THE TOOL SPACE

Designing Indirect Touch Input Techniques for Personal Multi-surface Computing Devices

DISSERTATION

an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von
Diplom-Medieninformatiker
HENRI PALLEIS

München, den 21. Februar 2017

Erstgutachter:	Prof. Dr. Heinrich Hußmann
Zweitgutachter:	Prof. Dr. Harald Reiterer

Tag der mündlichen Prüfung: 09.06.2017

To my parents and my brothers.

ABSTRACT

Visions of futuristic desktop computer work spaces have often incorporated large interactive surfaces that either integrate into or replace the prevailing desk setup with displays, keyboard and mouse. Such visions often connote the distinct characteristics of direct touch interaction, e.g. by transforming the desktop into a large touch screen that allows interacting with content using one's bare hands. However, the role of interactive surfaces for desktop computing may not be restricted to enabling direct interaction. Especially for prolonged interaction times, the separation of visual focus and manual input has proven to be ergonomic and is usually supported by vertical monitors and separate – hence indirect – input devices placed on the horizontal desktop. If we want to maintain this ergonomically matured style of computing with the introduction of interactive desktop displays, the following question arises: How can and should this novel input and output modality affect prevailing interaction techniques.

While touch input devices have been used for decades in desktop computing as track pads or graphic tablets, the dynamic rendering of content and increasing physical dimensions of novel interactive surfaces open up new design opportunities for direct, indirect and hybrid touch input techniques. Informed design decisions require a careful consideration of the relationship between input sensing, visual display and applied interaction styles. Previous work in the context of desktop computing has focused on understanding the dual-surface setup as a holistic unit that supports direct touch input and allows the seamless transfer of objects across horizontal and vertical surfaces. In contrast, this thesis assumes separate spaces for input (horizontal input space) and output (vertical display space) and contributes to the understanding of how interactive surfaces can enrich indirect input for complex tasks, such as 3D modeling or audio editing.

The contribution of this thesis is threefold: First, we present a set of case studies on user interface design for dual-surface computer workspaces. These case studies cover several application areas such as gaming, music production and analysis or collaborative visual layout and comprise formative evaluations. On the one hand, these case studies highlight the conflict that arises when the direct touch interaction paradigm is applied to dual-surface workspaces. On the other hand, they indicate how the deliberate avoidance of established input devices (i.e. mouse and keyboard) leads to novel design ideas for indirect touch-based input. Second, we introduce our concept of the tool space as an interaction model for dual-surface workspaces, which is derived from a theoretical argument and the previous case studies. The tool space dynamically renders task-specific input areas that enable spatial command activation and increase input bandwidth through leveraging multi-touch and two-handed input. We further present evaluations of two concept implementations in the domains 3D modeling and audio editing which demonstrate the high degrees of control, precision and sense of directness that can be achieved with our tools. Third, we present experimental results that inform the design of the tool space input areas. In particular, we contribute a set of design recommendations regarding the understanding of two-handed indirect multi-touch input and the impact of input area form factors on spatial cognition and navigation performance.

ZUSAMMENFASSUNG

Zukunftsvisionen thematisieren zuweilen neuartige, auf großen interaktiven Oberflächen basierende Computerarbeitsplätze, wobei etablierte PC-Komponenten entweder ersetzt oder erweitert werden. Oft schwingt bei derartigen Konzepten die Idee von natürlicher oder direkter Toucheingabe mit, die es beispielsweise erlaubt mit den Fingern direkt auf virtuelle Objekte auf einem großen Touchscreen zuzugreifen. Die Eingabe auf interaktiven Oberflächen muss aber nicht auf direkte Interaktionstechniken beschränkt sein. Gerade bei längerer Benutzung ist aus ergonomischer Sicht eine Trennung von visuellem Fokus und manueller Eingabe von Vorteil, wie es zum Beispiel bei der Verwendung von Monitoren und den gängigen Eingabegeräten der Fall ist. Soll diese Art der Eingabe auch bei Computerarbeitsplätzen unterstützt werden, die auf interaktiven Oberflächen basieren, dann stellt sich folgende Frage: Wie wirken sich die neuen Ein- und Ausgabemodalitäten auf vorherrschende Interaktionstechniken aus?

Toucheingabe kommt beim klassischen Desktop-Computing schon lange zur Anwendung: Im Gegensatz zu sogenannten Trackpads oder Grafiktabletts eröffnen neue interaktive Oberflächen durch ihre visuellen Darstellungsmöglichkeiten und ihre Größe neue Möglichkeiten für das Design von direkten, indirekten oder hybriden Eingabetechniken. Fundierte Designentscheidungen erfordern jedoch eine sorgfältige Auseinandersetzung mit Ein- und Ausgabetechnologien sowie adequaten Interaktionsstilen. Verwandte Forschungsarbeiten haben sich auf eine konzeptuelle Vereinheitlichung von Arbeitsbereichen konzentriert, die es beispielsweise erlaubt, digitale Objekte mit dem Finger zwischen horizontalen und vertikalen Arbeitsbereichen zu verschieben. Im Gegensatz dazu geht die vorliegende Arbeit von logisch und räumlich getrennten Bereichen aus: Die horizontale interaktive Oberfläche dient primär zur Eingabe, während die vertikale als Display fungiert. Insbesondere trägt diese Arbeit zu einem Verständnis bei, wie durch eine derartige Auffassung interaktiver Oberflächen komplexe Aufgaben, wie zum Beispiel 3D-Modellierung oder Audiobearbeitung auf neue und gewinnbringende Art und Weise unterstützt werden können.

Der wissenschaftliche Beitrag der vorliegenden Arbeit lässt sich in drei Bereiche gliedern: Zunächst werden Fallstudien präsentiert, die anhand konkreter Anwendungen (z.B. Spiele, Musikproduktion, kollaboratives Layout) neuartige Nutzerschnittstellen für Computerarbeitsplätze explorieren und evaluieren, die horizontale und vertikale interaktive Oberflächen miteinander verbinden. Einerseits verdeutlichen diese Fallstudien verschiedene Konflikte, die bei der Anwendung von direkter Toucheingabe an solchen Computerarbeitsplätzen hervorgerufen werden. Andererseits zeigen sie auf, wie der bewusste Verzicht auf etablierte Eingabegeräte zu neuen Toucheingabe-Konzepten führen kann.

In einem zweiten Schritt wird das Toolspace-Konzept als Interaktionsmodell für Computerarbeitsplätze vorgestellt, die auf einem Verbund aus horizontaler und vertikaler interaktiver Oberfläche bestehen. Dieses Modell ergibt sich aus den vorangegangenen Fallstudien und wird zusätzlich theoretisch motiviert. Der Toolspace stellt anwendungsspezifische und dynamische Eingabeflächen dar, die durch räumliche Aktivierung und die Unterstützung beid-

händiger Multitouch-Eingabe die Eingabebandbreite erhöhen. Diese Idee wird anhand zweier Fallstudien illustriert und evaluiert, die zeigen, dass dadurch ein hohes Maß an Kontrolle und Genauigkeit erreicht sowie ein Gefühl von Direktheit vermittelt wird.

Zuletzt werden Studienergebnisse vorgestellt, die Erkenntnisse zum Entwurf von Eingabeflächen im Tool Space liefern, insbesondere zu den Themen beidhändige indirekte Multitouch-Eingabe sowie zum Einfluss von Formfaktoren auf räumliche Kognition und Navigation.

ACKNOWLEDGMENTS

The work presented in this thesis would not have been possible without the support of many wonderful people. I want to express my deepest gratitude to everyone who has supported me over the last years.

First and foremost, I would like to thank my supervisor Prof. Dr. Heinrich Hussmann. He gave me the opportunity to work at the university, patiently guided me during the highs and lows of pursuing a dissertation and helped me to become a better researcher. Further, I thank Prof. Dr. Harald Reiterer for his genuine interest in my work and for co-supervising my thesis.

I really enjoyed working at the university and I want to thank my awesome colleagues. Although constantly changing, our group has always consisted of a bunch of interesting people and many have become friends. We undertook road trips in countries such as South Korea, USA or Canada, went skiing, visited concerts and festivals, sang and danced, or made music together. I like to thank Emanuel von Zezschwitz, Sarah Tausch, Axel Hösl, Tobias Seitz, Julie Wagner, Daniel Buschek, Renate Häuslschmid, Simon Stusak, Hanna Schneider, Sebastian Löhmann, Doris Hausen, Christian Mai, Maria Fysaraki, Fabian Hennecke, Bastian Pfleging, Alina Hang, Malin Eiband, Ceenu George, Mariam Hassib, Mohamed Khamis, Daniel Ullrich, Max Maurer, and Alexander Wiethoff. I am looking forward to what lies ahead!

I want to thank several students who contributed to this work in one or the other way. My special thanks go to Peter Yu, Mirjam Mickisch, Oliver Baumann, Sarah Aragon Bartsch, Katharina Sachmann, Manuel Demmler, Vanessa Niedermeier and Maximilian Meyer.

I would like to thank Prof. Dr. Andreas Butz and Prof. Dr. Florian Alt for providing valuable feedback during several IDC presentations and for running the B2Run with the team.

I really enjoyed working with Franziska Schwamb and Rainer Fink. I thank Franziska for her patience regarding barkeeper money, milk and coffee. I thank Rainer for always helping me with everything and for discussing cameras with me!

I furthermore would like to thank my friends. Special thanks go to Felix Schwarz for the many discussions that we had about our dissertations and for suffering setbacks and celebrating successes together.

Finally, I would like to express my deepest gratitude to my family for always being there for me. I am very grateful for the encouragement, the never-ending support and the guidance of my parents, Gabi Merz-Palleis and Robert Palleis, and my brothers, Robin and Charlie Palleis. Last but not least, I want to thank Martina Ponath for sharing good and bad times with me. Thank you for your love and patience.

TABLE OF CONTENTS

List of Figures **xv**

List of Tables **xxi**

1 Introduction **1**

1.1 Research Objectives 2
1.2 Research Approach 2
1.3 Contributions 3
1.4 Disclaimer and Conventions 3
1.4.1 Grammatical Choices 5
1.4.2 Review Boxes 5
1.5 Thesis Overview 5

I "BACKGROUND" **9**

2 Interactive Surfaces **11**

2.1 A Brief Introduction 11
2.1.1 Display Devices 12
2.1.2 Touch-sensitive Surfaces 13
2.1.3 Novel Paradigms 17
2.2 Direct Interaction 19
2.2.1 Semantic and Articulatory Distances 20
2.2.2 From Direct Manipulation to Natural User Interfaces 23
2.2.3 Fundamental Challenges of Direct Touch Input 24
2.2.4 Natural User Interfaces 27
2.3 Desktop Surface Computing 28
2.3.1 Novel Workspaces 29
2.3.2 Interaction with Dual-display Workspaces 31

3 Desktop Surface Computing **35**
3.1 Towards Desktop Surface Computing 35
3.1.1 Interaction Paradigms - Desktop Surface Computing 36
3.1.2 Design Factors 36
3.1.3 Thesis Focus 43

II "TOWARDS A NOVEL INTERACTION PARADIGM FOR PERSONAL MULTI-SURFACE COMPUTING" **45**

4 Explorative Case Studies **47**
4.1 Cross-Surface Object Movement 48
4.1.1 Direction Matters: Public Quiz Game 48
4.1.2 Exploring Cross-Display Interaction Techniques 53
4.2 3D Interaction on Dual-Display Setups 61
4.2.1 Background 62
4.2.2 The Prototype 64
4.2.3 Discussion 68
4.3 Chapter Summary 68

III "THE TOOL SPACE: MEDIATED INTERACTION IN DUAL-SURFACE SETUPS" **71**

5 The *Tool Space* Concept **73**
5.1 Background 73
5.1.1 Theoretical Foundations: Tools as Mediators 74
5.1.2 Tool Space: Design Rationale 77
5.1.3 Tool Space: A Precursor 79

6 *Tool Space* Case Studies **83**
6.1 A Toolspace for 3D Modeling 84
6.1.1 Touch-input for 3D-Modeling 87
6.1.2 Design Rationale 89
6.1.3 Prototype Setup 90
6.1.4 Tool Reification for Polygon Modeling 91

6.1.5 Initial User Feedback 101
6.1.6 Evaluation with Novice Users 104
6.2 A Toolspace for Editing Audioclips 108
6.2.1 Designing the *Tool Space* for Audio Editing 108
6.2.2 User Study: Mouse vs. *Tool Space* 109
6.3 Chapter Summary 115

IV ”UNDERSTANDING THE DESIGN OF THE TOOL SPACE” 117

7 Informing the *Tool Space* Design 119
7.1 Exploring Two-handed Multi-Touch Input 120
7.1.1 Background: Two-handed Computer Input 120
7.1.2 Experiment – A Two-handed Docking Task 124
7.2 Understanding the Impact of Form Factors and Transfer Functions 130
7.2.1 Background 130
7.2.2 Experiment 1 - Exploring Touch Input Area Size 132
7.2.3 Experiment 2 - Evaluating the Impact of a Non-linear Transfer Function 137
7.2.4 Chapter Summary 143

V ”REFLECTIONS ON THE TOOL SPACE” 145

8 Conclusion and Outlook 147
8.1 Summary of Contributions 148
8.1.1 Understanding the Requirements of Personal Dual-Display Interaction 148
8.1.2 Conceptualizing a Novel Interaction Paradigm 150
8.1.3 Informing the Design of the *Tool Space* 152
8.2 Limitations 153
8.3 Future Work 155
8.3.1 Application Contexts 155
8.3.2 Suggested Follow-up Studies 156
8.3.3 Directions for Future Research 157
8.4 Closing Remarks 158

Bibliography **161**

LIST OF FIGURES

1.1 Structure of this thesis 6

2.1 Origins of projection systems: (left) drawing of Athanasius Kircher, *Ars magna lucis et umbrae* (Rome, 1646) (from http://web.stanford.edu/), (right) hand-colored engraving of the series *Le Bon Genre* published by Pierre de la Mésangère in 1801 (from Gallica Digital Library). Both images are within the public domain. 13
2.2 Touch-sensitive controllers in musical instruments 14
2.3 Touch screen history 15
2.4 Examples of interactive surfaces 16
2.5 Touch-sensitive displays used throughout my thesis: (a) a pair of capacitive touch LED-displays from Dell, (b) the Curve – a non-flat touch display prototype based on back projection and FTIR, (c) Samsung Galaxy tablet with capacitive touch screen. 17
2.6 Ubiquitous computing 18
2.7 *DigitalDesk* 19
2.8 Surfaces in the context of tangible user interfaces 20
2.9 Gulf of execution and gulf of evaluation 21
2.10 Overview of interaction paradigms. 23
2.11 State-models for graphical input 24
2.12 Solutions for occlusion 26
2.13 TeslaTouch 28
2.14 Examples for touch-based NUIs 29
2.15 *Starfire* 30
2.16 Curved displays 30
2.17 Commercial dual-surface devices 31
2.18 Desktop surface computing 32
2.19 Switching between interaction styles 33
2.20 Themes of novel workspace interaction 34

3.1 Desktop Surface Computing. 37
3.2 Overview of interaction paradigms: (a) the *Magic Desk* (Bi et al., 2011): an integrative approach to surface desktop computing , (b) the *BendDesk* (Weiss et al., 2010): a large non-flat touch display that spans the whole reachable area of a person sitting at it. 38

4.1 Several application scenarios: a) Public intervention (Palleis and Hussmann, 2014), b) digital music production (Palleis, 2014a), c) music analysis, d) photobook layout, e) 3D interaction (Palleis et al., 2015a), and (f) MOBA gaming; (a) and e) are discussed in this chapter). 48
4.2 Conceptual overview: a) piled images on image area, b) semi-transparent image while being dragged onto the map, c) image with red border released on the wrong country, d) image released on the correct country with green border. 50
4.3 Two versions of the quiz game. 50
4.4 Image dragging trajectories from all participants. 54
4.5 Framework for a taxonomy of cross-display object movement interaction techniques with three levels (Nacenta et al., 2009b). 55
4.6 The five interaction techniques (from left to right): (1) Dragging, (2) Hold-and-Point, (3) WIM, (4) Flicking, and (5) Flicking via backdoor. 56
4.7 Task overview: the white source object had to be moved into target areas differing in size and position. 58
4.8 Mean task completion times per interaction technique for upward object movement. Error bars indicate SE. 59
4.9 Mean task completion times separated by different target sizes. Error bars indicate SE. 60
4.10 Examples of novel products targeted at the Maker scene. a) 123D design is an example of a simplified 3D creation and editing application that supports 3D printing. b) The HP Sprout allows to place physical objects on an interactive horizontal surface from where it can be scanned and transferred into the virtual realm. 63
4.11 Using the curved display area to render a seamlessly connected perspective extension of the physical horizontal surface. a) Schematic sketch of the visualization concept *Perspective+Detail* (Schwarz et al., 2012). b) The visualization concept employed to create *PerspectiveTable*, a remote collaboration scenario with a shared workspace (Hennecke et al., 2013). 64
4.12 (1) 3D scene that seamlessly extends into the depth, (2) horizontal touchpad for indirect touch control: the selected object can be translated by one-finger dragging within the XZ-plane and rotated with a two-finger rotation gesture around the Y-Axis, (3) vertical touchpad: the selected object can be translated within the XY-plane and rotated with a two-finger rotation gesture around the Z-Axis, (4) the virtual camera can be rotated around the scene by dragging it left or right. First rotating the scene and then using the vertical touchpad can achieve object rotations around the X-axis. 65
4.13 (left) The visualization of the 3D scene with selection widget and virtual touchpads, (middle) (1) translation constrained to one axis (global z-axis), (2) 2D-translation (xy-layer), (right) scene rotation by dragging the scene left or right. 65

4.14 The transformation manipulator for direct touch interaction (concept and detail). . . . 66

5.1 Conceptual overview of the *Tool Space*: manipulable objects are displayed on the vertical display, tools and meta-tools (tool bar on the left) on the horizontal display. . . . 78
5.2 a) Direct touch gesture event handling, b) gesture event handling for indirect interaction instruments. . . . 79
5.3 An exemplary tool for both axis-constrained and integral 2D object scaling: a) input area for integral scaling b) input area for scaling along the y-axis, and c) input area for scaling along the x-axis. . . . 80
5.4 *Tool Space* precursor (Buxton et al., 1985): (a) cardboard templates with cutouts divide one touch-sensitive surface into multiple tools, enabling spatial activation and task-specific input devices. (b) Later, the cardboard is replaced with virtual tools, which are conceived as invisible *windows*. . . . 81

6.1 Sutherland's Sketchpad system, described in his doctoral thesis at MIT in 1962 (Sutherland, 1963a) with an original drawing. . . . 84
6.2 Different 3D modeling paradigms exemplified in Blender. . . . 85
6.3 Novel 3D modeling paradigms. . . . 86
6.4 Multi-touch gesture set employed in Paper3D (Paczkowski et al., 2014). . . . 88
6.5 Axis-constrained rotation in screen space (Reisman et al., 2009). . . . 88
6.6 TouchTools: gesture design based on familiarity with physical tools (Harrison et al., 2014). . . . 89
6.7 The basic setup consists of the visual display of the scene (vertical screen) and the tool space (horizontal screen). . . . 91
6.8 The basic workflow: (a) adding a mesh to a scene by swiping it up within tool space, (b) selecting an object with direct touch, (c) selecting a tool from the toolbar, and (d) using touch widgets to control the object indirectly. . . . 91
6.9 Edge-loop scaling in Blender. . . . 92
6.10 The bimanual edge loop scaling workflow: (1) the non-dominant hand translates and (2) scales the selection volume, the dominant hand scales the contained edge loops (3). . . . 93
6.11 From left to right: first, the selection volume is positioned and scaled with the non-dominant hand, then the containing edge loops are scaled down. Then the view is slightly changed and the selection volume is repositioned to the top and scaled down. Subsequently, the newly selected edge loops are scaled up. A further change of the camera position shows the resulting 3D model from another viewpoint. . . . 94
6.12 Extrusion in Blender: first, polygons are selected, then extruded along their normal vector and finally translated. . . . 95
6.13 Extrusion tool concept: (a) the scene object, (b) the mesh selection tool, (c) the virtual touch pad for extrusion, and (d) the base layer for scene navigation. 95
6.14 The workflow for selecting surfaces and polygons for extrusion. . . . 96

6.15 Extrusion workflow: from a side view, the already extruded polygons are extruded again to the right, indicated by a dragging gesture on the horizontal bar to the right (left). Then, an upward movement in the vertical bar indicates that the newly created polygons facing up should be extruded (middle). The extrusion could then be continued with the polygons currently facing left, right or up (right). 97
6.16 Initial extrusion of selected polygons. 99
6.17 Creating bent shapes with simultaneous extrusion and camera control. α is the angle of camera rotation and corresponds to the sum of rotation angles of the automatically extruded polygons. 100
6.18 The shape that participants were asked to rebuild during the extrusion task. . 102
6.19 Subjective ratings for the edge-loop scaling and extrusion tasks. 103
6.20 Table modeling task results per participant with the *Tool Space* (top) and Blender/SketchUp (bottom). 103
6.21 Extrusion task results per participant with the *Tool Space* (top) and Blender/SketchUp (bottom). 104
6.22 Models from the combined task. 104
6.23 A participant during one of the sessions. (1) Side view to translate selected objects along the scene's y-axis, (2) top view to translate them in the scene's xz-plane and (3) scaling widget with axis- constraining bars (red, green, blue) at the borders. 105
6.24 Exemplary modeling results from our observational study with five 3D modeling novices. 106
6.25 Conceptual overview of the *Tool Space* for audio editing 109
6.26 Mouse input for audio editing: navigation is supported via the scroll-bar, audio cursor position and selection are controlled via left-click and click-and-drag, zoom is operated with the mouse-wheel. 110
6.27 The input conditions compared in the experiment: (left) the implementation of the *Tool Space*, (right) the mouse-based interaction with the application. . 111
6.28 Results of the questionnaires. Numbers indicate the amount of participants who answered with that score on the 5-Point Likert scale. 113

7.1 Early examples of designing two-handed input techniques. (a) permission requested from SRI, (b) from Sutherland (1963a) 121
7.2 Early examples of designing continuous two-handed input techniques. (a) from `http://www.billbuxton.com/`, (b) from Krueger et al. (1985) . . . 122
7.3 a): overview of the experimental setup. b): the 12 different docking tasks for right-handed and two-handed input. For left-handed input, the layout was flipped. 125
7.4 Mean task completion times per input condition. Error bars indicate SEM. . 126
7.5 Successful square alignment locations in screen pixel coordinates per docking (detail). 128
7.6 Personal preferences and assessments of the three input conditions. 128

7.7 Left: A participant of our experiment performing 2D panning with a virtual touch pad. Right: The visual grid with one spatial item configuration. The highlighted empty grid cells in the center represent the home position. . . . 133
7.8 The three different input conditions used during our experiment. 134
7.9 Navigation efficiency per input condition based on panning distance (error bars indicate standard deviation). 136
7.10 Our test setup: On the horizontal display, a virtual touch pad is used to navigate a view shown on the vertical display. 138
7.11 Left: the grid with the distribution of 18 graphical items and the highlighted empty home position. Right: detail of the view showing the red visual anchor to indicate the home position and the bounding box. 139

8.1 Direct *literal* technique vs. indirect technique for dragging images across the curved display connection: (a) literal techniques are easily detected and understood, but cause arm fatigue. (b) Exploration of an indirect technique: tapping and holding an image for a certain dwell time transforms it into a virtual track pad that allows indirect image movement. 149
8.2 Venues for future research: (a) How should domain objects be presented in a dual-surface workspace? (b) How can existing device ecologies be exploited to form temporary workspaces? 158

List of Tables

6.1 Distribution of ratings for the semantic differential. The numeric values represent the number of users. 108
6.2 Counter-balancing of the conditions 111

1
Introduction

Interactive surface technology has profoundly influenced the way we interact with computing devices. Most visibly, its adoption in mobile computing during the last decade has provided us with ubiquitous access to complex information architectures. *Mobile-first* is a much-heard slogan these days, advocating to cater electronic media primarily to mobile platforms. Further, we have become accustomed to large touchscreens deployed in public spaces, such as interactive exhibits in museums or information terminals at airports or malls. In 2008, the *Economist* speculated that "it may prove to be PCs, rather than hand-helds, that benefit the most from touch-screen technology", however, the core idea of direct interaction using one's own hands is probably not adequate for large-screen or dual-screen professional workplaces. As I will argue in this thesis, the main challenge for innovation in this context results from a *clash of interaction paradigms*: ergonomic workspace design is based on a matured set of physical input devices and graphical user interfaces, which are not easily compatible with direct touch input interfaces.

Adding multi-touch interaction to desktop computing workspaces entails a number of opportunities, such as the increased input bandwidth through two-handed input and multi-finger gestures, enriching the input vocabulary and facilitating compound tasks. We have become fluent with zooming in and out on maps, images or websites on mobile devices, and modern track pads (e.g., Apple's Magic Trackpad) allow us to transfer this vocabulary to the desktop computing context.

However, touchscreens imply a *direct* coupling of input gestures with visual feedback, which causes a variety of context-specific challenges, in particular with regard to ergonomics and precision. For instance, graphical editing tasks can profit from an increased input bandwidth (Latulipe et al., 2006), but finger input precision, occlusion and arm fatigue impede a straight-forward adoption of direct multi-touch editing on vertical touchscreens. Moreover, horizontal and vertical surfaces have different intrinsic properties, each catering well to particular tasks (Morris et al., 2007), which has yielded the recommendation to integrate horizontal and vertical touchscreens into novel interactive workspaces, which I will refer to as *dual-surface* computing workspaces throughout this thesis.

Related research has approached the idea of dual-surface computing workspaces differently: Arai et al. (1995) have proposed *temporal* switching between direct and indirect interaction. In contrast, Bi et al. (2011) presented a systematic approach to *spatially* integrate multi-touch widgets into an existing computer workspace desk. Moreover, Wimmer et al. (2010) emphasized novel possibilities for data exchange between horizontal and vertical touchscreens through a seamless connection (see chapter 2).

1.1 Research Objectives

The importance of *indirect input* styles in the context of dual-surface workspaces is emphasized by recent research, which despite some well-documented benefits of *direct touch input* (e.g., selection times for large targets (Forlines et al., 2007)) has started to systematically investigate *indirect* touch input techniques (e.g., Voelker et al. (2013)). The integration of touchscreens into existing input systems highlights the need to understand their effect on workspace ergonomics and matured indirect input styles. Hence, the first objective of this thesis is to contribute to a better understanding of adequate interaction styles for dual-surface workspaces by identifying relevant design factors.

The second objective of this thesis is to investigate adequate interaction techniques, in particular by providing and evaluating practical examples of *novel indirect touch input techniques*. These examples complement previous concepts from related research, which have explored different notions of combining direct and indirect input (e.g., Bi et al. (2011) or Voelker et al. (2015)).

1.2 Research Approach

To find answers to the research objectives illustrated above, I pursued an approach that incorporated the following aspects:

Literature Review and Definitions In order to understand the domain and establish a theoretical basis for developing and evaluating novel interaction techniques, I studied and iteratively revisited relevant literature regarding interactive surfaces, (indirect) touch input techniques, two-handed input techniques as well as applications-specific fields (e.g., touch-based 3D modeling). Engaging with the literature allowed me to derive working definitions and discover research opportunities.

Explorative Case Studies Exploration was another means I used to develop a sense for requirements and research opportunities in the context of novel computing workspaces incorporating multi-touch surfaces. Throughout several case studies, I developed various prototypes, sometimes with the help of students. To cover a broader spectrum of potential insights, I focused on different application areas, such as gaming or creative applications. Applying tools and methods known from user-centered design and Human-Computer Interaction, I gathered user feedback and conducted formative studies.

Conceptual Work Based on the formative case studies and literature revisits, I derived a conceptual interaction paradigm for personal dual-surface computing devices. The conceptual work constitutes an "inventive" step throughout my dissertation work, albeit its assumptions are well-grounded in related work and my own observations.

Evaluation In order to investigate some of the assumptions underlying the conceptual work and to demonstrate a way to derive design guidelines for the *Tool Space*, I conducted several formal lab experiments that involved interactive tasks, surveys and semi-structured interviews.

Retrospective Finally, in this thesis, I strive to complement the theoretical, explorative, conceptual and evaluative aspects of my work with a retrospective. I documented the tools and methods I used for prototyping, exploring and evaluating novel touch interaction techniques for multi-surface computing devices and derive a set of overarching comments on my work and provide an outline of future research directions.

1.3 Contributions

The contribution of this thesis is threefold:

Contribution 1: Case studies on user interface design for dual-surface computer setups First, I present a set of case studies on user interface design for dual-surface computer workspaces. These case studies cover several application areas, such as gaming or 3D interaction, and comprise formative evaluations. On the one hand, these case studies highlight the conflict that arises when the direct touch interaction paradigm is applied to dual-surface workspaces. On the other hand, they indicate how the deliberate avoidance of established input devices (i.e. mouse and keyboard) leads to novel design ideas for indirect touch-based input.

Contribution 2: *Tool Space* as conceptual interaction model for dual-surface setups Second, I introduce the concept of the tool space as an interaction model for dual-surface workspaces, which is derived from a theoretical argument and the previous case studies. The tool space dynamically renders task-specific input areas that enable spatial command activation and increase input bandwidth through leveraging multi-touch and two-handed input. I further present evaluations of two concept implementations in the domains 3D modeling and audio editing which demonstrate the high degrees of control, precision and sense of directness that can be achieved with our tools.

Contribution 3: Experimental results that inform the design of indirect touch tools Third, I present experimental results that inform the design of the tool space's tools. In particular, I contribute a set of design recommendations regarding the understanding of two-handed indirect multi-touch input and the impact of input area form factors on spatial cognition and navigation performance.

1.4 Disclaimer and Conventions

While the work presented in this thesis is mainly based on my own original work, I occasionally collaborated with colleagues and students. Further, parts of this thesis are based on papers that have been published at international peer-reviewed conferences. Some of the projects have been supported by students working on their Master's or Bachelor's theses. In the following, I list both the publications and the unpublished students' theses underlying my work and outline the respective contributors.

Part 2. Towards a Novel Interaction Paradigm for Personal Multi-Surface Computing

This section is based on several publications and collaborative projects with students.

(Palleis and Hussmann, 2014) is based on a formative project that I pursued without the help of students. The paper was written by me and revisions were made based on the feedback provided by Heinrich Hussmann.

(Palleis et al., 2015a) is based on a project where I was supported by Mirjam Mickisch, who wrote her Bachelor's thesis under my supervision. Starting with my own software prototype for 3D interaction on the curved display, the work was conducted in a collaborative manner and each step in the project was jointly discussed in weekly meetings. However, as a supervisor, I was the key decision maker on how to proceed in the project (e.g., study design, execution and evaluation). The paper was written by me; revisions to the paper were also made by me and were based on the feedback by Heinrich Hussmann.

Section 4.1.2 is based on a Master's thesis submitted by Peter Yu (Yu, 2015), who implemented a set of interaction techniques and conducted a user study. Again, as a supervisor, I was the key decision maker on how to proceed in the project and how to conduct and evaluate the user study.

Finally, (Palleis, 2014b) has been a doctoral consortium paper that summarized my formative projects and was written by me.

Part 3. The Tool Space: Mediated Interaction in Dual-Surface Setups

This section is based mainly on (Palleis et al., 2016), a paper written by me that establishes the Tool Space paradigm and exemplifies it using a self-developed 3D modeling application. Throughout the project, I had fruitful discussions with Julie Wagner – one of the co-authors and a post-doc at our lab back at the time –, who also supported me in finding participants for the first user feedback session. Revisions to the paper were made by me based on the feedback provided by Julie Wagner and Heinrich Hussmann. This paper further reports on a multi-session study that was designed and evaluated by me, but was conducted by the students Sarah Aragon Bartsch, Manuel Demmler and Katharina Sachmann during the course Advanced Topics in HCI.

Further, a second example of the Tool Space is given based on Oliver Baumann's Bachelor's thesis. Based on an audio editing software I had built for the curved display, he implemented a Tool Space that we devised collaboratively in weekly meetings. Further, I was the decision maker concerning the design, execution and evaluation of the user study conducted by him.

Part 4. Understanding the Design of the *Tool Space*

The first part of this section is based on (Palleis and Niedermeier, 2016), a paper I have written based on Vanessa Niedermeier's Bachelor's thesis. Under my supervision, she contributed both to the implementation of the study prototype and the study design, and conducted the study, which was then evaluated by me.

The second part is based on (Palleis and Hussmann, 2016), a paper based on a study designed, conducted and evaluated by me. The paper was written by me and revisions were based on the feedback provided by Heinrich Hussmann. Based on this project, student Maximilian Meyer wrote his Bachelor's thesis. Maximilian made slight adaptations to the existing prototype according to my specifications, conducted research concerning transfer functions, implemented a transfer function and executed a study, which was designed and evaluated by me. Again, as a supervisor, I was the key decision maker on how to proceed in the project and how to conduct and evaluate the user study.

1.4.1 Grammatical Choices

As described above, some projects were supported by students and colleagues. To acknowledge their invaluable contribution to the work presented in this thesis, I will use the scientific plural in the respective sections. For the work that I conducted alone, I will use the singular form.

1.4.2 Review Boxes

Review Throughout this thesis, I will repeatedly provide short summaries of sections and projects that will be contained in gray review boxes.

1.5 Thesis Overview

This thesis is divided into eight chapters and five main parts (see figure 1.1). In the first part, I provide an overview of existing research and outline factors for designing surface-enhanced workspaces. In the second part, I present a set of formative case studies, which complement the theoretical background. The third part introduces the *Tool Space*, a novel interaction paradigm for dual-surface workspaces, and exemplifies it with two case studies. The fourth part presents empirical findings with the goal to inform the *Tool Space* design. Finally, the fifth part concludes this thesis with a retrospective.

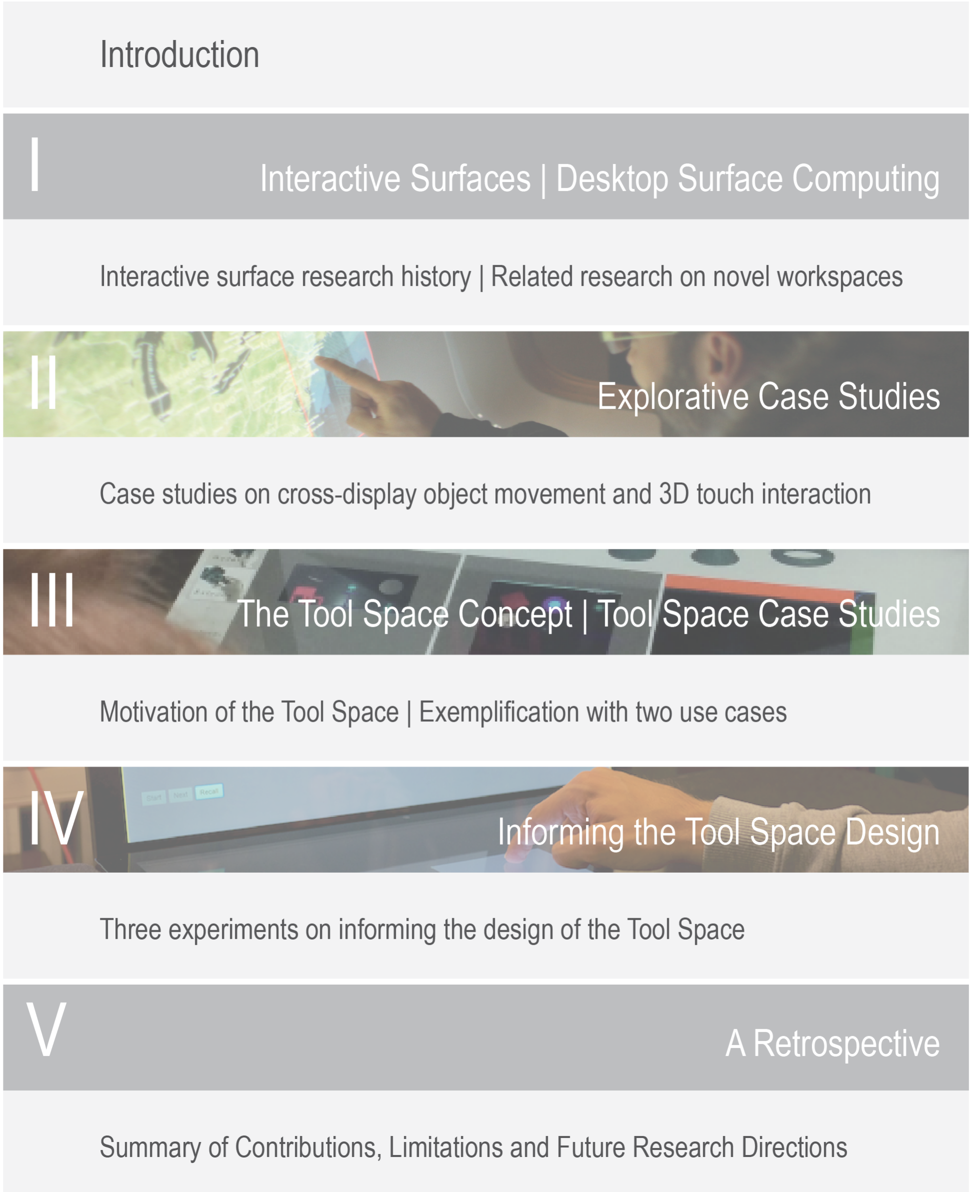


Figure 1.1: Structure of this thesis

Chapter 1: Introduction The first chapter motivates the topic of personal multi-surface computing devices. It further outlines the research objectives and the main contributions.

Part I: Background

Chapter 2: Interactive Surfaces In the second chapter, I provide an overview of interactive surface research history, novel interaction paradigms and discuss different notions of directness in the context of human-computer interaction. I further present particular challenges of surface input and motivate introduce related research on novel workspaces based on interactive surface technology.

Chapter 3: Desktop Surface Computing Based on the second chapter, the third chapter discusses various factors involved in the design of surface-enhanced workspaces. In particular, it introduces the notion of competing interaction paradigms and substantiates the focus of this thesis with a set of assumptions.

Part II: Towards a Novel Interaction Paradigm for Personal Multi-surface Computing.

Chapter 4: Explorative Case Studies In the fourth chapter, I present a choice of formative case studies that I have conducted in an early phase of my work. These studies include prototypes built in the context of a curved display and involve observations and lab studies. In particular, they concern cross-display object movement – i.e. moving objects between horizontal and vertical displays – and 3D touch interaction. The case studies have helped to understand the challenges that arise from building direct touch interaction techniques for dual-surface workspaces and led to a refinement of the research objectives.

Part III: The Tool Space*: Mediated Interaction in Dual-surface Setups*

Chapter 5: The *Tool Space* Concept In chapter five, I introduce the *Tool Space* paradigm – an interaction paradigm that separates dual-surface setups in input and output surfaces, and introduces the notion of virtual input devices, which are displayed and operated on a horizontal touch-sensitive *Tool Space*.

Chapter 6: *Tool Space* Case Studies Chapter six exemplifies the paradigm with two examples. First, a 3D modeling use case is introduced, which explored the reification of existing 3D mesh manipulation tools (edge-loop scaling and extrusion). Further, the findings from two preliminary user studies are presented – one involving expert, the other novice users. Second, the chapter discusses a prototypical *Tool Space* for audio editing and reports findings from a small experiment that compared the *Tool Space* to mouse-input.

Part IV: Understanding the Design of the Tool Space

Chapter 7: Informing the *Tool Space* Design In the seventh chapter, I present empirical results from three experiments, which aimed at informing the design of virtual tools for the *Tool Space*. The first experiment investigated our ability to cooperatively use both hands in tasks involving one- and two-finger touch gestures. The second and third experiment investigate the effect of input area size and non-linear transfer functions on a 2D navigation task.

Part V: Reflections on the Tool Space

Chapter 8: Conclusion and Outlook Chapter 8 summarizes the main findings and contributions of this thesis. Further, I discuss limitations of the work, in particular concerning the small sample sizes of the conducted experiments and the generalizability of the findings. Finally, I outline potential follow-up studies and future research directions.

