-dragging angle performance test.

gle and the direction for each operation. As before, we measured the length and the duration of the trajectories. We also assumed that a larger angle yields a lower dragging performance and higher deviation from the ideal dragging line. We postulate the following :

- H5: The dragging (a) duration and (b) trajectory increase with higher dragging angles.
- H6: The deviation from the direct connection between source and target increases with higher dragging angles.

6.4.1 Experimental Design

The participants' task was to dragging an object onto a target as fast as possible. The distance between the object and the target was 600 px (58.20 cm) in all conditions. The condition in this test was the dragging angle, i.e., the angle between the source-target axis and the y-axis. Our test setup is shown in Figure 6.7.

nine dragging operation from $-45°$ to $45°$ for each direction

We tested different angles in nine conditions: (1) $-45°$, (2) $-35°$, (3) $-25°$, (4) $-15°$, (5) $0°$ (vertical line), and (6) $15°$, (7) $25°$, (8) $35°$, (9) $45°$. For each condition, the users had to dragging the target upwards and downwards. Thus, a trial consisted of a total of 18 drag operations where we randomized the order of conditions in each trial. Each trial was repeated four times, twice for the dominant and twice for the non-dominant hand. This yielded a total of 1224 dragging operations, that was 72 dragging operations per participant. Due to a recording error in the test program we had to remove one participant from the evaluation of this test. To verify our hypotheses we defined the same dependent variables as in User Test 1 (6.3.1).

6.4.2 Methods

the task is do drag a circle into a ring

Our system displayed a white colored circle (the object) and a black colored circle inside a white ring (the target) for each dragging operation. Both circles had a diameter of 60 px (about 5.82 cm) and the thickness of the target ring amounts to 20 px (1.94 cm). Like in the previous test, each participant had to drag the object into the center of the target ring with a tolerance of 10 px (0.98 cm). The dragging length and duration was measured as in the previous test. After each successful drag operation, target and object disappeared and the next dragging task showed up. After the participants had conducted the dragging operation for all nine angles, they had to repeat the nine dragging operations in the other direction. On average, a participant spent about 60 seconds per trial. Figure 6.8 illustrates the test.

Before the first trial, each participant revived a handout in

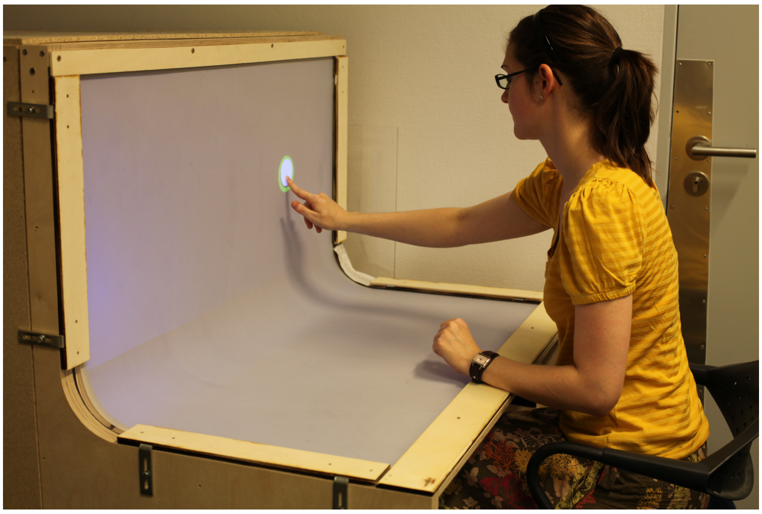


Figure 6.8: User conducting User Test 2.

which the task was described (Appendix B.2) and they had to conduct a training trial to familiarize herself with the task. In this training trial they had to conduct nine upward dragging operations.

6.4.3 Results

An evaluation of the repeated measures of the mean dragging duration time via ANOVA shows a significant effect $(F(8,128) = 2.656; p < 0.05)$. However, no significant difference can be found that shows that the dragging duration is longer for larger angles. This rejects hypotheses H5 (a).

no significant time differences between the angles

The repeated measures ANOVA for the mean dragging distances shows a significant effect $(F(8,128) = 8.947; p < 0.001)$, as illustrated in Figure 6.9. As shown in Figure 6.9 the trajectory depends on the *angle* condition. Between *angle* conditions $-45°$ and $0°$, as well as between $45°$ and $0°$ significant effects are found. This approves the hypothesis H5 (b).

dragging distance is longer for larger angles

The *angle* condition significantly influences the maximum $(F(8,128) = 11.662; p < 0.001)$ and the average $(F(8,128) = 10.516; p < 0.001)$ distance from the move-

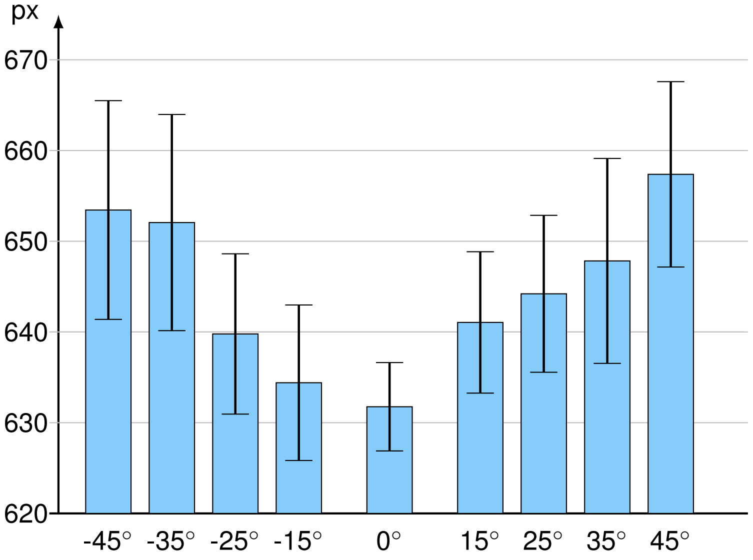


Figure 6.9: Average length of dragging trajectory depending on angle. Whiskers denote the 95% confidence interval.

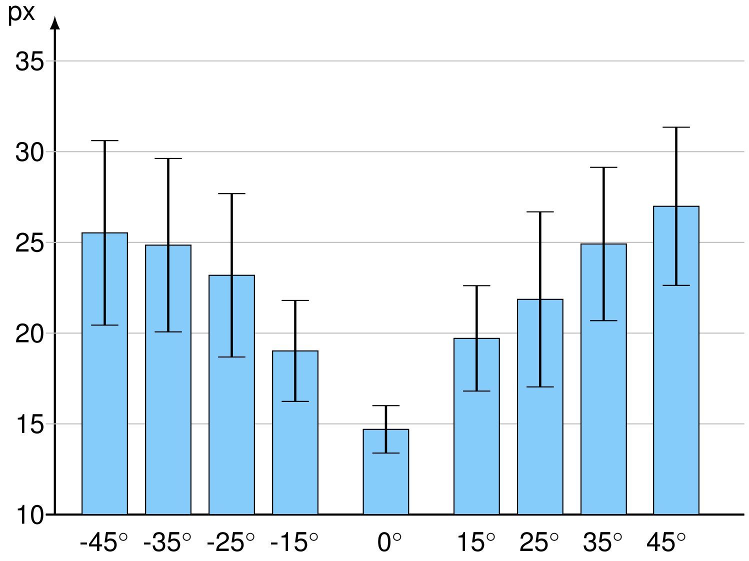


Figure 6.10: Average deviation from direct line between source and target depending on angle. Whiskers denote the 95% confidence interval.

ment trajectory to the direct line between the circle and the ring (Figure 6.10 and Figure 6.11). This confirms hypothesis H6.