I

"BACKGROUND"

2
Interactive Surfaces

As my research objectives involve a variety of topics from interactive surface research, this section introduces related work to establish the terms and theory relevant for this thesis. First, it provides a short introduction to interactive surface technology. Then, different surface computing paradigms are introduced by referencing seminal works from the 1990s, such as *Ubiquitous Computing*, *Augmented Reality*, or *Tangible User Interfaces*.

Subsequently, the question of input directness is discussed by providing an overview of interaction paradigms. In particular, different notions of directness will be discussed, such as the spatial configuration of input and output, the involved modalities and bandwidth, and user interface design.

Eventually, the working terminology *Desktop Surface Computing* is motivated, which comprises several key aspects of my work. Starting with an overview of existing approaches towards novel workspaces involving interactive surface technology, I then introduce different themes of research regarding the interaction in such contexts. More specific related work, which is especially relevant to my individual research projects, is presented in the beginning of the respective sections.

2.1 A Brief Introduction

The term *interactive surface* has not been well-defined in the literature and its emergence can rather be attributed to advancements in interactive display technology on the one hand, and conceptual work reinterpreting digitally enhanced physical surfaces on the other.

In 2006, the *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems Tabletop* defined its scope with the following description [1]:

"Interactive surfaces is emerging as an exciting new research area. Display technologies coupled with input sensors capable of enabling direct interaction, are being experimented with and embedded in tabletops, walls and floors to support a diversity of collaborative activities".

[1] `https://www.interaction-design.org/literature/conference_series/tabletop`

2.1.1 Display Devices

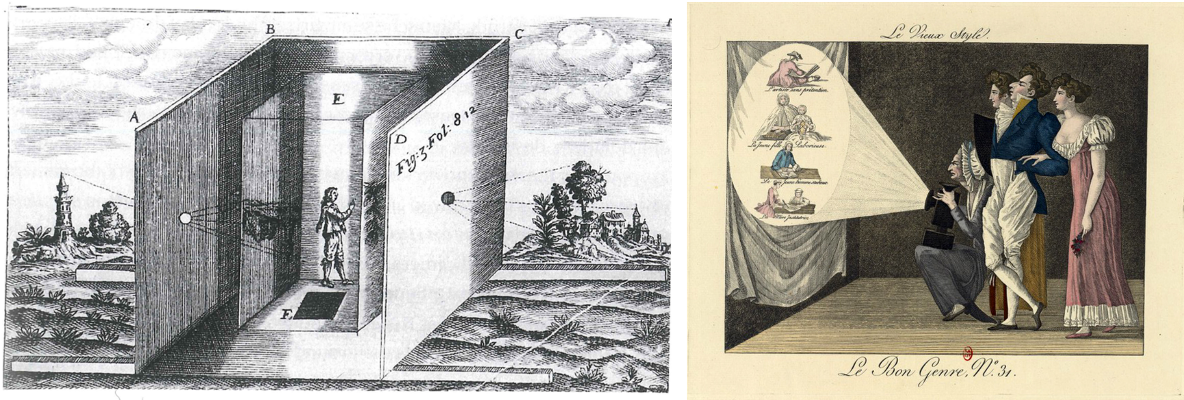
Regarding display technologies, it is distinguished between electronic visual displays, i.e. screens, and projection. While both technologies are based on a per-pixel generation of light, they differ regarding the spatial configuration of image generation and presentation: with screen technologies (e.g., CRT, LCD, plasma, LED), the light-emitting pixels are integrated into the presentation surface, whereas with projection technologies, images are projected onto arbitrary – passive – surfaces.

Screens The first screens were based on cathode rays, a term introduced in Goldstein (1876) for a phenomenon discovered earlier by German physicist Johann Wilhelm Hittorf. So-called CRT-displays have been widely used as television and computer displays throughout the 20th century. Several other technologies enabled the development of displays that adapt to novel contexts (e.g. mobile computing, public displays), such as liquid crystal displays, plasma displays, or light emitting diode displays. In contrast to these light emitting – hence *active* – screen technologies, passive screens include different electronic paper technologies. For an overview of screen technologies, refer to Meyer (1995) or Gurski and Quach (2005).

Projection The origins of passive display technologies can be traced back to the Arabic scholar Alhazen, author of a pioneering treatise on optics and inventor of the first camera obscura, who first described the principles of pinhole projection – a phenomenon known already by Aristotle and Euclid (Belting, 2008, p. 110). In contrast to the camera obscura, which projected an image of the real world outside its pinhole onto an interior surface, the so-called optical lanterns – later referred to as *Magic Lanterns* – were the first devices that projected exchangeable image carriers – typically glass slides mounted on a circular disk – against arbitrary physical surfaces (Friedberg, 2006, p. 70). Current projector technologies are based on the same principle, but use digital image carriers.

An extensive discussion about the history of visual display technologies is outside the scope of this thesis (see for example (Chen et al., 2012)), however, it is noteworthy that apart from technological aspects, the theory and practice of employing *windows as metaphors for displays*, framing our views on dematerialized or virtual worlds, has a long tradition in the Western world [2]. In *The Virtual Window. From Alberti to Microsoft.*, Friedberg (2006) gives an extensive account of how this metaphor has shaped our viewing experiences, and shows how both the transfer of the window metaphor to the virtual realm of the computer screen and today's ubiquity of electronic display has "brought the computer closer to the other predominant forms of visual imaging" (Friedberg, 2006, p. 222). In particular, she gives an insightful account on Douglas Engelbart's multiple-screen display approach (Engelbart and English, 1968), and Alan Kay's introduction of the terms *window* and *viewport* (Kay, 1969).

[2] In his treatise on painting from the 1430s, Leon Battista Alberti redefines the image as one of many possible intersections of the viewing frustum and its projection, which he refers to as *windows*.



(a) Camera obscura (b) Magic Lantern

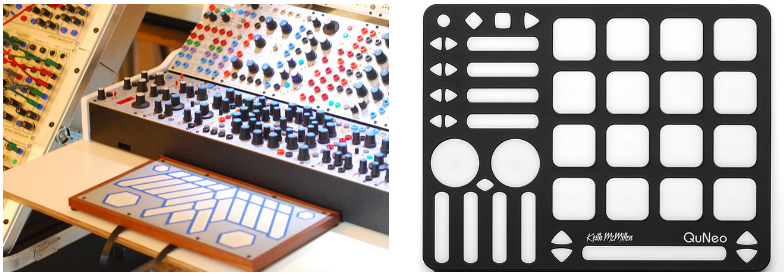
Figure 2.1: Origins of projection systems: (left) drawing of Athanasius Kircher, *Ars magna lucis et umbrae* (Rome, 1646) (from http://web.stanford.edu/), (right) hand-colored engraving of the series *Le Bon Genre* published by Pierre de la Mésangère in 1801 (from Gallica Digital Library). Both images are within the public domain.

2.1.2 Touch-sensitive Surfaces

Regarding touch-sensitive displays, a distinction between touchscreens and interactive surfaces seems useful. While touchscreens have emerged in the context of active display technologies and emphasize the screen as interaction device, interactive surface research has been concerned with transforming arbitrary physical surfaces, e.g. walls, tabletops, floors, or non-flat surfaces such as advertising columns into interactive displays. While the combination of passive display technology – i.e. projection – and optical touch sensing has been an enabler for prototyping interactive surfaces, advancements of active display technology increasingly enable the construction of large and even non-flat interactive surfaces (Gurski and Quach, 2005).

Touch pads

Buxton defines *directness* as a condition where "you touch exactly where the thing you are interacting with is" (Buxton, 2007). Given this definition, he differentiates between indirect touch pads or tablets and direct touch screens. While technically not displays, touch-sensitive control devices have a rich history that pre-dates the age of the personal computer, particularly in the context of musical instruments. In contrast to personal computing touch pads, musical controllers sometimes consist of a composition of multiple touch-sensitive areas (see figure 2.2(a)). With novel controllers like the QuNeo (see figure 2.2(b)), which combine touch pads with underlying multi-color LEDs, the adequacy of this definition may be challenged, as the pads might *temporarily* become "the thing" one is interacting with.



(a) Buchla 200e (b) QuNeo

Figure 2.2: Touch-sensitive controllers in musical instruments: (a) Buchla 200e from 1970, image by GeschnittenBrot licensed under Creative Commons Attribution-Share Alike 2.0 Generic. (b) QuNeo controller by Keith McMillen, image by Keith McMillen Instruments licensed under Creative Commons Attribution-Share Alike 4.0 International.

Touch screens

Touch screens integrate visual display and visually transparent contact sensing technologies. From a technological perspective, capacitive and resistive touchscreens are prevalent in today's devices. Additionally, there is a history of optical touchscreens (e.g., HP-150 [3]), which has regained relevance for large touch screens. As touch sensing technology is not in the focus of this thesis, I only present a short historical perspective. A more extensive overview of touch sensing technologies is provided for instance by Schöning et al. (2008).

E.A. Johnson is widely accepted as the inventor of the touchscreen. In the 1960s, he proposes and explains the construction of a capacitive touchscreen based on a CRT-display (Johnson, 1965, 1967) (see figure 2.3(a)). In the 1970s, capacitive sensing has also been successfully deployed in the Plato IV system (see figure 2.3(b)), an educational workstation equipped with a plasma display and a 16 x 16 grid of touch sensing locations that allowed students to answer questions using finger input (Buxton, 2007).

Touchscreen devices have long been niche products, embedded in ATM machines or public information kiosks (Buxton, 2010). In 2007, Apple's successful introduction of the iPhone has initiated a popularization of touchscreen devices (Lee, 2011), such as smartphones (figure 2.3(c) shows IBM's Simon Personal Communicator from 1993), tablets and also electronic book reading devices, which are now ubiquitous.

Interactive surfaces

Amongst the many usage contexts of the camera obscura, its early use as a drawing tool anticipated current challenges of using projection for surface input: in order to avoid occluding the projected image while drawing and to correct the optical left-right inversion, Leonardo da Vinci proposed the use of *back projection*, i.e. to use a semi-transparent sheet of paper and draw from the backside (Friedberg, 2006, p. 62).

[3] `https://en.wikipedia.org/wiki/HP-150`

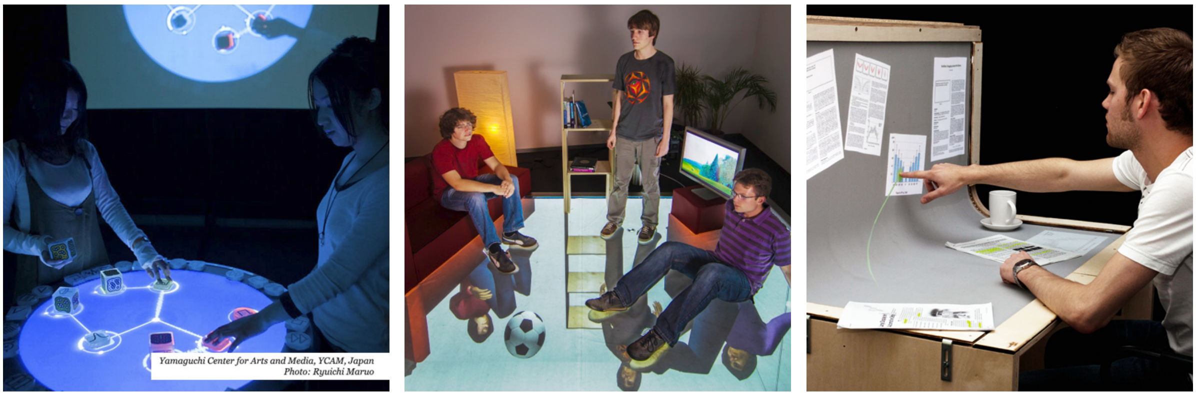
(a) First touchscreen (b) Plato IV (c) Smartphone predecessor

Figure 2.3: Touch screen history: (a) touchscreen presented by Johnson (1967), (b) Plato IV educational workstation with 16 x 16 capacitive grid and (c) *Simon*, smartphone predecessor with touchscreen by IBM and Bell South from 1993, both (Buxton, 2010).

The rediscovery of frustrated total internal reflection (FTIR) to enable optical multi-touch sensing (Han, 2005) has fueled the exploration of interactive surfaces in research because it allowed back-projection – a feature not provided by other touch sensing technology due to their opaqueness. Combined with a projection system, this technology allowed to inexpensively prototype large touch-sensitive surface displays. Subsequently, back projection in combination with optical touch sensing (see figure 2.4(a)) has been used in a wide variety of interactive surface prototypes, including tabletops (e.g., Jordà et al. (2007), figure 2.4(a)) , floors (e.g., Bränzel et al. (2013), figure 2.4(b)) , or curved displays (e.g., Wimmer et al. (2010) or Weiss et al. (2010), figure 2.4(c)). Other optical touch sensing technologies include camera-based tracking, e.g., (Jordà et al., 2007; Ullmer and Ishii, 1997), or video-based tracking in combination with acoustic sensing, such as in (Wellner, 1993).

Recent trends in interactive surface research are concerned with making projection-based touch displays portable, in order to render virtually any surface interactive. For instance, (Wilson, 2005) proposed *PlayAnywhere*, a self-contained projection and touch recognition system that can be placed on horizontal surfaces to create interactive tabletops ad-hoc. *OmniTouch* (Harrison et al., 2011a) employs a shoulder-mounted projection-tracking device that can turn any surface within reach interactive, including own body parts.

Beyond flat surfaces Graphical user interfaces are inherently capable of creating dynamic affordances (Norman, 2013), whereas tangible (or graspable) user interfaces may provide physical affordances (Ishii and Ullmer, 1997). A recent technological trend aims at building interactive systems that go beyond the flatness of surfaces and employ physical objects to dynamically display and manipulate information. For instance, Follmer et al. (2013) present a dynamic shape display: with a matrix of actuated height-changing bars, *physical dynamic affordances* become feasible.



(a) Tabletop (b) Interactive floor (c) Curved display

Figure 2.4: Examples of interactive surfaces: (a) the Reactable (Jordà et al., 2007), (b) GravitySpace (Bränzel et al., 2013), (c) BendDesk (Weiss et al., 2010)

Another body of recent work is concerned with building touch sensing (Gong et al., 2014) and touch screens (Olberding et al., 2014) into the design process of arbitrarily-shaped or flexible objects through printing, allowing to easily manufacture prototypes with built-in direct touch interaction.

Sensing properties Throughout the development of touchscreen and surface technologies, various sensing capabilities have evolved. Certain aspects, such as the increasing amount of simultaneously recognizable touch points, have been improved across various technologies. Others have been enabled by specific sensing hardware. For instance, the recognition of visual markers (Jordà et al., 2007) and contact shapes (Cao et al., 2008) has been enabled by optical sensing systems. In general, relevant properties include pressure (e.g. Lee et al. (1985)), force vectors (e.g., Herot and Weinzapfel (1978); Minsky (1984)), pre-touch properties such as angle of approach (e.g., Wang and Ren (2009) or Rogers et al. (2011)), user identification (e.g., Dietz and Leigh (2001)), and finger modes (e.g., Harrison et al. (2011b)).

Devices used for this thesis The starting point for this thesis is the idea of integrating interactive surfaces into the existing personal desktop computing environment. As such, there are no requirements that dictate the use of any specific surface technology, as long as multi-point input and graphical output is available. In fact, throughout the various projects presented in this thesis, different technologies have been employed: parts of the work presented in this thesis have been implemented using currently available touchscreens. For instance, the Dell S2340T[4] is based on an LED-display with capacitive sensing, capable of recognizing up to 10 simultaneous touch points and a set of touch gestures (figure 2.5(a)). The Samsung Galaxy Tab 2.0 is based on a LC-display and also uses capacitive sensing. Other projects have been conducted using an interactive surface prototype based on back-projection and optical touch sensing (figure 2.5(b)).

[4] `http://www.dell.com/ed/business/p/dell-s2340t-multi-touch-monitor`

(a) Dell S2340T (b) Curve (c) Samsung Galaxy Tab

Figure 2.5: Touch-sensitive displays used throughout my thesis: (a) a pair of capacitive touch LED-displays from Dell, (b) the Curve – a non-flat touch display prototype based on back projection and FTIR, (c) Samsung Galaxy tablet with capacitive touch screen.

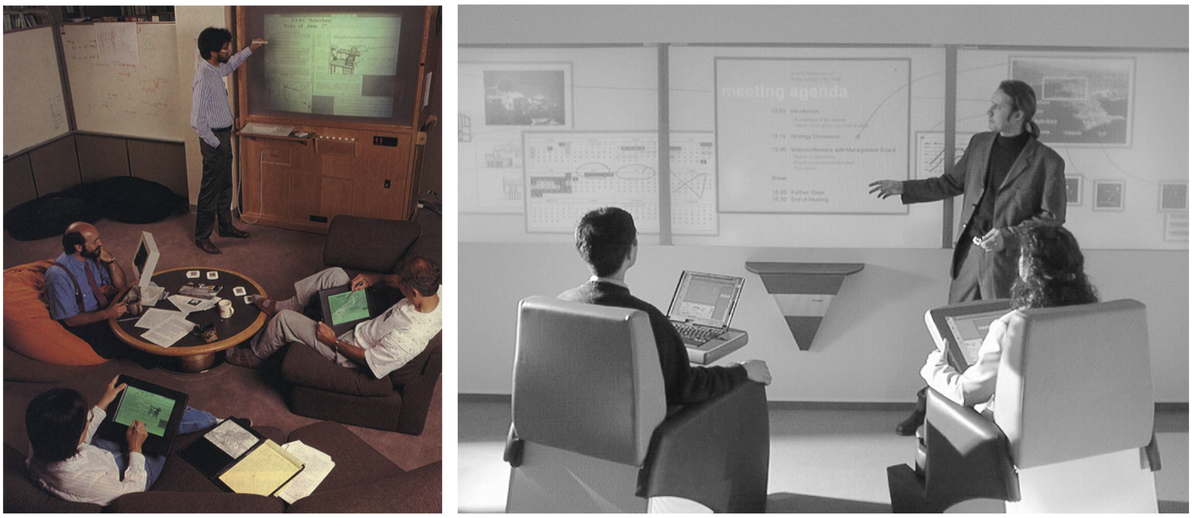
2.1.3 Novel Paradigms

The establishment of graphical user interfaces and the increasing ubiquitousness of interactive surface technology has inspired a variety of novel computing and interaction paradigms. Throughout the conception of these paradigms in the 1990s (Ishii and Ullmer, 1997; Weiser, 1991; Wellner et al., 1993), we can observe a strong opposition of *virtual reality* and a desire to interact with an increasing amount of interactive systems through the physical environment. In the following, I will briefly outline different conceptions of how interactive surface technology has been thought to impact the interaction with digital information systems.

The computer for the 21st Century

In *The computer for the 21st century* (Weiser, 1991), the term *ubiquitous computing* is coined to describe an era where each user operates a multitude of computing devices. Alluding to literacy as an information technology that does not require active attention and provides insights at a glance, Weiser argues for the disappearance of computers into the background of our lives, in a way that they "weave themselves into fabric of everyday life until they are indistinguishable from it". In his vision, the demanding focus of a single personal computer screen is replaced with numerous location-aware touchscreen devices of different scales: tabs, pads and boards have graphical user interfaces and are operated with fingers or pens. With regard to today's device ecologies – smartphones, tablet devices, interactive televisions, smartboards, or public displays – this vision has turned out astonishingly precise.

Weiser strongly opposes the idea of *virtual reality* and instead highlights the importance of the natural human environment, thus extending the task of interface design with architectural and social aspects. For instance, Streitz et al. (2001) proposed a design framework that embeds human-computer-interaction, ubiquitous computing, computer-supported cooperative work and augmented reality into architectural planning. Based on their framework, they developed *roomware®* components, which they define as "computer-augmented objects resulting from the integration of room elements – walls, doors, furniture – with computer-based information devices" (Streitz et al. (2001)).



(a) Tabs, Pads, Boards

(b) Roomware components

Figure 2.6: Ubiquitous computing: (a) Mark Weiser's vision of device ecologies (Weiser, 1991), (b) *roomware®* components, including wall displays and chairs with integrated touchscreens (Streitz et al., 2001).

Augmented Reality

In their article *Back to the Real World* (Wellner et al., 1993), motivate a similar desire of integrating human-computer interaction and the physical environment more tightly by asking: "From the isolation of our workstations we try to interact with our surrounding environment, but the two worlds have little in common. How can we escape from the computer screen and bring these two worlds together?" The seeming paradox created by Weiser's postulation of disappearance through device abundance (Cooperstock et al., 1995), is avoided by focusing on "appropriate tools that enhance our daily activities".

From a technical perspective, the key idea is to visually overlay digital information onto physical objects, for instance by means of *see-through head-mounted displays* (e.g., Feiner et al. (1993)), or interactive environments (e.g., Cooperstock et al. (1995)), where physical surfaces are augmented through visual projection.

A key project in this area is Pierre Wellner's *DigitalDesk* (Wellner, 1993). Here, a computer display is projected from the top onto the physical surface of an office desk including typically contained objects such as paper sheets, a coffee cup, or pencils (see figure 2.7(a) for a schematic overview). The *DigitalDesk* further enabled finger and pen interaction (see figure 2.7(b)), as well as reading paper documents.

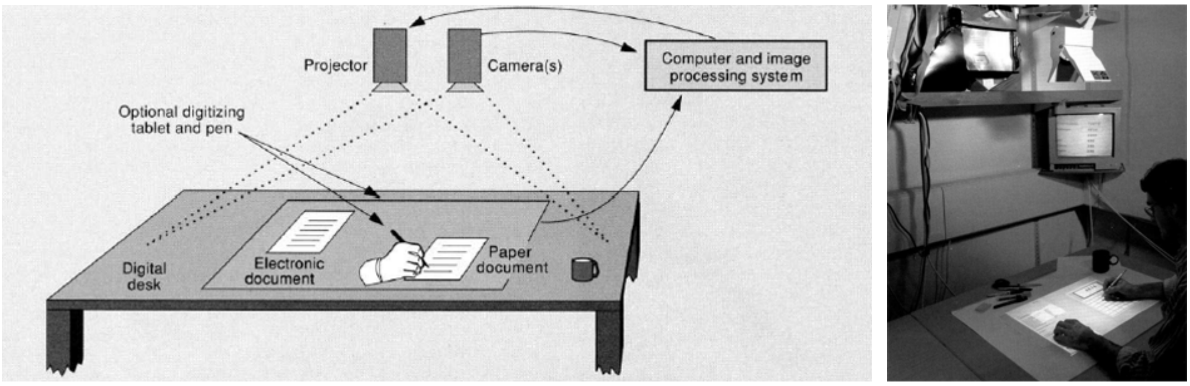


Figure 2.7: *DigitalDesk*: (a) schematic overview of the top projection and computer-vision setup, (b) interacting with the desk using finger and pen input (Wellner, 1993).

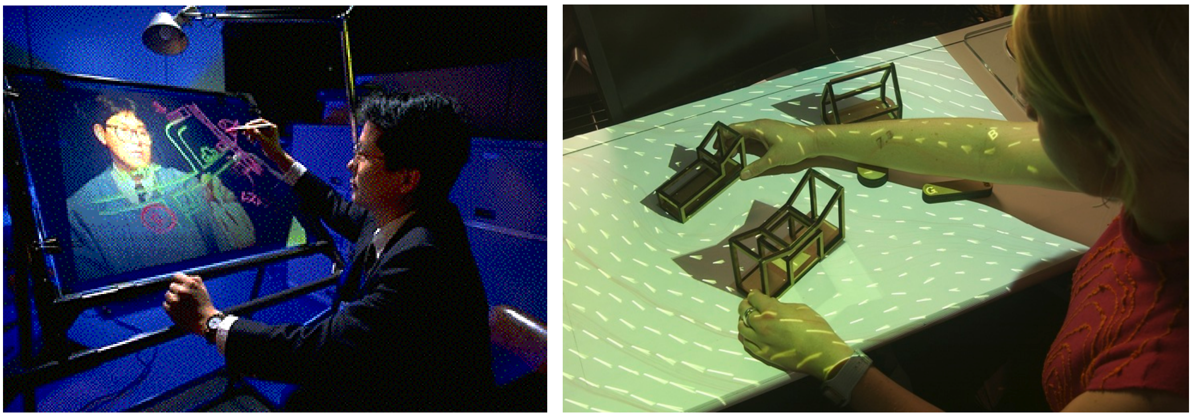
Tangible User Interfaces

Interactive surfaces also play a distinctive role in the concept of tangible user interfaces: the "Transformation of each surface within architectural space (e.g., walls, desktops, ceilings, doors, windows) into an active interface between the physical and virtual worlds" (Ishii and Ullmer (1997)) is a key concept of making digital information accessible through the physical environment. A difference to Weiser's vision outlined above is the interaction metaphor: while tabs, pads and boards are conceived as ubiquitous touchscreen computer devices operated through graphical user interfaces, Ishii and Ullmer (1997) aim at "richly-afforded" interaction through physical objects, surfaces and spaces. Further, compared to augmented reality, the focus is shifted from purely extending physical objects with graphical user interfaces to mediating interaction through the objects themselves.

Prior to establishing the concept of tangible bits, Ishii's project *ClearBoard* (Ishii and Kobayashi, 1992) already introduced the notion of transforming a passive wall into an active interface for remote collaboration (see figure 2.8(a)). *Urp* (see figure 2.8(b)) is an example for the combination of horizontal interactive surface and physical objects. Here, *tangible* architectural models can be placed and moved on the interactive surface to control an urban planning simulation. The surface itself provides visual feedback of the simulation, such as accurate shadows or wind patterns resulting from objects' positions.

2.2 Direct Interaction

The desire to integrate computing and our natural environment more tightly yields a novel importance of surfaces: the touchscreen as the successor of the screen becomes ubiquitous (Friedberg, 2006; Weiser, 1991), and physical surfaces, such as walls or tables, are transformed into active interfaces that allow *direct* access to digital information via touch input.



(a) ClearBoard (b) Urp

Figure 2.8: Surfaces in the context of tangible user interfaces: (a) Ishii's *ClearBoard* employs the mirror as a metaphor for remote collaboration (Ishii and Ullmer, 1997), (b) urban planning with *Urp*, image from `http://tangible.media.mit.edu/project/io-bulb-and-luminous-room/`

Yet, discussing directness in the context of human computer interaction is a challenging endeavor as it is a property used in different contexts and on a variety of levels. In the context of interactive surfaces, direct input is a prominent facet of interaction that allows users to manipulate displayed information directly in a *spatial* sense, i.e. "you touch exactly where the thing you are interacting with is" (Buxton, 2007). Further, directness has been understood as the *absence of artificial input devices* (e.g., Cas (2005) or Moscovich and Hughes (2006)), but also more broadly as a paradigm of embedding the interaction with computing devices more *seamlessly into the physical world* (Ishii and Ullmer, 1997; Jacob et al., 2008; Weiser, 1991).

However, directness more broadly connotes aspects of engagement and distinctive distances between human operators and computing interfaces. Hutchins et al. (1985) describe an *information processing distance* between user intents and according facilities provided by interactive systems on the one hand, and the relation between input and output vocabularies on the other hand. In the following, I will provide a short outline of how *directness* has been used to describe properties of human-computer interaction.

2.2.1 Semantic and Articulatory Distances

A holistic notion of directness is conveyed in (Hutchins et al., 1985). Here, semantic and articulatory distances influence both the *gulf of execution* and the *gulf of evaluation* (see figure 2.9).

Semantic distances are conceived as gaps between user and interface vocabulary. Regarding input, the gap's size is determined by the similarity between *conceptual* user task and the

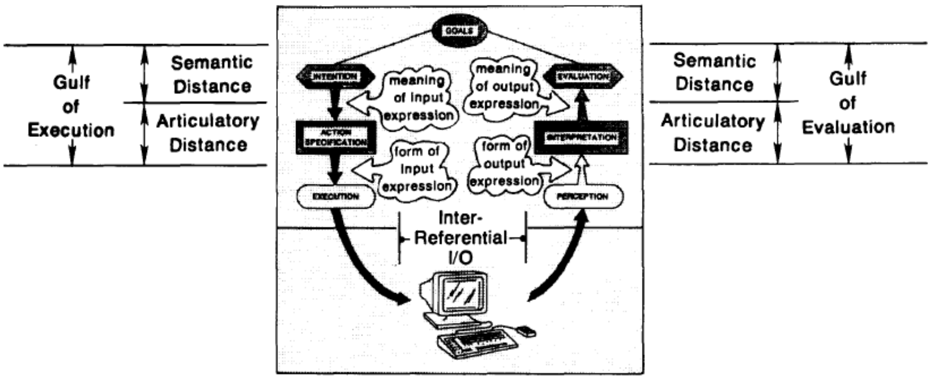


Figure 2.9: Forming intentions and interpreting results in interactive computing systems: semantic and articulatory distances determine both the gulf of execution and the gulf of evaluation (Hutchins et al., 1985).

language provided by an interface for a specific task. Similarly, the semantic distance between system output and human operator describes the amount of reasoning a user has to do in order to properly evaluate the outcome of previous actions.

With regard to articulation, techniques that are built upon spatial mimesis of the user's intent on the one hand, and directly coupled graphical feedback on the other – for instance moving a cursor with a finger on a screen – exhibit a smaller articulatory distance than structural or arbitrary relationships.

Directness as the absence of mediation In his account of *user interaction generations* , John Walker, founder of *Autodesk*[5], points out that early mainframe computers with physical knobs and dials operated by experts exhibited a rather direct relationship: "Since the user was the operator of the machine and controlled it with little or no abstraction, there was essentially no mediation between the computer and its expert user." (Walker (1990))

The directness provided by the touchscreen in this sense has early been considered as one of its biggest benefits (Sears et al., 1992): in particular, it has been conceived to reduce attention shifts and hand mappings between input device and objects of interest and to support gestural interaction. In contrast, separate loci of input and output often refer to indirect input techniques, such as using a touch tablet for input and a monitor for visual output (Buxton, 2007).

Indirectness as a consequence of abstraction The introduction of batch processing introduced new levels of mediation and abstractions, which resulted in semantic and articulatory distances: commands were specified with punchcards (figure 2.10(a)), which were prepared using specialized machines and then handed over to an operator. Feedback was received only after the computer had performed a whole batch of cards, resulting either in success or cryptic failure printouts (Walker, 1990). The development of command line interfaces (figure 2.10(b)) did decrease the semantic and articulatory distances by choosing non-arbitrary command names and providing instant feedback.

[5] `https://www.fourmilab.ch/autofile/www/chapter2_69.html`

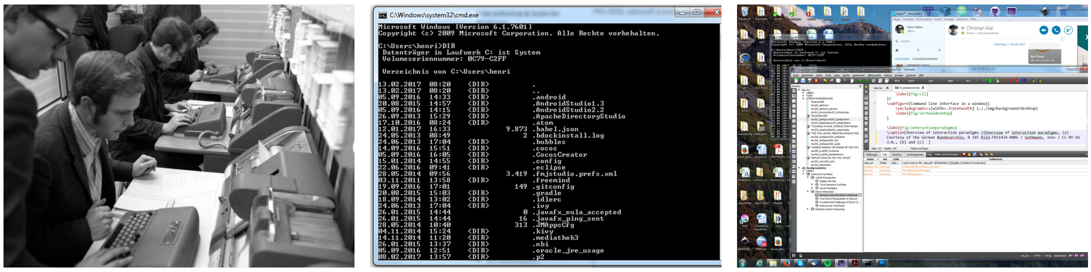
With the emergence of the personal computer and the invention of the virtual desktop, the WIMP interaction paradigm reduced the offsets inherent in previous command specification paradigms. WIMP interaction is based on a graphical user interface, employs real-world metaphors (e.g., the desktop metaphor) and reflects features of the Direct Manipulation paradigm introduced by Shneiderman (1997). Most importantly, Direct Manipulation postulates the visibility of objects and the reversibility of actions, facilitating explorative interaction (figure2.10(c)) and establishing ease of use, learning, and knowledge transfer as valuable design goals (van Dam, 1997).

However, as outlined by Moscovich and Hughes (2006), cursor-based interaction inherently involves indirection: "The cursor itself is an indirection in our interaction: we control some device like a mouse, which in turn controls a cursor, which has power over either some model (a text document, a drawing) or our view of it".

Further, graphical user interface elements represent intermediary entities, as well as the input devices used to operate them. This is reflected in criticisms of the WIMP paradigm, which – amongst other issues – have pointed out its time requirements on handling user interface elements instead of the task (van Dam, 1997). Similarly, Green and Jacob (1991) formulate requirements for non-WIMP interaction, which according to the authors' definition includes *Notebook*-computing (alluding to the Dynabook concept (Kay and Goldberg, 1977)), an interaction style that is based on handwriting and gestural input. The requirements all relate to different notions of indirectness present in WIMP interaction systems and concern their limited input bandwidth, the low amount of degrees of freedom provided, and missing real-time and continuous response and feedback mechanisms.

Increased directness through spatial activation Beaudouin-Lafon (1998) establishes a notion of post-Wimp interaction, explicitly describing *level of indirection* as a measure of user interface elements, such as dialogs, scroll-bars or widgets. In his work, indirectness is related to the command activation scheme. In WIMP systems, activation with a pointing input device requires a previous association of the virtual cursor with a graphically represented command. Both the graphical tool representation and the physical input device configuration can employ spatial activation as a mean to increase directness. In the first case, graphical command representations such as scrollbars are activated spatially instead of using temporal schemes (i.e. modes) (Beaudouin-Lafon, 1998). In the latter case, input devices contain multiple controllers, each directly mapped onto specific commands, as for example in an audio mixing console. This idea also underlies the work of Fitzmaurice et al. (1995), who introduced the idea of the graspable user interface, which is based on physical handles that allow the direct control of virtual objects.

Towards dynamic physical representations A further concept of directness is related to computer-mediated representations. Using the example of a flight engineers panel, Hollan et al. (Hollan et al., 2000) describe the cognitive shifts that occur between referring to the user interface as a mere representation of the fuel system or as a substitute. While the paper was published in the year 2000, more recent developments further challenge our thinking about representations in user interfaces. Representations of things assimilate to the things



(a) 1970: punchcard writing at TH Aachen (b) Command line interface (c) Command line interface in a window

Figure 2.10: Overview of interaction paradigms. (a) Courtesy of the German Bundesarchiv, B 145 Bild-F031434-0006 / Gathmann, Jens / CC-BY-SA 3.0., (b) and (c) own images.

they represent, increasing the subtlety of discrepancies between the virtual and the physical spaces. This is observable for interaction styles based on graphical output, and efforts like *Radical Atoms* (Ishii et al., 2012) strongly push forward the vision of dynamically changeable properties of physical objects, eventually enabling dynamic physical representations.

2.2.2 From Direct Manipulation to Natural User Interfaces

In 1993, Sears et al. (1992) summarize the advantages of touchscreens, based on their use in "public access situations": First, due to the lack of an additional device and a cursor, touchscreens allow for fast selection times [6]. Second, the touchscreen's immediacy supports ease-of-learning, as neither spatial reorientation between hand and cursor movements, nor complex hand-eye coordination is required. In the same year, Shneiderman himself concludes that touchscreens provide "unrivaled immediacy, a rewarding sense of control, and the engaging experience of direct manipulation" (Shneiderman, 1993).

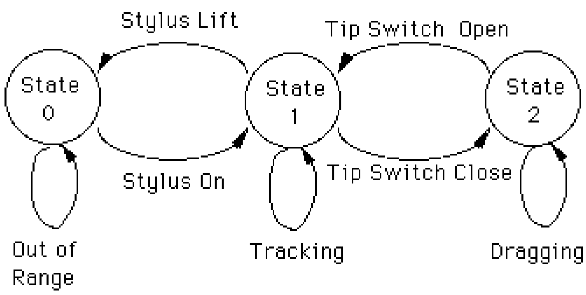
Further, touch-sensitive displays capable of multi-touch sensing allow for an increased input bandwidth through multi-finger, gestural and two-handed input techniques, which has resulted in a variety of novel interaction techniques (see section 7.1.1 for an overview of bimanual input). Moreover, blending physical with interactive surfaces transfers social characteristics into the user interface domain: for instance, shared tabletop displays support mutual awareness and their screen space is allocated to private and shared territories (e.g., Scott et al. (2004)).

Despite its suggested adequacy for direct manipulation systems, the properties of direct touch input (e.g., with regard to input modality and states, ergonomics) challenge a straightforward adoption of user interface paradigms that have established for mouse-operated personal computers.

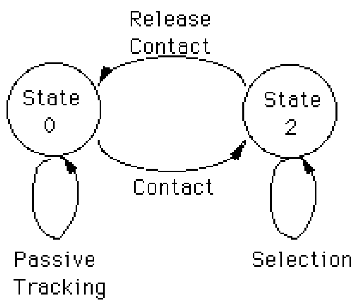
[6] This *speed advantage* is not undisputed: more recently, Forlines et al. (2007) conclude from an experiment that for unimanual input, there are no speed advantages for direct touch compared to mouse input.

2.2.3 Fundamental Challenges of Direct Touch Input

Alluding to the "long nose" of innovation (Buxton, 2008), touch screen technologies have been employed for decades in niche products, such as ATM machines or point-of-sale devices before becoming a ubiquitous product feature (Buxton, 2010). Therefore, the particularities and limitations of direct touch input have been recognized early. For instance, a particularity of touchscreen interaction is its two-state input model. Buxton (1990) described that – compared to a three-state input model (see figure 2.11) – the lack of a tracking state requires novel procedures for many standard tasks. Potter et al. (1988) list a variety of further basic challenges: making precise selections with our fingers is difficult and we occlude content with our limbs while gesturing at interactive surfaces. Also, the reach is naturally limited by our arms' length and depending on the situation and task, direct input can cause fatigue and physical discomfort. Buxton et al. (1985) described further challenges of using touch tablets as input devices: the lack of haptic feedback that may result in increased visual and auditive attention demands and the friction between the input surface and the finger. These fundamental challenges have sparked ongoing research activities in a variety of contexts to both facilitate input techniques based on touch and elaborate more adequate user interface paradigms.



(a) State 0-1-2 Transaction



(b) State 0-2 Transaction

Figure 2.11: State-models for graphical input (Buxton, 1990).

Input states

Since 2015, pressure sensation is incorporated in Apple's iPhones, adding a third dimension to touch sensing [7]. Yet, ideas on how to extend the touch input *vocabulary* with pressure sensing, particularly through the measurement of excess capacitance, have been expressed much earlier (e.g., Lee et al. (1985), or Sasaki et al. (1981)). A method for large-scale pressure sensing based on resistance is described in (Rosenberg and Perlin, 2009). With optical tracking, the utilization of the depressed finger's contact size enables a low-cost technical alternative to pressure sensing (Han, 2005). However, precise pressure control on interactive surfaces relies on adequate visual feedback (Buxton et al., 1985; Ramos et al., 2004).

[7] `https://developer.apple.com/ios/3d-touch/`

Regarding Buxton's input state models (Buxton, 1990), Voelker et al. (2013) conducted a study which compared different state-switching methods for cursor control with touch surfaces as input device. Their results indicate superior performance and user preference for *lift-and-tap*, an explicit movement-based switch explored already by Potter et al. (1988), compared to pressure and dwell-time control.

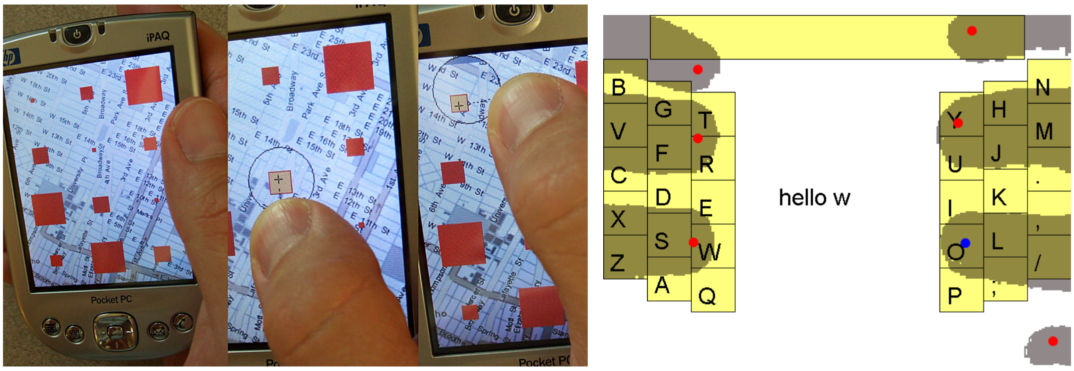
Precision and occlusion In addition to the requirement of three input states, WIMP interfaces involve small visual elements, which can be targeted more easily with a mouse-controlled cursor than with finger input, because we occlude targets with our fingers and arms, and because of the low resolution of finger input, which makes hitting targets smaller than our finger width challenging (Albinsson and Zhai, 2003; Benko et al., 2006; Ren and Moriya, 2000). For this reason stylus or pen input is an established high-precision input modality for touch screens, but its precision is not on par with mouse cursor input (Ren and Moriya, 2000).

Several solutions for the occlusion problem – sometimes called *fat-finger problem* (Wigdor et al., 2007a) – have been proposed, for instance (Potter et al., 1988; Sears and Shneiderman, 1991) introduced *Take-Off*, a precise finger selection technique for cursor-based input. Here, the cursor is displayed with a fixed offset above the finger touch point and lifting the finger selects the target currently underneath the cursor. The drawbacks of this technique were identified by Vogel and Baudisch (2007) and particularly include the inability to reach every screen position with a fixed offset as well as conceptual issues that arise from the offset: aiming for the actual target is not possible and novices will hardly anticipate the offset.

Later, Albinsson and Zhai (2003) presented several alternative widget-based techniques to make small target selections with finger input, using manipulable precision handles and virtual keys allowing to adjust the control-display ratio, i.e. to change the the ratio of input movement and output effect. Similarly, Benko et al. (2006) introduced a set of selection techniques aimed at precise selection for multi-touch screen. Their techniques also build upon a manipulable control-display ratio, but instead of a graphical widget, their idea is to involve a secondary finger to adjust cursor offsets and control-display ratio while the primary finger controls the cursor.

For direct input without cursor control – the common interaction style with mobile touch-screen devices – Vogel and Baudisch (2007) presented *Shift*, a technique which instead of the cursor offsets the screen content to avoid the drawbacks of the *Take-Off* technique described above and leads to significantly better targeting performance. Another technique called *LucidTouch* (Wigdor et al., 2007a) aimed at solving the occlusion problem within the mobile domain is based on the idea of *back-of-device* input. Here, silhouettes of the hands operating on the backside of the screen are displayed underneath screen content.

Holz and Baudisch (2011) showed that finger occlusion results in an offset between sensed and perceived touch position, which is due to the visual features of targets that users align with. These features "are located on top of the user's fingers, not at the bottom, as assumed by traditional devices". Subsequently, the authors show that this offset-error can be reduced by using precise camera-based tracking systems. Yet, high-resolution touch sensing ampli-



(a) *Shift* (b) *LucidTouch*

Figure 2.12: Solutions for occlusion on small touchscreens: (a) callouts shift content to an unobstructed screen position in case of input ambiguity (Vogel and Baudisch, 2007), (b) back-of-device interaction with simulated silhouettes (Wigdor et al., 2007a).

fies the instability of release (Wang and Ren, 2009), which describes unintended manipulation occurring while lifting the fingers from the screen, impeding for instance pixel-precise target selection using take-off techniques (e.g., Potter et al. (1988); Voelker et al. (2013)).

A related challenge concerning multi-touch input techniques is the *integrality* of operations (Jacob et al., 1994). While single-point input techniques (e.g., common cursor-based interaction) rely on sequences of separate actions for tasks that require more then two degree of freedom, common two-finger gesture techniques allow to simultaneously rotate, scale and translate objects. Yet, in cases when separate and precise control of these operations is desired, special interaction techniques are required, allowing to specify only a subset of manipulations (Nacenta et al., 2009a).

Ergonomics The spatial coincidence of manual input and visual feedback may yield in "literally the most 'direct' form of HCI" (Albinsson and Zhai, 2003), making touch screen usage intuitive and adequate for novice users. However, in many cases this form of input deviates from established ergonomic designs of interactive systems. Early surface input techniques using light pens, such as Ivan Sutherland's *SketchPad* system, led to arm fatigue after few minutes of operation (MacKenzie, 2013, p. 6), initiating the search for more comfortable pointing devices that could be operated on the desktop in proximity to the text keyboard.

Arm fatigue occurs also with horizontal interactive surfaces (the so-called "gorilla-arm-effect") (Ahlström et al., 1992; Ryall et al., 2006). In addition, prolonged interaction with horizontal surfaces lead to neck strain, causing a desire to adjust the surface angle (Benko et al., 2009). Slight inclinations (20°to 30°) of touchscreens also reduce arm fatigue and yield high user preference ratings (Ahlström et al., 1992). Further, Bi et al. (2011); Wimmer et al. (2010) provide a set of recommendations on the ergonomics of non-flat interactive surfaces.

With increasing surface sizes, reachability becomes a challenge. Solutions from related work include proxy widgets (e.g., (Bezerianos and Balakrishnan, 2005; Voelker et al., 2011)), but also involve additional devices, such as laser-pointer interaction (Olsen and Nielsen, 2001), or forms of remote controlling using mobile devices (e.g., Boring et al. (2010); Gilliot et al. (2014a)).

Haptic feedback

In their seminal paper about *Direct Manipulation Interfaces*, Hutchins et al. (1985) note that "In order to have a feeling of direct engagement, the interface must provide the user with a world in which to interact. The objects of that world must feel like they are the objects of interest, that one is doing things with them and watching how they react."

Benko et al. (2009) report that the most missed feature for users of tabletop systems is a physical keyboard. The inadequacy of so-called *soft keyboards* for text input (also reported by Ryall et al. (2006); Wigdor et al. (2007b)) highlights a further challenge of surface input: communicating visual feedback during interaction cycles that require a lot of hand-eye coordination, including mid-air targeting and finger movements, results in a visually highly demanding interaction (Buxton, 2007).

Tangible or graspable user interfaces aim at enriching surface interaction with non-visual feedback modalities by creating a symbiosis between the surface and physical objects – embodiments of digital information– which can be sensed by the surface and provide haptic cues to the users' hands, "safeguarding user intent" Fitzmaurice et al. (1995). Technical progress with computationally enhanced materiality – aggressively pursued by Hiroshi Ishii's group working towards their vision of *Radical Atoms* (Ishii et al., 2012) – promises to solve some of the involved limitations, such as the static (or passive) character of physical objects that does not match the dynamics of graphical displays.

Another approach presented by Bau et al. (2010) and Poupyrev and Maruyama (2003) is to equip a touch sensitive surface with actuators inducing dynamic tactile feedback. In the *TeslaTouch* project, a touch surface is enhanced with tactile feedback trough electrovibration (see figure 2.13), a technique that allows to superimpose sensable "textures" onto surfaces (Strong and Troxel, 1970). In particular, this technology allows users to feel different tactile sensations while moving their fingers across an interactive surface.

2.2.4 Natural User Interfaces

The above-mentioned challenges illustrate well that simply exchanging input devices and retaining prevailing user interface styles is neither straight-forward nor beneficial. Instead, we can observe adaptations of existing user interfaces to touch or gestural input. These efforts can be seen as part of a broader quest for innovation in human-computer interaction, including gestural and speech interaction. It is fueled by the "idea that it isn't going to be mouse and keyboard forever" (Goth, 2011). The goal of this quest is to achieve *naturalness*, a rather elusive description of interaction, which, however, is not intended to indicate

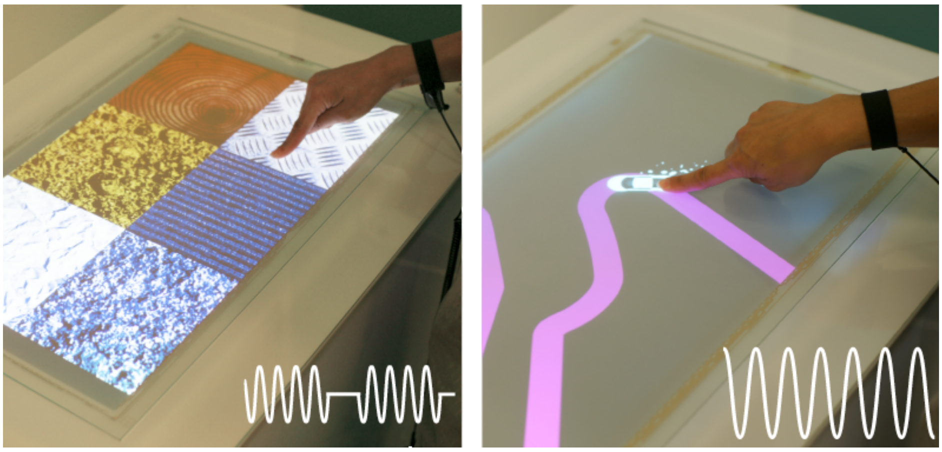


Figure 2.13: *TeslaTouch* (Bau et al., 2010): (left) different textures induce different sensations, (right) semantic use of variable friction in a racing game.

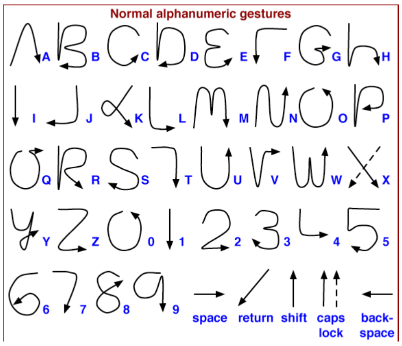
mimicry of real-world properties (Wigdor and Wixon, 2011). Instead, it has been used to describe input devices as controllers that one forgets about during operation (Goth, 2011), and according to Wigdor and Wixon (2011) refers "to the way users interact with and feel about the product, or more precisely, what they do and how they feel while they are using it" (p. 9). Further, they define a natural user interface as one that is both *easy-to-learn* and *easy-to-master*, allowing a "virtuoso" (p. 12) experience from novice to expert.

According to Wigdor and Wixon (2011), an important goal of developing natural user interfaces is to leverage the particular strengths of input modalities. With respect to multi-touch input, we can observe how its spatial immediacy and increased bandwidth are reflected in interaction techniques such as *pinch-to-zoom* or *two-finger rotation* (see figure 2.14(b)), which have become widely used after the popularization of multi-touch through Apple's release of the iPhone. A less-adopted example is handwriting. The Palm Pilot's *Graffiti* (see figure 2.14(a)) text input language is a good example of how an abstraction from real-world activities may nonetheless result in a natural user interface.

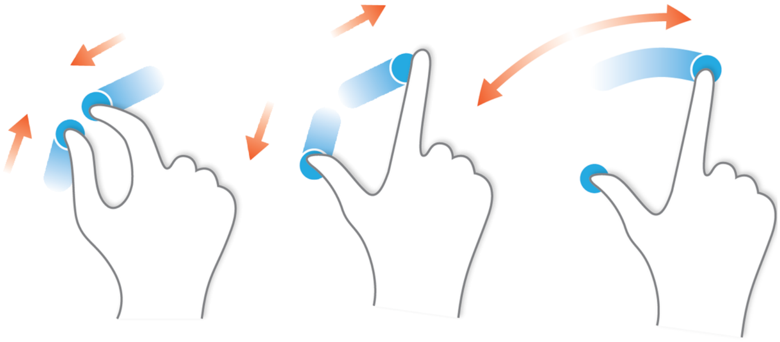
2.3 Desktop Surface Computing

Following Pierre Wellner's seminal work *DigitalDesk* (Wellner, 1993) (see section 2.1.3), different lines of research have approached the idea of blending physical and interactive surfaces in the context of desktop computing. A lot of the work on *computer-augmented desks* has been driven by technical problem-solving, such as questions of ambient projection (e.g., Sukthankar (2005)) or non-planarity (e.g., Weiss et al. (2010); Wimmer et al. (2010)). Other projects, for instance *Magic Desk* (Bi et al., 2011) or Voelker's work on gaze input (Voelker et al., 2015), are targeted more at understanding interaction in such contexts.

In this section, I will introduce related work with a focus on integrating interactive surfaces into personal computing workspaces. In the most cases, this implies a stationary context and



(a) *Graffiti*



(b) Two-finger *pinch*, *zoom* and *rotation*

Figure 2.14: Examples for touch-based NUIs: (a) *Graffiti*, a handwriting abstraction introduced with the Palm Pilot in 1996, image by IMeowbot licensed under the GNU Free Documentation License; (b) two-finger gestures popularized with Apple's release of the iPhone in 2007, image by GRPH3B18 licensed under Creative Commons Attribution-Share Alike 3.0 Unported.

typically involves a desk, at which a user may take a seated position to operate a desktop computing device. It is noteworthy, that many of the proposed approaches build upon the existing configuration of horizontal desktop and vertical monitor planes (e.g., Arai et al. (1995); Bi et al. (2011); Voelker et al. (2015); Weiss et al. (2010); Wimmer et al. (2010)).

2.3.1 Novel Workspaces

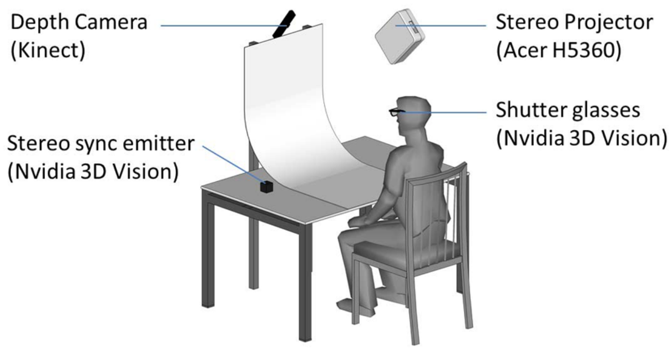
An often-cited example for a novel computer workspace targeted at knowledge workers is the *Starfire* video prototype (see figure 2.15) (Tognazzini, 1994). It features a large touchscreen spanning the horizontal desk surface and seamlessly extending into an upright display that slightly bends around the user (probably to support reachability). While its look and many of the proposed interface ideas may have appeared as rather futuristic in 1994, the goal of this project was to envision an interactive workspace scenario that – technically – would be feasible within a decade. Themes of the video include direct manipulation of virtual objects on the horizontal surface and observing multiple juxtaposed streams of information on the upright part of the display.

Little more than a decade after the *Starfire* video had been produced, several hardware prototypes picked up its theme and realized curved surfaces that seamlessly combine differently oriented displays. Two very similar prototypes are *Curve* (Wimmer et al., 2010) (see figure 2.16(a)) and *BendDesk* (Weiss and Voelker, 2009) (see figure 2.4(c)), which are based on dual back-projection and differ mainly concerning the inclination of the vertical display (75 °vs. 90 °). Further, both projects highlight the curved connection between the displays as an independent area that acts as a physical means for seamlessness, but also as a place for dedicated user interface elements (e.g., clipboard (Weiss and Voelker, 2009)). A different construction with a similar display shape has been presented by Benko et al. (2012), whose

Figure 2.15: Scene from the *Starfire* video prototype (Tognazzini, 1994). Image from Bruce Tognazzini's website `http://asktog.com/images/strfr.jpg`.

prototype *MirageTable* (see figure 2.16(b)) uses a seamless bent display plane in combination with stereoscopic projection to evoke the illusion of a spatial continuum that extends from the user's perspective into a virtual world.

(a) *Curve*



(b) *MirageTable*

Figure 2.16: Curved displays: (a) the *Curve* (Wimmer et al., 2010) is based on bent acrylic glass and dual back-projection, (b) the *MirageTable* (Benko et al., 2012) involves a bent display, depth-sensing and stereoscopic rendering to create a visual continuum between physical and virtual space.

Bonfire (Kane et al., 2009) presents a different approach towards augmented workspaces. The authors describe their system as a self-contained integration of a laptop and a tabletop system. The system is based on two pairs consisting of a laser micro-projector and a camera, mounted on the sides of a laptop. This way, an interactive display space is projected to either side of the laptop keyboard, enlarging the interaction space of traditional laptop computing and enabling direct gestural input on the desk surface. Moreover, the systems allows the recognition and augmentation of physical objects through computer vision.

Eventually, there have been occasional references to dual-surface workspaces in commercial products. For instance, the now discontinued Acer *Iconia 6120* [8] dual touchscreen notebook (see figure 2.17(a)) featured a second capacitive touchscreen that replaced the physical keyboard. A similar currently offered device is Lenovo's *Yoga Book* [9] (see figure 2.17(b)), which is available both with mobile (Android) and desktop (Microsoft Windows) operating systems and features a large second touch pad based on a back-lit LCD display, which in addition to finger touch also supports pen input and can display a keyboard and cursor trackpad as needed. The HP *Sprout* [10] (see figure 2.17(c)) is a third device that combines a vertical touchscreen with an interactive horizontal surface. Here, a top-mounted device both projects a display on the touch-sensitive horizontal surface and captures 3D scans from objects placed beneath.

(a) *Acer Iconia* (b) *Lenovo Yoga Book* (c) *HP Sprout*

Figure 2.17: Commercial dual-surface devices: (a) the Acer *Iconia 6120* features two identical capacitive touchscreens, (b) Lenovo's *Yoga Book* allows both finger and pen input, (c) the HP *Sprout* is targeted at creative workflows that involve 3D object scanning and editing. It allows finger and pen input on a projected horizontal surface.

2.3.2 Interaction with Dual-display Workspaces

Combining two or more interactive surfaces in a spatially continuous configuration, for instance such as proposed with the angular setups introduced in the previous section, extends prevailing interaction styles with novel opportunities such as finger or pen input (e.g., (Wellner, 1993)), enlarged information spaces (e.g., (Bi et al., 2011)) or the integration of real-world objects (e.g., (Patten et al., 2001)). In the following, I will outline themes of questions that have been explored in such setups which are relevant for my work.

[8] https://en.wikipedia.org/wiki/Acer_Iconia_6120

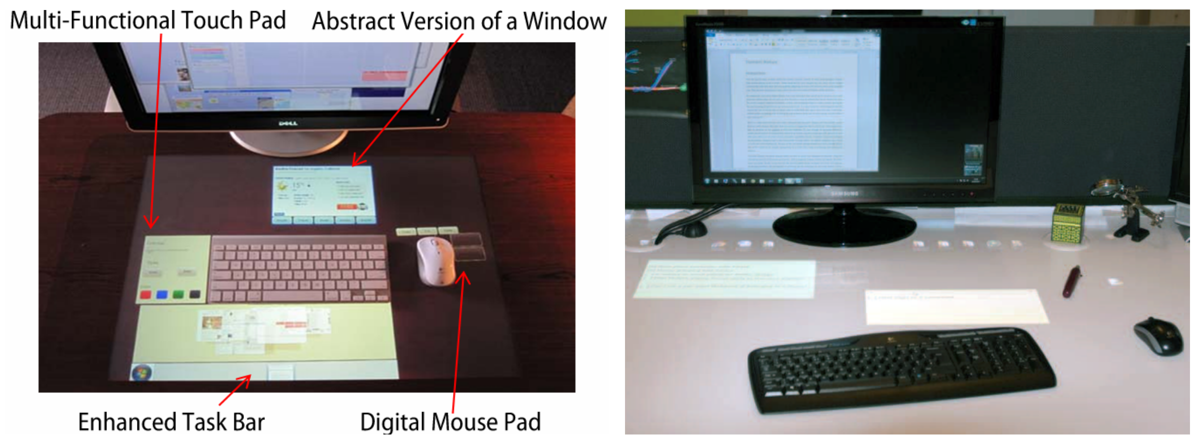
[9] http://shop.lenovo.com/de/de/tablets/lenovo/yoga/

[10] http://www8.hp.com/us/en/sprout/home.html

Integration of interactive surfaces into existing workspaces

Bi et al. (2011) presented an exploration of integrating multi-touch surfaces into a desktop computing workspace based on Microsoft Windows in order to give users an additional input channel. Guided by the idea of extending the prevailing interaction paradigm, they first systematically investigated the adequacy of the regions on the desktop surface surrounding the physical keyboard for one- and two-finger input. Based on their experimental findings, the authors then presented a set of interaction techniques, such as an enhanced task bar or a digital mouse pad (see figure 2.18(a)).

A subjective perspective on this topic is provided by Hardy (2012), who reports on his long-term experience with an interactive office desk based on a modified desktop computer (see 2.18(b)). Among many findings, he reported that despite the novel input modality, mouse and keyboard remained the dominant input modality, and that switching between modalities requires mental effort. In his case, the different display areas serve specific purposes: the upright monitor was an area for tasks requiring continuous focus, whereas the horizontal desk surface was used for peripheral, organizational or social aspects.

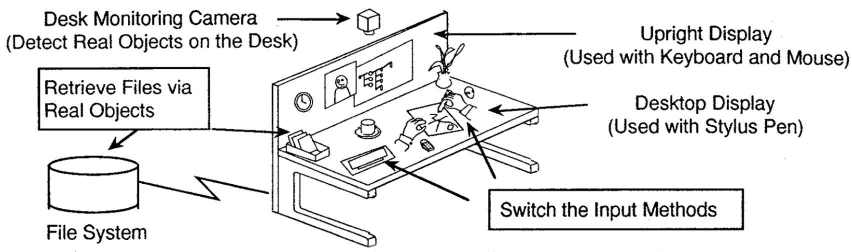


(a) *Magic Desk* (b) Modified desktop computer

Figure 2.18: Desktop surface computing: (a) the *Magic Desk* prototype is based on a Samsung SUR40 tabletop and a conventional desktop PC (Bi et al., 2011), (b) John Hardy's desktop computer modified with a top-projected interactive desk surface (Hardy, 2012).

Switching between different input modes

In 1995, Arai et al. (1995) presented *InteractiveDESK*, an computer-augmented desk with an upright display and a large horizontal desktop display operated with pen input (see figure 2.19(a)). Instead of extending desktop computing with novel multi-touch widgets, the desk is based on a rationale of *switching* between different interaction styles: to ensure that "work is carried out effectively", users may either operate mouse and keyboard in combination with the upright display, or use the pen to directly manipulate virtual documents displayed on the horizontal desktop surface. The keyboard's placement on the desk is decisive for the selected interaction style – without the keyboard, pen input is used.



(a) *InteractiveDESK*

(b) Using gaze

Figure 2.19: Switching between interaction styles: (a) the *InteractiveDESK* combines an upright display and a large horizontal desktop display, (b) using gaze to switch between direct and indirect input styles (Voelker et al., 2015).

More recently, the rationale of switching between to interaction styles in a dual-display setup consisting of a combination of vertical and horizontal displays has been investigated by Voelker et al. (2015). Based on the assumptions that the horizontal touchscreen may alternately act as touch pad, i.e. an indirect input device, or as direct interaction device, and that visual focus is a reliable indicator for the user's intention, gaze tracking is used to perform the switches between interaction styles. In a user study, different gaze-based selection techniques were compared, resulting in a recommendation for relative cursor mapping in the indirect mode, i.e. conceiving the horizontal touchscreen as a large cursor trackpad when the vision is focused on the upright screen.

Cross-display object movement

Rekimoto and Saitoh (1999) proposed *Augmented Surfaces* (see figure 2.20(a)), an augmented environment integrating notebooks, projected tabletop and wall displays, as well as physical objects. The project explored how to smoothly exchange information between these entities. The proposed concept *Hyperdragging* allows to move the mouse cursor across the notebook screen border to the projected tabletop and wall displays, which is a spatial technique (Nacenta et al., 2009b), i.e. the different parts of the screen form a logically connected interaction plane, despite being potentially separated and arranged in different angles. Section 4.1.2 presents a more detailed overview of cross-display object movement techniques in dual-surface display setups.

Cross-display object movement is closely related to switching between different input modes, which implies a shift of visual focus (Voelker et al., 2015). If we want to interact with virtual objects both directly and indirectly, their representations need to be either redundant (e.g., a document is displayed on the both the horizontal and the upright display), or movable between displays.

Visual metaphor for seamlessness

A seamless connection between horizontal and vertical touch displays enables the visual representation of a physical-to-vertical spatial continuum (Benko et al., 2012; Hennecke et al., 2013; Schwarz et al., 2012). This theme seems especially relevant for remote collaboration, as illustrated by Raskar et al. (1998), who introduced the idea “to use real-time computer vision techniques to dynamically extract per-pixel depth and reflectance information for the visible surfaces in the office including walls, furniture, objects, and people, and then to either project images on the surfaces, render images of the surfaces, or interpret changes in the surfaces” (see figure 2.20(b)).

Section 4.2 contains further references to relevant related work. However, I did not focus on the remote collaboration aspect, which in the context of dual-surface workspaces has been investigated in (Hennecke et al., 2013) and (Hennecke et al., 2013), but used the visualization technique in an exploratory study on 3D interaction (see section 4.2).

(a) *Augmented Surfaces* (b) *The Office of the Future*

Figure 2.20: Themes of novel workspace interaction: (a) dragging virtual objects across different screens and surfaces (Rekimoto and Saitoh, 1999), (b) visual seamlessness for remote collaboration in the *The Office of the Future* (Raskar et al., 1998).

3 Desktop Surface Computing

The previous chapter has given an overview of the requirements and particularities of designing interaction techniques for interactive surfaces. It has been shown that a wealth of quite different directions exist into which the desktop computer may evolve. While interactive surfaces have been explored predominantly in mobile, collaborative or other special contexts, the idea of blending them with the personal computing context has recently started to spark interest among researchers.

The goal of this chapter is to elaborate the design factors relevant to integrating interactive surfaces into personal desktop computing. In particular, it highlights the challenges that arise from the convergence of different interaction paradigms and substantiates the focus of this thesis by presenting theoretical assumptions that have shaped the conceptual contribution of this thesis.

3.1 Towards Desktop Surface Computing

In 2008, shortly after the introduction of the iPhone, the Economist got carried away with speculations about the potential of touchscreen technology: “Despite the iPhone’s success, it may prove to be PCs, rather than hand-helds, that benefit the most from touch-screen technology. That is because touch screens, like mice, are best suited to manipulating information, rather than inputting it in the first place – an area in which keyboards remain unchallenged. PCs with keyboards and touch screens (not to mention mice or trackpads too) could offer the most flexibility, letting users choose the appropriate input method for each task.” ((Economist, 2008))

The quote illustrates a vision of desktop computing, which enhances the existing apparatus of input and output devices with touchscreen technology, in a way that amplifies productivity by combining the strengths of different interaction paradigms.

3.1.1 Interaction Paradigms - Desktop Surface Computing

Interaction Paradigms are one source of knowledge to inform the design of interactive systems. Dix et al. (2003) regard interaction paradigms as “successful interactive systems [are] commonly believed to enhance usability and, therefore, serve as *paradigms* for the development of future products”. Similarly, Heim (2007) defines the term as a “conceptual framework that serves as a model for thinking about human-computer interaction”. His classification of interaction paradigms is derived from considering both innovation in computing technology and computing environments.

In the broadest sense, interaction paradigms reflect computing paradigms, which are constituted by major technological and conceptual shifts in computing. A common classification distinguishes between mainframe computing, personal computing and mobile computing. As illustrated in (Heim, 2007), further paradigms can be derived from the intersections of these basic paradigms. For instance, *Ubiquitous Computing* (see section 2.1.3) can be localized at the intersection of networked computing, personal computing and mobile computing.

In the personal computing paradigm, the term desktop is symbolically charged, as its metaphoric use as virtual desktop marks an important milestone in the evolution of graphical user interfaces and is a central aspect of WIMP interaction [1]. The relation between the virtual desktop and surface computing, an umbrella term that refers to different interaction paradigms based on touchscreens, is not easy to assess: emancipated pixels (Underkoffler et al., 1999), reality-based interfaces (Jacob et al., 2008), the computer for the 21st century (Weiser, 1991), tangible user interfaces (Ishii and Ullmer, 1997) – many ideas have emphasized the potential of touchscreens to render the interface invisible and thus overcome the "boundaries" of indirect and artificial interaction as well as metaphors. Yet, observing the legacy of both workspace ergonomics and graphical user interfaces, which have fundamentally shaped today’s information work, reveals an area of conflict for the integration of surface and desktop computing. These conflicts arise from the clash of interaction paradigms and map out design opportunities for developing adequate concepts (see figure 3.1).

3.1.2 Design Factors

In the following, I will outline the design factors for designing novel surface-enhanced desktop computing setups.

Workspace Composition

Interaction paradigms involving interactive surfaces usually emphasize the *directness* of input, which has been thought to help users “escape from the computer screen” (Wellner et al., 1993) and create an “active interface between the physical and virtual worlds” (Ishii and

[1] WIMP - Windows, Icons, Menus, Pointer

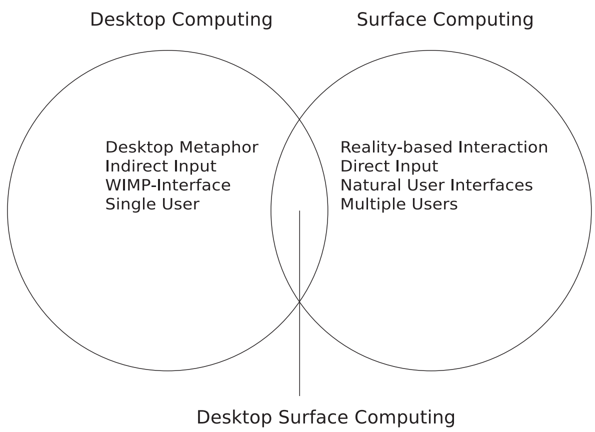


Figure 3.1: Desktop Surface Computing.

Ullmer, 1997). While conventional desktop computing clearly separates between the physical desk surface as input space and upright monitors as output space, the integration of interactive surfaces questions the strictness of this division. On a fundamental level, it affects the composition of computing workspaces, i.e. the arrangement of furniture and electronic computing devices. In the following, I will argue that *the choice of arrangement shapes our conceptions, expectations and design approaches*.

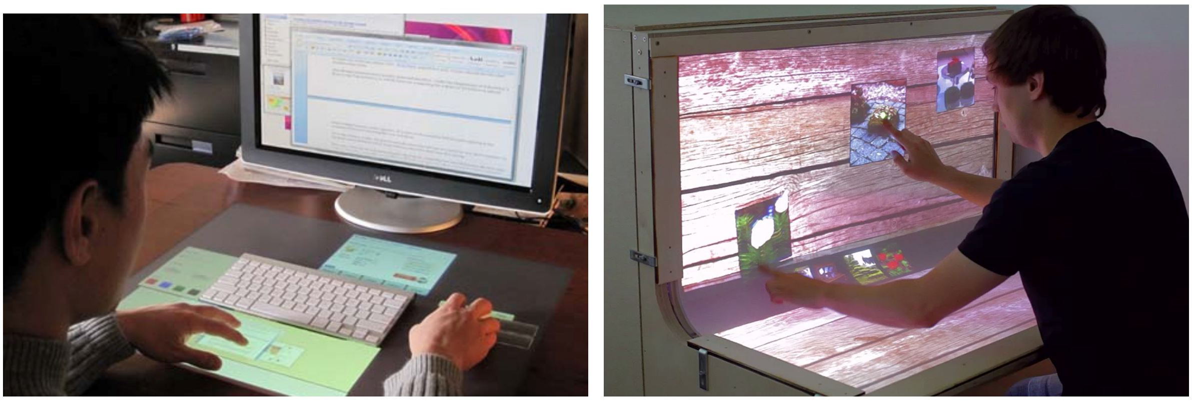
With regard to the composition of dual-surface workspaces, it can be distinguished between merged and composite approaches: with merged approaches, computer periphery and workspace are united in one device, whereas with composite approaches, touchscreen devices are added to existing computing workspaces. A brief look onto the recent related work illustrates both composite and blended design approaches.

With composite approaches, interactive surfaces are used as add-ons to work spaces, extending or recomposing rather than replacing the existing device ecology. An example is Magic Desk (Bi et al., 2011), which follows a top-down approach and extends a conventional version of Microsoft Windows with multi-touch widgets on the desk's surface that are specifically designed to complement mouse and keyboard with direct touch interface elements (figure 3.2(a)). A commercial example is Duet [2], which connects iPads as additional touchscreens to Apple computers or PCs.

With merged approaches, both input and output spaces are blended completely, resulting in a large touchscreen setup that both serves as a desk and singular computing device at the same time. Examples are prototypes like *Curve* (Wimmer et al., 2010) or *Benddesk* (Weiss et al., 2010) (figure 3.2(b)), which realize the vision of a non-flat display and explore its effects on basic interaction properties such as pointing or dragging performance.

The merged approaches are *conceptual* and concern questions such as form factors, ergonomics and novel interaction techniques. In particular, they do not resort to established

[2] `http://www.duetdisplay.com/de/`



(a) Magic Desk (b) Benddesk

Figure 3.2: Overview of interaction paradigms: (a) the *Magic Desk* (Bi et al., 2011): an integrative approach to surface desktop computing , (b) the *BendDesk* (Weiss et al., 2010): a large non-flat touch display that spans the whole reachable area of a person sitting at it.

interaction styles based on existing input devices, but rather explore the transfer of touch interaction styles to novel touchscreen contexts (e.g., Weiss et al. (2010), Voelker et al. (2012) or Hennecke et al. (2013)).

While not all workspace concepts necessarily adhere to the idea of sitting at a desk with arms rested on the desktop surface and vision directed forward to a display (such as for instance the *DigitalDesk* (Wellner, 1993)), the mentioned examples from research employ non-planarity as a feature of interactive surfaces to both unify desktop and display surfaces and maintain an ergonomic style of computer work.

In contrast, the composite approach is *pragmatic* and predominantly concerns the integration of interactive surfaces into a preallocated physical space. For instance, Magic Desk (Bi et al., 2011) assumes a vertical display, mouse and keyboard and systematically explores desk areas around the keyboard regarding their suitability for direct touch interaction. Similarly, Bonfire (Kane et al., 2009) introduces a technical solution to create such areas ad-hoc, without the need for an interactive tabletop display.

In summary, approaches to novel surface computing workspaces can be *both merged and composite*: on a conceptual level, it is assumed that multi-touch sensing and display capabilities will be integrated into parts of workspace surfaces and that a division between horizontal and vertical surfaces will be maintained for ergonomic reasons. On a pragmatic level, there seems to be a necessity of partitioning the surfaces, because they serve multiple purposes: as a desk, the horizontal surface supports our arms and holds physical objects, while as an interactive surface it provides a novel input and output channels for computer work.

Interaction Style

"The term 'interaction style' refers to the way in which we interact with computers" (Heim, 2007). Established interaction styles include for instance command line interfaces, menus, dialogs, and direct manipulation (Dix et al., 2003).

Conceptual workspaces based on touchscreens highlight novel styles of input: here, interactive surfaces enable direct interaction styles, which are often subsumed under terms like *Natural User Interfaces* (NUIs), *Reality-based Interaction* (RBIs) or *Tangible User Interfaces* (TUIs), where virtual objects are *directly* manipulated using finger, pen or object input, without the need of abstract *mediating* input devices (e.g., mice, track pads etc.) and menus or dialogs.

Yet, as outlined by Moscovich and Hughes (2006), both direct and indirect (in particular cursor-based) interaction styles have their distinctive strengths. Particularly, they support the assumption that established desktop computing involves many tasks, for which "direct manipulation would be inappropriate". Therefore, *pragmatic* approaches must (at least partially) adhere to the established *WIMP* interaction style that is based on indirect pointing and *Graphical User Interfaces* (GUIs), and add direct interaction elements as extensions. As discussed in section 2.3.1, potential solutions are either based on *temporal* switching between interaction styles (e.g., Arai et al. (1995)) or on *spatial* juxtapositions of differently operated interfaces (e.g., (Bi et al., 2011)).

The work presented in this thesis therefore builds upon the idea of integrating both established and novel interaction styles with surface-enhanced desktop computing.

Input Modality

Finger touching is the prevalent, but not the only input modality associated with interactive surfaces. Other common modalities for surface input include pens, shape and hover gestures, or tangibles. In the context of dual-surface workspaces, for instance Hennecke et al. (2012) explored the use of tangible objects that can be operated on the upright part of the display though adhesion. In general, tangibles provide haptic cues that benefit eyes-free interaction and can create physically affordances (see section 2.1.3).

Second, depending on the touch sensing hardware, pen input can be understood as a variant of touch input with altered properties, such as increased precision. In particular, the combination of pen and touch input has emerged as a distinct input style for interactive surfaces (e.g., Hinckley et al. (2010) or Pfeuffer et al. (2015)), but the primary use of pen input for indirect pointing in desktop contexts is less common (e.g., as a feature of graphic tablets), which may be due to a more complicated acquisition (Vogel and Baudisch, 2007) and therefore higher switching costs compared to the mouse.

Mid-air gestures have been explored to create an *off-screen workspace* (Hausen et al., 2013), allowing users to interact with invisible content on the desk via hovering gestures. Based on computer vision techniques, Malik and Laszlo (2004) presented a touch pad with additional detection of hovering hand gestures. Further, mid-air gestures have also been explored in the context of facilitating 3D interaction, in particular with a combination of stationary see-through displays and depth cameras (e.g., Hilliges et al. (2012) and Lee et al. (2013)). A commercial product which allows to add gesture recognition to personal computers is the *Leap Motion* [3]. Eventually, eye-tracking has been employed in the context of dual-surface workspaces to switch between direct and indirect input based on eye-focus (Voelker et al., 2015).

Further, natural language is a widely-studied input modality that has been employed in different interactive surface contexts, such as mobile interaction (e.g., Apple's Siri [4]) or automotive user interfaces (e.g., Graham and Carter (2000)).

Output Modality

The majority of surface-related research is focused on established two-dimensional display technologies (see section 2.1.1). In addition, there is related research exploring stereoscopic rendering, which raises fundamental questions about adequate selection and manipulations techniques (e.g., Benko and Feiner (2007)). In the context of dual-surface devices, Simeone and Gellersen (2015) have presented a combination of stereoscopic projection onto a vertical display and indirect touch input on a horizontal input surface. Further, transparent screens have been employed to create stationary augmented reality setups (Hilliges et al., 2012; Lee et al., 2013).

Physical output has been explored, for instance by coupling transparent tangibles with screen output (Weiss et al., 2009), with shape-changing interfaces (e.g., Follmer et al. (2013)), or with computationally enhanced materials (Ishii et al., 2012). Other relevant feedback modalities include tactile (see section 2.2.3) as well as auditory feedback.

Input Directness

For a detailed discussion about different notions of directness in human-computer interaction, please refer to section 2.2.1.

While direct touch input is a prevalent notion of interactive surface input, indirect touch input systems have been used for decades (i.e.track pads and graphic tablets). Moreover, *indirectness* has been recognized as one factor able to mitigate different shortcomings of direct touch interaction, such as occlusion or covering distances exceeding arm reach Forlines et al. (2007), but also arm fatigue Stellmach and Dachselt (2012). Examples for indirect touch systems include prototypes for novel interactive workspaces that integrate a touch screen into the desktop surface (e.g., Voelker et al. (2013)), automotive interfaces that aim to leverage touch input without drawing the driver's attention away from the street (Döring et al., 2011) or mobile interaction with interactive walls (Gilliot et al., 2014a). Recently, Apple introduced an indirect touch mode for the iPad's QuickType keyboard that lets users transform the keyboard into a touch pad for controlling a cursor to ease text selection.

[3] `https://www.leapmotion.com/`

[4] `http://www.apple.com/de/ios/siri/`

On the one hand, precision and ergonomics are of importance in work contexts, on the other hand, there seems to be continued need for *mediated* input styles (Moscovich and Hughes, 2006), complementing direct input styles. In the context of dual-surface devices, one can observe the re-introduction of *mediated* cursor-like interaction instruments. For instance, Schmidt et al. (2009) compared direct and indirect multi-touch input, based on two identical large touchscreens and an absolute mapping in the indirect condition. To provide visual pre-touch feedback in the indirect condition, contours of the hands hovering over the horizontal surface were displayed on the output screen. Moscovich and Hughes (2006) presented design ideas on how to generalize the cursor to multi-finger input. Voelker et al. (2013) explicitly assumed the necessity of a tracking or hovering state (see section 2.2.3), and compared different touch techniques to switch states.