6.4.4 Discussion

users tended to minimize dragging distance in the curve

The results show that the angle has no significant effect on the dragging duration, but we found a significant increase of the trajectory length at larger angles. To analyze this effect, we plotted the trajectories for each angle. Figure 6.12 illustrates only the upwards direction as an example, the downwards direction looks very similar. Two effects can be clearly identified: First, the higher the angle the higher is the spreading of the trajectories besides the direct line between the object and the target. This is also confirmed by Figure 6.10 and Figure 6.11 that indicates that not only the maximum deviation is higher for higher angles. Furthermore, the average deviation is higher for higher angles. The second effect is that the participants tended to minimize the dragging distance on the curve for higher angles. Some users tried to follow a path that has a minimized length in the curve to the price of a longer path length on the planar surfaces. We assume that is one of the reasons why the spreading is higher for higher angles.

downward dragging was more comfortable

Additionally, our observations show that most of the participants tried to reduce muscle exertion by optimizing their dragging operations. Dragging downwards some of them let their arm fall straight downwards through the curve before they changed the direction to the target. This is illustrated in Figure 6.13(a). In upwards movements most of the participants used a stiff bent arm to drag object across the curve. They finished the dragging by turning the hand and the lower arm with the upper arm as rotation axis (Figure 6.13(b)). Two user mentioned that dragging an object straight upwards or downwards through the curve was more comfortable as dragging it diagonally through it. We noticed that users chose a convenient movement trajectory over the task to drag as fast as possible.

6.5 User Test 3

does the curve influence the perception of lines?

The second user test showed that dragging with a flat angle through the curve resulted in a longer dragging trajectory as with a steep angle. This pointed out that the curve area is a barrier for dragging operations. In this test we wanted to investigate whether the curve is not only a barrier for dragging operation, but also influences the perception of straight lines and directions in the user interface. The perception of directions is especially important for flinging gestures. In this test the participants had to hit a target by defining a straight line with two fingers. To evaluate the accuracy of the targeting operation we defined the *aiming error* as the distance of the defined line to the target (Figure 6.15). We hypothesized the following outcomes:

- H7: The aiming error is higher, if source and target are shown in different areas, rather than if both are positioned on the same surface.
- H8: The less orthogonal the angle of the source-target line, the higher the aiming error.
- H9: An uniform grid displayed in the background supports lowering the aiming error.

6.5.1 Experimental Design

the task is to hit objects from a distant position

The participant's task was to hit a target by specifying a straight line in a source area. We tested ten different conditions varied by the angle between the source-target line and the y-axis of the table. The angles were between 0° and 90° (see Figure 6.15). As illustrated in Figure 6.15, in conditions (1) to (3) the target was placed on the same surface as the source, whereas in condition (4) the target was put on the curve. Finally, in conditions (5) to (10) the target was positioned on the opposite surface compared to the source's one. The distance between the source area and the target was for all conditions the same. Additionally, we varied the background as a second variable:

(1) Solid color

(2) Uniform grid

Furthermore, we varied the position of the source area that could be bottom left, bottom right, top left, and top right. Therefore, a trial consisted of 40 aiming tests. Each participant had to conduct four trials: two with a solid color in the background and two with a grid. This resulted in a total of 2880 aiming operations; 160 per user. To verify our hypotheses we recorded the following dependent variables:

- **Distance to target center:** The distance between the straight line and the center of the target.
- **Distance to target:** The distance to the target center minus the target radius. If the distance to the target center is shorter than the radius, the distance to the target is set to null.
- **Hit:** Indicates if the target is hit or not.

6.5.2 Methods

For each aiming task a gray filled circle with a diameter of 200 px (19.5 cm) (the source area) and a white circle with a diameter of 60 px (5.82 cm) were displayed on the table. The background for the *solid color* condition is blue. For the *uniform grid* condition a white 26 × 26 grid with a cell size of about 40 px × 40 px (3.9 cm × 3.9 cm) was shown on the blue background. The user's task was to touch two points inside the target area, such that the straight line that was defined by these two points also hits the target. Beneath each touch a surrounding circle was displayed. The line was only displayed when both fingers are released from the table. The line was shown for two seconds to give the user a feedback if she had hit the target. Then the next target appeared. After the user had conducted ten aiming tasks for a source area position the source area changed the position. The test is shown in Figure 6.16.

the user has to create a line with two fingers that runs through the target

Before the first trial each participant got a handout in which the task was described (Appendix B.2) and she had to conduct a training trial where she had to aim at ten targets for situated in the right source area position.

6.5.3 Results

In this test we conducted for each dependent variable a 2 (*background*) x 10 (*angle*) repeated measures ANOVA and pairwise comparisons for each independent variable.

for both orthogonal angles the mean distance to target is the lowest

Significant effects for the *angle* condition ($F(9, 158) = 17.241; p < 0.001$) and for the interaction of both conditions ($F(9, 153) = 2.61; p < 0.01$) are shown by the ANOVA. The pairwise comparison of the different *angle* conditions shows significant effects only between the *angle* 90° and all other conditions. Therefore, hypothesis H7 does not hold. Furthermore, additional significant effects can be found between the *angle* condition 0° and the conditions 90°, 20°, 30°, 40°, 50°, and 60°. This confirms hypothesis H8.

a grid increases the accuracy of targeting only for $90°$ angles

The results of the pairwise comparison of the interaction shows only significant effects between both *background* conditions for the *angle* conditions 0° ($F(1, 17) = 17.179; p < 0.001$) and 90° ($F(1, 17) = 17.854; p < 0.001$) (Figure 6.17). This confirms hypotheses H9 only for both orthogonal *angle* conditions. For the 90° *angle* condition the mean distance to the target with displayed grid is (mean 4.55 px) is 65% (9.792 px) shorter than without the grid (mean 13.138 px). For the 0° *angle* the mean distance to the target with the grid (mean 13.911 px) is 57% (18.946 px) shorter than without the grid (mean 32.857 px).