Naturalness of Interaction

Naturalness is a much desired, but also elusive property of interaction. Interaction in the era of personal computing is often characterized as artificial, based on devices and paradigms that need to be learned and that provide limited input bandwidth and little understanding of the user's intentions. Natural user interfaces, in turn, are intended to enable the direct manipulation of virtual objects via natural language or body gestures and to reduce the need of learning by leveraging our intuitions about cause and effect in the real world. However, concerns about the terms naturalness and intuition have been raised repeatedly, for instance by Jeff Raskin and Donald Norman. In *The Humane Interface* (Raskin, 2000), Raskin argues that neither naturalness nor intuition are clearly defined and quantifiable concepts, and that they usually refer to a given similarity to other common human activities or express that something is easy to learn. Norman's critique evolves around the role of gestures in interactive systems. In particular, he argues that gestural input techniques require as much learning as other techniques and, due to their ephemeral character, are hard to memorize. He concludes that: "Are natural user interfaces natural? No. But they will be useful".

Another notion of naturalness in HCI is conveyed in the ever increasing complexity of information computing devices can sense from their users and process in real time, such as natural language, non-verbal behavior (e.g., mid-air gestures or eye gaze) or brain activity, allowing to approach richer or more human-like interaction between users and computers. Considering communication, the Media Naturalness Theory suggests human face-to-face communication as a baseline and establishes the amount of exchanged communicative stimuli as a measure to determine the naturalness of communication media. While this theory may have direct applications in computer-mediated communication, such as speech-based interaction, the importance of natural communication for productive and personal work, e.g., using office or creative applications, remains unclear.

The goal of this thesis is to explore novel input technology and vocabulary in the desktop computing context – a domain where the "use of the mouse itself is often claimed to be natural and intuitive" (Raskin, 2000). Naturalness (see also section 2.2.4) can refer to notions of *directness* or *mimicry*, however, a more assessable theme of naturalness is aiming for *learnability* and *expressivity throughout different levels of mastery*.

Mappings

Mappings refer to the relation between input and output spaces. With interactive surface workspaces, mappings can be direct or indirect. Indirect mappings can be distinguished between *absolute* and *relative*. Further, with indirect mappings, the shape, size and number of input areas are further design factors. With regard to novel workspaces, horizontal interactive surfaces provide space for both physical and virtual objects (e.g., Wimmer et al. (2010)), however, the potential mappings of virtual input areas are under-explored.

Mappings can be direct, such as proposed in projects which explore the effect of a curved display on direct interaction techniques like dragging (Weiss et al., 2010) or flicking (Voelker et al., 2012). Here, the *input space equals and overlays the output space*.

Schmidt et al. (2009) and Voelker et al. (2013) introduced setups, where *the input space equaled the output space in size and aspect ratio, but were spatially separate*. They employed an *absolute mapping* during their experiments, i.e. input signal positions corresponded directly with the associated visual cursor position. Further, Gilliot et al. (2014b) have explored the *influence of size and aspect ratio of rectangular touch pads* on indirect absolute pointing performance and found correlations between form factors and performance.

Bi et al. (2011) have identified four *distinct input regions* for surface desktop computing (before, behind, left, and right of the text keyboard) and assessed their adequacy for finger input. They propose different interaction techniques based on these areas, including a multi-functional touch pad, which might be used to control a second cursor with a *relative mapping*, i.e. the input signal and cursor positions can be offset with a variable mapping.

The Role of Hands and Fingers

We cooperatively use our hands for a wide variety of real-world tasks, but with regard to desktop computing, the non-dominant hand is mostly used to control the text keyboard. It has early been argued to extend the input device ecology in order to allow two-handed continuous input, i.e. to control two cursors (e.g., Engelbart and English (1968) or Buxton and Myers (1986)) (see section 7.1 for a more extensive overview).

Interactive surfaces naturally support two-handed input and research on novel workspaces takes this into consideration (e.g., Bi et al. (2011) or Voelker et al. (2013)). Two-handed cooperation is either asymmetric (Guiard, 1987), or symmetric (Balakrishnan and Hinckley, 2000). Some interactive surface technologies further support multi-finger sensing, theoretically increasing input bandwidth to ten individual cursors.

Two-handed and multi-finger input has been employed differently with interactive surfaces. There are accounts of employing two hands to control two distinct input devices (e.g., Bi et al. (2011) or Pfeuffer et al. (2015)), to control multiple virtual devices with multiple fingers (e.g., Buxton et al. (1985)), or to involve both hands in the operation of a single virtual tool to allow high-degree-of-freedom input, such as 3D manipulations (e.g., Reisman et al. (2009)).

Tasks

Basic tasks, such as pointing or object movement, are the fundamental building blocks of *Direct Manipulation* tasks, i.e. tasks that involve visual representations of objects and manipulations. In the context of interactive surface workspaces, such tasks have been explored both with regard to *cross-display* interaction (e.g., Hennecke et al. (2013), Voelker et al. (2012), or Weiss et al. (2010)), and concerning *indirect input* (e.g., (Schmidt et al., 2009) and Voelker et al. (2013)).

More complex tasks require the manipulation of many degrees of freedom, such as integrally translating, scaling and rotating domain objects. Such *compound* tasks have been studied for instance by Buxton and Myers (1986) and more recently by Moscovich and Hughes (2008). On a system level, interacting with an WIMP operating system (Bi et al., 2011) involves a variety of tasks, such as window management, hierarchical navigation, or menu selection.

A wider notion of task is involved when considering specific application areas: for instance, interacting in virtual 3D environments involves object manipulation and navigation tasks (Jankowski and Hachet, 2013). In this regard, dual-surface workspaces have been used to allow a coordinated display of top-view and 3D view to enable immersive navigation through virtual environments (Wu et al., 2011).

Further, Morris et al. (2007) have specifically investigated active reading tasks, which involve reading and writing subtasks. They identified dual-surface setups as promising devices to exchange documents between vertical and horizontal surfaces: vertical surfaces are well-suited for reading tasks, and horizontal surfaces are well-suited for handwriting tasks.

In summary, tasks can be atomic or compound. One measure of complexity is the number of degrees of freedom required to control the task. Specific interaction paradigms and application contexts comprise a wide variety of tasks and complex tasks should be taken into consideration when designing for desktop computing contexts.

3.1.3 Thesis Focus

Due to the number and complexity of design factors, it was necessary for this thesis to restrict its scope from the beginning, by making certain assumptions on how the design factors are explored. In the following, I will outline the reasons underlying these preconceptions.

First, this thesis investigates novel interaction techniques for dual-surface workspaces based on *touch input*. I assume that with the introduction of multi-touch surfaces, touch will be an obvious modality to consider. Further, as touch input has become widely used during the last decade, I hypothesize that *a certain touch vocabulary can be presumed* (e.g., pinch gesture). In contrast, alternative modalities based on additional hardware have either established rivals in a desktop computing context (e.g., pen vs. mouse) or are less suited for prolonged work (mid-air interaction). Yet, both these assumptions and a systematic assessment of alternative modalities are not addressed in this thesis and subject to future research.

Second, the work presented in this thesis is limited to visual feedback created with two-dimensional display technologies, which primarily is a pragmatic choice to confine the investigation of novel techniques to a manageable number of design factors.

Third, I assume a logical partition of dual-surface workspace into a horizontal plane and a vertical plane, which will serve specific purposes. In particular, ergonomics suggest to design interfaces which reduce direct interaction with the vertical surface and to limit visual focus on one surface (Morris et al., 2008).

Fourth, in addition to the logical partition of horizontal and vertical surfaces, I assume a spatial approach of allocating touchscreen areas which provide a persistent input style and potentially complement other objects and devices on a desk surface, incorporating its double function acting both as physical table and interactive surface (Morris et al., 2008; Wimmer et al., 2010). In particular, this thesis does not investigate switching between direct and indirect input styles.

Last, a more general view underlying this thesis is that *naturalness is an elusive property of interaction*. The interaction techniques presented in this thesis do not aim at real-world mimicry and do not employ frameworks such as *Reality-based Interaction* (Jacob et al., 2008). Instead, the focus of this thesis is to employ the *assumed touch-gesture vocabulary* to create an additional input channel that extends established interaction styles. The primary goal is to *facilitate learning and performing* existing tasks through increased input bandwidth and visual input structures.

II

"TOWARDS A NOVEL INTERACTION PARADIGM FOR PERSONAL MULTI-SURFACE COMPUTING"

4 Case Studies

I complement my theoretical approach described in the previous chapter with my own observations and experiences, which have developed during an extended phase of exploration. Exploratory research can be useful in early stages of work and help to push "not-yet-formed or preliminary questions forward" (Shields and Rangarajan, 2013). In my case, the preliminary question was related to the Curve, a prototype with a non-flat touchscreen: *How can we effectively make use of a seamless curved connection between horizontal and vertical touchscreens?* In order to derive more concise research questions, I identified several specific application scenarios and transferred them to the non-flat touchscreen by crafting special-purpose prototypes (figure 4.1 shows a selection of case study impressions). By building and evaluating interactive prototypes, I gathered a hands-on understanding of potential challenges and trade-offs, which ultimately led to a shift of focus: instead of focusing on the type of connection between horizontal and vertical touchscreens, I recognized the need to understand more basic questions first: *What is a suitable interaction paradigm for such a display setup?* and *What are adequate interaction techniques?*

The construction of the Curve's non-flat touchscreen as well as a set of practical approaches for building interactive applications have been described by Hennecke in his Dissertation (Hennecke, 2014). Section 2.3.1 contains further references to relevant related work. For developing the prototypes I mostly used a customized version of the application framework *Multi-touch for Java (MT4J)*[1], an open source, cross platform Java framework for building applications with 2D and 3D user interfaces that supports optical multi-touch tracking through the implementation of the TUIO prototcol.

In the following, I present a selection of the results gathered during this phase of my work. In particular, I focus on two themes that have shaped the further path of my research: *Cross-Surface Object Movement* concerns the question of moving objects across the curved part of the display, and *3D interaction* explores interaction techniques to navigate in 3D scenes and manipulate 3D objects.

[1] The framework was released by Fraunhofer IAO in 2009. In 2011, the last official version (v0.98) was released. `https://de.wikipedia.org/wiki/Multi-touch_for_Java`

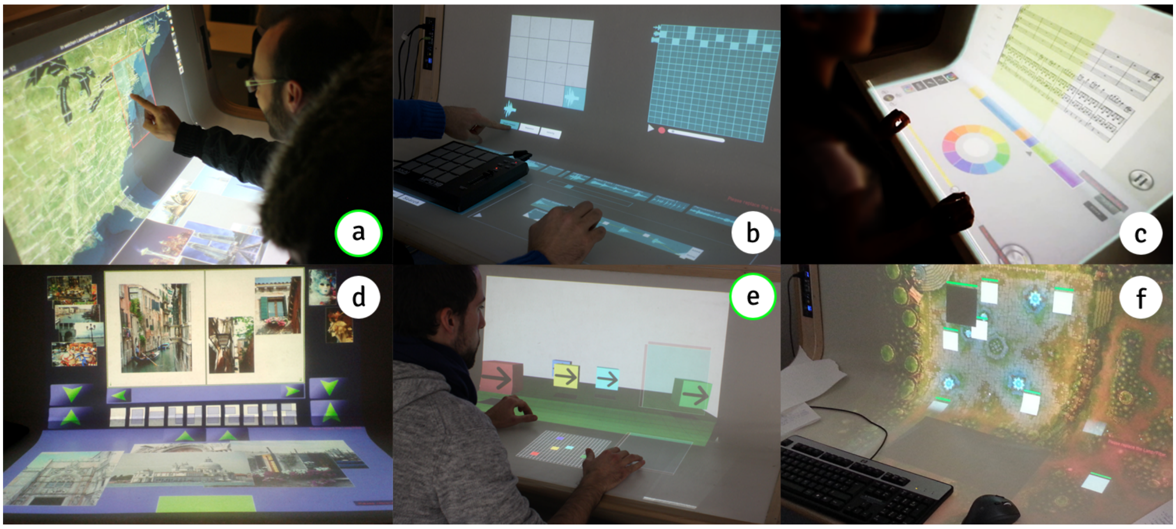


Figure 4.1: Several application scenarios: a) Public intervention (Palleis and Hussmann, 2014), b) digital music production (Palleis, 2014a), c) music analysis, d) photobook layout, e) 3D interaction (Palleis et al., 2015a), and (f) MOBA gaming; (a) and e) are discussed in this chapter).

4.1 Cross-Surface Object Movement

Early studies in the context of display transitions between horizontal and vertical display areas show that a curved connection supports object movement on both physical and cognitive levels (Hennecke et al., 2012), but negatively influences the performance of established touch-based object movement techniques like dragging (Weiss et al., 2010) or flicking (Voelker et al., 2012). Also, although both the *Curve* and the *BendDesk* are built with ergonomics in mind, there has been preliminary evidence that object movement across the curve quickly causes arm fatigue (Weiss et al., 2010). For these reasons, I set out to better understand the affordance of a curved display transition to move objects across differently angled display areas and its implications on the design of adequate interaction techniques.

4.1.1 Direction Matters: Public Quiz Game

Interacting with installations in public spaces like museums or libraries are one scheme of interaction that is often based on large touchscreens (e.g., Geller (2006)). Our assumption was that non-flat display arrangements provide interesting characteristics for this domain as the arrangement can be used to structure the interaction with digital assets and information in a way that considers the distinctive affordances of horizontal and vertical surfaces, for instance sorting images on a desk surface and pinning them on a wall. On the one hand, I wanted to understand if the visual and haptic continuity of the display effectively implies to a user that digital objects can be dragged across the curve. On the other hand, I explored how the spatial layout of the graphical user interface influences the interaction with such a display.

To gather first insights into these questions, I developed a geography quiz application for the Curve that made distinctive use of the horizontal and vertical parts of the display and required dragging digital objects across them. In particular, the game featured two main components: an *image area* on the horizontal surface and an interactive *world map* on the upright part of the display. The *image area* contained images of famous places, which could be dragged onto the interactive *world map* and should be placed onto the respective countries they were taken in. In a first step, I made the application accessible to the public during an annually held open lab day, where I observed visitors walking up and playing the game. In a second step, I conducted an experiment with the goal to compare the mapping between the two game components and the two display surfaces.

Game Design

The quiz asks players to assign the places shown on the images to their corresponding location on the world map by dragging the image from the image area across the curved part of the display onto the according country on the world map. In the original design, the image area was shown on the plane horizontal part of the display and the map was placed on the curved and the vertical parts (see figure 4.2). Primarily, this decision was based on the analogies of photo browsing on a table and wall-mounted maps featuring pins. Additionally, an ergonomic assessment of the application layout was not obvious at first, since both tasks required continuous input.

At game start, the map was zoomed out to show the whole world. It could be manipulated with one- and two-finger touch gestures (two-finger pinch and zoom, one-finger dragging). Images were initially piled at the center of the image plane and could then be dragged around the whole image area. When dragged and held over the map, the image was rendered semi-transparent to support a precise placement on the map, with the dragging finger acting as cursor. Feedback on the player's answer was given as colored border (see figure 4.2). In addition, when released over the correct country on the map, the image was scaled down to a miniature and fixed to the map.

In total, I prepared six image sets (e.g. nature, ancient wonders, architecture etc.), each containing between seven and 13 images depicting places from three to five continents. I added a simple menu and navigation to create a complete gaming experience (see figure 4.3(b)).

Public Exhibition of the Game

Each year, our lab organizes the so-called *Open Lab Day*, a four-hour evening event where recent research results and prototypes are presented to the interested public. In 2013, I invited visitors passing by to play the quiz game without giving them instructions how to operate the game (see figure 4.3). While they interacted with the game I observed them, taking handwritten notes using a predefined matrix that contained categories such as recognition

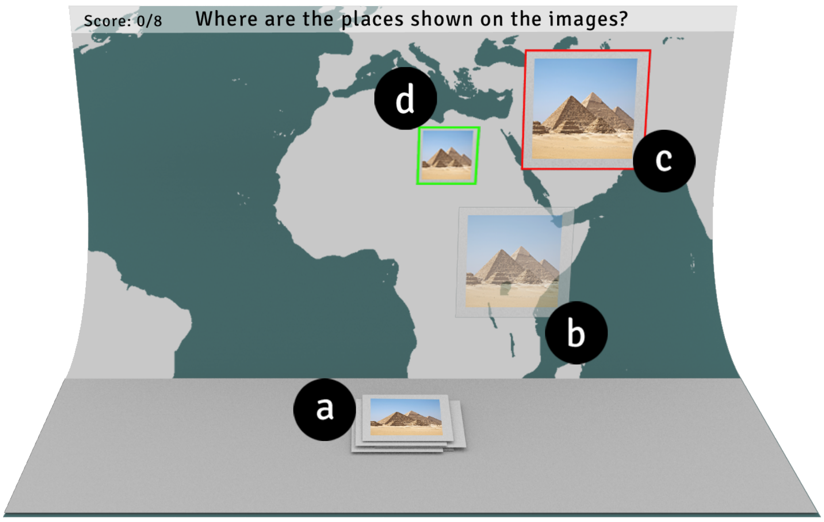


Figure 4.2: Conceptual overview: a) piled images on image area, b) semi-transparent image while being dragged onto the map, c) image with red border released on the wrong country, d) image released on the correct country with green border.

of the game's interaction principle, recognition of applicable multi-touch gestures, game strategies or problems. Upon finishing and leaving the game, I asked them to fill out a short questionnaire, provided they had completed at least one set of questions. Next to basic demographics, the questionnaire asked for prior touchscreen experience and contained subjective ratings using a five-point Likert scales (1 = "I strongly disagree", 5 = "I strongly agree"). In cases where multiple people played the quiz collaboratively, I asked the player who sat on the provided chair to fill out the questionnaire.

(a) A visitor spontaneously playing the quiz game. (b) The inverted quiz game used during the experiment.

Figure 4.3: Two versions of the quiz game.

Participants Six individuals and six groups consisting of two to five people completed at least one set of questions. Hence, twelve players (five female) filled out the post-questionnaire. Two players were in the age range 14-19 years, nine in 20-29 years and one in 40-49 years. All of them stated to regularly use smartphones and to have experience with tablet devices.

Game principle In ten out of twelve cases, the game principle – answering questions by dragging images across the curve – was recognized without asking for advice. In two cases, players tried to answer quiz questions by tapping sequences. The interactive features of the map (i.e. panning and zooming capabilities) were detected in ten cases without help.

Image browsing strategies Further, players exhibited varying image browsing behavior. About half of the players used the image plane to spread and sort the piled images before dragging them onto the map. The other players applied a batch processing strategy and worked through the piled images one after another without using the image area for browsing. None of the players or groups used the map component to put down images temporarily. Instead, when not released over the correct country, images were always dragged back down onto the image area.

Group strategies Group interactions involved one player sitting on the provided chair and at least one more player standing aside with access to the touchscreen. Additional group members standing behind the active players acted as advisors or commentators. General group strategies can be distinguished between *taking turns* and *sharing control*. In the first case, one player operates the game at a time, controlling all aspects of the game. In the second case, one player browses and drags the images and the other one simultaneously navigates the map. When sharing control, players communicated to help each other, for instance to find a country on a map, to agree on a picture handover, to issue commands (e.g., "*Zoom in!*") or to argue (e.g., "*You have to do it that way!*").

Subjective ratings Subjective data indicates that players experienced dragging pictures across the curve to answer questions as *fun* (median=5). The game was considered *easy-to-use* (median=4) and both the interaction with the images and the world map was rated *easy* (median=4). Overall, players did not consider the interaction with the vertical map as *physically demanding* (median=3), however ratings were less clear in this case.

Discussion In general, the geography quiz game was an adequate tool to make observations in the wild, because it attracted visitors to play. The gameplay sparked their ambition to score well and made them play more than two question sets on average. I received many positive comments ("*It's so cool that I can touch it!*", "*I can't stop*" etc.), and combined with the data from the questionnaires and the long gaming times, I conclude that the game principle led to a joyful experience. The observation showed that players unfamiliar with the curved display had no difficulties to pick up the principle of the game quickly. This indicates that the display's haptic and visual continuity suggest its cross-display interaction capability, despite a clear visual separation between the image area and the map in the user interface. Also, this separation possibly indicated not only a visual, but also a logical structure, as players did not keep images on the world map in case of wrong answers. However, the negative feedback (i.e. red image border) when releasing images on the map might have influenced this behavior.

Regarding the group interactions, I was surprised about the variety of observed behaviors: the spontaneous sharing of control, the simultaneous interaction and the communication between the cooperators indicate that this type of display may enable new forms of cooperative work, despite its original design as personal workspace.

Although I observed that the interaction with the map on the vertical part of the display outweighed the interaction in the horizontal image area, I did not observe signs of exhaustion and players did not complain about arm fatigue. Originally, I had chosen the game layout based on real world analogies, but being aware that literature suggests to concentrate touch interaction on the horizontal part of the display (e.g., Voelker et al. (2013)), I then became interested in an inverted game layout and its effect on the gaming experience.

Lab Study

Following the observed exhibition at the public event, I conducted a small lab experiment. Therefore, I created an alternative version of the quiz, exchanging the vertical layout of image area and world map. The sizes of the areas remained the same, which means that the map filled the horizontal and the curved parts of the display and the image area filled the vertical display plane (see figure 4.3(b)).

Experiment design I used a within-subjects design in our experiment with game layout as counterbalanced independent variable with the two levels *OLD* (the original layout) and *NEW* (the inverted layout). From the six question sets I chose four sets (A, B, C, D) and fixed the number of images to eight in each set. Sets A, B and C contained images from four continents, D contained images from two continents. I randomized the order of the sets during the experiment. The order of the images within the sets (i.e. the order of the piles of images) was not altered.

Procedure Upon arrival, participants were asked to sign a consent form and fill out a demographic questionnaire. Then, they read a written explanation and schedule of the experiment. Aware of the influence of individual knowledge, I tried to reduce the potential fear of being embarrassed by the questions by assigning participants the role of testers evaluating a new game for museums. In addition, I informed them that knowing all the places shown on the images was neither important nor probable and encouraged them to ask for unfamiliar places and locations. To further evaluate the self-explanatory nature of the game, they were instructed to figure out the game principle themselves.

Collected Data and Hypotheses All touch events were logged in order to determine the amount of image and map interactions, to determine the dragging distances of both map and images, and to draw trajectories of the dragging interactions.

In addition, I asked the participants to fill out a paper and pencil NASA RTLX questionnaire (Byers et al., 1989) after each condition to assess workload and collected subjective data with five-point Likert scale questions (same scale as previously). Finally, I conducted a short interview, during which I asked the participants for personal preferences regarding game layout and dragging direction. Our working hypotheses were:

H1 Condition NEW causes less workload than *OLD*.

H2 Participants prefer condition *NEW*.

Participants I recruited twelve participants (four female). Eleven were between 20 and 29 years and one was between 30 and 39 years old. None had prior experience with the *Curve*, but all had experience with smartphones and tablet devices.

Results The RTLX scores are lower for condition *NEW* (29.01, sd = 7.9) than for condition *OLD* (32.40, sd = 6.5). A one-tailed paired t-test results in a p-value of 0.008, which indicates support of H1. In addition, the *NEW* condition was preferred by ten out of twelve participants. Given reasons include that playing the game in this condition is less exhaustive and more comfortable (n=7), that dragging pictures from the vertical display towards the body feels "more natural" (n=4), and that the horizontal placement of the map feels better (n=3), because this is the "natural" way to interact with maps, improving overview and readability. One participant was undecided and preferred condition *NEW* concerning *ease of interaction*, but *OLD* for a better overview. One participant clearly preferred condition *OLD*, because this setting felt more natural. Ten participants stated that dragging images downwards felt better than upwards.

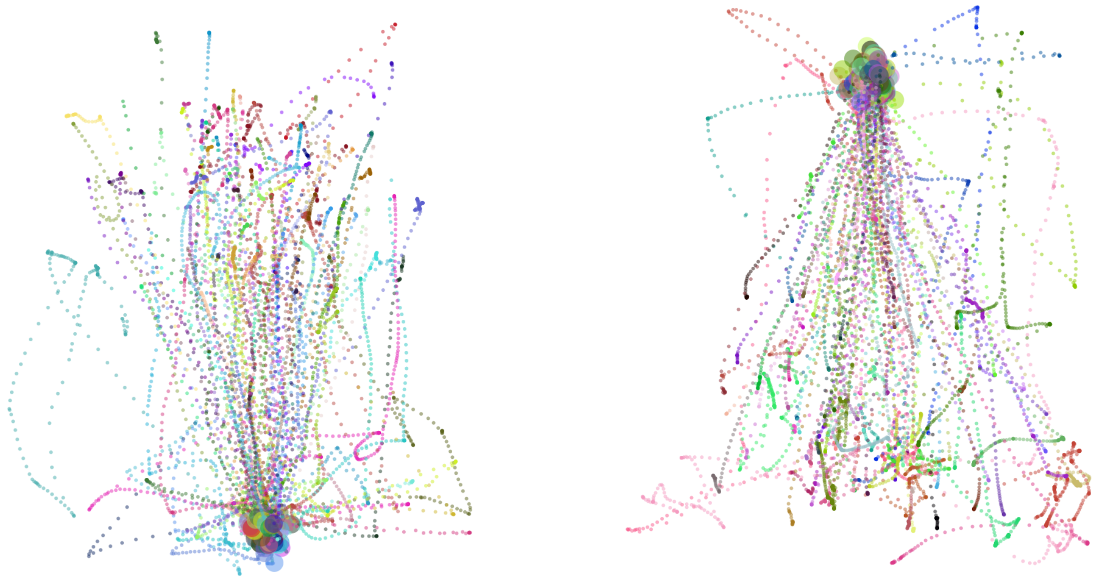
The subjective data does not indicate differences between the conditions. In both conditions participants rated dragging pictures onto the map to answer questions as *fun* (median=5) and *easy* (median=4 for *OLD*, 5 for *NEW*). The interaction with the map was considered *easy* (median=4) as well as the interaction with the images (median=4 for *OLD*, 5 for *NEW*).

The logged data reveals that the total map dragging distance measured in pixels is higher in condition *NEW*, however, I did not run a statistical test. Further, the absolute vertical dragging distance of images is lower in condition *OLD*. Visualizing the trajectories of the image movements (figure 5 and 6) also hints at a slightly higher image movement within the image area in condition OLD (see figure 4.4).

Discussion I outline three preliminary findings: First, dragging objects across the curved display connection is an obvious option even for first time users and thus *suitable for walk-up-and-use contexts*; *dragging downward is preferred* over upward dragging. Second, the Curve is suitable for *side-by-side group interaction*. In contrast to tabletops, this setup avoids orientation problems, opening up a range of dyadic cooperation scenarios. Third, designing user interfaces with task-to-surface mappings can be informed both by real-world analogies and the degree of subtask interactivity (e.g., writing on the horizontal and reading on the vertical display) and further supported by visual structures. However, in many cases, there will not be an *obvious task-to-surface assignment*.

4.1.2 Exploring Cross-Display Interaction Techniques

Following the exhibition of the quiz game, I continued the work on this topic in cooperation with Peter Yu, who wrote his Master's thesis under my supervision (see section 1.4). We



(a) Condition *OLD*. (b) Condition *NEW*.

Figure 4.4: Image dragging trajectories from all participants.

evaluated different touch interaction techniques aimed at facilitating cross-surface object movement. In particular, we considered *dragging* and *flicking* as baselines for closed-loop and open-loop control and compared them to other techniques proposed in related work.

Related Work

The first reported experiment conducted on the *BendDesk* explored the influence of the curved display shape on dragging performance (Weiss et al., 2010). The authors observed that dragging across the curve leads to slight discomfort, impairs spatial perception, results in higher curvature in hand paths and is slower than planar dragging. They argue that on the cognitive level, the curve may pose an obstacle in planning and executing the movement. On the motor level, hand movements trough the curve are more complex than planar movements.

In contrast to dragging, touch flicking is based on a physics simulation and users set an object into motion with a directional "flicking" gesture that applies a force to the object – feedback about its target position is only available once the motion has come to an end. Voelker et al. (2012) explored the effect of display connection type on flicking accuracy in a horizontal-vertical display setup and found that a curved connection yields a higher accuracy than an edge or gap between displays. Further, they found that errors in the motor execution stage have the highest influence on the accuracy and outlines the influence of surface location on motor execution. In particular, they showed that flicking gestures started in close proximity to the curved part of the display are most error-prone.

Both studies show that compared to planar surfaces, the curved shape – despite its seamlessness – imposes a “light barrier” (Weiss et al., 2010) on planning and executing movements through it. In addition, they conform with our own observations that downward is preferred over upward movement by users and that prolonged interaction causes arm fatigue. Therefore, our goal was to consider alternative interaction techniques, potentially avoiding some of these challenges. Understanding the *Curve* as a combination of two distinct surfaces, we turned to the related field of *Multi-Display Environments* (MDE). Nacenta et al. (2009b) define an MDE as an “an interactive computer system with two or more displays that are in the same general space (e.g., the same room) and that are related to one another in some way such that they form an overall logical workspace”. They provide a taxonomy of cross-display object movement interaction techniques (in turn partly inspired by large display research) which served as a basis for choosing adequate techniques for our experiment.

The framework underlying the taxonomy (see figure 4.5) comprises three levels: regarding *referential domain*, we were interested in techniques based on spatial display referencing (as opposed to non-spatial techniques, e.g. list-based referencing). Concerning *Display Configuration*, both planar and literal techniques were relevant: with planar configurations, a combination of displays – even if arranged in a non-flat fashion – forms a logical plane, and the input is modeled accordingly. An example is the *world-in-miniature* technique (WIM). Literal configurations are related to direct input: the input is based on the physical properties of the actual display configuration, such as direct dragging across two angular displays. In terms of *Control Paradigm* we were looking for both open-loop (continuous feedback) and closed-loop (delayed feedback) techniques.

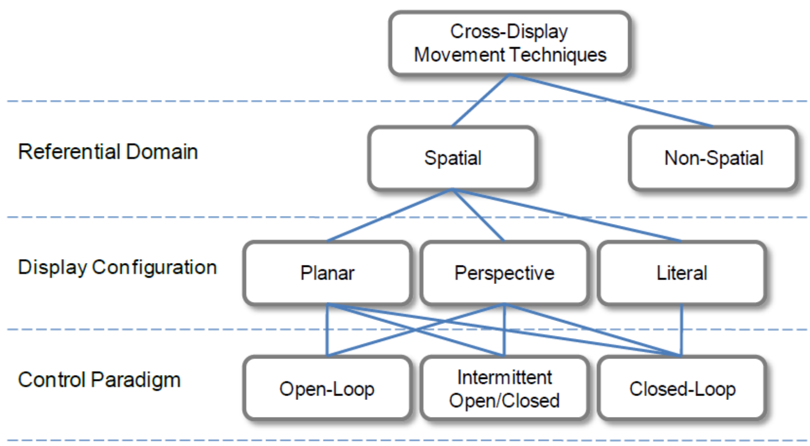


Figure 4.5: Framework for a taxonomy of cross-display object movement interaction techniques with three levels (Nacenta et al., 2009b).

The Interaction Techniques

Dragging Touch dragging is a well-established interaction technique for moving objects on touchscreens. With regard to the framework of Nacenta et al. (2009b), it is a spatial, literal, and closed-loop technique and serves as a baseline for the other closed-loop techniques investigated in this study.

Hold-and-Point Hold-and-point is a closed-loop, spatial, and planar technique. By holding draggable objects for a certain amount of time (we choose 800ms, based on informal tests), a virtual touch pad is activated and displayed as a semi-transparent object overlay. The touch pad preserves the aspect ratio of the curved display area and allows to position and move objects with absolute pointing. Visual feedback is provided only through the actual objects and not within the touch pad overlay (see figure 4.6 (2)). After three seconds without input signal, the touch pad fades away.

WIM Similar to Hold-and-point, WIM is a rectangular touch input area with absolute mapping, preserving the aspect ratio of the curved display. However, it is displayed continuously at a fixed position (depending on handedness) and contains visual information. In particular, it shows a scaled representation of the curved display area with its objects and targets. Hence, operations on objects and targets can be executed exclusively using the WIM. In our case, the WIM supported dragging operations and may therefore be classified as a closed-loop, spatial, and planar technique (see figure 4.6 (1)).

Flicking Flicking is a known open-loop, spatial and literal technique and serves as a baseline for an alternative open-loop technique.

Flicking via backdoor A backdoor introduces an alternative way to move an object towards a target. In desktop computing window managers, cursor wrapping allows to reduce target distance in pointing tasks. This idea has been formally described and evaluated by Huot et al. (2011), who conceptionally turn the desktop into a torus, allowing the cursor to take shortcuts from one edge to another. In our study, we introduce such a backdoor for flicking, which allows to alternatively flick objects through the bottom edge of the display instead of across the curve (see figure 4.6 (3)), avoiding the error-prone movement planning through the curve (Voelker et al., 2012).

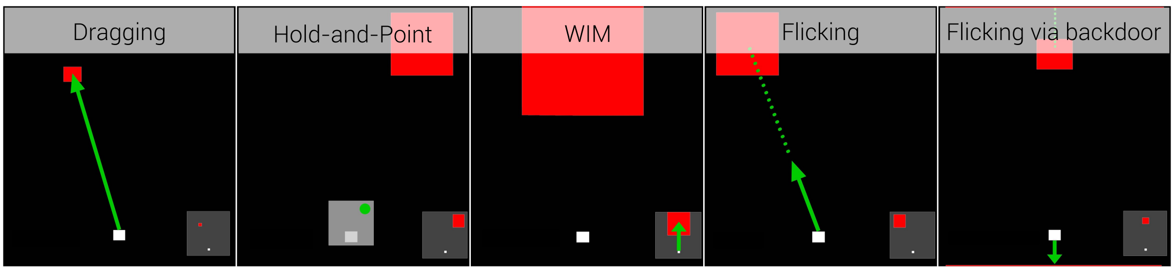


Figure 4.6: The five interaction techniques (from left to right): (1) Dragging, (2) Hold-and-Point, (3) WIM, (4) Flicking, and (5) Flicking via backdoor.

The Experiment

The experiment's goal was to compare the different interaction techniques in a basic object movement task, where a source object displayed on the horizontal surface shall be moved into a target area on the vertical part of the screen. To simplify the experiment design and based on previous findings, we considered only the upward direction.

Independent variables: Next to the five interaction techniques, we introduced four target area sizes (150x150, 300x300, 500x600, 960x1080 pixels), as well as three target area positions (left, center, right). For the largest target area, there where only two positions (left, right).

Dependent variables Task completion time (ms), NASA RTLX score; for open-loop techniques: accuracy.

Other measures Additionally, we collected subjective data with a Likert-scale questionnaire (five-point, 1 = do not agree at all, 5 = fully agree) and a short structured interview, which asked for personal preferences, comments on the techniques usefulness, potential application scenarios, further ideas and a technique ranking.

Working hypotheses

H1 Dragging and flicking will be outperformed by the proposed alternative techniques with regard to task completion time, workload and subjective preference.

H2 Depending on required accuracy (target size), different techniques will be preferred.

The object's starting position was the same across all trials. The positions of the goals (i.e. left, center, right) were chosen in a way that kept object-to-target distances constant across the varying target area sizes, except for the extra large targets, which had only two target positions and filled the whole height of the vertical display area. Additionally, both Flicking via backdoor and Hold-and-Point were techniques that inherently enabled shortcuts (see figure 4.7).

Time measurement was implemented through software logging. It started with the first registration of object movement and ended as soon as the object was fully contained within the target area. As the open-loop techniques did not guarantee an instantly successful object movement, they involved a second movement phase to complete the movement after the flicks, during which objects could be dragged into target areas. Further, in these cases, the distance between targets and flicked objects were recorded to measure flicking accuracy.

Procedure The experiment consisted of four parts and took approximately 30 minutes to complete. In the beginning, participants filled out a consent form and a demographic survey and were introduced to the nature of the task by the experimenter. In the second part, the participants executed the task. In two cycles, they had to complete a randomly generated sequence of 12 object movements (four target sizes, three target positions) with each of the five techniques, resulting in 60 object movements per cycle; the sequence of interaction

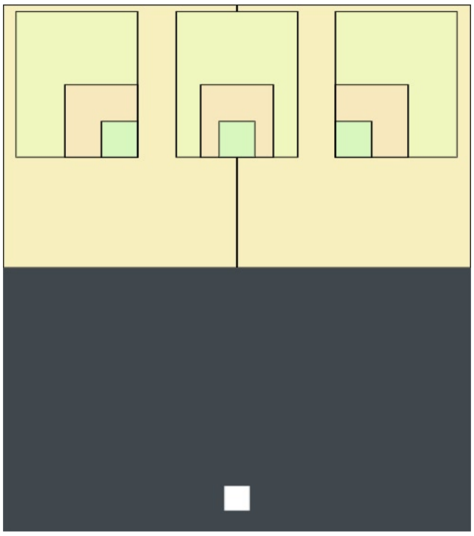


Figure 4.7: Task overview: the white source object had to be moved into target areas differing in size and position.

techniques was counterbalanced with a latin-square design. The first cycle was considered as training phase and only data from the second cycle was used for analysis. In the third part, participants filled out questionnaires and eventually the interviews were conducted.

Participants We recruited 15 participants (3 female, aged 18-33) who were compensated either with a voucher for an online store or with additional study credits. 14 were right-handed, one was left-handed; 13 had a background in media informatics.

Results

Figure 4.8 illustrates the mean task completion times for upward object movement for all target sizes. A one-way repeated measures ANOVA with a Greenhouse-Geisser correction determined that mean task completion times differed statistically significantly between interaction techniques ($F(2.309, 30.012) = 9.935$, $p < 0.0005$). Post hoc tests using the Bonferroni correction revealed that the WIM technique (1541.643 ms, SE = 187.361) was significantly faster than Dragging (2119.714 ms, SE = 169.668)($p = .044$) and Hold-and-Point (3080.786 ms, 257.187)($p < .005$), which was significantly slower than Dragging ($p = .042$) and Flicking via backdoor (1994.571 ms, SE = 241.136)($p < .005$).

Figure 4.9 shows the task completion times separated by target size. For small targets, a one-way repeated measures ANOVA with a Greenhouse-Geisser correction determined that mean task completion times did not differ statistically significantly between interaction techniques ($F(1.427, 14.273) = 2.804$, $p = 0.106$). For medium targets, the ANOVA determined statistically significant differences ($F(5, 65) = 3.564$, $P = 0.007$). Bonferroni-corrected post hoc tests showed that WIM (1448.9286 ms, SE = 118.740) was significantly faster than Hold-and-Point (3017.928 ms, SE = 202.839)($p < .005$) and also Hold-and-Point without dwell-time (2217.928 ms, SE = 202.839)($p = 0.025$).

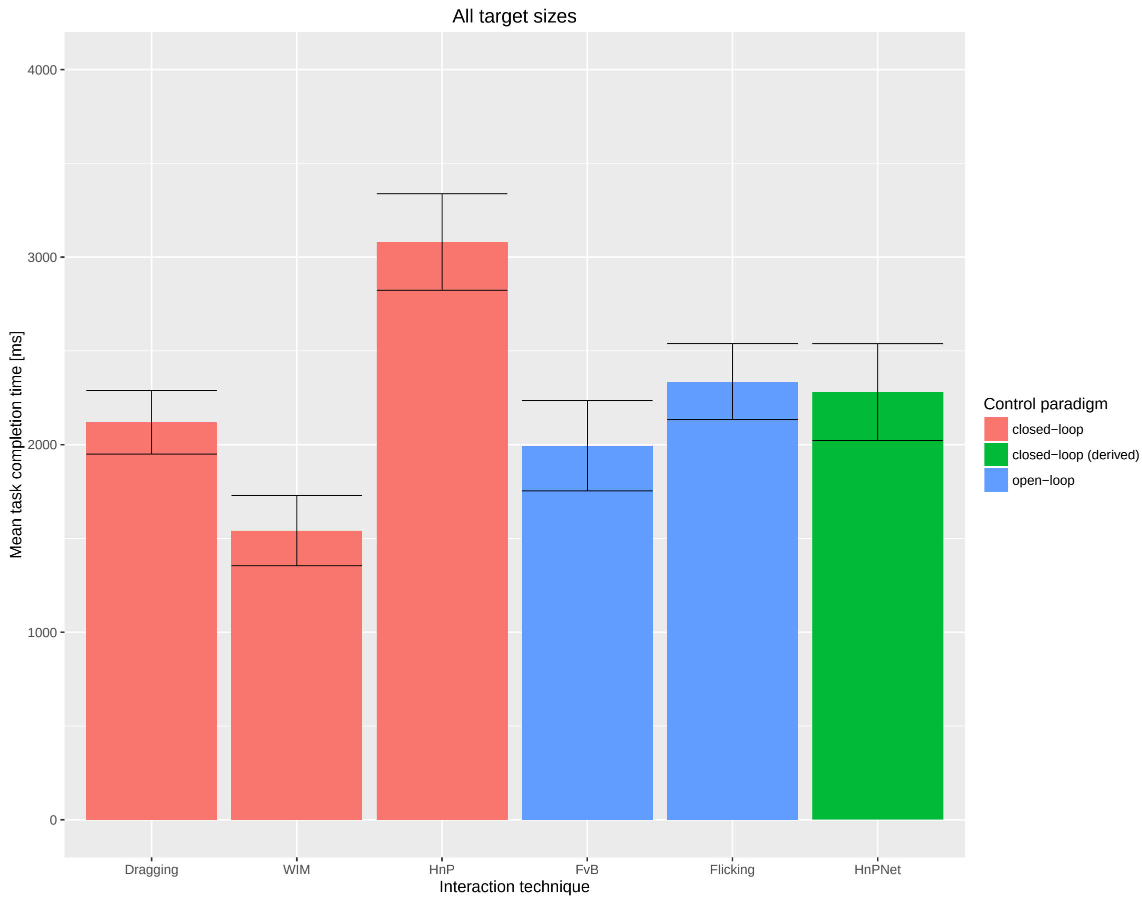


Figure 4.8: Mean task completion times per interaction technique for upward object movement. Error bars indicate SE.

For large targets, the ANOVA determined statistically significant differences ($F(5, 45) = 3.680$, $p = 0.007$). Bonferroni-corrected post hoc tests showed that WIM (1486.9286 ms, SE = 118.740) was significantly faster than Hold-and-Point (3017.928 ms, SE = 202.839)($p = .011$).

For extra large targets, the ANOVA determined statistically significant differences ($F(5, 50) = 3.680$, $p < 0.005$). Bonferroni-corrected post hoc tests showed that WIM (963.273 ms, SE = 113.163) was significantly faster than Hold-and-Point (2241.818 ms, SE = 174.828)($p = .002$). Also, flicking via backdoor (684.636 ms, SE = 38.752) was significantly faster than Hold-and-Point ($p < 0.005$).

The Nasa RTLX scores indicate that both Hold-and-Point (4,83) and WIM (3,78) result in less workload than dragging (8,23). For open-loop techniques, Flicking via backdoor (8,52) resulted in less workload than normal flicking (10,54).

The questionnaires show, that WIM (median = 5) and Hold-and-Point (median = 4) are rated best in terms of perceived time advantage, followed by dragging (median = 3) and flicking

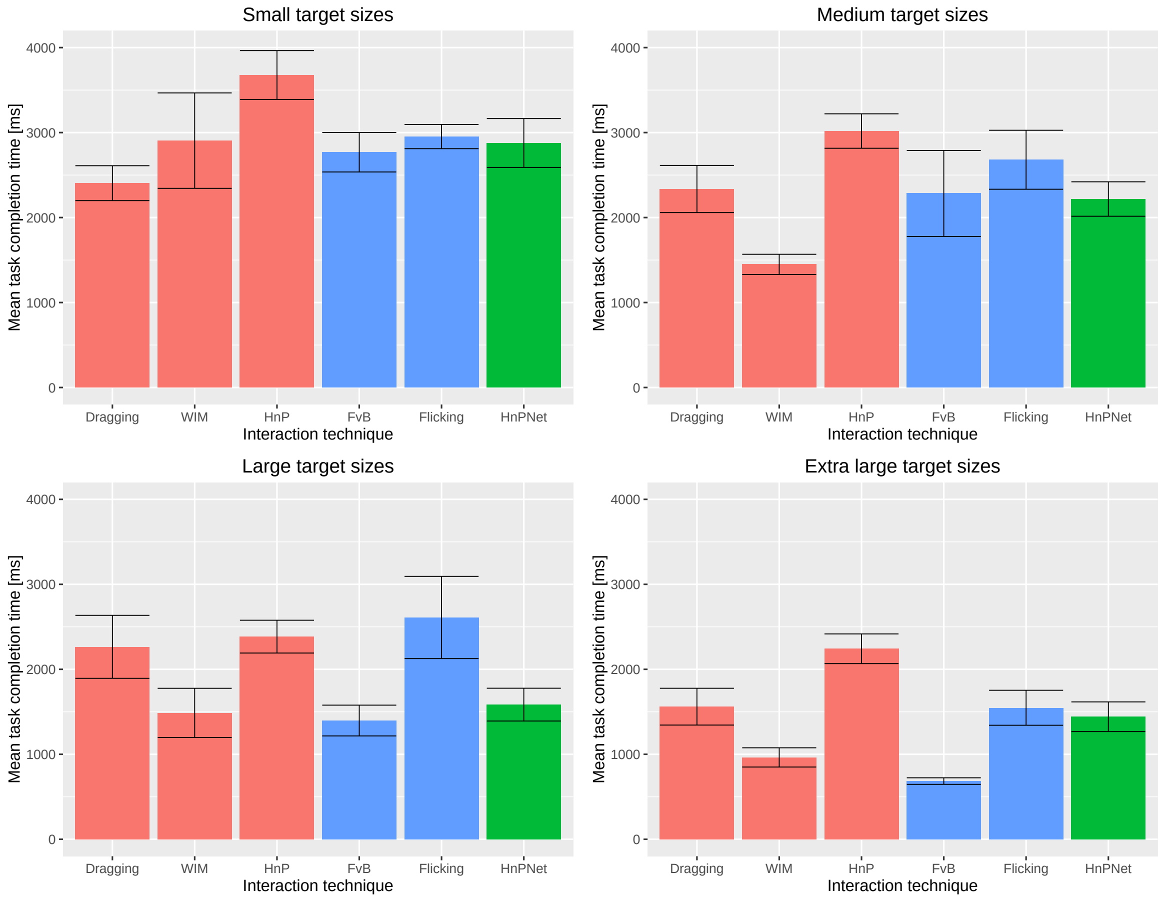


Figure 4.9: Mean task completion times separated by different target sizes. Error bars indicate SE.

via backdoor (median = 3), and flicking (median = 2). Regarding perceived accuracy and efficiency for large targets, all techniques were perceived well. WIM and Hold-and-Point are rated best (all medians = 5). Dragging was rated as very accurate (median = 5), but slightly less efficient (median = 4). Flicking via backdoor was rated as very efficient (median = 5), but slightly less accurate (median = 4). Normal flicking was rated as both accurate and efficient (medians = 4). For medium target sizes, the perception of accuracy and efficiency changes only for the open-loop techniques. Flicking is perceived rather inaccurate and inefficient (medians = 2); flicking via backdoor is rated slightly better regarding accuracy (median = 3) and positively regarding efficiency (median = 4). For small targets, dragging is perceived best in terms of accuracy (median = 5), followed by WIM and Hold-and-Point (medians = 4). All three techniques are rated as efficient (medians = 4). Flicking via backdoor is rated rather negative (medians = 2), but still better than normal flicking (medians = 1).

In the subjective rankings collected during the interview, the most preferred techniques were Hold-and-Point (n = 8), WIM (n = 5), and flicking via backdoor (n = 2). The second most preferred techniques were dragging (n = 5), WIM (n = 4) and flicking via backdoor (n = 3). Hold-and-Point and WIM were liked for the smaller physical distances that had to be covered and the resulting smaller movements required. Further, the perceived aiming effort was lower for flicking via backdoor compared to flicking, which was perceived as error-prone by 13 participants.

Discussion

Regarding time, the closed-loop techniques outperformed the open-loop techniques for smaller target sizes, which required higher precision. This was not surprising, as the nature of open-loop control delays the evaluation and correction of an action. However, for large target areas, i.e. little required precision, the open-loop techniques performed well. In particular, flicking via backdoor was more accurate and preferred across all target area sizes compared to normal flicking.

Considering all collected data, both WIM and Hold-and-Point are well-perceived techniques, yet they differ in measured performance: while WIM yields faster task completion times and less workload than dragging with all but the smallest target size, Hold-and-Point was the slowest of the three closed-loop techniques across all target sizes. If the dwell time needed to activate the touchpad is factored out (a hypothetical twist – the actual dwell time may never be zero), the significant differences fade away. However, the subjective data also suggests that participants did not perceive the dwell time as a penalty compared to the always available WIM. This indicates that for low to medium precision, planar techniques requiring less arm movement can outperform literal dragging, which is only advantageous for small targets.

This suggests further room for improvement: on the one hand, effective input strategies for cross-display object movement in such dual-display setups may need to comprise several techniques, e.g. a combination of dragging and WIM. On the other hand, a more systematic assessment of novel interaction techniques may also lead to further improvements regarding accuracy. For instance, we chose one size for WIM and Hold-and-Point based on informal tests, but to further inform the design of such techniques, the effect of form factors (e.g., size) on performance measures (e.g., time, accuracy) should be explored systematically.

In summary, the choice of interaction technique depends on the required accuracy. For coarse interactions, planar techniques that map input from a flat and scaled-down representation of the display environment to the actual displays (e.g. WIM), but also flicking via backdoor can clearly outperform literal techniques. However, in case high accuracy is needed, literal dragging turned out to be the best trade-off.

4.2 3D Interaction on Dual-Display Setups

The motivation to explore 3D interaction in this phase of my work was two-fold: firstly, due to advancements in personal fabrication, which transitions from an early adopter to a mainstream phenomenon, simplified 3D content creation and editing tools are gaining in importance. Secondly, controlling interactive 3D environments with 2D touch input signals poses a fundamental challenge that has inspired a variety of research (e.g., Jankowski and Hachet (2013)). The goal of my project was to employ an existing visualization concept based on the Curve to give users the impression of sitting at a real desk with a virtual 3D environment on top of it and explore suitable interaction techniques to interact with the virtual 3D scene. In particular, I wanted to address the challenge of mapping 2D input signals to more degrees of freedom (DoFs) in a novel way by utilizing the non-flatness of the *Curve's* display. The work presented in this section was done in collaboration with Mirjam Mickisch, who wrote her Bachelor's thesis under my supervision (see section 1.4).

4.2.1 Background

In recent years, cheap 3D printers have become available on the consumer market, accompanied by the emergence of Maker or DIY subcultures. This has created novel needs for products that support the intertwined handling of both virtual and physical 3D objects, which increasingly permeates amateur and hobbyist activities. These needs are addressed for instance with simplified 3D modeling software, such as Tinkercad [2] or Autodesk's suite of 123D apps [3], but also with novel devices, such as Hewlett Packard's Sprout [4], a dual-surface device that allows to scan and edit 3D objects (see figure 4.10(b)).

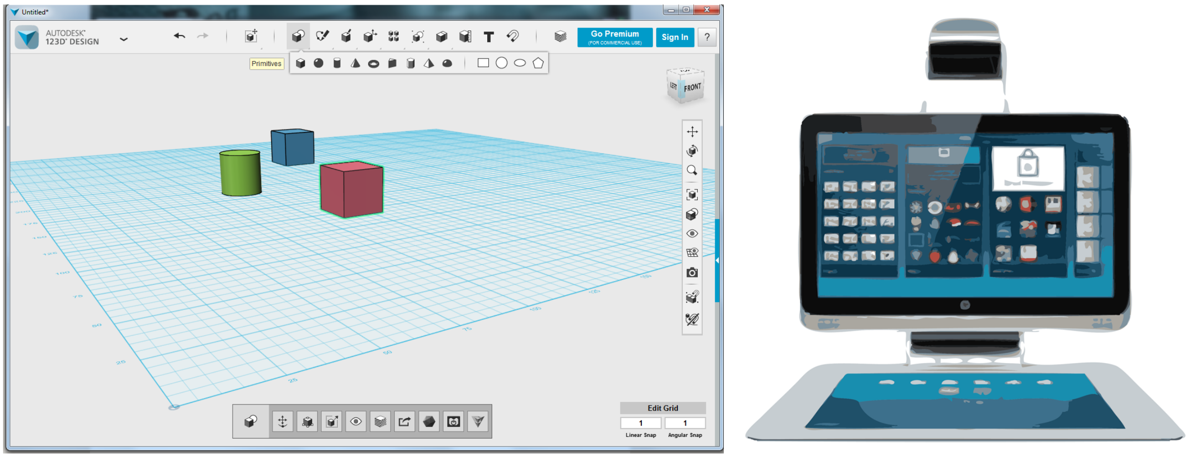
The dual-surface character is interesting in this context as it has the potential to move physical and virtual workspaces closer together. For instance, Schwarz et al. (2012) introduced a visualization concept based on the *Curve* that extended an *overview+detail* view with a perspective extension of the overview plane rendered on the horizontal part of the surface, extending through the curved display segment and thus growing "naturally into the space behind the display surface" (Schwarz et al., 2012). This visualization technique was later adapted to a remote collaboration scenario, where *Curve* and *BendDesk* jointly formed the *PerspectiveTable*, each extending its physical horizontal touch surface seamlessly into a virtual representation of its counterpart (Hennecke et al., 2013), thus creating a virtual shared workspace with physical outlets.

In their extensive survey of interaction techniques for interactive 3D environments, Jankowski and Hachet (2013) distinguish between three fundamental tasks: (1) navigation, (2) object selection and manipulation, and (3) application/system control tasks. In our case, navigation was limited to rotating a virtual camera around the center of the scene, a simplification of orbiting, which is a widespread technique originally proposed by Chen et al. (1988) and Shoemake (1992). The focus of this project was on exploring object manipulation tasks, i.e. rotating, scaling and translating geometric primitives within a scene.

[2] `https://www.tinkercad.com/`

[3] `http://www.123dapp.com/`

[4] `http://www8.hp.com/us/en/sprout/home.html`

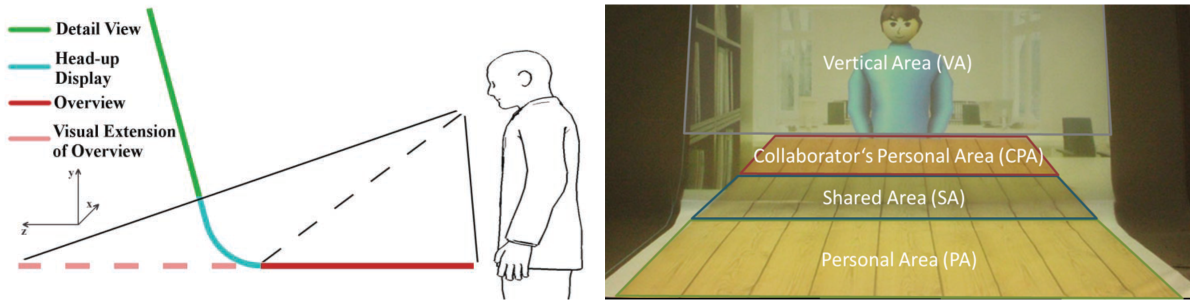


(a) *Autodesk's 123D Design* (b) *HP Sprout*

Figure 4.10: Examples of novel products targeted at the Maker scene. a) 123D design is an example of a simplified 3D creation and editing application that supports 3D printing. b) The HP Sprout allows to place physical objects on an interactive horizontal surface from where it can be scanned and transferred into the virtual realm.

Touch manipulation of objects contained in 3D environments has been addressed either with multi-touch gesture sets (e.g., Edelmann et al. (2009), Liu et al. (2012a)), or with graphical handles, sometimes called manipulators, widgets, or gizmos (e.g., Cohé et al. (2011), Schmidt et al. (2008)). While multi-touch gesture sets allow for high-bandwidth (integral control of multiple degrees of freedom, no explicit mode switches), they do not offer visual guidance and need to be internalized (Norman, 2010). In contrast, handles coupled with 3D objects externalize commands through visual structures. However, their often filigree appearance catered to cursor operation needs to be adapted to match the precision of finger input.

Application or system control tasks refer to the interaction style underlying an interactive 3D application. For instance, WIMP application control is based upon established concepts, such as menus or keyboard shortcuts and sometimes extended with screen-space representations of commands, e.g. virtual head-up displays. Post-WIMP system control (as in Jankowski and Hachet (2013)) is a broader term encompassing concepts that go beyond WIMP concepts, for instance graphical handles or menus displayed in 3D object space instead of 2D screen space. This topic directly relates to other chapters, where the role of different interaction paradigms 2.2.1 and their effects on 3D interaction 6.1 are discussed in more detail. In this project, we explored a novel screen-space input widget that considered the screen angle in its mapping from screen to object space and contrasted it with a widely used manipulator control.



(a) *Perspective+Detail* (b) *PerspectiveTable*

Figure 4.11: Using the curved display area to render a seamlessly connected perspective extension of the physical horizontal surface. a) Schematic sketch of the visualization concept *Perspective+Detail* (Schwarz et al., 2012). b) The visualization concept employed to create *PerspectiveTable*, a remote collaboration scenario with a shared workspace (Hennecke et al., 2013).

4.2.2 The Prototype

We developed a prototype for viewing and interacting with 3D content (see figure 4.12). The horizontal surface resembles a desk surface and is virtually extended into a plane that is projected into the curved part of the display. The virtual horizontal plane appears "in the depth" and constitutes a ground plane. The virtual 3D scene unfolds above this ground plane and contains 3D objects that are displayed on the vertical part of the display.

Input techniques, tasks and measures

Our explorative input method (*TNEW*) is based on two virtual track pads that spatially imitate the display's arrangement. The rationale behind this design is that established and widespread multi-touch gestures (dragging, pinch-to-zoom, and two-finger-rotation) can directly be mapped to different planes in the 3D object space using the pads' orientation as an additional input parameter. In particular, input on the horizontal pad manipulates selected objects in the scene's XZ-layer, while input on the vertical one manipulates objects in the global XY-layer.

To allow both integral and separate control of translation DoFs, the touch pads employ visual structures: dragging gestures initiated on visual bars on the sides of the pads (see figure 4.13 middle) restrict the corresponding translation to the respective axes. Further, to facilitate object selection, a WIM of the 3D scene was displayed on the horizontal part of the surface (figure 4.13 left), allowing indirect object selection by tapping on the minified objects in the WIM. As described above, scene navigation was possible through an axis-constraint version of camera orbiting: by applying one-finger dragging gestures applied to the green ground layer, the scene was rotated around its global y-axis (see figure 4.13 right).

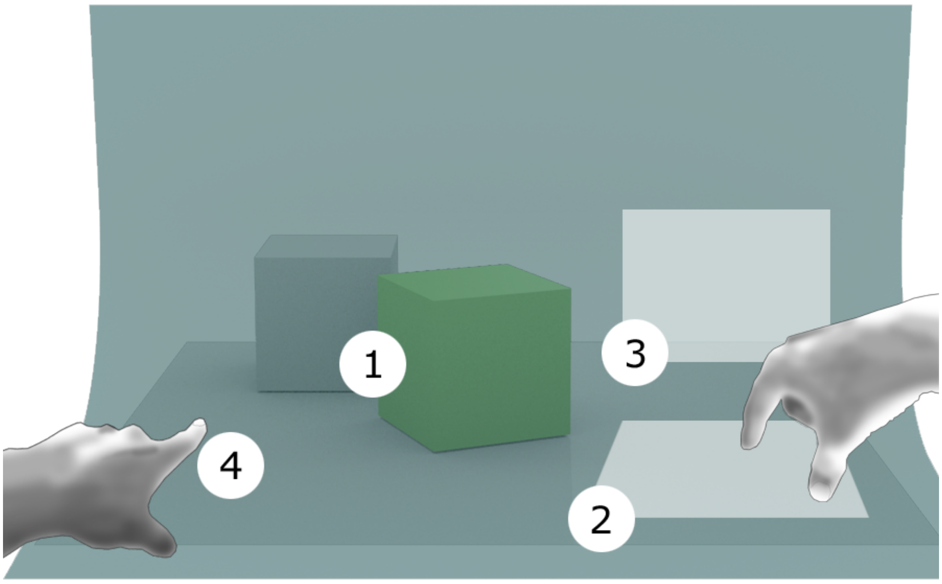


Figure 4.12: (1) 3D scene that seamlessly extends into the depth, (2) horizontal touchpad for indirect touch control: the selected object can be translated by one-finger dragging within the XZ-plane and rotated with a two-finger rotation gesture around the Y-Axis, (3) vertical touchpad: the selected object can be translated within the XY-plane and rotated with a two-finger rotation gesture around the Z-Axis, (4) the virtual camera can be rotated around the scene by dragging it left or right. First rotating the scene and then using the vertical touchpad can achieve object rotations around the X-axis.

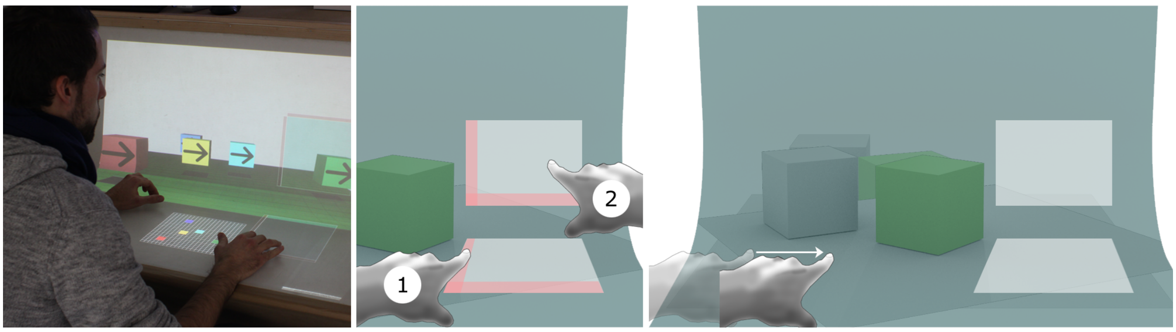


Figure 4.13: (left) The visualization of the 3D scene with selection widget and virtual touchpads, (middle) (1) translation constrained to one axis (global z-axis), (2) 2D-translation (xy-layer), (right) scene rotation by dragging the scene left or right.

As a baseline technique (*TOLD*), we implemented a transformation manipulator known from most 3D modeling applications (4.14). In translation mode, a dragging gesture started on one of the manipulation axes translated the object in the chosen dimension. In rotation mode, surrounding circles in the three coordinate dimensions could be used to rotate the object. To ensure comparability, a semi-transparent sphere around the manipulator's origin allowed to transform the object integrally in the plane parallel to the display plane. In contrast to WIM selection, objects were selected with direct taps and scene rotation worked as described above. Manipulation modes were switched using the keys *t* or *r* on a standard keyboard. To simplify the study design, we omitted scaling in both conditions, because at this stage, we were not interested in finding a complete and well-performing input technique, but in gathering first observations and feedback on the general concept.

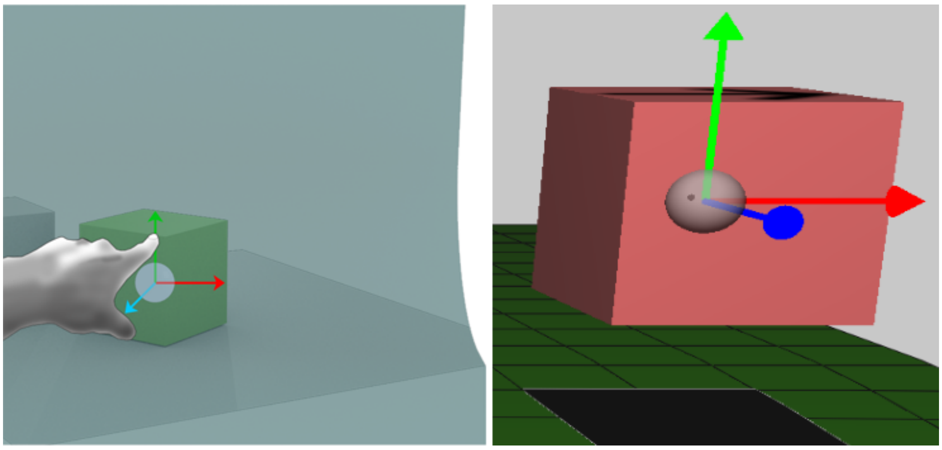


Figure 4.14: The transformation manipulator for direct touch interaction (concept and detail).

The following research question was investigated: *How does the indirect, screen space input technique based on the virtual touchpads affect the performance of rotation and translation compared to direct touch manipulation using an established object space tool?*

Docking Task Introduced by Zhai and Milgram (Zhai and Milgram, 1998), the docking task asks a user to align a manipulable cube with a fixed target cube. In our case, the available operations to achieve dockings were object translations and rotations. Further, we introduced two difficulty levels: L1 included transformations in only one coordinate plane (i.e. 2D translation or 1D rotation), whereas L2 included unrestricted 3D translations and rotations. Further, we used different distances and angles, resulting in 18 different docking tasks.

Selection Task In the selection task, the participants had to copy a given figure consisting of five differently colored cubes. One of the cubes had a fixed position and the other ones had to be selected and then translated to their according position.

Compound Task The third task was similar to the docking task but additionally involved barriers between the movable and the target cube. The barrier consisted of four cubes that were not allowed to be intersected. The position of the barrier was chosen in a way that required a rotation of the whole scene in order to complete the task.

We measured task completion times and the number of applied transformations. For each transformation we recorded completion time and distance/rotation. Additionally, we recorded the amount of scene rotations. Post-questionnaires were used to assess the participants' opinion regarding *ease of use*, *understandability* and *ergonomics*, which were assessed with five-point Likert scales ranging from 1 (best) to 5 (worst).

Design and Procedure

We used a within-subjects design with *input* method as the main factor and *task* as secondary factor. Input technique was counterbalanced, and with both input techniques the participants had to fulfill all three tasks, starting with docking, followed by selection and ending with the compound task. During the docking task, the order of the single docking operations was randomized.

Each participant was seated centered in front of the prototype. A short training phase preceded each task. Qualitative data was collected with paper questionnaires after each group of tasks and at the end of the study.

Results

We recruited 17 right-handed participants (9 female, aged between 21 and 61, 9 students and 8 participants with various occupational backgrounds) via an announcement posted on our lab's facebook page and compensated them with a 10 Euro voucher for an online store or alternatively with extra study credits.

Task completion time A two-way repeated measure ANOVA with *input* method (*TNEW*, *TOLD*) and *task* (docking, selection, compound) as factors did not reveal a significant main effect of input method ($F(1,13) = 2.432$, $p = .143$). For *task*, there was a significant main effect ($F(1.2, 15.606) = 80.325$, $p < .001$), however, this is not surprising due to the different task designs. Further, there was a significant interaction effect between *input* and *task* ($F(2,26) = 10.857$, $p < .001$), indicating that selection takes longer with *TOLD* than with *TNEW*.

Perceived performance and accuracy Here, performance reflects subjective ratings of task completion time. Participants rated their task completion time separately for translation and rotation. For translation, performance was perceived slightly better with *TNEW* (median = 2) than with *TOLD* (median = 3). For rotation, the participants' perception did not differ between *TNEW* and *TOLD* (median = 3). For both the selection and the navigation task, performance was perceived as equal (median = 2).

Like performance, perceived accuracy in the docking task was rated separately for translation and rotation. Overall, ratings did not differ between *TNEW* and *TOLD*, but translation (median = 2) was perceived as more accurate than rotation (median = 3). For the selection task and the navigation task, accuracy was perceived as equal (median = 2).

Convenience At the end of each task, participants were asked with which input method they felt more comfortable. 53% stated that they preferred *TNEW* for the docking tasks, 65% preferred *TNEW* for the selection task and 53% preferred *TNEW* for the navigation task.

Further findings For rotations, we observed that the visual guidance offered by the manipulator in *TOLD* supported the participants in their spatial planning of transformations. In contrast, with *TNEW*, participants needed extra rotations to determine if they resulted in the desired effect.

4.2.3 Discussion

The preliminary results indicate that for the chosen tasks, our explorative input method (*TNEW*) and the direct touch manipulator (*TOLD*) yield similar task completion times. This indifference is also reflected in the subjective ratings of perceived performance. Taking into account the maturity of the manipulator as an interaction instrument in this context, this indifference can be seen as encouraging.

While the subjective data indicate a slight preference for *TNEW*, more research is necessary to better understand the reasons for it. During the experiment, the manipulator's graphical axes and rotation circles seemed to better support the spatial planning and execution of transformations, whereas *TNEW* seemed to be more comfortable. This trade-off suggests that integrating similar visual guidance into the virtual touch pads of *TNEW* may improve its overall performance.

The study results also indicate that in our setup the indirect selection technique was faster than direct touch selection. While at the current state the selection widget resembled the WIM from the previous section and was conceived as a separate tool, an integration of both instruments seemed interesting, as the transformation of objects with the touch pads requires a previous selection of an object.

4.3 Chapter Summary

Apart from the three case studies presented above, I have conducted several further explorations, which for reasons of brevity and focus are not detailed in this thesis. However, an important outcome of this explorative phase has been a precise definition of my research focus. Starting with a vague question (*How can we effectively make use of a seamless curved connection between horizontal and vertical touchscreens?*), the different prototypes and small experiments (also the ones not detailed in this thesis) helped us to gain an impression of more important questions. In particular, we identified several aspects that will confront application designers with trade-offs in this device context.

Presentation trade-off The seamlessness of the *Curve's* display, its shape that puts virtually every corner of the display within arms' reach, and its classification as multi-touch device raise questions relevant for the design of adequate graphical user interfaces. Structural elements, such as physical display bevels, but also virtual windows have to be rethought and classical layouts are extended with the question of physical orientation. As a starting point, we tried to work with physical analogies, both in the quiz game scenario and with the interactive 3D environment. We found that imposing a visual structure dividing horizontal and vertical areas does not destroy the affordance that objects can be dragged across areas through the curve. However, defining suitable structures for dual-surface devices remains a design challenge. It may be informed by existing task/surface assignments, or experimentally, but as the case of the quiz game indicates, there might not always be a clear solution.

Input technique trade-off For established *literal* touch techniques, such as touch dragging or flicking, the dual-surface context introduces novel challenges: dragging involves large arm movements and leads to arm fatigue and flicking is error-prone. While the public exposition of the quiz game indicates that they can be employed in spontaneous *walk-up-and-use contexts*, where a potential novelty effect outweighs slight physical discomfort, the idea of a renewed desktop workspace also requires support for prolonged interaction with complex applications. Here, the duality of the interactive surfaces seems more important than the curved connection. By turning to the field of *Multi-Display Environments*, we have started to explore complementary input techniques with the goal to provide comfortable and precise input. In particular, we have seen that *planar* techniques – scaled-down representations of the display area – can outperform *literal* techniques for object pointing and selection tasks in terms of task completion time. They may also increase input bandwidth by incorporating the information of the angle between the displays into the mapping of 2D input signals to 3D environments.

Adding visual features to such input areas is a valuable, yet not well-understood design resource: in the object movement tasks, the WIM was the fastest and preferred input techniques and in the 3D selection task, it outperformed direct selection. In the 3D environment, the visual subdivision of the input areas allowed spatial access to both constrained and unconstrained 3D transformations, rendering the indirect touch pads as fast as the established 3D manipulator. However, introducing virtual *direct touch* input devices also raises questions: firstly, in the object movement case study we observed the well-known fat-finger problem: for the smallest target size in the object movement task, completion times with WIM increased beyond direct dragging. This indicates opportunities for exploring touch-based indirect pointing techniques that allow comfortable, precise and fast performance. Secondly, the characteristics of eyes-free interaction with virtual touch pads needs to be explored in more detail: on the one hand, the interactive surface does not provide haptic cues that facilitate target acquisition. On the other hand, when the touch pads involve dynamic visual representations of screen content, a shift of visual attention is required, but may still result in superior performance.

Interaction style trade-off In general, a setup consisting of a connected pair of horizontal and vertical touchscreens raises questions about adequate interaction models. Firstly, unifying both surfaces, either logically or physically, more strongly emphasizes touch as the prevailing input modality than touch-enabled displays in PC or laptop devices, where WIMP interaction prevails and limits the impact of direct touch input. However, many features of WIMP interaction seem desirable in such setups: from an ergonomic perspective, arm fatigue is prevented by confining physical manipulation mostly to the horizontal surface. From a mapping perspective, the existence of dedicated input device spaces allows to support interaction through transfer functions. For instance, touch dragging vectors can be gained and then added to object positions in order to cover larger distances with smaller physical movements.

Secondly, the proliferation of touch interaction styles through mobile devices has empowered people to use one- and two-finger touch gestures to an extent that they expect touchscreens to support them. The observation of the quiz game revealed that despite the novel form factor, dragging objects across the surface and navigating the map using zoom and pan gestures were assumed features. While some track pads, for instance Apple's Magic Trackpad, also support touch gestures in WIMP environments, relatively little is known about how user interfaces for large touchscreens may incorporate them in ways that differ from direct touch paradigms, where their applicability cannot be inferred from other contexts (e.g., mobile). This uncertainty is reflected in existing programming toolkits: gesture event handling usually assumes that event listeners are registered to virtual objects and that received signals are used to manipulate the objects directly. However, an event handler can also be registered to one object receiving the touch signals and then manipulating other objects accordingly [5]. For instance, applying touch gestures indirectly on the horizontal surface for controlling 3D objects displayed on the vertical surface again raises a question about mapping, which we explored in a further experiment not detailed in this thesis (Guererro, 2014): should the transfer of input signals be based on the horizontal manipulation space or the vertical output space?

Review By building and evaluating interactive prototypes, we gathered a hands-on understanding of potential challenges and trade-offs, which ultimately led to a shift of focus: instead of focusing on the type of connection between horizontal and vertical touchscreens, we recognized the need to understand more basic questions regarding dual-surface configurations first: *What is a suitable interaction paradigm for such a display setup?* (Palleis, 2014b) and *What are adequate interaction techniques?* Based on the findings and observations, I identified a potential solution in designing dedicated input areas, which allow indirect control of domain objects.

[5] In some cases, e.g. with Microsoft's .NET framework, the direct touch concept is indeed hard to bypass.

III

"THE TOOL SPACE: MEDIATED INTERACTION IN DUAL-SURFACE SETUPS"

5
Tool Space

Dual-surface workspaces evoke a question of tool use: the space on a physical desktop is traditionally used to store and operate physical objects and tools (e.g. paper, hole puncher, computer input devices etc.). Added touchscreen technology may be used to replace physical objects and devices with virtual clones, or to more closely couple existing or novel physical objects with digital content (see section 2.3.1).

Based on the theoretical findings from the sections 2 and 3, which show that indirect touch input is an important but underexplored modality in dual-surface research, and the related user interface design trade-offs identified in the previous chapter, I developed a set of assumptions about an adequate interaction model for dual-surface workspaces. This chapter introduces these assumptions, which form the basis for a preliminary interaction model with the title *Tool Space*. Its core idea is to maintain a logical separation between horizontal and vertical surfaces. In particular, the horizontal surface is conceived as an input device space while the vertical display remains predominantly an output device. In contrast to physical input devices, like mouse and keyboard, the input devices used in the *Tool Space* are *virtual*, rendering them *dynamic*, but also *flat*.

In this chapter, I will first discuss further related work which informed the concept of the *Tool Space*. Also, I will briefly outline the technical foundations underlying the implementation of the tools discussed in this thesis. Subsequently, the concept is illustrated with two case studies which apply and discuss core ideas of the *Tool Space* using concrete application scenarios.

5.1 Background

Sections 2 and 3 discussed the theoretical implications of introducing interactive surfaces to computer workspaces, including the different notions of *directness* conveyed by different interaction paradigms, and the previous chapter has complemented these with practical evidence of resulting trade-offs. In particular, the explorative case studies have shown that reintroducing input devices as virtual mediating instances, i.e. visual elements that transduce input information from their input space to a connected output space, can facilitate interaction in a dual-display context. Given the dynamics provided by graphical output and potentially large physical dimensions, the horizontal surface can provide *multiple* such instances, which *adapt* to both applications contexts and *user characteristics*.

5.1.1 Theoretical Foundations: Tools as Mediators

The mediating role of tool usage has been analyzed by the philosopher Don Ihde, who described how humans develop relationships with the world through the use of tools (Ihde, 1990). For instance, *embodiment relationships* are formed with tools that – through mastery – become extensions of the human body. In activity theory, which has been employed in HCI early on, the notion of activities as the basic unit of analysis is established, and activities are understood as mediated actions, where actors transform objects using tools (Kuutti, 1995).

Tools are central in human-computer interaction: In their book *The Psychology of Human-Computer Interaction*, Card et al. (1983) describe computer systems as tools, that have many uses and thus involve their users in an open-ended dialog. Unlike most analogue tools, tools in HCI are comprised of a physical body (i.e. input device) and a "symbolic system of computational design"(Magnusson, 2009), or a physical and a logical device (Beaudouin-Lafon, 1998). Symbolic systems or logical devices are the *virtual* part of a tool and include algorithms, representations and feedback mechanisms that enable users to create or manipulate digital information such as text, images, or sound.

The compositional and mediating character of tools in interaction paradigms is further elaborated in the *Instrumental Interaction* paradigm (Beaudouin-Lafon, 1998), where tools are defined as *interaction instruments*, which are "mediator[s] or two-way transducer[s] between the user and domain objects".

In establishing the *Instrumental Interaction* paradigm, Beaudouin-Lafon (1998) describes central aspects of *interaction instruments*:

Activation Activation is the act of associating the physical with the virtual part of an interaction instrument. It is distinguished between *spatial* and *temporal* activation. With spatial activation, visible representations of tools that require screen space, e.g. scrollbars, are acquired and manipulated. With temporal activation, a *mode* is activated (e.g. by a button press) and persists until another mode is activated, which takes time and is less direct than spatial activation.

Reification On a basic level, reification is the transformation of computer commands into interaction instruments. For instance, the scrollbar is a reification of the commands that scroll a document. On a second level, reification describes the potential of interaction instruments to become objects of interest themselves. In particular, it introduces the idea of meta instruments which are used to manipulate interaction instruments, i.e. to modify their properties or organize them (e.g. menus or tool palettes).

Properties (Beaudouin-Lafon, 1998) introduces the following three properties of *interaction instruments*:

- The *degree of indirection* refers to a measure of spatial and temporal offsets generated by an interaction instrument describing a continuum between direct and indirect manipulation. The lowest degree of indirection can be achieved by minimizing the offsets between interaction instruments and domain objects.

- The *degree of integration* refers to the ratio between the number of degrees of freedom (DOFs) that users can control simultaneously in the application and and the number of DOFs captured by an input device. For many integral tasks (Jacob et al., 1994), for instance simultaneous 2D image rotation and scaling or 3D-positioning, the degree of integration provided by single-pointer devices is below one.

- The *degree of compatibility* is a measure for the similarity between the physical actions of users when performing interaction techniques and the visual feedback generated by the manipulated virtual object.

Surface Tools

From a technological perspective, touchscreens inherently are general-purpose input devices – they mediate interaction with dynamically changing screen content. However, advances in sensing technology have facilitated various input modalities, which influence the concept of tools as composite mediators between users and objects.

Graphical Tools Tools in this category mainly employ existing interaction instruments and comprise both mimicry of physical input devices, such as soft keyboards or virtual mice, and virtual tools, such as scrollbars or sliders. With multi-touch sensing, several instruments can naturally be activated simultaneously, allowing for spatially-multiplexed tool usage.