6.5.4 Discussion

The results show that the orthogonal angles ($0°$, $90°$) have the lowest aiming errors. This can be explained by the fact that the user could use the table borders as alignments guides. Additionally, the results show that the higher the deviation from the orthogonal angles the higher is the *aiming error*. The *aiming error* is maximal if the target for the *angle* conditions 30° and 40° that are both on the other planar surface as the source area. We assume that the performance depends on the different coordinate systems. The table uses a 2D coordinate system in a 3D space. The users had to map the 3D space they interact in to the 2D space of the GUI. We think that this mapping between the two spaces is especially difficult in the curve, because the curve is larger than it is perceived. This could be one of the reasons why targeting an object that is placed on a different surface as the source yields in a higher *aiming error*. Another effect that influences the aiming accuracy is the perspective distortion on the plane. In contrast to the study by Wigdor et al. [2007b] that says that this effect is stronger on a horizontal surface than on a vertical surface, we found no differences between both areas.

mapping from 3D space into 2D GUI space difficult

The background grid does only improve the aiming accuracy for both orthogonal angles significantly. This is obvious, because the participants had only to position their fingers onto a grid line that runs through the target and the source area. For all other angle conditions the grid has no effect on the aiming accuracy.

Most of the users reported that this test is more like a competitive game. Some of them pointed out that this technique could be used to create an entire game and demanded a high score for this test.

6.6 User Test 4

One of the major benefits of a multi-touch system is that it supports direct touch, which is also one of the major drawbacks because interacting over a long time with the system can be exhausting. The users have to use their arms and cannot interact with such a system with the arm lying on the table. In the BendDesk system the user has not only to reach over a horizontal area, furthermore she has to lift her arm to interact with the vertical area. We believed that these movements would lead to muscle fatigue in short time.

how long can users conduct dragging operations without any signs of fatigue?

Therefore, we developed a fourth test to gain a rough estimate how long users can conduct dragging operations without any signs of fatigue. In this test we asked the users to repeat the second user test as long as they felt no muscle fatigue. Furthermore, we asked users to express any signs of fatigue during the test.

Results

signs of fatigue on average after 7:30 min

No participants reported any signs of fatigue in the first four minutes. The first participants (6/18) that expressed signs of fatigue in their upper arms did it after four minutes. Additionally, they pointed out that their fingers were warm. The average time each participant conducted the dragging task was 7:30 minutes. However, after twelve minutes two participants commented that they could do the test "the whole day". Both stopped the test after about 15 minutes without any symptoms of fatigue.

16 of the 18 participants perceived the downward dragging as the more comfortable interaction direction, since in this case they could almost let their arms fall down. Compared to this, as shown in Figure 6.13(b), most users had to rotate their hands to do a comfortable motion in the upwards movement.

the users tried to minimize the dragging way in the curve

Dragging diagonal through the curve was very inconvenient for 13 participants. They pointed out, that they rather increase the dragging path on the planar surface in order to cross the curve with a very steep angle than dragging diagonal through it. 10 participants got the impression that their dragging speed inside the curve area was much slower than on the planar areas. Five of them thought that they had to use more pressure on the curve to drag the object.

Discussion

We only tested 18 participants, which is not a representative population for an ergonomic analysis of the BendDesk system. Furthermore, most users were male and between 24 and 32 years old. Nevertheless, we could show, that all participants were able to perform the dragging task for a rather long period without any signs of fatigue.

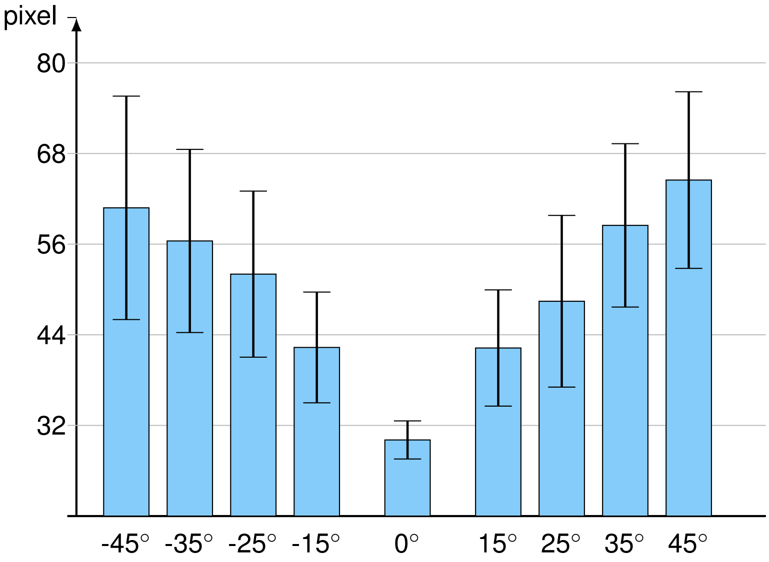
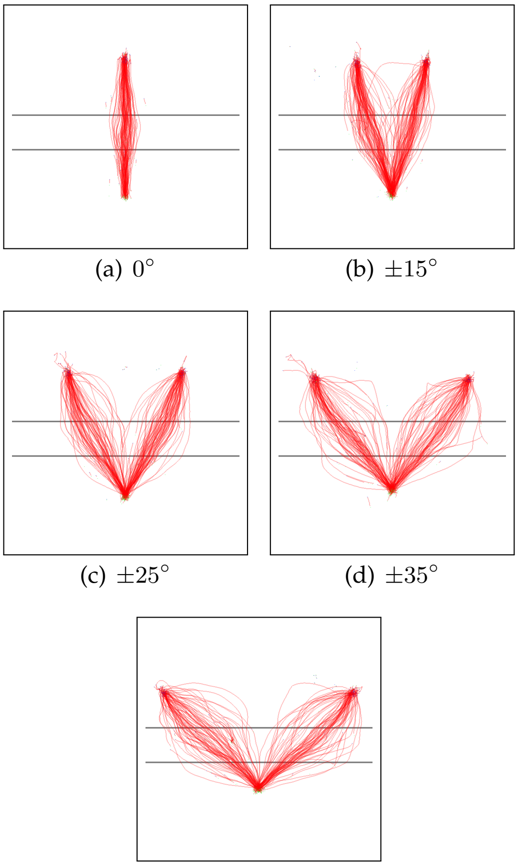


Figure 6.11: Maximum deviation from direct line between source and target depending on angle. Whiskers denote the 95% confidence interval.



(a) 0°

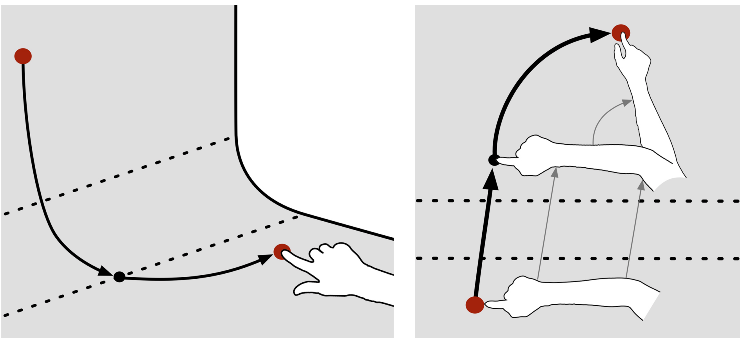
(b) ±15°

(c) ±25°

(d) ±35°

(e) ±45°

Figure 6.12: Dragging trajectories for upward dragging across the curve for different angles. Variance significantly increases with higher angles.