As discussed in section 2.2.3, finger precision is lower than cursor precision, thus generally requiring larger minimum target sizes. Therefore, conceiving touchscreen interaction instruments for spatial activation results in increased screen-space demands, which for instance in the mobile context conflicts with decreased screen dimensions.

For contexts with larger touchscreens, spatial activation is confined by the arms' reach. There have been attempts to virtualize the mouse (e.g., Bartindale et al. (2011)) – an adequate hardware solution to cover large distances with small movements – but here the challenge with hand-eye coordination resulting from missing haptic feedback and increased visual attention demands (section 2.2.3) is carried to extremes.

To improve the usability of touch-based graphical tools, either pen input is used (e.g., Ren and Moriya (2000)), visual representations of existing interaction instruments are adapated to better suit the requirements of finger input (e.g., Cohé et al. (2011)), or indirectness is employed to avoid finger occlusion (e.g., Benko et al. (2006)).

Gestural Tools As discussed in section 2.2.4, the overlay of input and output spaces has created an input style that does not require artificial devices, empowering users to operate with their fingers directly on graphical output. However, also the virtual parts of tools have been influenced profoundly: for instance, the input bandwidth provided by multi-finger input can make handles to scale and rotate 2D graphical objects needless (e.g., Reisman et al. (2009)). Gesture vocabularies – ranging from abstract to literal, from discrete to continuous – put all power to the users' hands, which through their gestures indicate both tool activation and manipulation, removing the need to graphically display tool representations. Regarding activation, this interaction style is both spatial – in the absence of instruments the input is directed to the object themselves – and often *quasi-temporal*, as different modes are entered implicitly through varying gestures. The degree of indirection is low, and both the degree of integration and compatibility depend on the respective gesture vocabulary.

In *TouchTools* Harrison et al. (2014), a trained classifier is used to distinguish between multiple hand gestures mimicking physical tool operation, such as holding a camera to activate the snapshot tool or sliding an eraser. Tool activation is spatial and quasi-temporal, and the degree of indirection is low as the manipulation occurs directly on the object. The degree of integration is low for tasks requiring more than two DoFs, as the classifier can infer which tool is activated, however, the activated tool always provides a maximum of two DoFs. The degree of compatibility is high, as the manual movements mimic real-world tool usage.

A comparison with other gesture sets reveals a trade-off between degree of integration and compatibility. For instance, the screen-space formulation for 2D and 3D direct manipulation (Reisman et al., 2009) increases DoFs with two-handed multi-finger touch gestures that enable both 2D and 3D object transformation. However, the two-dimensionality of screen-space results in finger movements that have less resemblance of according movements in the real world. Also, there are cases where compatibility may be hard to establish due to a lack of physical analogies.

In general, with gestural touch input both physical and virtual tool representations are gone. This lack of externalization is reminiscent of Donald Norman's concerns about mental load and the naturalness of gestural input (section 3.1.2). It further raises reification questions: *How can tools without externalized representations be modified?*, and *How can they be organized?*

Tangible Tools Interactive surfaces have also been a facilitator for tangible user interfaces (e.g., Underkoffler and Ishii (1999)), which are either generic (e.g. pucks) or specific physical objects acting as input devices. They are often linked to theories of embodiment and move tool representations from the virtual into the physical realm, drawing on the idea of tools as physical extensions of the body, leveraging existing intuitions about cause and effect in the real world (i.e. natural affordances).

Tangibles and graphical multi-touch widgets share many properties: they are closely linked to interactive surfaces, they allow for spatially-multiplexed and two-handed input, and invite collaboration. Tuddenham et al. (2010) have compared tangible with multi-touch tools and found that tangibles are faster to acquire, more precise during manipulation and they prevent *exit errors*, i.e. small unintended input signals occurring when removing fingers from graphical tool representations.

Tangible tools have also been investigated in the context of Curve, where – with the adequate adhesion technique – they become *Vertibles* (Hennecke et al., 2012), i.e. translucent tangible control elements that stick to the vertical display.

While active (i.e. movable, shape-changing) tangibles are being researched, they cannot adapt to the display rate of graphical output yet. This challenges their effectiveness to act as specific tools in interactive systems (e.g. *Photohelix* by Hilliges et al. (2007)), where switching applications with varying interaction instruments pose high requirements on the dynamics of both tool and object representations.

5.1.2 Tool Space: Design Rationale

The *Tool Space* is greatly influenced by the *Instrumental Interaction paradigm* (Beaudouin-Lafon, 1998). In particular, the added screen-space in dual-surface workspaces provides room for the *reification* of tools, or a way to rethink how computer commands are *mediated* by graphical control structures. The *Tool Space* deliberately avoids a discussion about virtualization of physical devices or physicalization of logical devices, but focuses on a conceptual adaption of interaction instruments to novel conditions.

Mediated Input and Spatial Activation The *Tool Space* adheres to the concept of interaction instruments as mediators or two-way transducers and spatial activation: objects displayed on the vertical screen are manipulated using dedicated graphical logical devices. In contrast to direct touch interfaces, these logical devices are spatially separate from the objects and displayed on the horizontal surface that is easily reachable and suited for prolonged finger input. A simple example is a scrollbar, which is moved away from the side of a document to be operated comfortably with one's fingers. This may lead to a high *degree of indirection* (Beaudouin-Lafon, 1998), which is taken into account to improve ergonomics. Further, instead of applying cursor-mediated indirect input (e.g., using a mouse) to direct manipulation systems (Moscovich and Hughes, 2006), the goal of the *Tool Space* is to increase the feeling of directness by providing a direct coupling between the logical input devices and specific functions.

Logical Device Design The logical devices fulfill *activity-specific* functions. Thus, activities – or applications – have specific sets of logical devices. They combine generic with spatially-multiplexed characteristics: the internal structure of logical devices can exhibit spatial-multiplexing, e.g. to enable direct access to several object-related commands, whereas throughout an activity, the device may be associated with different objects over time. Further, logical devices are static, i.e. they do not move (although their placement may be adapted or configured), in order to support hand-eye coordination despite the lack of haptic cues. Further, logical devices may involve configurable object representations, and are organized per activity.

Finger Input and Bimanuality In order to meet the requirements of finger pointing (Albinsson and Zhai, 2003), logical devices are conceived as rectangular areas that are easy to acquire and operate with fingers. Further, in addition to one-finger input (i.e. tapping, lifting and dragging), logical devices may employ two-finger rotation and pinching gestures. On the one hand, they increase the input bandwidth in a simple and widely established way, on the other hand, they have already been employed in a mediating or indirect sense in commercially available track pads. Further, indirect gestural input has been shown to be more precise (Knoedel and Hachet, 2011) compared to direct input. Further, the design of the logical devices can integrate bimanual operations to different degrees. They may be central to the tool reification process, e.g. by conceiving novel instruments to simultaneously transform selections and selected objects based on Guiard's principles (Guiard, 1987). They may also allow for a smooth transition from beginner to expert through the spatial layout of the instruments within the tool space, which may encourage the transition from one-handed to two-handed input styles. Also, by using two-finger gestures and potentially two-handed input, the *degree of compatibility* may be increased.

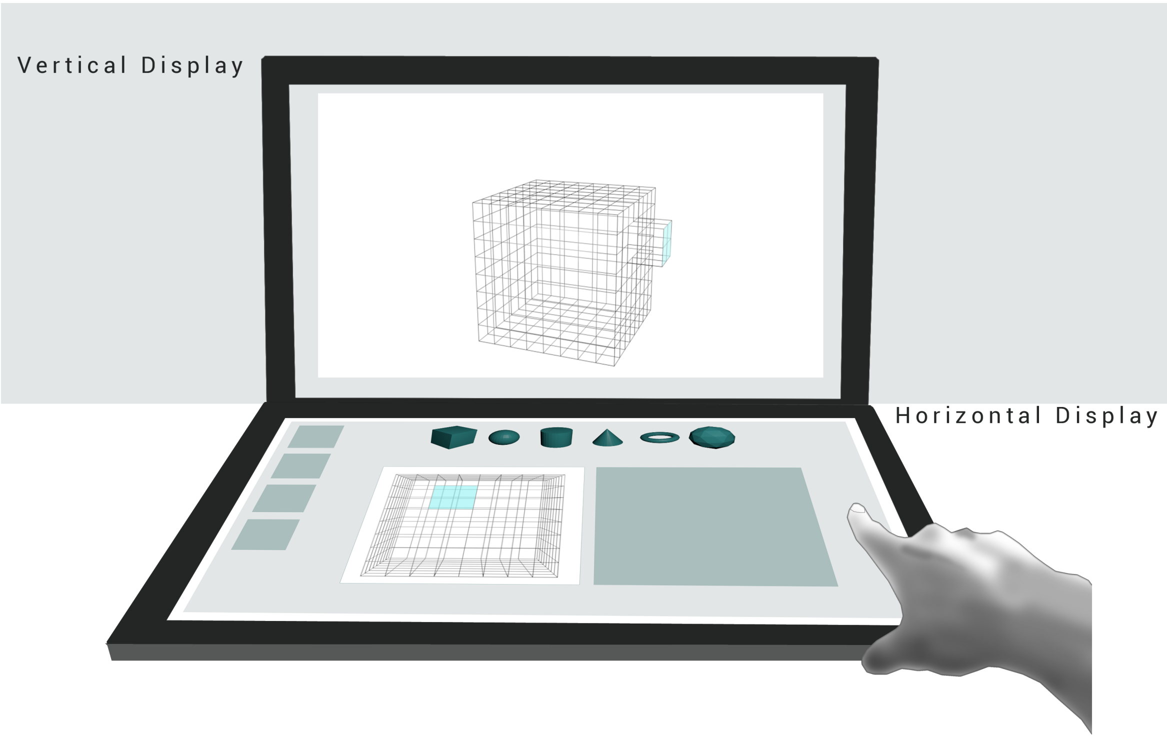


Figure 5.1: Conceptual overview of the *Tool Space*: manipulable objects are displayed on the vertical display, tools and meta-tools (tool bar on the left) on the horizontal display.

Technical Foundations

I want to briefly discuss the technical foundations of the logical devices in the *Tool Space*. The explorative case studies with the Curve (see section 4) have been mostly based upon *MT4J*, a Java framework for building multi-touch applications. Like many other software development kits that support multi-touch (e.g. Java, .NET), its event architecture provides a set of gesture events, such as zoom and rotation events. The idea of direct interaction paradigms is to register according event handlers to objects and then manipulate these objects based on the event data (see Algorithm 1 for a pseudocode description).

The central idea of the interaction instruments in the *Tool Space* is to register gesture event handlers on *tool* objects, which transduce the event data to other objects. In Algorithm 2,

Algorithm 1 Pseudocode for handling two-finger rotation events with direct touch (see figure 5.2 a)

```
rect1.setOnRotate(RotateEvent event -> rect1.setRotation() = rect1.getRotation() +
event.getAngle())
```

the handler is registered on one rectangle, but the manipulation information contained in the gesture event data is transduced to a second rectangle.

Algorithm 2 Pseudocode for handling two-finger rotation events with interaction instrument (see figure 5.2 b)

```
rect1.setOnRotate(RotateEvent event -> rect2.setRotation() = rect2.getRotation() +
event.getAngle())
```

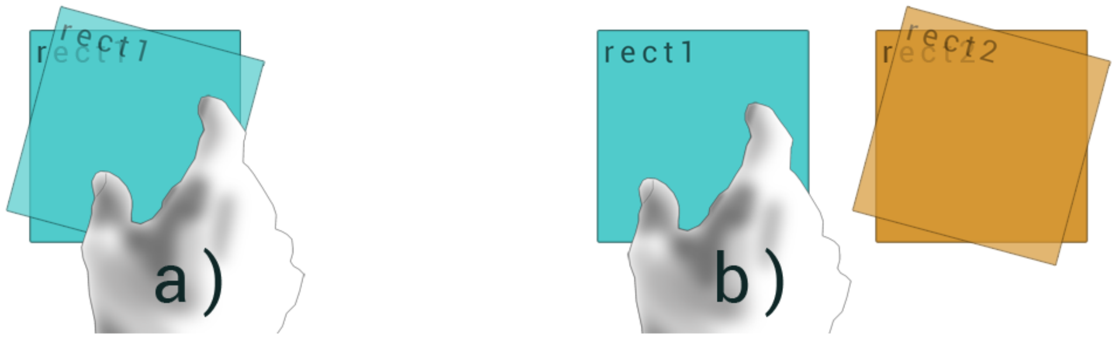


Figure 5.2: a) Direct touch gesture event handling, b) gesture event handling for indirect interaction instruments.

Building upon this simple idea, interaction instruments can be understood as visual structures that employ layering and stacking of *input areas*, each with the ability to register and implement their own gesture event propagation.

Figure 5.3 exemplifies how graphical structure might be used in the tool reification process: a rectangular area (a) registers a scale event handling routine that takes the scale factor contained in the event and sends it to a 2D object to be scaled uniformly along both its axes. The overlaid bars (b) and c) operate in the same way, however, event data captured by their handlers is interpreted only for one axis, adding axis constrained scaling capability.

5.1.3 Tool Space: A Precursor

At a late stage of my work on this thesis, I discovered a conceptual precursor to the *Tool Space*. In early work on touch input, Buxton et al. (1985) developed the idea to use a touch-sensitive tablet as a sensor and overlay it with a cardboard template. Cutouts subdivided the underlying surface into several autonomous input areas, each acting as a controller directly

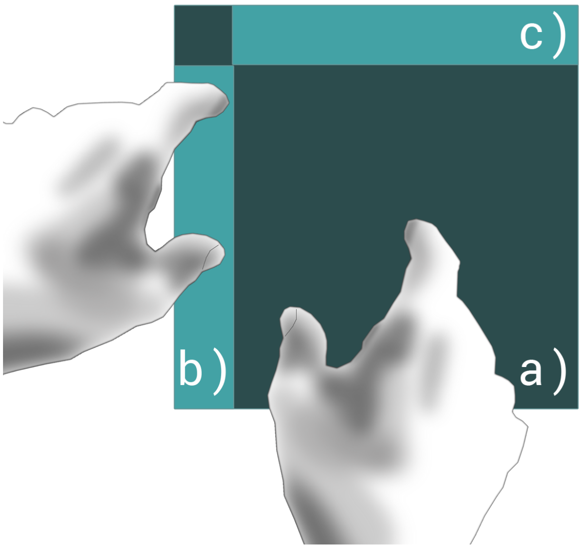
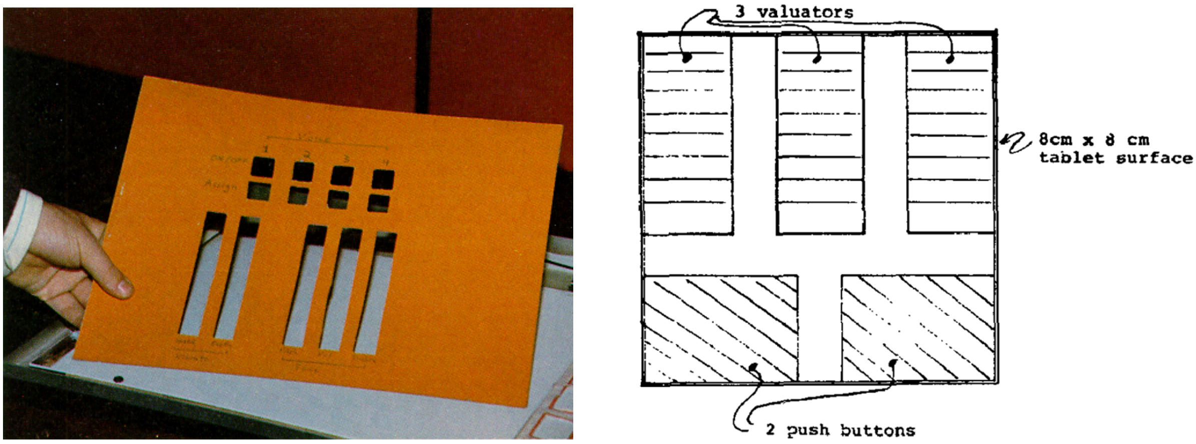


Figure 5.3: An exemplary tool for both axis-constrained and integral 2D object scaling: a) input area for integral scaling b) input area for scaling along the y-axis, and c) input area for scaling along the x-axis.

coupled to a specific function. The bevels resulting from the cutouts were meant to provide haptic guidance. By exchanging templates, each application could have its own specific set of controllers, which the authors referred to as "virtual operating console" (see figure 5.4(a)).

Buxton et al. (1985) further discovered that, "when the tablet surface is small, and the partitioning of the surfaces is not too complex", no template is needed at all and "users very quickly (typically in one or two minutes) learn the positions of the virtual devices relative to the edges of the tablet". With regard to the idea of virtual devices, the authors proposed to conceive the virtual devices as *windows*, which should be managed by the operating system (see figure 5.4(b)).

The idea of the *Tool Space* picks up this notion of virtual devices as input areas and spatial activation. It extends it with actual display capability and multi-touch gesture support. Buxton et al. (1985) already envisioned multi-touch operation of virtual devices, but in a sense of multi-tool operation, such as operating three virtual sliders at once. The *Tool Space* explicitly integrates the idea of using two-finger gestures and promotes novel tool compositions.



(a) (b)

Figure 5.4: *Tool Space* precursor (Buxton et al., 1985): (a) cardboard templates with cutouts divide one touch-sensitive surface into multiple tools, enabling spatial activation and task-specific input devices. (b) Later, the cardboard is replaced with virtual tools, which are conceived as invisible *windows*.

6
Tool Space Case Studies

To exemplify the concept of the *Tool Space* and discuss first user feedback, I present two case studies. The first case study has evolved from the exploration phase (see section 4), where I have started to look at touch interaction techniques for virtual 3D environments on the curved display. I identified interactive 3D environments as an interesting use case, because of its functional complexity and the challenges arising from the dimensional mismatch of 2D touch input and virtual 3D content. In addition, the angled touch pads, but also techniques like WIM and Hold-and-Point, already suggested a subsequent focus on designing tools as mediators.

After the exploration phase, it became clear that the focus of this work is on interaction paradigms for dual-surface workspaces, independent of whether there is a curved transition between the two surfaces. In order to avoid the shortcomings of working with a research prototype (Palleis et al., 2013) and in order to achieve more generally applicable results, further investigations were carried out on a setup with two orthogonal touch screens instead of the Curve system.

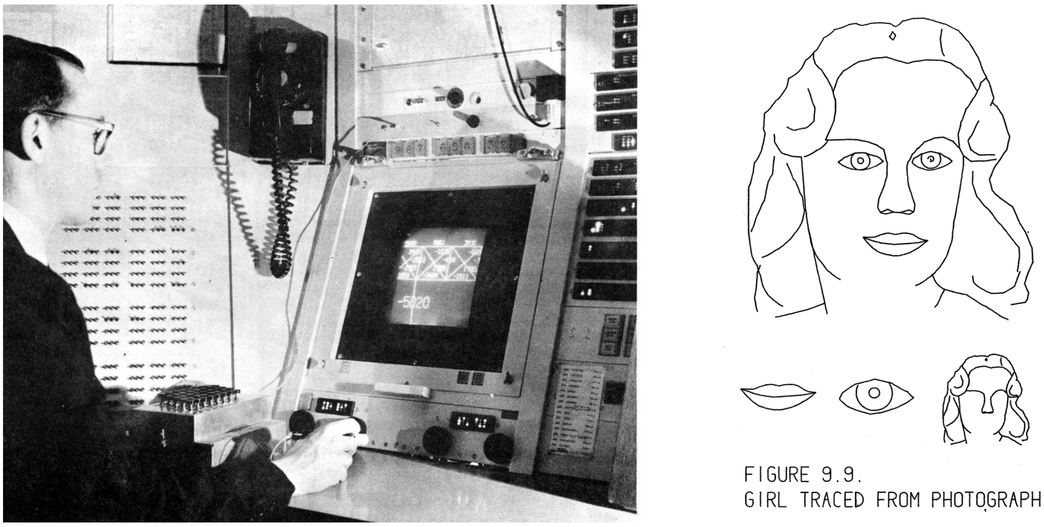


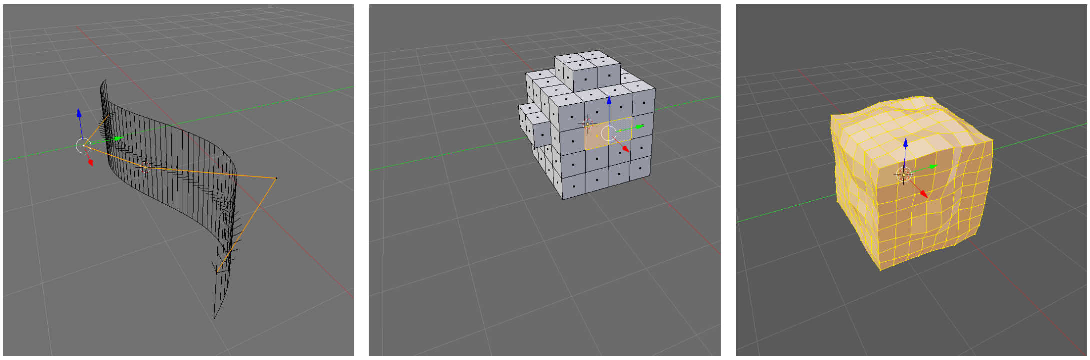
Figure 6.1: Sutherland's Sketchpad system, described in his doctoral thesis at MIT in 1962 (Sutherland, 1963a) with an original drawing.

6.1 A Toolspace for 3D Modeling

One of the prime examples of graphical user interfaces – Ivan Sutherland's *SketchPad* (Sutherland, 1963b) – illustrated the potential of *direct* 2D computer-aided design using an interactive screen operated with a light pen. According to Sutherland's original speculations, his interaction technique would "generalize nicely to three dimensional drawing", a project pursued simultaneously by Sutherland's colleague Johnson (Johnson, 1963). Since then, the computer-aided design of digital three-dimensional objects has become a core task of production workflows in diverse industries such as engineering, industrial design, architecture, movie making, game design, and many more. Existing software tools can roughly be distinguished between solid surface modeling software and CAD (computer-aided design) software: while the former is used predominantly by creative professionals to create imagery from solid 3D models, the latter enables engineers to model the shape of 3D objects along with its physical characteristics.

The focus of this case study is to apply the tool space concept to the object shaping process, i.e. the actual modeling. With regard to 3D modeling, different basic 3D construction and modeling paradigms have evolved and established, which are distinguished by the way in which 3D objects are represented. On the one hand, with *curve-based* modeling 6.2(a) shapes are represented by a set of control points that specify curvatures (e.g. Bézier-curves, NURBS). Accordingly, the shaping process predominantly involves the manipulation of control points. In contrast, with *polygon modeling* 6.2(b) objects are represented as meshes of polygons and the shaping process involves the manipulation of mesh components, i.e. vertices, edges and faces.

Apart from these basic paradigms, *digital sculpting* 6.2(c) has established as a second-tier paradigm, mostly used to generate very detailed and fine-grained 3D structures on top of prefabricated models. According applications offer virtual tools, such as chisels, that are conceived after physical examples and operate in analogy to brushes known from 2D image processing tools.



(a) Curve-based modeling (b) Polygon modeling (c) Digital sculpting

Figure 6.2: Different 3D modeling paradigms exemplified in Blender.

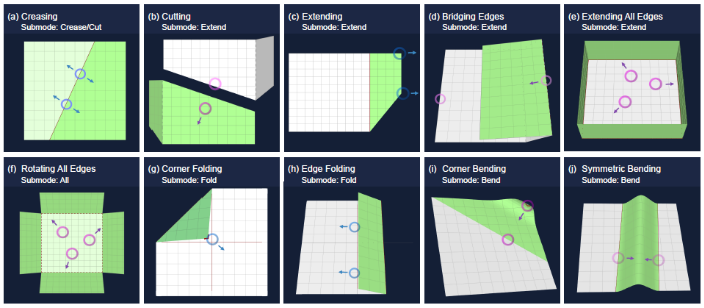
Typically, 3D modeling software built for WIMP systems (e.g., Blender) decomposes task complexity by introducing a variety of modes and virtual tools, often resulting in complex user interfaces and a sequential style of interaction that bears little resemblance to hand crafting in the real world and further presents a high entry barrier to novice users. Therefore, the emergence of novel modeling paradigms can be both technique-driven or rooted in user-centered research approaches.

For instance, curve-based modeling was developed in the context of automotive engineering, but operating on control point structures to achieve shapes exhibits a rather large gap to conceptual approaches, such as drawing sketches, employed by creative professionals or laymen (Schkolne et al., 2001). For such reasons, making modeling more accessible has led to novel paradigms, such as automatic conversion of hand-drawn sketches to 3D model representations (Igarashi et al., 2007; Zeleznik et al., 2007). With his surface drawing paradigm, Schkolne et al. (2001) adapted curved-based modeling to virtual reality, trying to make it more accessible to creative usage by using hand-held controls to draw 3D shapes in mid-air (see figure 6.3(a)).

Wang and Kaufman (1995) and Galyean and Hughes (1991) pioneered digital sculpting by exploring ways to imitate manual sculpting operations. More recently, Paper3D (Paczkowski et al., 2014) – a tablet-based app – employed a paper craft analogy based on folding and cutting to create 3D models using touch input (see figure 6.3(b)).

Further, advancements in input sensing technology and interaction paradigms have produced a variety of ideas on how 3D object manipulation may be facilitated by employing novel input modalities, such as multi-touch (Walther-Franks et al., 2011), pen sketching (Bae et al.,

(a) Surface drawing



(b) Paper 3D

Figure 6.3: Novel 3D modeling paradigms: (a) using a head-mounted display and mid-air gestures to paint in 3D space (Schkolne et al., 2001), (b) a paper-folding modeling paradigm for tablets (Paczkowski et al., 2014).

2008), force-feedback devices (Foskey et al., 2002) or free-hand gestures (Grossman et al., 2001).

Similarly, advances in output technology have introduced novel questions, in particular concerning the interaction with stereoscopic displays (Benko and Feiner, 2007; Strothoff et al., 2011) or virtual reality environments (Jayaram et al., 1997). Interestingly, advances in personal fabrication technology may generate reciprocal effects, bringing digital modeling techniques to physical material.

Instead of adapting or evolving a paradigm to certain requirements (e.g. specific user group, use case or modality), I wanted to employ an existing approach to explore how the idea of the *Tool Space* and its underlying *Instrumental Interaction Paradigm* (Beaudouin-Lafon, 1998) relates to an inherently complicated application area. In particular, I did not intend to contribute to the field of 3D modeling, but to the understanding of how indirect touch tools may extend established input vocabulary of existing applications. My choice for polygon modeling was a pragmatic one, as the direct manipulation of polygon meshes does not involve the further abstraction of shaping objects indirectly through curve specification.

In polygonal modeling, users often start from geometric primitives, e.g. boxes or cylinders, and operate on these primitives to shape out more detail – a process sometimes called *box modeling* (Russo, 2005). Common interaction instruments in this process are mesh subdivision, extrusion and affine transformations. For instance, to model a single-legged table, users might start from a cylinder, pull out a thin pole from the center (i.e. extrusion) and scale the top edge-loops of the pole to form a table surface (i.e. edge-loop scaling).

In a reification process, I developed novel interaction instruments for *edge-loop scaling* and *extrusion* functions. To have full control over the design of novel interaction instruments, I implemented my own 3D modeling software, developed in Java. Apart from the conceptual character of the work, this process has also led to technical innovations regarding polygon extrusion (Palleis et al., 2015b). While both techniques evolved within the context of dual-display workspaces – qualitative results show that users like this setup for its partitioning character – my techniques are primarily designed to explore the potential of the *Tool Space* to support complex tasks. As such, an adaption of the ideas presented in this section to slightly different settings, for instance a large tilted display, is well worth considering.

6.1.1 Touch-input for 3D-Modeling

Like other interactive 3D environments, modeling applications support a basic set of tasks, which include object *selection and manipulation*, *navigation* and *system control* tasks. In general, object manipulation tasks refer to affine transformations of solid objects, such as rotating or scaling a cube. Similarly, navigation tasks concern transformations of the virtual camera, such as providing distinct perspectives on objects. System control tasks refer to actions that are not situated in virtual 3D environments itself, such as changing application states or interaction modes. There is a large body of relevant work on interaction techniques for all three task classes, which cannot be detailed in this thesis. However, Jankowski and Hachet (2013) provide an extensive and up-to-date survey.

Despite precision and occlusion problems, direct multi-touch gesture input is frequently used for 2D rotation, scale and translation (RST) object manipulation tasks (Rekimoto, 2002) and view manipulation tasks (Hancock et al., 2006), such as pan-and-zoom navigation. Several techniques in the literature generalize 2D techniques to 3D environments, exploring both 3D object manipulation (Au et al., 2012; Hancock et al., 2007; Martinet et al., 2010a; Reisman et al., 2009), and 3D navigation tasks (Hachet et al., 2009; Klein et al., 2012).

Essentially, all 3D touch interaction techniques face the problem of a low *degree of integration* (Beaudouin-Lafon, 1998): the degrees of freedom (DoFs) necessary for three-dimensional object manipulation are not naturally provided by two-dimensional interactive surfaces. There are at least two solutions to this problem: first, the input-bandwidth may be increased through bimanual and gestural interaction to enable simultaneous control of multiple parameters, and second, the control may be constrained to meet the requirements of a given input bandwidth.

Increased input bandwidth This approach involves the design of specific input vocabularies (see figure 6.4 for an example), often employing multi-finger and bimanual input in order to increase input bandwidth (Liu et al., 2012b; Paczkowski et al., 2014; Reisman et al., 2009). For instance, in the *three finger rotation* technique (Hancock et al., 2007), two fingers of one hand define an axis and the second hand's index finger rotates the object around it.

Constrained dimensions The second approach often exploits the type of data and the context of the application to constrain the manipulation of an object in a meaningful way: for instance, using the *z-technique* (Martinet et al., 2010a), users can first directly select an object with one finger and then use another finger to translate the selected object in a plane parallel to the viewing plane, effectively constraining the manipulation to two dimensions. Reisman et al. (2009) computationally establish a screen-space formulation for 2D and 3D direct manipulation that enables axis-constrained 3D manipulation (see figure 6.5) by solv-

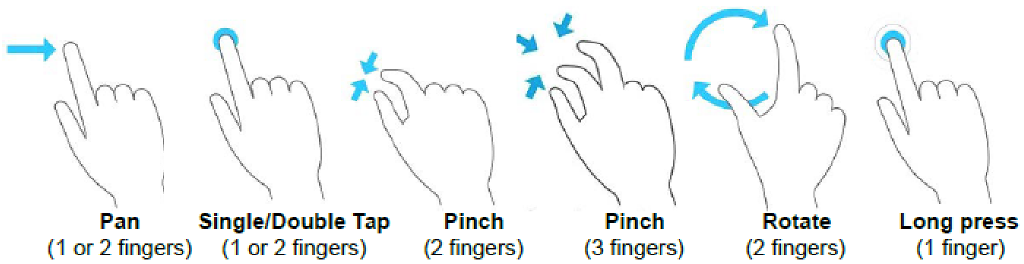


Figure 6.4: Multi-touch gesture set employed in Paper3D (Paczkowski et al., 2014).

ing constraints given by users' fingers. A further example is the adaption of established GUI transformation widgets to touch input (e.g., (Cohé et al., 2011)).

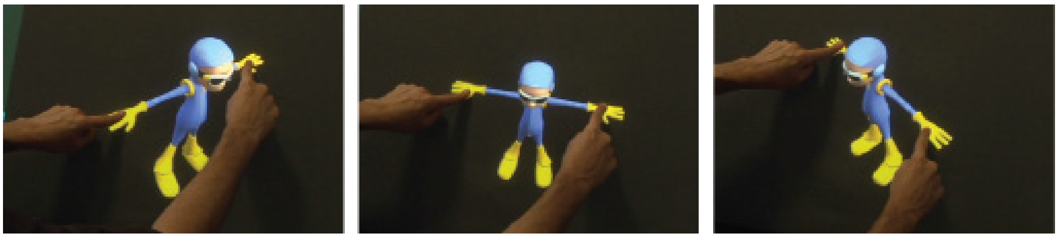


Figure 6.5: Axis-constrained rotation in screen space (Reisman et al., 2009).

Integrality vs. separability A fundamental question concerns whether integrality is actually beneficial to 3D object manipulation. While Jacob et al. (1994) suggests that interaction techniques should generally adhere to perceptive task structure, Martinet et al. (2010b) has shown in the context of 3D object manipulation that a separation of control dimensions is preferred by users and outperforms integral control of parameters, although their manipulation is perceived integrally in the real world.

Indirect touch for 3D interaction Knoedel and Hachet (2011) studied the impact of directness on user performance in multi-touch RST tasks. Their findings indicate that especially for 3D interaction, indirectness improves precision and efficiency, with the cost of being slower than direct input. Notions of indirectness are also present in other works, for instance with the *z-technique* (Martinet et al., 2010a) only the object selection is direct, while the translation can occur anywhere on the screen. Indirect touch is further used with stereoscopic rendering. In this context, (Simeone and Gellersen, 2015) compared direct and indirect touch input techniques and found that indirect input results in less errors due to reduced occlusion. (Giesler et al., 2014) showed precision benefits for indirect touch input based on shadows cast by virtual objects onto a touch-screen surface compared to in-air interaction technique.

6.1.2 Design Rationale

With my techniques, I predominantly address the sequential workflow exhibited by many of the 3D object-shape manipulation techniques in today's desktop environments. The techniques are designed with the three properties of interaction instruments in mind (Beaudouin-Lafon, 1998): degree of indirection, integration, and compatibility.

Degree of indirection An optimal, low degree of indirection could be achieved by, e.g. performing the modeling tasks directly on the scene object. However, due to fatigue effects, touch imprecision and occlusion issues (Albinsson and Zhai, 2003), I decided to use indirect touch interaction, which has been shown to be more precise, efficient and comfortable. Touch input is therefore not directed to objects of interest, but to specialized tools represented by dedicated input areas and additional visual representations of the scene object in a separate tool space.

Degree of integration refers to the ratio between the number of DOFs that users can control simultaneously in the application and the number of DOFs provided by an input device. For instance, specifying a 3D object translation using a one-finger touch gesture inherently exhibits a degree of integration below one due to the mismatch between 2D surface input signals and 3D position values. With the *Tool Space*, the ratio can be raised through the design of multi-digit or bimanual interaction techniques based on spatially multiplexed tool palettes.

Degree of compatibility is arguably the most elusive property: it describes a measure for the similarity between the physical actions of users performing the interaction technique and the visual feedback of the virtual object. Paczkowski et al. (2014) have achieved a high degree of compatibility by shifting the manipulation into the two-dimensional domain of paper-folding. With *TouchTools* 6.6, Harrison et al. (2014) have taken the idea of designing for compatibility to the extremes, arguing that many of the existing multi-finger gesture and chording approaches were too simplistic and not taking into account the familiarity and fluency with physical tools.

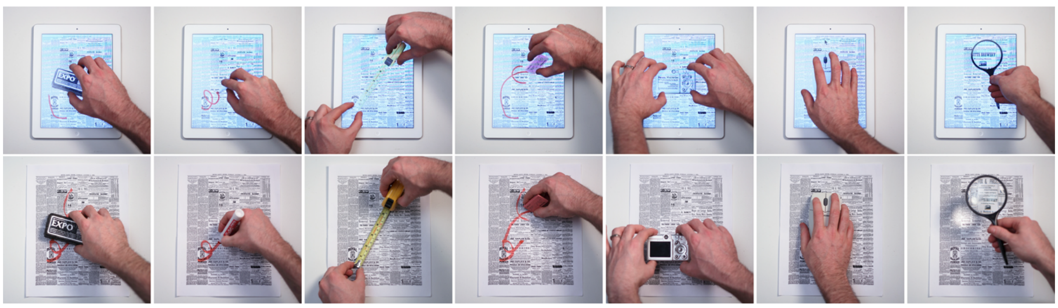


Figure 6.6: TouchTools: gesture design based on familiarity with physical tools (Harrison et al., 2014).

With the *Tool Space* for polygon modeling, I tried to approach compatibility by restricting the gesture vocabulary to established - yet arguably simplistic – one- and two-finger gestures, but applying them in a meaningful way. For instance, visually squeezing a cylinder may be performed with various gestures, but a two-finger pinch gesture may be one that exhibits a high degree of compatibility.

6.1.3 Prototype Setup

The prototype consists of two conventional touch screens, arranged as a pair of connected horizontal and vertical surfaces (see figure 6.7(a)). The form factor of this setup is intended to better emphasize the integrative character of my approach, exploring extensions to existing interaction techniques rather than inventing completely novel setups. In particular, the horizontal touch display is conceived as an extension to keyboard and mouse input, as proposed also in the Magic Desk project (Bi et al., 2011).

The horizontal screen constantly displays a toolbar and the available geometric primitives (see figure 6.7(b)) and otherwise leaves space required for the virtual tools introduced below. Geometric primitives can be added to the scene using a swipe-up gesture, adhering to the conceptual model of both displays forming a unified interaction surface similar to curved displays (Hennecke et al., 2012).

The prototype is developed based on JavaFX 8 and has been developed on a HP desktop PC with a 2.8 GHz Intel Core i7 CPU, an ATI Radeon HD5670 graphics card and two Dell S2340T touch displays with a resolution of 1920 x 1080 pixels each. Although some established 3D modeling applications offer programming or scripting interfaces, I decided to develop the functionality myself, because of the greater freedom to map input signals to commands. I have looked into the programming interfaces of Blender and Google SketchUp, but while they allow to build custom functionality or to integrate novel input signal channels, they do not allow to change the fundamental logic of the application.

In contrast to many modeling software packages, the meshes I used are based on triangles. This is due to mesh representation in JavaFX, which currently only supports triangles. The geometric primitives I used within the applications were created in Blender, exported as Collada-files and imported into JavaFX using an existing model loading library[1].

Basics: Scene Navigation and Object Positioning

The prototype enables *panning*, *zooming*, and *orbiting* scene navigation: two-finger dragging is mapped to panning operations, pinch and zoom gestures to zooming and one-finger dragging orbits the virtual scene camera around the selected object. These gestures can be performed anywhere on the *base layer of the Tool Space* and within the 3D scene.

Figure 6.8 shows an interaction sequence for basic object instantiation and manipulation in 3D – translating, rotating and scaling objects. Users add an object to the scene by performing a swipe-up gesture on the according icon or select an already added object on the

[1] `http://www.interactivemesh.org/models/jfx3dimporter.html`

Figure 6.7: The basic setup consists of the visual display of the scene (vertical screen) and the tool space (horizontal screen).

vertical screen by tapping on it, invoke the object manipulation widget, and transform the object using a pair of interactive orthographic views (front view, top view) and a dedicated rotation/scaling widget.

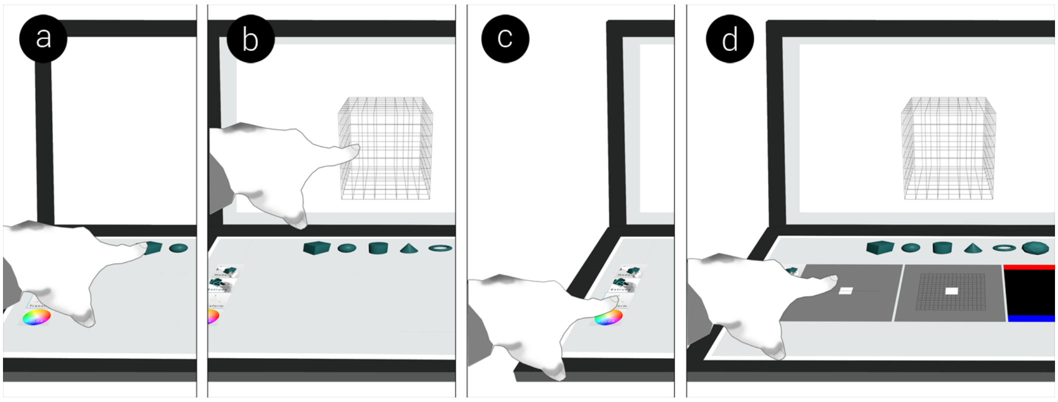


Figure 6.8: The basic workflow: (a) adding a mesh to a scene by swiping it up within tool space, (b) selecting an object with direct touch, (c) selecting a tool from the toolbar, and (d) using touch widgets to control the object indirectly.