(a) downwards

(b) upwards

Figure 6.13: Observed dragging trajectories that reduce exertion.

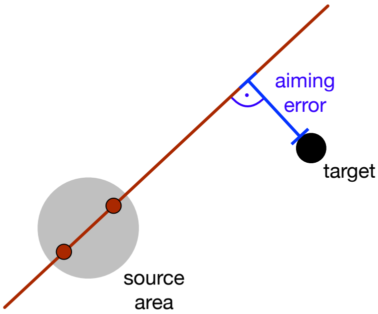


Figure 6.14: Two touches (red dots) in the source area define a straight line. The aiming error is the distance between this line and the target circle.

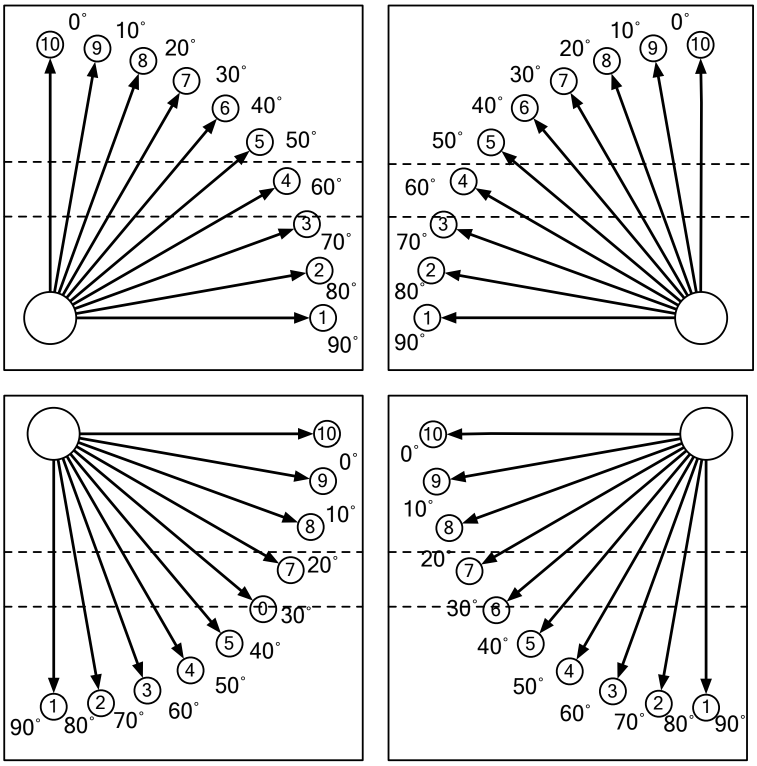


Figure 6.15: Experimental design for User Test 3.

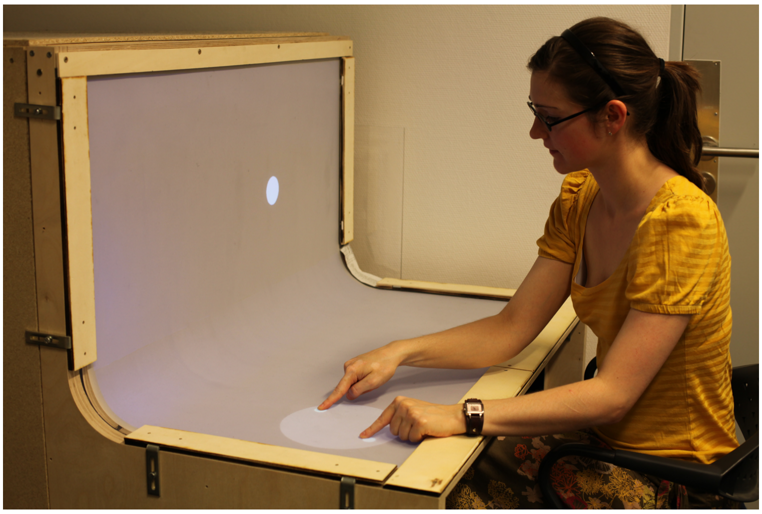


Figure 6.16: User conducting User Test 3.

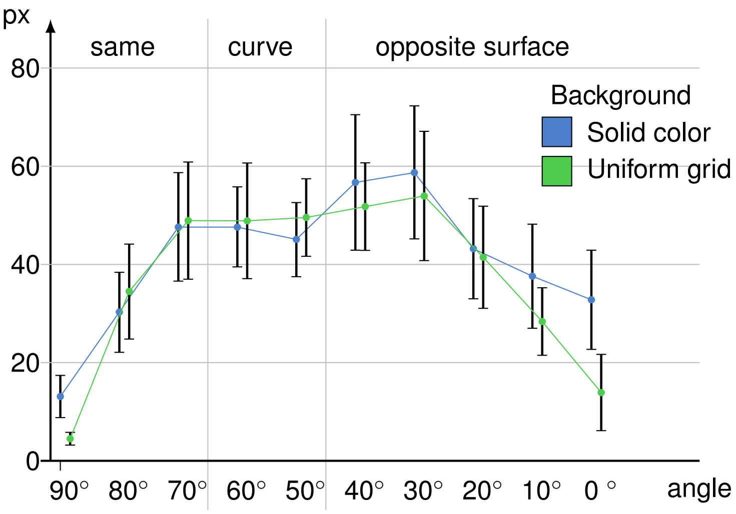


Figure 6.17: Average distance from target depending on angle. Whiskers denote the 95% confidence interval.

Chapter 7

Summary and Future Work

In the previous chapters we explained the ideas, the design, and the developing process of BendDesk. Furthermore, we presented a user study that explores dragging and targeting interactions on BendDesk. This last chapter summarizes our work and gives an outlook on future research.

7.1 Summary and Contributions

BendDesk is an interactive multi-touch desk environment system that combines a horizontal area with a vertical area by a curve into one seamlessly connected touch sensitive surface. This offers a very large but still entirely reachable interactive area, where for each task the user can choose which area she wants to use. Furthermore, the system is designed in such a way that users can sit at the system and use it comfortably.

BendDesk is a multi-touch desk

We presented the design considerations about the form factors and the hardware setup that had to be evaluated to create this system. Furthermore, we conducted an user test with the prototype to determine the form factors of BendDesk. With the results of this test we created the hardware of the final system and developed the agent application that allows us to use the system. This agent application consists of two different parts:

Visual Output: Since BendDesk has a non-planar surface the software has to compensate the distortion of the GUI in the curved area.

Tracking: To determine where the user has touched the interactive surface the software has to analyze the images of the camera.

Both parts can be calibrated by the agent application.

dragging performance on the curve is slow

curve as storage space

horizontal area should be main workspace

Additionally we conducted a user study to analyze the task of dragging and targeting task on BendDesk. Our user study shows that there are differences in how users interact with the different areas. It shows that the dragging performance on the curve is inferior to the performance on the planar surfaces. The results also show that dragging with a flat angle should be avoided. That points out that this area should be used only as transition area or as storage area. Additionally, the tests show that users are most accurate on the horizontal area, which suggests that this area should be used as main working area. The vertical area performs not so well as the horizontal area. It could be used as an overview area where users interact less than on the horizontal area.

curve influences targeting operations

Especially User Test 3 points out that aiming operations like flinging where the direction is not orthogonal to table borders have a very low accuracy. In the application's development such operations should be avoided. Furthermore, reaching techniques such as the I-Grabber technique by Abednego et al. [2009] that uses a very similar method to target an object is not suitable on BendDesk.

7.2 Future Work

This thesis is an initial work that focuses on the design and the development of the system. Since the system is ready to being used now, there is a lot of research that can be done with BendDesk. The following possible future research is based on ideas that arouse during the development process, on the features we could not implement, and on the feedback from from our user studies.

7.2.1 Improving Tracking Algorithms

parallel implementation of the spot detection algorithm

Since our algorithm for the spot detection is nearly straight forward the performance maybe be increased by a parallel implementation the algorithm. Furthermore, it could be possible to compute this algorithm on an Field Programmable Gate Array (FPGA). This would reduce the system load and also increase the performance of the spot detection. We assume that using an *local maximum algorithm* instead of an *connected component algorithm* would increase the performance as well.

improving the predictive tracking algorithm

Additionally our *predictive tracking algorithm* can be improved with the following methods: As first improvement the algorithm could use the real time between each, hence this time is not always the same. Second, we could use specific heuristics to distinguish between two touches that are very close together.

7.2.2 Perception of the System

does the curve influence the user's perception?

The targeting user test has revealed that the users had problems to perceive direct lines crossing the curve. In a following study this effect could be analyzed in more detail and how it influences the possible usage of the system. Furthermore, the dragging test showed that users perceived the curve as a barrier. This could also be analyzed in more detail.

7.2.3 Investigate Pointing and Other Gestures

does Fitts's Law hold on BendDesk?

In our user study we investigated dragging and targeting operations on BendDesk but we intend to explore pointing as well. For most surfaces pointing is well covered by Fitts's law by [Fitts, 1992], hence, we want to investigate if Fitts's law holds on BendDesk. Additionally we want to explore other gestures such as flinging or tapping as well. A possible use for flinging could be to move objects from one planar area into the other one. We assume that this gesture would make the interaction with BendDesk more comfortable.

7.2.4 Pen as Additonal Input Device

pen and touch input on BendDesk

Multi-touch systems have the problem that touch input is not very precise. Therefore, we intend to use a pen to solve that problem. As Brandl et al. [2008] proposed, using an Anoto pen [H:Anoto2000] in combination with touch input is a very practical combination that the users already know of interacting with real paper sheets. To use an Anoto pen on BendDesk we only have to replace the diffuser layer on the table with a diffuse Anoto pattern.

7.2.5 Developing Desk Workplace Applications

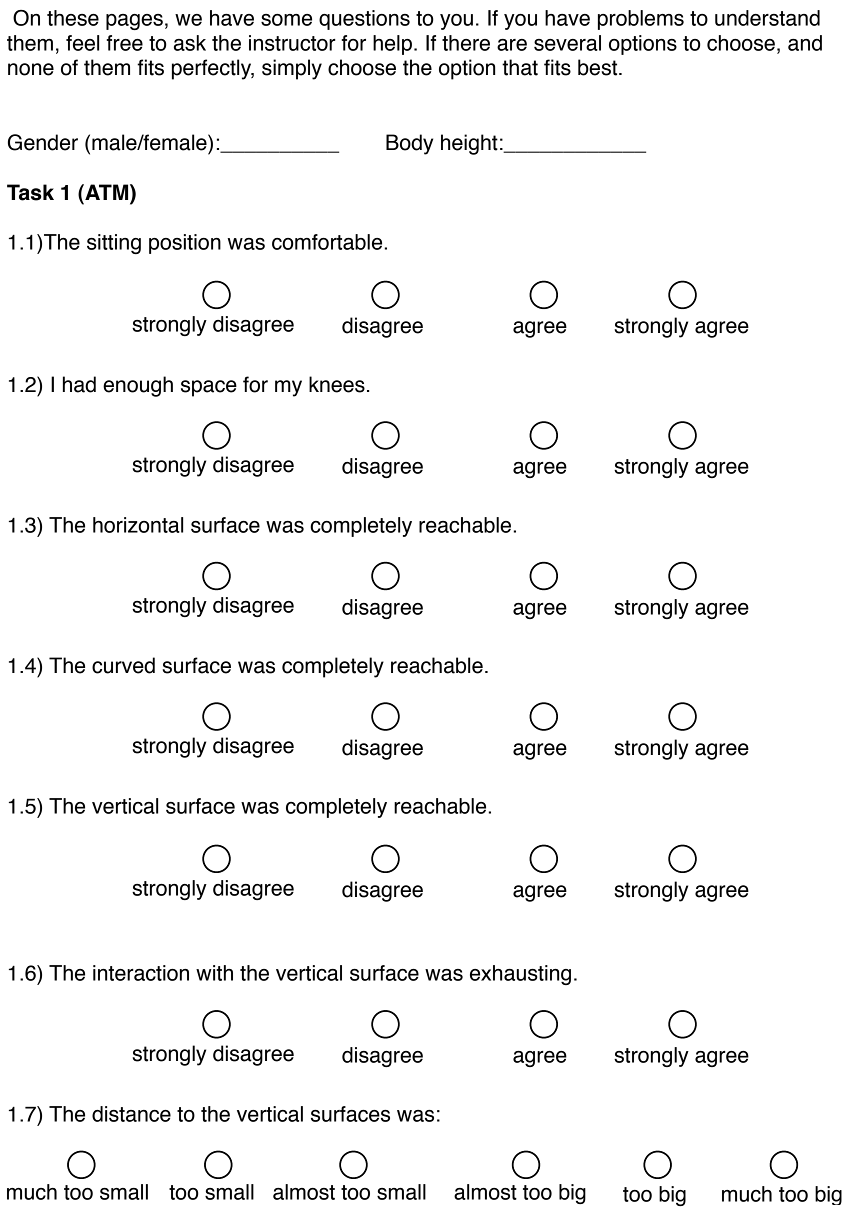
can BendDesk be used for everyday tasks?

After building the system and developing software that allows to use BendDesk as a multi-touch table, we intend do create a real desk working environment such that it can be used as a desk workspace to perform everyday tasks. We want to offer the users the possibility to work with documents and other digital objects. Furthermore, we intend to explore whether BendDesk could be used as a normal desk.

Appendix A

Form Factor Prototype Questionnaire

This appendix contains the questionnaire that was used in the form factor prototype user test.