6.1.4 Tool Reification for Polygon Modeling

One well-established approach to build polygonal models is box modeling Russo (2005). Here, a modeler starts from geometric primitives such as boxes or cylinders and refines them using different tools. Common tools include *mesh subdivision* to increase the number of polygons in order to allow for more detailed changes, *affine transformations* (translation, rotation, and scaling) of selected sub-meshes or *extrusion*, i.e. the process of adding new surface meshes (edges and vertices). Beyond these basic tools, existing polygon modeling software (e.g., Blender [2], Maya [3], 3D Studio Max [4] etc.) offers a variety of specific tools, such as constructive solid geometry, curve-based or sculpting elements, or procedural modeling.

Performing edge-loop scaling and extrusion in conventional 3D authoring tools requires users to frequently switch modes resulting in a sequential workflow of alternating selection and scaling commands: a single mesh transformation operation involves the switch into a mode allowing editing parts of the mesh (as opposed to transforming the whole object), and a subsequent transformation requires the user to (a) decide if she is applying the transformation on vertices, edges or faces, (b) choose a transformation to apply and (c) determine axes of the object's coordinate system to apply the chosen transformation. These choices are either made by selecting menu items, applying keyboard shortcuts or operating graphical handles.

Using the *Tool Space*, my intention was to explore how dynamic graphics and spatial multiplexing of touch input areas (i.e. virtual tools) may incorporate some of these complexities and enable novel access to these operations.

Tool Reification A: Edge-loop Scaling

One approach for transforming sub-meshes is *edge-loop scaling*: in order to create an hour-glass from a cylinder primitive, the designer selects closed *edge-loops* in the center area of the cylinder – an edge loop is a sequence of connected edges that reaches around the surface of an object. By scaling down the size of the selected loops, the resulting model becomes an hour-glass (see an example in Blender in figure 6.9). This task involves both a selection and a scaling task.

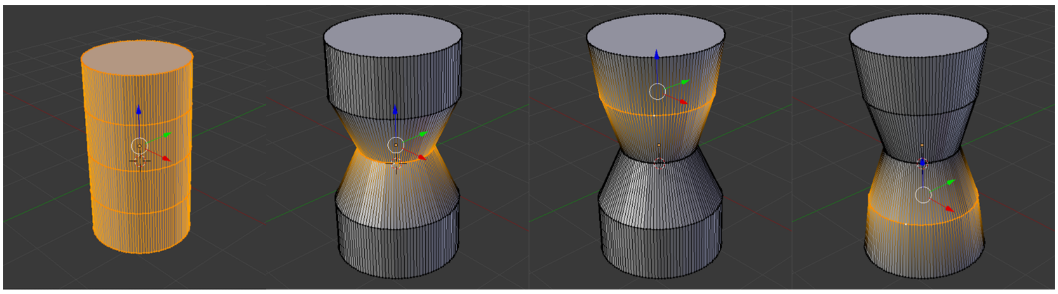


Figure 6.9: Edge-loop scaling in Blender.

I designed a bimanual indirect touch interaction technique for scaling edge loops with the goal to increase the degrees of both integration and compatibility (Beaudouin-Lafon, 1998). The bimanual workflow adheres to Guiard's (Guiard, 1987) observations on asymmetric

[2] `https://www.blender.org/`

[3] `http://www.autodesk.de/products/maya/overview`

[4] `http://www.autodesk.de/products/3ds-max/overview`

bimanual actions in the physical world: the non-dominant hand sets the frame-of-reference, starts the interaction sequence and performs coarse actions. In particular, the non-dominant hand performs the selection task and the dominant hand scales the selected edge-loops (refer to section 7.1.1 for related work on two-handed input). To increase compatibility, both hands employ one- and two-finger pinch gestures to shape objects.

Figure 6.10 outlines the workflow: touching the thin bar with the non-dominant hand triggers the appearance of a mesh-selection volume which can be translated along the y-axis (vertical bar) or x-axis (horizontal bar) with one-finger and can be resized using a two-finger pinch gesture. Performing a pinch gesture with the dominant hand in a dedicated rectangular widget controls the scaling of edge-loops previously or simultaneously selected with the non-dominant hand.

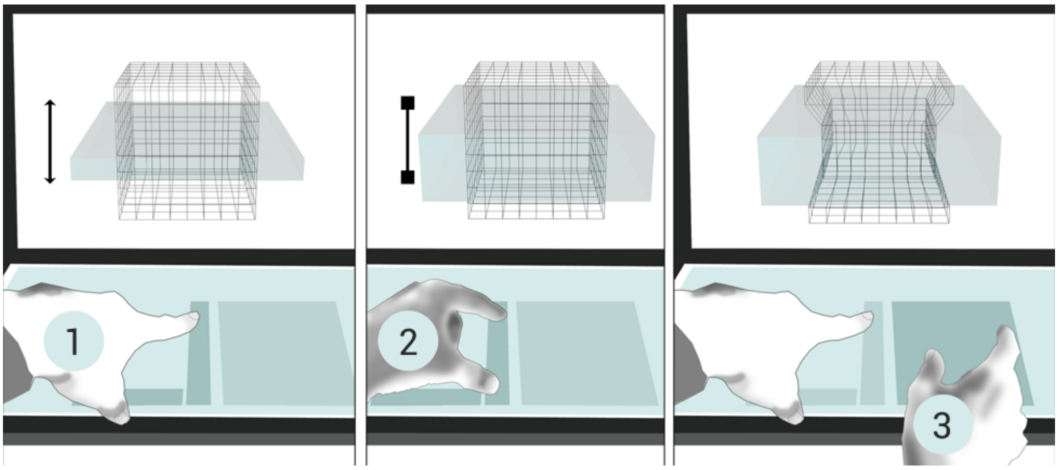


Figure 6.10: The bimanual edge loop scaling workflow: (1) the non-dominant hand translates and (2) scales the selection volume, the dominant hand scales the contained edge loops (3).

It is important to note that edge-loop scaling operations assume the existence of edge-loops. Normally, edge-loops are inserted with mesh subdivision commands. However, as the subdivision itself was not the focus of this tool reification case study, I prepared meshes that where already subdivided and therefore had a predefined amount of edge-loops.

Both, the dragging and pinch gestures used to translate and resize the selection volume are mapped relatively to the last state of the selection volume, so that translating and resizing can be interrupted and resumed without causing the selection volume to jump. Further, position and size of the selection volume are constrained by the selected object's dimensions, allowing quick movements towards the respective ends. In order to minimize the need for clutching and yet enable precise control, the mapping between finger movements and the transformation of the selection volume is based on a discrete gain change function:

$$\triangle trans = \begin{cases} \triangle pos \div bar\text{-}length \cdot size & \text{if } \triangle p \leq 10 \\ \triangle pos \cdot 2 \div bar\text{-}length \cdot size & \text{if } 10 < \triangle p \leq 20 \\ \triangle pos \cdot 3 \div bar\text{-}length \cdot size & \text{if } \triangle p > 20 \end{cases}$$

$\triangle trans$ describes the translation delta of the selection volume along the respective axis of the 3D object over a time interval of 15 ms relative to the respective *size* of the 3D objects. $\triangle p$ describes the relative pixel distance traveled within the length p of the control bar (here $500px$).

The angles of the dominant hand's pinch gestures are not taken into account, in order to enable a comfortable hand positioning. Figure 6.11 shows screen-shots from a short sequence of modeling with the edge loop scale tool.

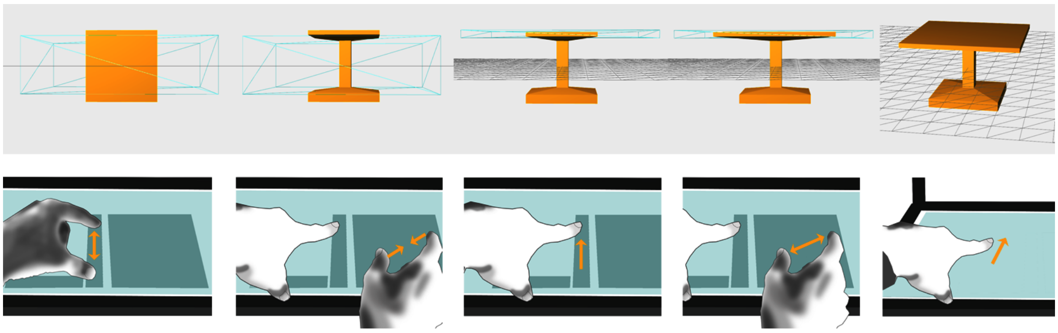


Figure 6.11: From left to right: first, the selection volume is positioned and scaled with the non-dominant hand, then the containing edge loops are scaled down. Then the view is slightly changed and the selection volume is repositioned to the top and scaled down. Subsequently, the newly selected edge loops are scaled up. A further change of the camera position shows the resulting 3D model from another viewpoint.

Tool Reification B: Extrusion

Extrusion adds new polygons to a mesh and complex shapes can be created by iteratively extruding and transforming sets of polygons. It requires two basic steps: first, the selection of a set of polygons to extrude, and second, the directed extrusion itself [5] (see figure 6.12 for a Blender example).

When used iteratively to create shapes that follow a three-dimensional path, a constant shift between navigation and extrusion is necessary. Further, in WIMP-based user interfaces, this requires the explicit de-selection of polygons and the selection of novel polygons, often with an intermediary change of the camera viewpoint. Again, my goal was to explore the potential of spatial tool multiplexing to support an existing complex workflow with an increased *degree of integration*.

Figure 6.13 gives an overview of the interface elements of the extrusion tool.

[5] Usually, the direction is constrained to the normal vector of the selected polygons and altered in a second step by transforming the extruded geometry.

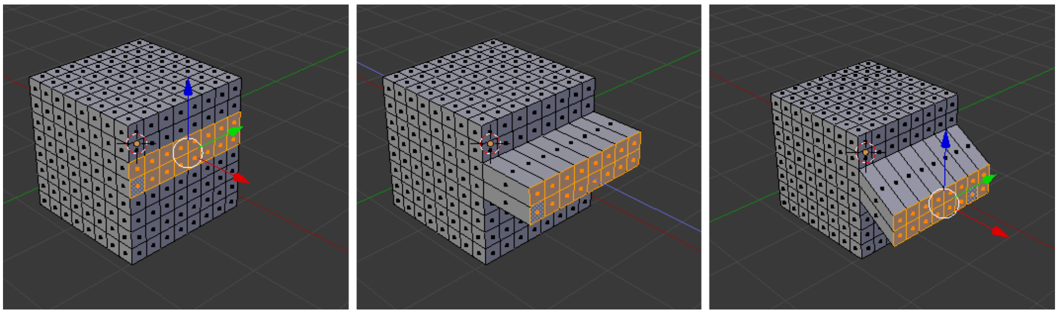


Figure 6.12: Extrusion in Blender: first, polygons are selected, then extruded along their normal vector and finally translated.

Figure 6.13: Extrusion tool concept: (a) the scene object, (b) the mesh selection tool, (c) the virtual touch pad for extrusion, and (d) the base layer for scene navigation.

Selecting Surfaces and Polygons Users select a surface of the scene object on the vertical screen via direct touch (figure 6.13 a). Then, the *polygon selection tool* (figure 6.13 b) displays an animated virtual camera motion towards the selected surface. Its camera moves toward a viewpoint position that allows the user to see the entire surface represented within the polygon selection tool area, with the viewing plane parallel to the selected polygons' surface plane (similar to the work of Hachet et al. (2009)). The distance between virtual camera and 3D object can be further adjusted using the pinch gesture within the selection tool.

The animated camera motion is intended to support the user's spatial awareness, as the view point of the main scene camera is not changed. I decided to add the additional polygon selection tool in the horizontal tool space to reduce arm fatigue when making precise sub-mesh selections. Moreover, the two separated views of the 3D object shall help to keep an overview of the scene while selecting polygons.

Within the polygon selection tool, users can select and deselect single polygons of the selected object by tapping and also – similar to finger painting – by dragging. The selected polygons are highlighted both in the polygon selection tool and in the scene. Figure 6.14 shows a schematic overview of these steps.

Technically, the polygon selection is based on a ray casting algorithm. For every touch input signal, a ray is sent from the virtual camera of the polygon selection tool onto the selected surface. A swipe-up gesture performed on the polygon selection tool de-selects all polygons.

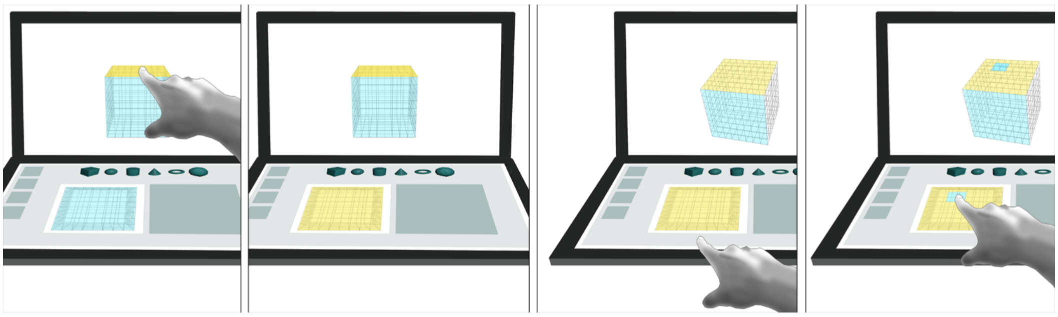


Figure 6.14: The workflow for selecting surfaces and polygons for extrusion.

Perspective-dependent Extrusion Once a set of polygons is selected, it can be extruded using dragging gestures on the extrusion touch pad (figure 6.13 c).

The extrusion touch pad consists of a square area containing a horizontal and a vertical bar, as well as a free-form manipulation area (see figure 6.15). Gestures within the horizontal and vertical bar extrude the selection constrained the respective axis, whereas gestures on the pad's free-form area additionally apply a translation using both deltas from the two-dimensional translation vectors provided by the touch input signals (see figure 6.16 on the right).

The interpretation of the two-dimensional dragging vectors – i.e. their mapping to the 3D scene – depends on the current orientation of the scene. Changing the scene's camera viewpoint during extrusion operations will cause updated automatic polygon selections, reassigning polygons continuously to the possible extrusion directions *up*, *down*, *left*, and *right*. In particular, selecting a set of polygons and then rotating the camera will make the selection flip as soon as a certain threshold angle is reached.

For instance, figure 6.15 shows the interpretation of the rightward gesture in a side-view: differently colored polygons mark alternative polygon selections on the up- and down-side of the extruded end, indicating how a subsequent up- or down-ward gesture would be interpreted in the current camera orientation. The visual indication of directly available alternative extrusion operations is continuously updated with the manipulation of the camera viewpoint.

The concept assumes a two-handed operation during which the non-dominant hand controls the camera perspective, which in turn – based on an initial selection – continuously determines a set of directly available polygon sets that can be extruded using the dominant hand.

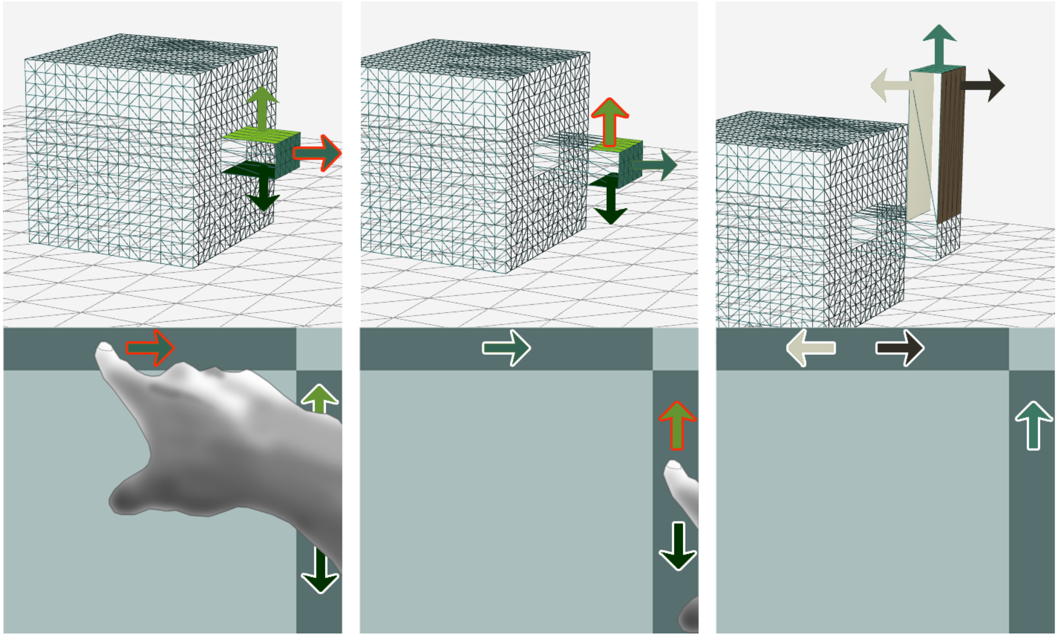


Figure 6.15: Extrusion workflow: from a side view, the already extruded polygons are extruded again to the right, indicated by a dragging gesture on the horizontal bar to the right (left). Then, an upward movement in the vertical bar indicates that the newly created polygons facing up should be extruded (middle). The extrusion could then be continued with the polygons currently facing left, right or up (right).

In theory, this workflow should reduce activation costs (Beaudouin-Lafon, 1998), which seems especially useful for repetitive task shifting.

Implementation of the Extrusion Upon touching the extrusion touch pad, a sequence of routines is started to infer the specific command intended by the user.

onTouchStart When a touch gesture is initiated on the extrusion touch pad, the 2D touch coordinates are used to determine whether the touch occurred on one of the constraining bars or not. Further, a time stamp is set.

onTouchMoved Algorithm 3 illustrates the procedure that runs on every touch movement update.

During the first 100ms of a touch gesture, the algorithm decides which polygons to extrude. This is necessary to distinguish between up and down, as well as left and right. Therefore, the variable *timeDiff* saves the difference between the current system time and the time stamp set at the *onTouchStart* event handler. As long as it is smaller or equal to 100 ms, the touch movement deltas are summed up in the variables *sDeltaX* and *sDeltaY*. After the first 100 ms, these sums of the two deltas are used in combination with the current scene camera's orientation to determine which polygons to extrude. Then, the insertion of the new vertexes is triggered once with the *extrude()*-function.

Algorithm 3 The Pseudocode for the *OnTouchMoved* event handler of the extrusion pad

if $timeDiff \leq 100ms$ **then**
  $sDeltaX+ = deltaX$
  $sDeltaY+ = deltaY$
else
  if $!extruded$ **then**
    $determineSelectedFaces(sDeltaX, sDeltaY)$
    $extrude(selectedFaces)$
    $extruded = true$
  else
    $translateExtrudedFaces(deltaX, deltaY)$
  end if
end if

After the insertion, the deltas of the ongoing touch movement are used to translate the new polygon. It can be constrained to the surface normal (the polygon's local z-axis) by starting the dragging movement in one of the constraining bars of the extrusion pad. In this case, a vector that is the result of a multiplication of the surface normal and a factor depending on the touch movement delta is added to each vertex. Whether the horizontal or the vertical touch movement delta is used also depends on the viewing orientation of the selected polygons. If the dragging is not initiated within one of the constraining bars, then the new vertexes are additionally translated by either the polygon's local x- or y-axis, also depending on the current viewing orientation of the polygon.

onTouchFinished After each extrusion movement (as well as after each change of the camera orientation), all newly created polygons are assigned to one of the following *future* directions: left, right, up or down. This is done by constructing local coordinate axes of the camera and calculating angles between them and the polygon's surface normals.

The camera's local z-axis is calculated as the vector from its current position to the center point of the selected 3D object (as the camera orbits around the object, it always looks at this point). Its local X-axis can be calculated as the cross product of an up-vector (0,1,0) and its z-axis, and its y-axis can be calculated as the cross product of its z-axis and its x-axis.

Subsequently, the angles between the camera's x- and y-axis and the polygon's surface normals are calculated and used to determine whether, from the camera's perspective, the polygons face to the right or left, up or down (see algorithm 4).

Here, *angleX* is the angle between the camera's x-axis and the polygon's surface normal, and *angleY* the angle between the camera's y-axis and the polygon's surface normal.

Inititial Extrusion The initial extrusion of polygons selected in the polygon selection tool is a special case, as there are no automatic alternative polygon selections facing in other directions that can be extruded yet. In this case, the touch input of the extrusion pad is used

Algorithm 4 The Pseudocode illustrating how polygons are assigned to the four directions left, right, up, and down.

if $angleX > 315 \,||\, angleX < 45$ **then**
  $left.add(polygon)$
else if $angleX > 135 \,\&\, angleX < 225$ **then**
  $right.add(polygon)$
end if
if $angleY > 135 \,\&\, angleY < 180$ **then**
  $up.add(polygon)$
else if $angleY > 0 \,\&\, angleY < 45$ **then**
  $down.add(polygon)$
end if

to insert the new geometry into the mesh and to translate the new polygons in the manner described above (see figure 6.16).

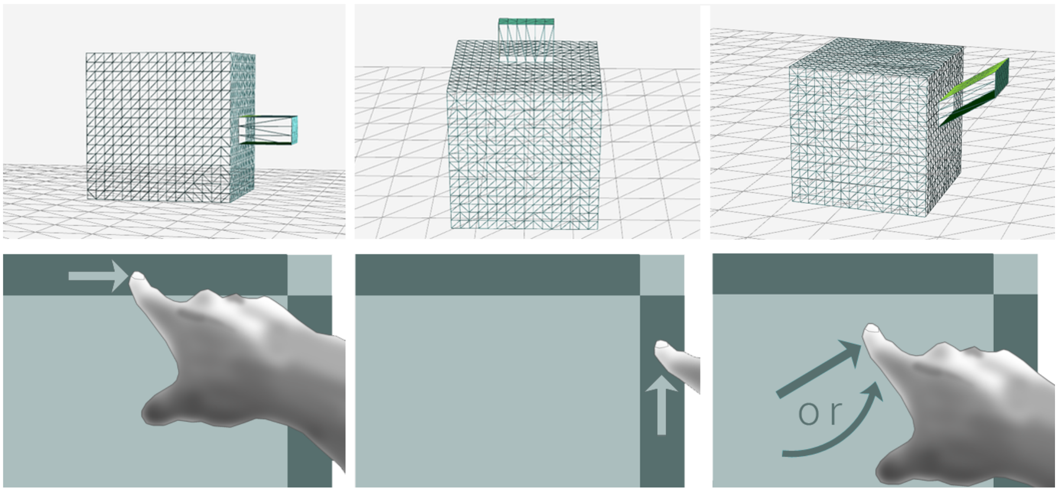


Figure 6.16: Initial extrusion of selected polygons.

Bent Extrusion with Scene Navigation

The sequential style of alternating extrusion and transformation commands make it a non-trivial task to create bent shapes, as many extrusions and rotations are involved. Therefore, this task is often approached with special commands that allow to extrude along a previously specified curved line, which usually requires the user to explicitly set the number of divisions (extrusions): the higher the amount of divisions, the closer the curved line is approximated by the extruded geometry.

While this approach is integrated in many software packages, the simplification it offers goes along with the necessity of learning a new tool that partitions the modeling process in a sequence of steps and requires abstract knowledge, as the result will only be visible after creating a line and applying the respective extrusion command.

My idea was to explore a more integral way of extruding bent shapes that does not require a special tool and is more explorative, in particular through real-time feedback. Therefore, I built upon the previously described idea of two-handed perspective-dependent extrusion and developed a workflow which allows to create bends and twists by rotating the camera with the non-dominant hand while simultaneously extruding polygons with the dominant hand. Figure 6.17 illustrates the interaction technique: while the right hand extrudes to the left, the left hand performs a camera rotation around the object's y-axis. During the ongoing camera movement, the right hand continues the extrusion movement.

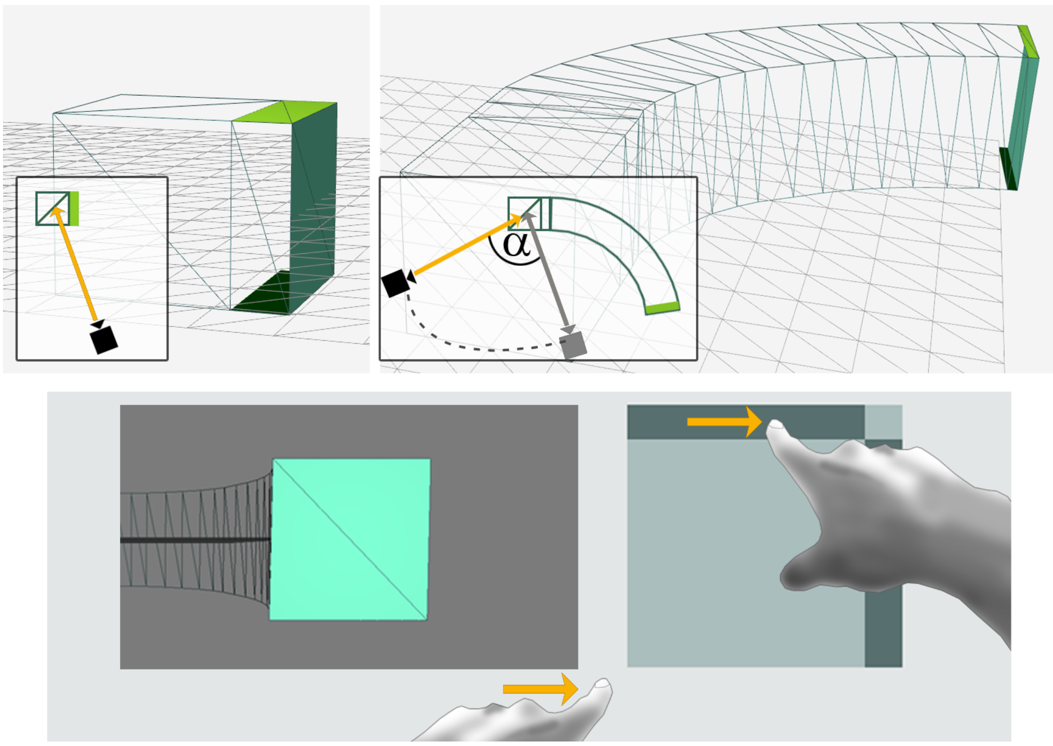


Figure 6.17: Creating bent shapes with simultaneous extrusion and camera control. α is the angle of camera rotation and corresponds to the sum of rotation angles of the automatically extruded polygons.

Implementation While the camera is moved during an extrusion touch movement, the polygons currently being translated are rotated and extruded automatically in small regular intervals. This is achieved by measuring the distance between the polygons currently being translated and the polygons that were extruded before, and by repeating the extrusion after a certain distance threshold has been reached. The rotation of the automatically extruded faces is dependent on two angle deltas: the camera's rotation around the object's x- and y-axis since the last extrusion (see algorithm 5). This way, after a bent extrusion has been finished, the sum of the camera's angle deltas corresponds to the sum of the newly created polygons' rotation angles.

Algorithm 5 Continued event handling code from algorithm 3 that illustrated how the camera angles are used to rotate the extruded polygons.

if $isCameraRotated$ **then**
  if $currentExtrusionDistance > 0.5$ **then**
    $angleDX = angleCamX - angleExtrX$
    $angleDY = angleCamY - angleExtrY$
    $angleExtrX = angleCamX$
    $angleExtrY = angleCamY$
    $rotateFaces(selectedFaces, angleDX, angleDY)$
    $extrude(selectedFaces)$
    $currentExtrusionDistance = 0$
  end if
end if

6.1.5 Initial User Feedback

In order to gather first user feedback on the tools, I conducted a test with six participants who had prior polygon modeling experience. During the test, they were guided through a series of small modeling tasks using both my system and their preferred commercial system, whereas the order of system was counter-balanced. Qualitative data was collected through questionnaires and conducted a semi-structured interview at the end of the session. The sessions lasted about one hour and were recorded on video.

The participants (two female, aged between 23 and 59, all right-handed) had various backgrounds: three were computer science students who had both private and educational experience with 3D modeling in Blender, one was an art student who teaches 3D modeling (mostly 3D Studio Max, but also Blender) to fellow students, one was a professional graphic designer with experience in Blender and one was a high school teacher with expert knowledge of and teaching experience with Google SketchUp [6]. Three of them worked on ten or more models during the last year, whereas three stated to have worked on less than five models.

Procedure Participants were asked to perform two modeling tasks with both systems, and subsequently two tasks only with the *Tool Space*. The *Tool Space* system and all relevant tools were demonstrated to the participants beforehand. Time was not constrained and participants were asked to model until they felt satisfied with their results. After the first two tasks, they rated *ease-of-use*, *perceived performance time* and *precision* using five-point Likert-scale items.

[6] http://www.sketchup.com

Edge-loop scaling Here, the task was to model a simple rectangular table with a single centered leg without further specification.

Extrusion This task asked to copy the shape shown in figure 6.18 which was displayed on paper during the task. The commercial modeling software of their choice was adapted to their preferences first (user interface settings) and available keyboard shortcuts were displayed during the task on paper.

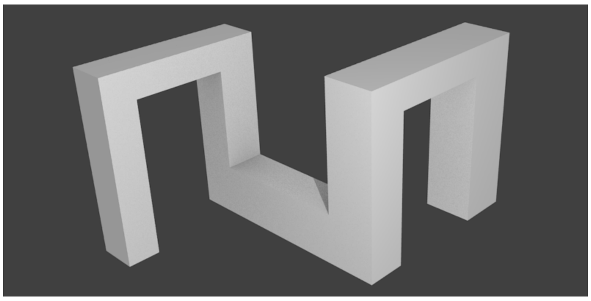


Figure 6.18: The shape that participants were asked to rebuild during the extrusion task.

Combined task Participants were asked to model a scene containing a table, a wine glass and optionally a chair without further specifications, this time only using the *Tool Space*.

Bent extrusion Finally, they were asked to perform a bent extrusion starting from a simple cube.

Results Figure 6.19 shows an overview of participants'subjective ratings: in the *edge-loop scaling task*, users experienced the *Tool Space* as easy and fast compared to the desktop tool. Three participants stated that modeling using the bimanual edge-loop scaling tool almost felt like working with "modeling clay" or "pottery", indicating a high degree of compatibility.

Regarding precision, users rated the *Tool Space* as less precise compared to their preferred tool. However, the more experienced modelers reported to be surprised about their own capabilities in creating shapes with it (see figure 6.22 for an overview of models created in the third task).

This relates to the statement of another user regarding a positive *sense of achievement*: shapes created with the edge-loop scaling tool exhibit similar stylistic properties, yet resulted in more "beautiful shapes" compared to the shapes created with the expert tool (see figure 6.20).

In the *extrusion task*, the participants experienced both workflows as comparably easy, fast and precise (see figure 6.19, figure 6.21 for the modeling results). One participant stated that extruding with the *Tool Space* felt like "painting in space", whereas especially the more experienced modelers did not see an instant benefit. On the one hand, this might be due to the task that required the participants to repeatedly focus on the print-out in order to copy the model. On the other hand, only two participants started to use both hands towards the end of the task, indicating that adopting a two-handed style of work might need more training.

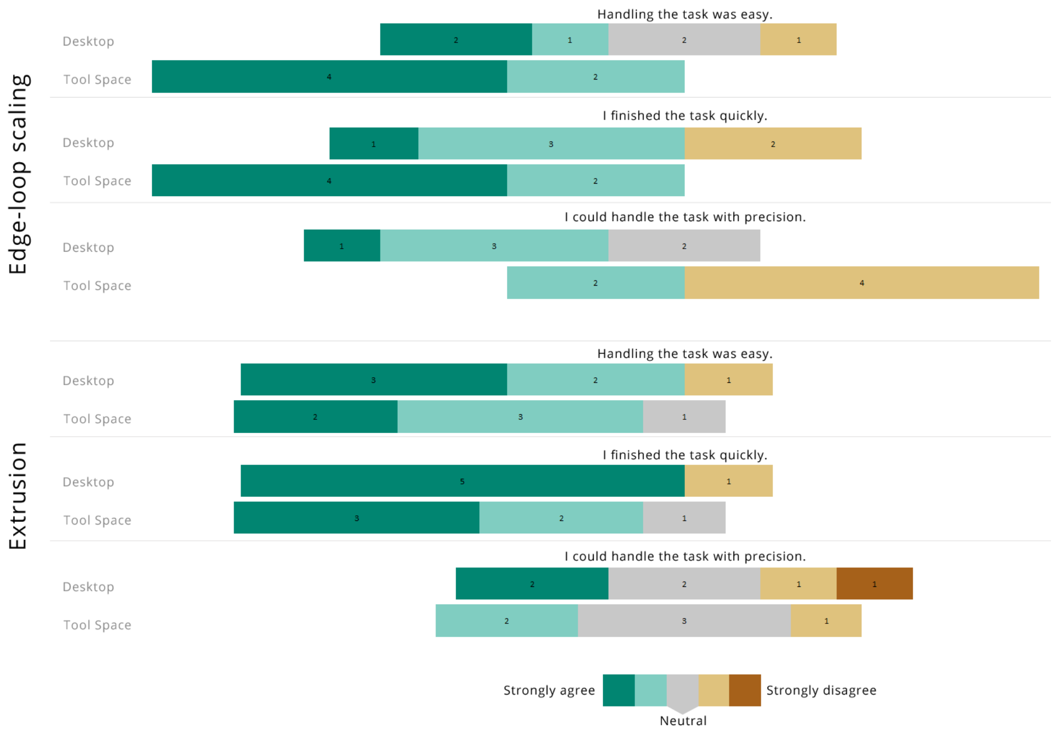


Figure 6.19: Subjective ratings for the edge-loop scaling and extrusion tasks.

Figure 6.20: Table modeling task results per participant with the *Tool Space* (top) and Blender/SketchUp (bottom).

The indifferent ratings regarding precision were surprising as the *Tool Space* does not provide features that enable precise modeling. However, the task did not require precise operation and participants did not aim for precision with the desktop tools either.

In general, participants had no difficulties with bimanual control when it was enforced in the edge-loop scaling task. However, two participants repeatedly tried to control selection and the scaling sequentially with one hand. This is reflected in the statements from the interview, indicating that this style of input is “enjoyable” and enriches the modeling experience, but also requires learning.

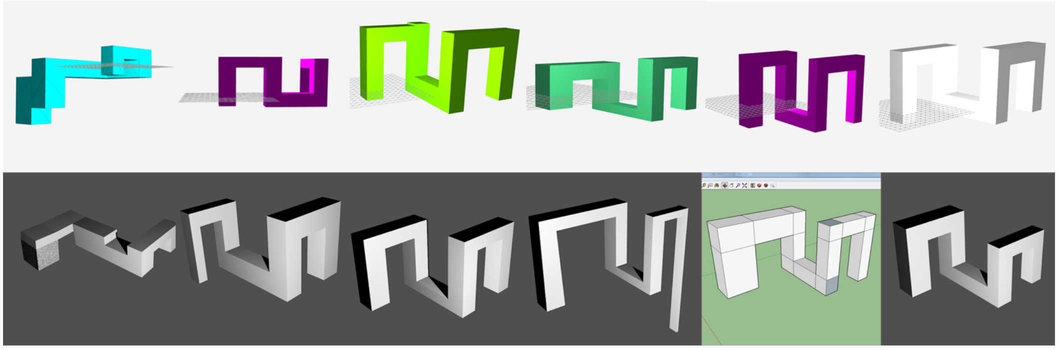


Figure 6.21: Extrusion task results per participant with the *Tool Space* (top) and Blender/SketchUp (bottom).

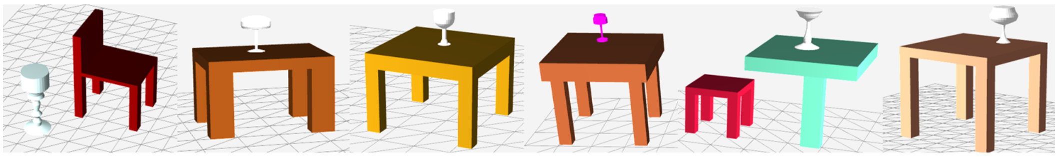


Figure 6.22: Models from the combined task.

Creating bent extrusions was perceived as more complicated than normal extrusion, yet "still manageable". I assume that the cognitive effort to plan the coordinated input movement is too high for explorative tool usage. However, users had little training with our tool.

6.1.6 Evaluation with Novice Users

The initial user feedback implied that the tools designed for polygon modeling are manageable by users with previous domain knowledge. In particular, the observations indicate that the techniques might stimulate exploration and allow for a quick sense of achievement. I concluded that one useful next step would be to expose the techniques to novice users without 3D modeling experience.

RQ1 How well do novices learn to use the *Tool Space* across multiple usage sessions?

RQ2 How do they reflect on the tools' role in their learning process of a complex application domain?

To answer these questions, an observational user study with five 3D modeling novices (four female), aged between 17 and 25, all students from varying educational backgrounds (business, sociology, cultural science and computer science) was conducted.

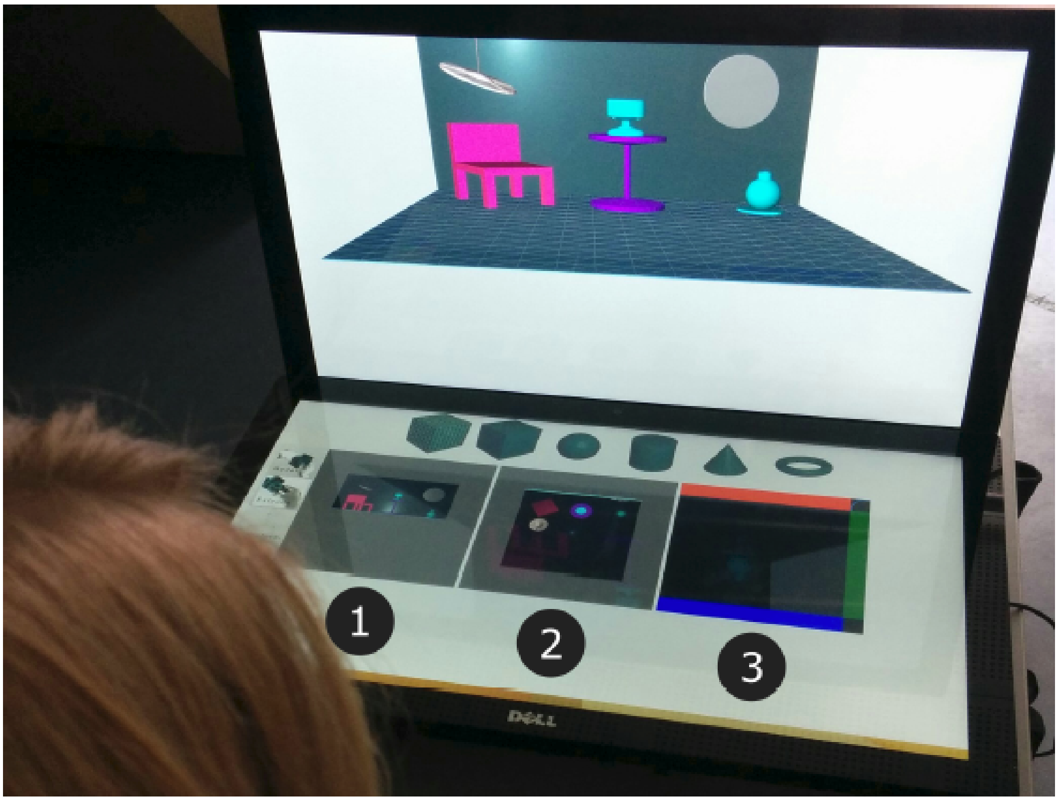


Figure 6.23: A participant during one of the sessions. (1) Side view to translate selected objects along the scene's y-axis, (2) top view to translate them in the scene's xz-plane and (3) scaling widget with axis- constraining bars (red, green, blue) at the borders.