On these pages, we have some questions to you. If you have problems to understand them, feel free to ask the instructor for help. If there are several options to choose, and none of them fits perfectly, simply choose the option that fits best.

Gender (male/female):__________ Body height:____________

Task 1 (ATM)

1.1)The sitting position was comfortable.

○ strongly disagree ○ disagree ○ agree ○ strongly agree

1.2) I had enough space for my knees.

○ strongly disagree ○ disagree ○ agree ○ strongly agree

1.3) The horizontal surface was completely reachable.

○ strongly disagree ○ disagree ○ agree ○ strongly agree

1.4) The curved surface was completely reachable.

○ strongly disagree ○ disagree ○ agree ○ strongly agree

1.5) The vertical surface was completely reachable.

○ strongly disagree ○ disagree ○ agree ○ strongly agree

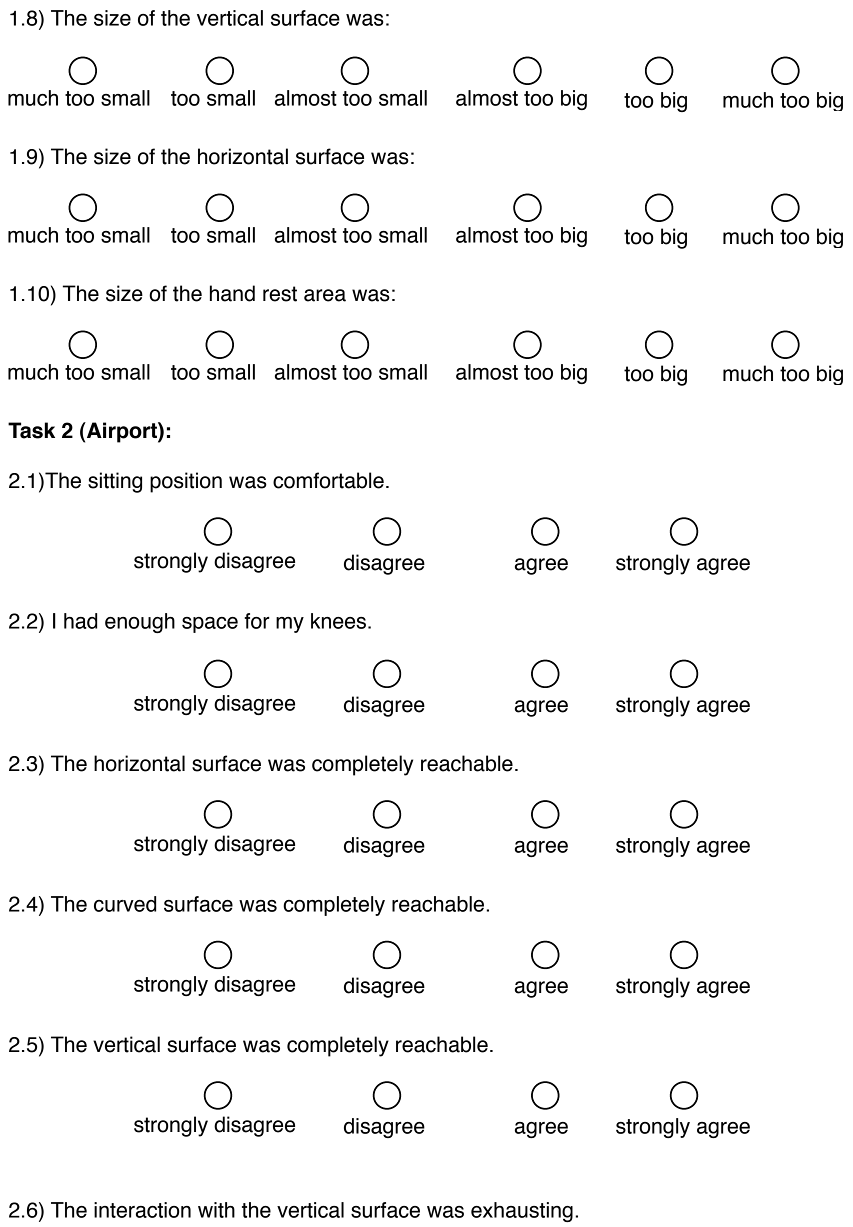
1.6) The interaction with the vertical surface was exhausting.

○ strongly disagree ○ disagree ○ agree ○ strongly agree

1.7) The distance to the vertical surfaces was:

○ much too small ○ too small ○ almost too small ○ almost too big ○ too big ○ much too big

Figure A.1: Form factor prototype questionnaire page 1



1.8) The size of the vertical surface was:

○ much too small ○ too small ○ almost too small ○ almost too big ○ too big ○ much too big

1.9) The size of the horizontal surface was:

○ much too small ○ too small ○ almost too small ○ almost too big ○ too big ○ much too big

1.10) The size of the hand rest area was:

○ much too small ○ too small ○ almost too small ○ almost too big ○ too big ○ much too big

Task 2 (Airport):

2.1)The sitting position was comfortable.

○ strongly disagree ○ disagree ○ agree ○ strongly agree

2.2) I had enough space for my knees.

○ strongly disagree ○ disagree ○ agree ○ strongly agree

2.3) The horizontal surface was completely reachable.

○ strongly disagree ○ disagree ○ agree ○ strongly agree

2.4) The curved surface was completely reachable.

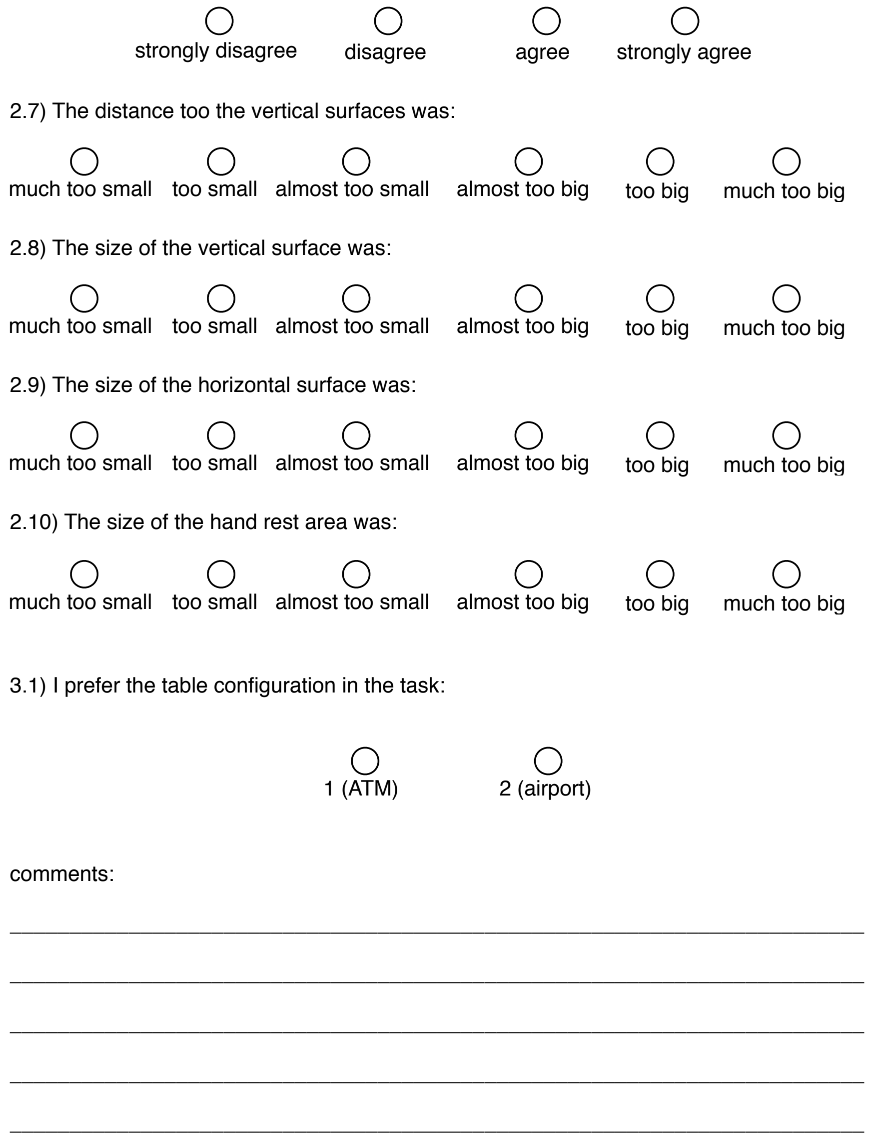
○ strongly disagree ○ disagree ○ agree ○ strongly agree

2.5) The vertical surface was completely reachable.

○ strongly disagree ○ disagree ○ agree ○ strongly agree

2.6) The interaction with the vertical surface was exhausting.

Figure A.2: Form factor prototype questionnaire page 2



○ strongly disagree ○ disagree ○ agree ○ strongly agree

2.7) The distance too the vertical surfaces was:

○ much too small ○ too small ○ almost too small ○ almost too big ○ too big ○ much too big

2.8) The size of the vertical surface was:

○ much too small ○ too small ○ almost too small ○ almost too big ○ too big ○ much too big

2.9) The size of the horizontal surface was:

○ much too small ○ too small ○ almost too small ○ almost too big ○ too big ○ much too big

2.10) The size of the hand rest area was:

○ much too small ○ too small ○ almost too small ○ almost too big ○ too big ○ much too big

3.1) I prefer the table configuration in the task:

○ 1 (ATM) ○ 2 (airport)

comments:

____

____

____

____

____

Figure A.3: Form factor prototype questionnaire page 3

Appendix B

User Test Handouts

This appendix contains the texts that are handed out for each use test as task description.

B.1 User Test 1 Handout

This test investigates the dragging of objects on BendDesk. You can drag an object by placing one of your fingers onto it. Then, it follows your fingertip until you released the finger again.

Your task is to drag a white quad as fast as possible into the white frame for several positions on the screen.

When the object is inside the target area, its border changes from red to green. We measure the dragging duration from the moment you touch the object until you release your finger. After you have successfully dragged a quad into the frame, a new object and a new target appear.

Please use the index finger for each drag. During the test, you will be asked to change your hand.

We start with a short 1-minute practice trial. After these trials the actual test begins. The test takes about 4 minutes.

B.2 User Test 2 Handout

This test investigates the dragging of objects on BendDesk. You can drag an object by placing one of your fingers onto it. Then, it follows your fingertip until you released the finger again.

Your task is to drag a white circle as fast as possible into the white ring for several positions on the screen.

When the object is inside the target area, the ring changes from white to green. We measure the dragging duration from the moment you touch the object until you release your finger. After you have successfully dragged a circle into the ring, a new object and a new target appear.

Please use the index finger for each drag. During the test, you will be asked to change your hand.

We start with a short 1-minute practice trial. After these trials the actual test begins. The test takes about 4 minutes.

B.3 User Test 3 Handout

This test investigates the accuracy of targeting an object on the BendDesk. A white circle and a gray circle are displayed on the table.

Your task is to define an invisible line by touching two positions in the gray circle, such that the resulting straight line hits the target. Be as accurate as possible!

Inside the gray circle you can change the position of your fingers until you think the resulting straight line hits the target. If you release both of your fingers the color of the gray circle changes to white and the straight line is show so you can see if you have hit the target.

We start with two short 1-minute practice trials. After these trials the actual test begins. The test takes about 5 minutes.

Appendix C

Online Resources

This chapter contains the URLs of online resources that have been referenced in the text.

Anoto2000 Anoto Group AB: *Anoto - THE PEN* (established 2000, accessed at July 10, 2010)
http://www.anoto.com/the-pen.aspx

Apple2010 Apple Computer, Inc: *Apple - NSNotificationCenter* (established 1987, accessed at July 10, 2010)
hhttp://developer.apple.com/mac/library/documentation/Cocoa/Reference/Foundation/Classes/NSNotificationCenter_Class/Reference/Reference.html

Evonik2004 Evonik Degussa GmbH: *EndLighten* (established 2004, accessed at July 10, 2010)
http://www.acrylite-magic.com/

EnvironmentalLights2006 Environmental Lights : *InfraRed LED Strips* (established 2006, accessed at July 10, 2010)
http://www.environmentallights.com/products/12705/

Nuigroup2006 NUI Group Community: *Tinkerman's method* (established 2006, accessed at July 10, 2010)
http://nuigroup.com/forums/viewthread/2383/

openGL.org2005 openGL.org: *Framebuffer Object* (established 2005, accessed at July 10, 2010)

http://www.opengl.org/wiki/GL_EXT_framebuffer_object

Vicon2000 Vicon Motion Systems: *Vicon Camera* (established 2000, accessed at July 10, 2010)

http://www.vicon.com/products/cameras.html

Bibliography

Martha Abednego, Joong-Ho Lee, Won Moon, and Ji-Hyung Park. I-grabber: Expanding physical reach in a large-display tabletop environment through the use of a virtual grabber. In *Proc. of ITS '09*, pages 61–64, November 2009.

Patrick Baudisch, Edward Cutrell, Ken Hinckley, and Robert Gruen. Mouse ether: Accelerating the acquisition of targets across multi-monitor displays. In *Proc. of CHI '09*, pages 1379–1382. ACM, 2004.

Hrvoje Benko. Beyond flat surface computing: Challenges of depth-aware and curved interfaces. In *Proc. of MM '09*, pages 935–944. ACM, 2009.

Hrvoje Benko, Andrew D. Wilson, and Ravin Balakrishnan. Sphere: Multi-touch interactions on a spherical display. In *Proc. of UIST '08*, pages 77–86. ACM, 2008.

Xiaojun Bi, Seok-Hyung Bae, and Ravin Balakrishnan. Effects of interior bezels of tiled-monitor large displays on visual search, tunnel steering, and target selection. In *Proc. of CHI '10*, pages 65–74. ACM, 2010.

Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proc. of AVI '08*, pages 154–161. ACM, 2008.

William Buxton, George W. Fitzmaurice, Ravin Balakrishnan, and Gordon Kurtenbach. Large displays in automotive design. *Computer Graphics and Applications, IEEE*, 20 (4):68–75, August 2000.

Dempsey Chang, Laurence Dooley, and Juhani E. Tuovinen. Gestalt theory in visual screen design: a new look at an old subject. In *Proc. of the Seventh world conference on computers in education conference on Computers in education*, pages 5–12. Australian Computer Society, Inc., 2002.

Fang Chen, Peter Eades, Julien Epps, Serge Lichman, Benjamin Close, Peter Hutterer, Masa Takatsuka, Bruce Thomas, and Mike Wu. ViCAT: visualisation and interaction on a collaborative access table. In *Proc. of TABLETOP '06*, pages 59–62. IEEE Computer Society, 2006.

Francois Coldefy and Stphane Louis-dit-Picard. Digitable: An interactive multiuser table for collocated and remote collaboration enabling remote gesture visualization. In *Proc. of CVPR '07*, pages 1–8. IEEE, 2007.

Paul Dietz and Darren Leigh. DiamondTouch: a multi-user touch technology. In *Proc. of UIST '01*, pages 219–226. ACM, 2001.

Florian Echtler, Andreas Dippon, Marcus Toennis, and Gudrun Klinker. Inverted FTIR: easy multitouch sensing for flatscreens. In *Proc. of ITS '09*, pages 29–32. ACM, 2009.

Katherine Everitt, Chia Shen, Kathy Ryall, and Clifton Forlines. MultiSpace: enabling electronic document micromobility in table-centric, multi-device environments. In *Proc. of TABLETOP '06*, page 8 pp., 2006.

Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. 1954. *J Exp Psychol Gen*, 121(3):269–262, 1992.

Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *Proc. of CHI '07*, pages 647–656. ACM, 2007.

Stefan Hafeneger, Malte Weiss, Gero Herkenrath, and Jan Borchers. Pockettable: Mobile devices as multi-touch controllers for tabletop application development. In *Ext. Abstr. of Tabletop '08*, 2008.

Jefferson Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc. of UIST '05*, pages 115–118, October 2005.

Otmar Hilliges, Shahram Izadi, Andrew D. Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. Interactions in the air: Adding further depth to interactive tabletops. In *Proc. of UIST '09*, pages 139–148. ACM, 2009.

David Holman and Roel Vertegaal. Organic user interfaces: Designing computers in any way, shape, or form. *Organic User Interfaces*, 51(6):48–55, 2008.

David Holman, Roel Vertegaal, Mark Altosaar, Nikolaus Troje, and Derek Johns. Paper windows: interaction techniques for digital paper. In *Proc. of CHI '05*, pages 591–599. ACM, 2005.

International Organization for Standardization. ISO 9241-5, ergonomic requirements for office work with visual display terminals: Workstation layout and postural requirements, 1998.

Jakob Leitner, James Powell, Peter Brandl, Thomas Seifried, Michael Haller, Bernard Dorray, and Paul To. FLUX: a tilting multi-touch and pen based surface. In *Proc. of CHI '09*, pages 3211–3216. ACM, 2009.

Nobuyuki Matsushita and Jun Rekimoto. HoloWall: designing a finger, hand, body, and object sensitive wall. In *Proc. of UIST '97*, pages 209–210. ACM, 1997.

Merrie Morris, A.J. Bernheim Brush, and Brian R. Meyers. Reading revisited: Evaluating the usability of digital display surfaces for active reading tasks. In *Proc. of TABLETOP '07*, pages 79–86, 2007.

Merrie Morris, Bernheim A. J Brush, and Brian Meyers. A field study of knowledge workers' use of interactive horizontal displays. In *Proc. of TABLETOP '08*, pages 105–112, 2008.

Christian Muller-Tomfelde, Anja Wessels, and Claudia Schremmer. Tilted tabletops: In between horizontal and vertical workspaces. In *Proc. of TABLETOP '08*, pages 49–56, October 2008.

A. Naamad and W. L Lee. On the maximum empty rectangle problem. *Discrete Applied Mathematics*, 8(3):267–277, 1984.

Miguel A. Nacenta, Regan L. Mandryk, and Carl Gutwin. Targeting across displayless space. In *Proc. of CHI '08*, pages 777–786. ACM, 2008.

Jun Rekimoto and Masanori Saitoh. Augmented surfaces: A spatially continuous work space for hybrid computing environments. In *Proc. of CHI '99*, pages 378–385. ACM, 1999.

Kathy Ryall, Meredith Ringel Morris, Katherine Everitt, Clifton Forlines, and Chia Shen. Experiences with and observations of direct-touch tabletops. In *Proc. of TABLETOP '06*, pages 89–96. IEEE Computer Society, 2006.

Bruce Tognazzini. The "Starfire" video prototype project: A case history. In *Proc. of CHI '94*, pages 99–105. ACM, 1994.

Aaron P. Toney and Bruce H. Thomas. Modeling reach for use in user interface design. In *Proc. of AUIC '07*, pages 27–30. Australian Computer Society, Inc., 2007.

Julie Wagner. SLAP, silicone illuminated active peripherals. Master's thesis, RWTH Aachen University, 2009.

Pierre Wellner. The DigitalDesk calculator: tangible manipulation on a desk top display. In *Symposium on User Interface Software and Technology*, pages 27–33. ACM, 1991.

Daniel Wigdor, Chia Shen, Clifton Forlines, and Ravin Balakrishnan. Effects of display position and control space orientation on user preference and performance. In *Proc. of CHI '06*, pages 309–318. ACM, 2006a.

Daniel Wigdor, Chia Shen, Clifton Forlines, and Ravin Balakrishnan. Table-centric interactive spaces for real-time collaboration. In *Proc. of AVI '06*, pages 103–107. ACM, 2006b.

Daniel Wigdor, Gerald Perm, Kathy Ryall, Alan Esenther, and Chia Shen. Living with a tabletop: Analysis and observations of long term office use of a multi-touch table. In *Proc. of TABLETOP '07*, pages 60–67, 2007a.

Daniel Wigdor, Chia Shen, Clifton Forlines, and Ravin Balakrishnan. Perception of elementary graphical elements in tabletop and multi-surface environments. In *Proc. of CHI '07*, pages 473–482. ACM, 2007b.

Daniel Wigdor, Hao Jiang, Clifton Forlines, Michelle Borkin, and Chia Shen. WeSpace: the design development and deployment of a walk-up and share multi-surface visual collaboration system. In *Proc. of CHI '09*, pages 1237–1246. ACM, 2009.

Ryder Ziola, Melanie Kellar, and Kori Inkpen. DeskJockey: exploiting passive surfaces to display peripheral information. In *Proc. of IFIP TC '07*, pages 447–460. Springer-Verlag, 2007.

Index

Typeset May 5, 2011