Procedure

The study was organized in three sessions per participant, each lasting 60 to 90 minutes, with breaks of 2-5 days between sessions. The goal of the first two sessions was to familiarize the participants with the functions of the *Tool Space* for polygon modeling and give them time to learn the usage of the tools.

First Session During the first session, participants were instructed on the general system workflow and had the possibility to explore all functionalities freely.

Second Session The second session started with a recap of the system functions and tools, followed by a free modeling task.

Third Session In the last session, the participants were asked to apply their knowledge from the first two sessions and model an imaginary room (see figures 6.23 and 6.24).

Each session was concluded with a survey and a semi-structured interview. The survey asked to rate specific aspects concerning the system's usability and learnability on a seven-point semantic differential and answer related five-point Likert scale questions concerning the learnability of the system. These aspects were further discussed in the semi-structured

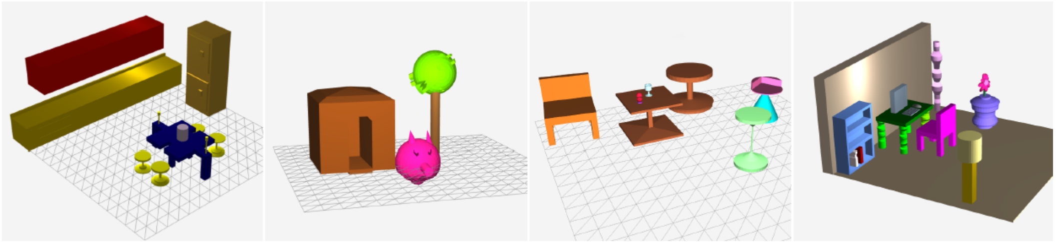


Figure 6.24: Exemplary modeling results from our observational study with five 3D modeling novices.

interviews. A more comprehensive perspective on the course of the three sessions was discussed during the final interviews conducted after the third session, where the participants were asked to reflect on their experience with the *Tool Space*.

Observations and Feedback

In general, all participants perceived the interaction with the *Tool Space* as easy across all sessions (see table 6.1). After the instruction on how to use the tools, they became "quite intuitively usable" (P1) even if they seemed rather complex when they were first introduced by the experimenter. Some users were initially skeptical about their ability to learn the tools, which is reflected in the semantic differential ratings: after the first session, three of five users rated the tools as "hard to learn", but after the second session four users changed their rating to "easy to learn" (table 6.1). Additionally, the interviews of the second sessions revealed that some were surprised (P1, P3, P4) of how well they were able to remember how to work with the tools from the first session (median = 2 for the statement "It requires a lot of time to learn how to use the prototype" across all participants and sessions).

Reproducibility The evaluation of the questionnaires implies that our techniques facilitate the reproducibility of the workflow: the statements *The prototype requires me to remember a lot of details* (median = 2, across all participants and sessions) and *The prototype is designed in such a way that learned steps are easily remembered* (median = 4, across all participants and sessions) support this assumption.

Display Setup All participants considered partitioning the interface into a horizontal *Tool Space* and a vertical display as a beneficial property. The upright display was mostly conceived and operated as overview, and the role of the horizontal surface as input area was well understood. However, all participants except P4 used the vertical display for scene navigation, although this was primarily meant to be supported through the base layer of the *Tool Space*. Addressing this observation during the final interviews revealed that the indirectness caused some confusion. On the one hand, the most common response was that the vertical screen was understood as the area providing a scene overview, hence it felt logical that input was supported there. On the other hand, three of the participants thought that a non-touch screen would have been sufficient for providing a visual overview. A further recurring topic was the desire for more functionality on the vertical screen, in particular allowing direct manipulation of the domain object.

Role of Hands and Touch The observation showed that all participants operated both the *Tool Space* and the upright display almost exclusively with their dominant hand. Only when explicitly required by the *edge-loop scaling* tool, both hands were involved. When this was addressed, the participants reasoned that dominant-hand control feels quicker and more accurate, even though some realized the potential to optimize hand movements between input elements by additionally involving the non-dominant hand. P3 commented that the system should more clearly visualize hand-task mappings. P5 thought that the layout of the *edge-loop scaling* tool was error-prone: the need to either switch or cross hands can arise because "you would start selecting the area with your right hand in the left field and then realize you need to operate the field on the right as well".

Employing touch input techniques instead of mouse and keyboard input was regarded as positive, which is underlined with the "hands-on feeling" they provide and comparisons to "working with clay". Yet, the observation also revealed well-known challenges specific to touch input: tasks requiring fine-grained input movements, such as polygon selection, led to a manifestation of the *fat-finger problem*. In this regard, P3 suggested alternative indirect selection mechanisms, particularly referring to selection brushes with variable diameters and a polygon selection view with more controls. Participants generally felt that the system was missing features to support precise modeling, such as numerical input, or grid snapping.

Further Observations While the interaction was perceived as *easy-to-use* in general, some participants felt that the insert mechanism for objects was cumbersome: "swiping it in is a nice metaphor, but in the computer context I still have clicking in mind" (P2). Also, during the study participants repeatedly tried to add objects to the scene with tapping sequences and only remembered the swipe-move after a series of failures – an observation I already made during the exposition of the quiz game 4.1.1. Further, sometimes participants swiped downwards instead of upwards, which was not recognized as valid input by the system. In terms of adequate alternative usage contexts, participants mentioned educational contexts, arguing with the systems capability to support shape exploration and visual thinking. Moreover, some participants thought that the "direct interaction" provided by the system would be a good way to support teaching 3D modeling to non-technically-minded users. Further potential for the dual-display setup was identified in contexts like *purpose-built games* or *creative work* in general, and particularly in video or photo editing.

Review The main goal of this section was to introduce the *Tool Space* concept. Using the example of 3D modeling, I illustrated how tool reification can result in novel virtual input devices that extend the input vocabulary for existing application domains. Despite their novelty, these tools are manageable by novice and expert users, and, despite being *indirect*, mediate a rather hands-on feeling.

		Score								
		3	2	1	0	1	2	3		
incomprehensible	Session 1	-	-	-	-	3	2	-	Session 1	*comprehensible*
	Session 2	-	-	-	-	3	-	2	Session 2	
	Session 3	-	-	-	-	1	3	1	Session 3	
easy to learn	Session 1	1	1	-	-	1	2	-	Session 1	*hard to learn*
	Session 2	2	1	1	-	-	1	-	Session 2	
	Session 3	1	3	-	-	-	1	-	Session 3	
unpredicatble	Session 1	-	1	-	1	2	1	-	Session 1	*predictable*
	Session 2	-	-	1	2	-	2	-	Session 2	
	Session 3	-	-	-	1	3	1	-	Session 3	
quick	Session 1	1	-	-	1	2	2	-	Session 1	*slow*
	Session 2	1	2	1	1	-	-	-	Session 2	
	Session 3	-	3	-	1	-	1	-	Session 3	
complicated	Session 1	-	-	-	-	3	1	1	Session 1	*easy*
	Session 2	-	-	-	-	4	-	1	Session 2	
	Session 3	-	-	-	-	3	1	1	Session 3	
clear	Session 1	1	1	3	-	-	-	-	Session 1	*confusing*
	Session 2	2	1	1	1	-	-	-	Session 2	
	Session 3	1	3	1	-	-	-	-	Session 3	

Table 6.1: Distribution of ratings for the semantic differential. The numeric values represent the number of users.

6.2 A Toolspace for Editing Audioclips

Implicitly, some of the ideas behind the *Tool Space* concept were already formed in projects I realized in an early phase of my dissertation work. For instance, in the concept of an audio workstation for the *Curve* (Palleis, 2014a), the notion of distinct task-display mappings was already present, but not yet formulated. In this specific case, audio clips appeared in the curved part of the display and could be dragged either downwards to an clip-editing area or upwards to a sequencer window. After having established the concept of the *Tool Space*, I revisited the original idea of a music workstation and designed an indirect touch interaction technique for two handed selection and editing of audio clips in a tool reification process based on the existing prototype. In the following section, I first introduce the interaction technique and then report on the results of a user study conducted by a student in the course of his Bachelor's thesis (see section 1.4). This study compared the novel technique with mouse input during a simple audio editing task.

6.2.1 Designing the *Tool Space* for Audio Editing

Based on the *Curve*, a two-handed indirect touch interaction technique was designed that supported a basic set of functions to work with audio files, such as navigation, selection and common domain functions. The newly designed interaction technique picked up the idea of distributing different commands between both hands by using spatial tool multiplexing. Figure 6.25 outlines both the application and tool components.

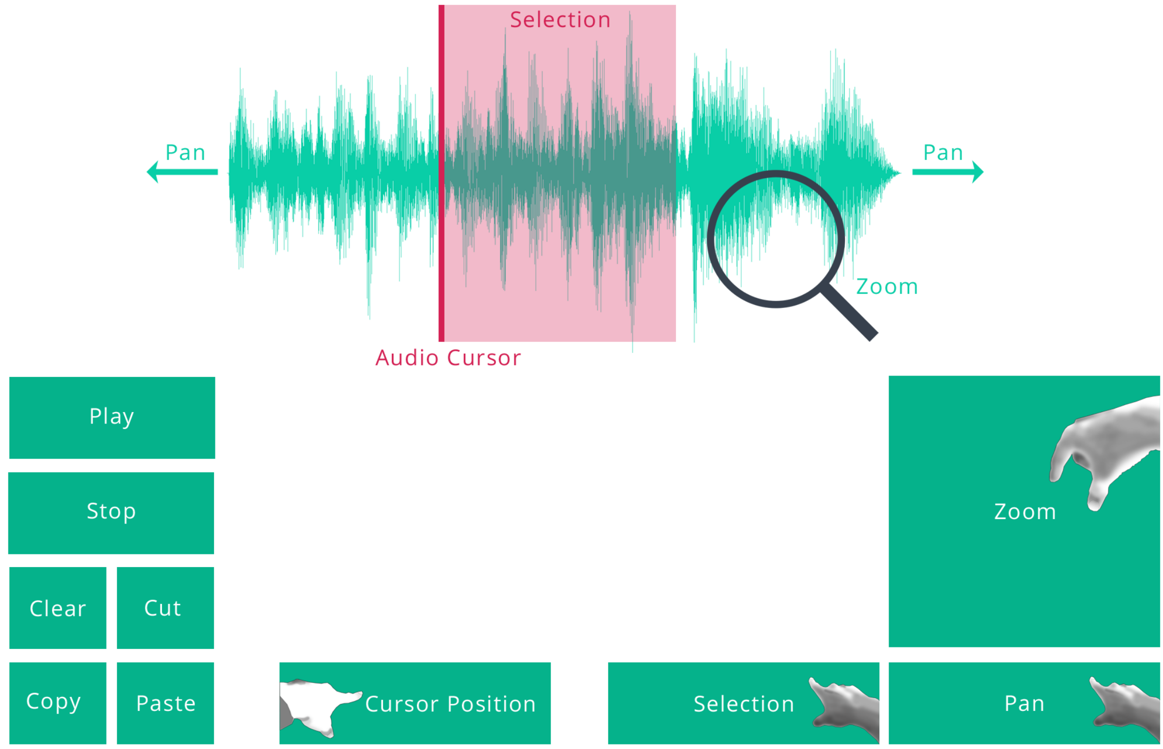


Figure 6.25: Conceptual overview of the *Tool Space* for audio editing

Audio waveform view The file view component (figure 6.25 top) consisted of a waveform visualization of the loaded audio file. The actual application included time indicators, a horizontal scrollbar for navigation feedback, selection highlighting and supported zooming.

Tool palette The tool palette (figure 6.25 bottom) included buttons to play and stop the audio file, as well as to cut, copy, paste and clear selections. Further it contained input areas to control the audio cursor position, to make selections, as well as to pan and zoom the waveform representation.

6.2.2 User Study: Mouse vs. *Tool Space*

In a lab experiment, the newly conceived interaction technique was compared to mouse input, which served as baseline. Figure 6.26 outlines the mouse-based interaction implemented in the prototype, which is based on existing software tools (e.g., Audacity, Adobe Audition etc.). The commands for cursor positioning, selection, navigation and zooming are executed by common mouse events, such as left-click, dragging and mouse-wheel scrolling. The buttons for copy, paste and cut have been replaced to a position close to the waveform in order to avoid covering large distances and traversing the curve with the mouse pointer. The clear button has been removed, as de-selection is achieved by clicking with the mouse outside the selected area - a common practice in existing GUIs.

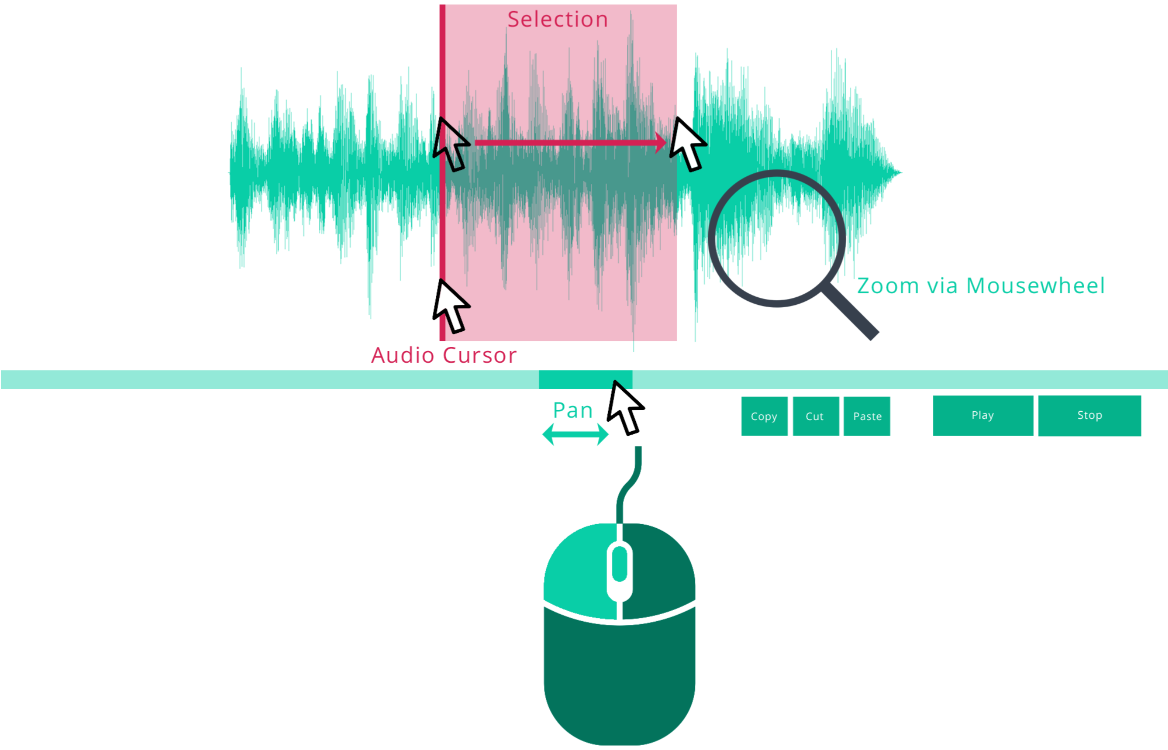


Figure 6.26: Mouse input for audio editing: navigation is supported via the scroll-bar, audio cursor position and selection are controlled via left-click and click-and-drag, zoom is operated with the mouse-wheel.

Task The task required participants to rearrange a given audio sequence by means of searching certain parts of the audio, cutting it out and pasting it at another location within the sequence. In particular, two audio sequences (A and B) containing spoken words (each around 45 minutes) from the university podcast “Tonspur Forschung” [7] were prepared. Using an audio editor, the sequences were disarranged by cutting pieces of the audio and pasting them at a different location. During the study, the participants' task was to restore the original sequence. The complexity of both task was intended to be comparable: each task required four cut/paste cycles and the length of disarranged subsequences was comparable.

Procedure A within-subjects experiment was designed, with input technique and audio sequence as counter-balanced independent variables (see table 6.2). In the beginning of the experiment, participants were introduced to the audio editing user interface as well as the rearrangement task. Before each condition, they were given time to freely familiarize with it.

[7] http://www.uni-muenchen.de/aktuelles/publikationen/tonspur/index.html

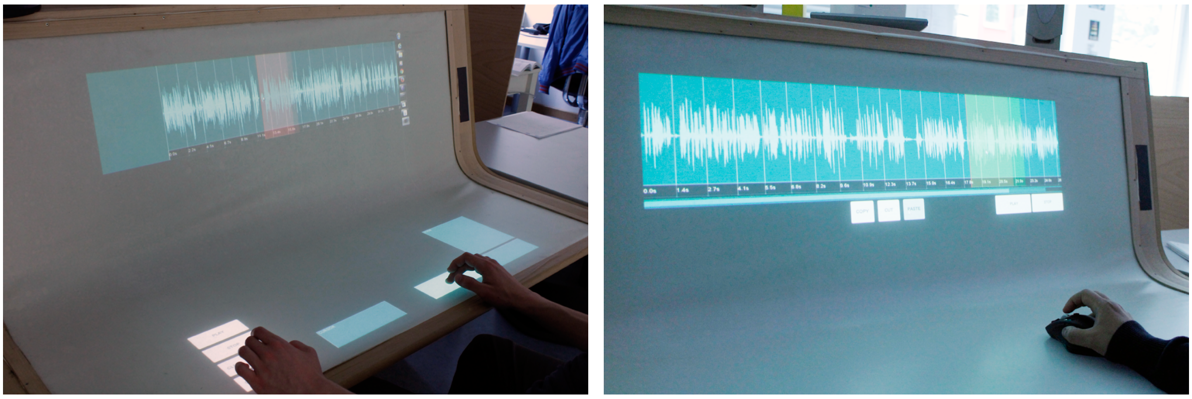


Figure 6.27: The input conditions compared in the experiment: (left) the implementation of the *Tool Space*, (right) the mouse-based interaction with the application.

Table 6.2: Counter-balancing of the conditions

Interface 1	Task 1	Interface 2	Task 2
Mouse	A	*Tool Space*	B
Mouse	B	*Tool Space*	A
Tool Space	A	Mouse	B
Tool Space	B	Mouse	A

Also, they were told about the necessary amount of cut/paste operations. After each condition, participants gave subjective ratings via questionnaires. Upon completion, demographic data was collected and a semi-structured interview was conducted that took up specific observations from the experiment, but included also more general questions concerning the novel touch-interface.

Variables and Measures The independent variables were the two interaction techniques (*Tool Space*, Mouse, see figure 6.27) as well as the two audio sequences. Further, the experiment included the following dependent variables and measures:

Task completion time Task completion time (TCT) was logged through the prototype application.

Workload Workload was assessed with unweighted NASA TLX questionnaires, filled out after each condition.

Subjective ratings Five-point Likert scale items were used to gather subjective data. The scale ranged from 1 (“I do not agree at all”) to 5 (“I fully agree”).

Hypotheses Due to spatial multiplexing of input areas, which allowed a higher input bandwidth, the following hypotheses were formulated:

H1 *Tool Space* interaction results in faster task completion times than mouse input

H2 *Tool Space* interaction is perceived faster than mouse input

Further, due to the pervasiveness of mouse input, as well as the physical affordance and transfer functions it provides, the following hypotheses were added:

H3 Navigation is perceived as more accurate with mouse input

H4 Selection is perceived as more accurate with mouse input

Participants Eight participants (two female, aged between 21 and 36 years, mean = 26 years, sd = 4.75) were recruited from students and alumni of media informatics, which took place at our lab. All participants were right-handed and experienced users of touch screens (7 smartphone, 4 tablet, 3 notebook). Two had previously participated at a user study involving the curve, but none could recall specific patterns of two-handed touch input usage.

Results

Due to the small amount of participants, the report and discussion of the results is focused primarily on the qualitative findings.

Task completion time Overall, task completion times were higher for *Tool Space* (14.23 minutes, sd = 7.15) than for mouse input (11.41 minutes, sd = 5.31). However, due to the small amount of participants, no statistical tests were applied.

Workload Workload was compared both for task and for input condition. For task, B resulted in a higher workload (47.5, sd = 18.89, se = 6.67) than A (41.25, sd = 20.18, se = 7.13). However, the difference was small and differences were well within standard errors, indicating a comparable level of task complexity. Regarding input technique, differences were even smaller, with a mean score of 43 (sd = 20.47, se = 7.24) for *Tool Space* and 45.75 (sd = 19.05, se = 6.73) for mouse input.

Subjective ratings

Overall, the subjective ratings show little differences for the different input techniques. Figure 6.28 illustrates the results of the questionnaires. For both input conditions, participants largely complied with the statement “Handling the task was easy” (median = 1.5 for *Tool Space* and median = 2 for mouse). Further, “I finished the task quickly” was rated more neutral (median = 3 for both input systems). Frustration was rated slightly higher with the *Tool Space* (median = 4), compared to mouse input (median = 4.5). Regarding selection and cursor placement, both input techniques resulted in comparable ratings.

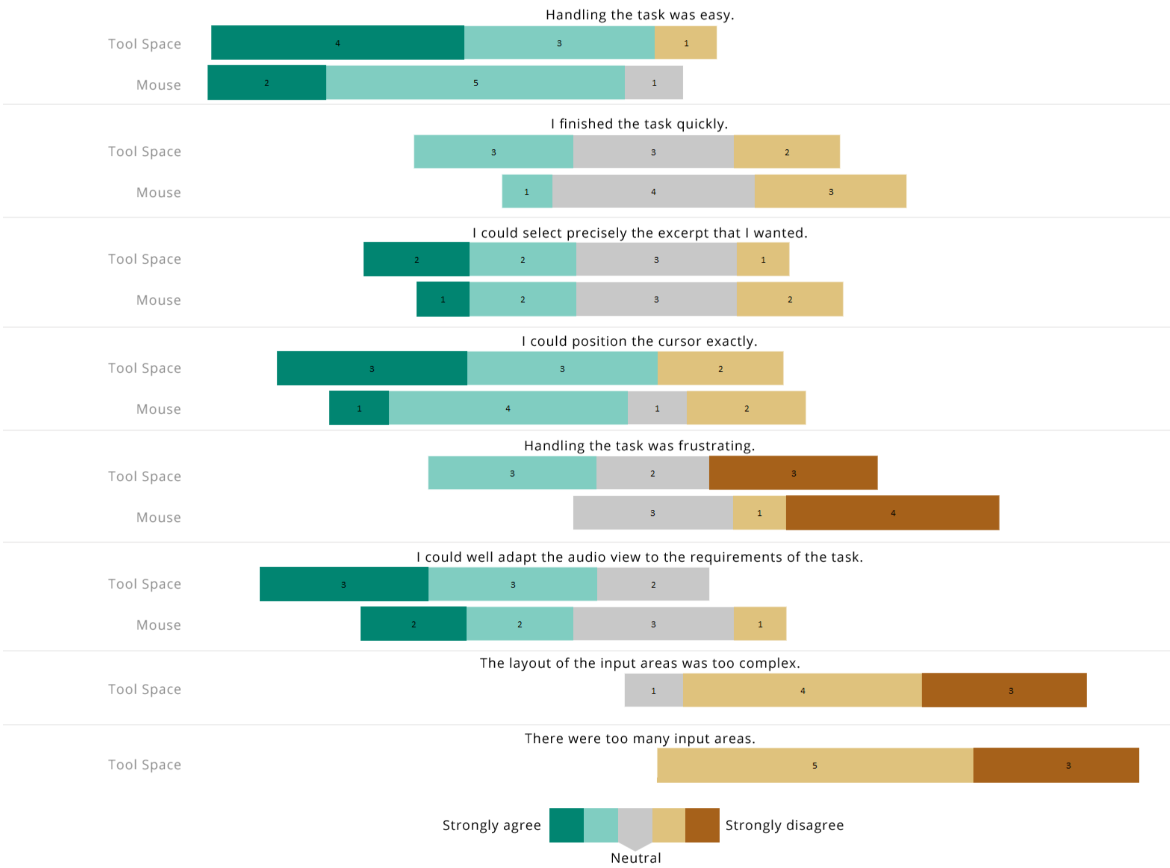


Figure 6.28: Results of the questionnaires. Numbers indicate the amount of participants who answered with that score on the 5-Point Likert scale.

Discussion

Workload and TCT Because of the small sample size, both the measured task completion times and the relatively low difference between task load indices need to be regarded with care. The task completion times were higher with the *Tool Space*, which was not surprising given the participants' familiarity with mouse interaction. However, the subtle difference in task load indicates that neither the increased expenditure of time, nor the involvement of both hands acting on spatially spread visual input areas per se lead to an increased workload.

Subjective sense of time While the measured times are shorter for mouse input, the subjective data reveal an opposed trend: "I finished the task quickly" resulted in more agreement with *Tool Space*. One reason could be *novelty*: learning to control a new interface could have influenced the participants' sense of time in favor of the *Tool Space*. Another reason could be related to the direct command mapping of the *Tool Space*: while mouse input was familiar, it inherently required explicit cursor-command-associations, such as activating navigation and selection with the same device by moving the cursor to the respective screen locations.

Frustration While the novelty may have positively influenced the sensation of time, it might have negatively influenced the perceived frustration, which was rated higher for *Tool Space*. In particular, the deviation from common touch interfaces caused initial confusion: for instance, some participants were observed trying to move the input areas onto the audio waveform to activate commands. Further misconceptions were caused by ill-conceived design choices, such as the strict separation between cursor positioning and navigation, which did not enable auto-scrolling by moving the cursor to the border of the waveform. In general, however, participants quickly learned the control mechanisms.

Perceived difficulty For both interfaces, the handling of the task was perceived as easy, indicating that the choice of input technique did not influence the overall task difficulty. A closer look into the data revealed that strong agreement with *ease-of-use* did not depend on the order of input condition. In particular, half of the strong agreements for *Tool Space* were given by participants who had started with mouse input, toning down the obvious assumption of a novelty effect.

Precision of selection and navigation Interestingly, the ratings regarding precision are slightly more positive for *Tool Space*, despite the familiarity with mouse input and the known detrimental effects of touch input. A potential explanation could be seen in effects enabled by parallelism: the two-handed control of the *Tool Space* allowed to simultaneously navigate and position the cursor, which was not possible in the mouse condition. This might have been perceived as beneficial, and thus resulted in positive ratings, although parallel or two-handed input was rarely observed during the study.

***Tool Space* layout** In general, the layout was not perceived as complex. Interestingly, in the concluding interview some participants stated that they had forgotten about the two-handed control, although it was demonstrated during the experiment. Thus, there were little insights on how the layout of the input elements facilitated two-handed input. Further, although the *Tool Space* contained one input area for every command, the amount of input areas was not perceived as too high.

Review Despite a small sample size, the results of the experiment are a further indicator for the *feasibility* of the *Tool Space*. The audio editor prototype introduced in this section comprised several common operations, such as horizontal scrolling and selection, which are found in other application as well (e.g., text or video editing). Although execution times were higher with the novel interface, participants neither indicated to be overburdened with the amount and layout of tools, nor did they perceive the *Tool Space* as inferior regarding performance compared to mouse input. Therefore, extending the *Tool Space* concept with more tools seems to be a viable approach in order to support more or more complex operations.

6.3 Chapter Summary

This chapter introduced the *Tool Space* concept, a novel interaction paradigm for dual-surface workspaces. In two projects, the general feasibility of the concept was explored by developing novel tools in the context of existing application domains in reification processes, and by gathering first user feedback in several small user studies. In the following, the main findings are summarized.

- The *Tool Space* is an interaction paradigm that stresses the tool as mediator for interacting with domain objects. Instead of *directly* applying commands to objects (e.g., using finger gestures), touch input signals are spatially multiplexed with visual structures, such as rectangular areas that resemble *virtual* touch pads.
- Tool reification is a process that thinks about the adaption of tools to application domains. In several such processes, novel tools for 3D modeling and audio editing tasks were conceived. Results and observations from initial user feedback indicate the general feasibility of the approach.
- In particular, participants of our studies perceived the *Tool Space*-mediated interaction as enjoyable and empowering, evoking comparisons to manual crafting. This highlights its ability to design for a high *degree of compatibility* (Beaudouin-Lafon, 1998).
- The *Tool Space*-inherent potential to support two-handed input techniques may influence the design of input techniques in two ways: first, it may be enforced by requiring simultaneous input from both hands, as illustrated by the example of the *edge-loop-scaling* tool. Second, it may be understood as a vehicle for transitioning from novice to expert, as illustrated by the *extrusion tool*. Here, homing both hands on distinct input areas would allow skilled users to perform rapid successions of camera and mesh operations. Yet, during the user studies, participants only rarely adopted a two-handed style of control.
- In a small comparative study, participants needed more time to complete a task using the *Tool Space* than using a standard mouse-operated WIMP interface, however, they did not perceive the novel input technique as inferior or more demanding.

IV

"UNDERSTANDING THE DESIGN OF THE TOOL SPACE"

7 Informing the *Tool Space* Design

The first user studies yielded positive feedback regarding *Tool Space* usage in specific contexts, such as its ability to mediate a hands-on feeling and to stimulate exploration. Also, the proposed tool compositions did not seem to overburden the participants of our user studies with complexity – in fact, the tools were easy to learn and remember. Yet, there is little information that guides the design of such tools.

While the size of embedded hardware track pads is often limited by system form factors (e.g., notebooks or VR controllers), the *Tool Space* as a conceptual interaction paradigm confronts interface designers with a range of design opportunities: size and shape, external and internal tool composition, mappings, one- or two-handed control.

The goal of the following sections is to provide a starting point for experimentally deriving knowledge which can inform the design of tools. As the design opportunities mentioned above are too numerous to be investigated in this thesis, I describe insights on two specific aspects: with one experiment, I explored the ability of the non-dominant hand to perform two-finger touch gestures – both independently and in cooperation with dominant hand input. Since some of the well-received tools presented in section 6.1 build upon bimanual two-finger input, a systematic assessment of this capability is important to motivate further conceptual work in this area.

In an additional series of experiments, I investigated the effect of virtual touch pad size and transfer functions on 2D navigation, which is a common task in graphical user interfaces. Input size and transfer functions are related: smaller input sizes allow for shorter physical input movements, which make it more tedious to cover large distances in 2D information spaces. Both increased input area sizes or transfer functions with higher gains may be used to support navigation, and the goal of my experiments was to contribute to a systematic understanding of how these factors influence the navigation performance.

7.1 Exploring Two-handed Multi-Touch Input

Within the field of human-computer interaction, two-handed (or bimanual) interaction is a well researched topic. It has been shown that carefully designed two-handed computer input can have beneficial effects on both performance measures and cognitive aspects (e.g. Kabbash et al. (1994) or Leganchuk et al. (1998)). However, while we use both hands during a wide variety of real-life activities, involving both hands to specify and invoke computer commands has mostly been confined to certain patterns: we use our hands cooperatively for text input and in the interplay of mouse input and modifier keys (e.g., shortcuts). Concerning stationary devices, this may be related to the *WIMP* (Windows, Icons, Menus, Pointer) interaction paradigm that still prevails in the realm of desktop computing, fostering the role of established input devices.

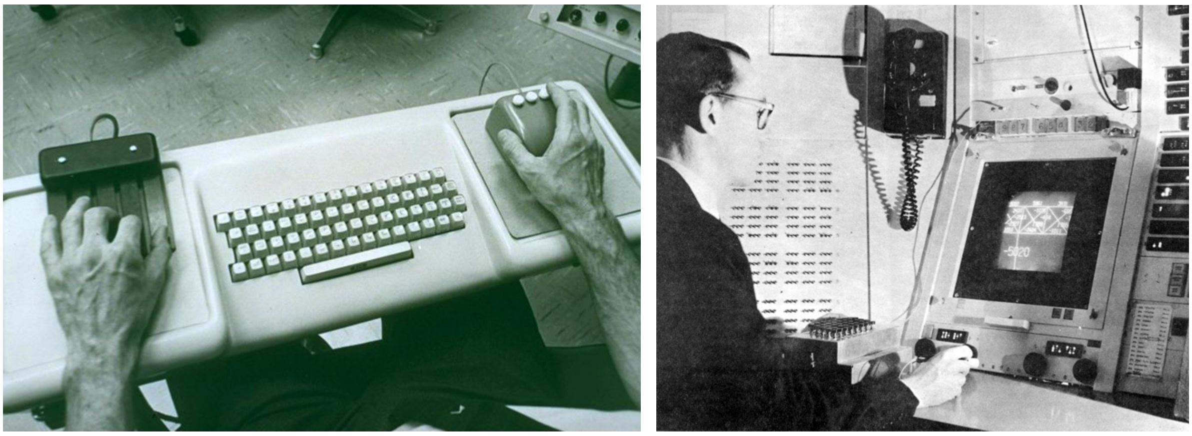
Until now, neither the research on (multi-) touch input, touchscreen technology and novel related interaction models, nor the commercial breakthrough of mobile touchscreen devices have caused a major transformation of the desktop computing paradigm. Mobile touchscreen interaction models are not adequate to be transferred into this domain without considerate adaption, as they are based on the assumption of one-handed input, due to the necessity of supporting the device with one hand. In contrast, self-supporting interactive surfaces, such as tabletop displays, can naturally be operated by two (or more) hands.

New devices – both research prototypes like Curve (Wimmer et al., 2010) or the MagicDesk Bi et al. (2011) and commercial products like HP's Sprout – exhibit touchscreen technology integrated into the horizontal layer of desks. However, often this primarily demonstrates a technical feasibility of certain form factors and provides little evidence how the technology may be reflected in adequate interaction models. Particularly, a systematic understanding of our hands' ability to both individually and cooperatively perform well-established two-finger touch gestures can inform the design of novel tools for the *Tool Space* paradigm proposed in this thesis.

In this section, I present the result of an experiment that explored our hands' ability to perform a 2D rectangle docking task both individually and cooperatively. Therefore, a standard desktop computer was extended with two tablet devices placed left and right of the keyboard, acting as input areas. During the experiment, participants were asked to align one square with another one using translation (one-finger drag), scaling (two-finger pinch) and rotation (two-finger rotation) operations. The work presented in this section was supported by Vanessa Niedermeier, who wrote her Bachelor's thesis under my supervision (see section 1.4).

7.1.1 Background: Two-handed Computer Input

There seem to be two stories of two-handed computer input. The first could be subtitled with "It has always been there." and refer to button interfaces, such as keyboards. Next to referring to text input keyboards, this story could also feature Ivan Sutherland's Sketchpad (Sutherland, 1963a), a light-pen operated predecessor of graphical CAD user interfaces, and Douglas Engelbart's prototype of the computer mouse (Engelbart and English, 1968). Both projects from the 1960s introduced novel interaction techniques focusing on pointing interactions, whereas Engelbart's invention of the computer mouse would turn out to be one of the most influential input devices ever conceived. However, both projects also envisaged a special device for the non-dominant hand (see figure 7.1). These button controllers allowed to type in text, activate modes or quasimodes (Raskin, 2000, p. 55) – a concept nowadays supported through modifier-keys on the keyboard.



(a) Douglas Engelbart's setup with chord handset on the left and a three-button mouse on the right.

(b) Ivan Sutherland's box with push buttons

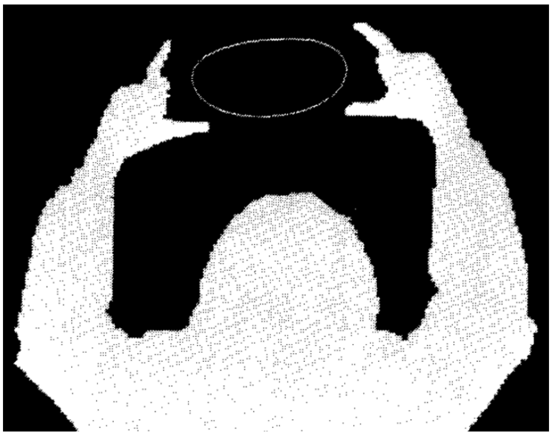
Figure 7.1: Early examples of designing two-handed input techniques. (a) permission requested from SRI, (b) from Sutherland (1963a)

In contrast, a second story could be subtitled with "We still do not have it, although it would benefit us". This story takes a slightly different perspective on two-handed input, in particular with regard to the type of input provided by the hands. Here, input from the non-dominant hand is conceived as a modality not only capable of actuating buttons, but instead providing continuous spatial input signals (Kabbash et al., 1993). In 1986, Buxton and Myers (1986) presented a series of experiments – gathered in a setup similar to ours (see figure 7.2(a)) – which demonstrated how user interfaces can benefit from explicitly designing for two-handed continuous input. In particular, their experiments show that splitting continuous compound tasks into sub tasks and distributing them between both hands improves performance, even if operated sequentially – indicating benefits beyond time savings. In this regard, Hinckley et al. (1997) have shown that in contrast to one-handed manipulation, two-handed operations can create a stable frame of reference independent of visual feedback.

Already a year earlier, Krueger et al. (1985) presented their work on *VIDEOPLACE*, the culmination of a decade-long work on artificial reality. In their setup, users' silhouettes were tracked and displayed via wall projections, allowing to interact with digital objects using both hands. This work is often referenced as the origin of the pinch-to-zoom two-finger gesture, but its use of a visual representation of the human body to interact with digital objects can also be seen as an early example of embodiment, enabling *interpretable* spatial interactions (Shoemaker et al., 2007).

Kabbash et al. (1993) were the first to systematically investigate movement time and accuracy of non-preferred hand input. For a set of input devices commonly used with the preferred hand (mouse, stylus, trackball), their experimental results show that the operation with the non-preferred hand does not inevitably lead to inferior performance. Especially for more coarse positioning tasks, the performance of both hands is comparable.

(a) Buxton's experimental environment for investigating two-handed document navigation with a computer in 1986.



(b) Hand silhouettes from VIDEOPLACE

Figure 7.2: Early examples of designing continuous two-handed input techniques. (a) from `http://www.billbuxton.com/`, (b) from Krueger et al. (1985)

Kinematic chain model While the work of Buxton and Myers (1986) was driven by the observation of how people use their hands in real life tasks, Guiard (1987) timely added a motor behavior perspective onto two-handed cooperation. His theoretical framework for understanding asymmetry in the context of bimanual action included a model – the kinematic chain model (KCM) – that conceived the human hands as two motors, cooperating as if they formed a kinematic chain. Based on this model, Guiard formulated three principles on asymmetric division of labor in bimanual action that have informed the design of many subsequent two-handed interaction techniques (e.g., Bier et al. (1993)):

1. the non-dominant hand's actions precede the dominant hand's actions
2. the non-dominant hand's actions set the frame of reference for the dominant hand's actions
3. the hands operate in different spatio-temporal scales

Guiard's model has been important to HCI research, because it provides evidence as basis for design guidelines concerning two-handed interaction techniques that match human motor and cognitive abilities, i.e. that facilitate parallelism and support cognition (Leganchuk et al., 1998). Based on the KCM, Bier et al. (1993) introduced the toolglass widgets – movable tool palettes that can be positioned with the non-dominant hand and provide click-through buttons for the cursor operated with the dominant hand. In a graphical design context, Kurtenbach et al. (1997) introduced a novel GUI paradigm based on similar ideas, observing not only performance increases, but also beneficial qualitative effects on the designers' input vocabulary. In contrast to real world tasks, these concepts of two-handed computer input involved two input devices defining distinct spatial frames of reference. Balakrishnan and Hinckley (1999) showed that Guiard's principles are robust and work even when these frames of reference are completely decoupled from each other, as long as coherent visual feedback is provided.

Kabbash et al. (1994) explored a drawing task and showed that an "asymmetric dependent" distribution of subtasks between hands – in particular the toolglass technique – yield a superior performance compared to unimanual input and poorly conceived bimanual input techniques, which can worsen performance. Similarly, Balakrishnan and Kurtenbach (1999) showed how distributing control of virtual camera and object manipulation in a virtual 3D environment can result in performance benefits of up to 20%.

More recently, Hinckley et al. (2010) presented work on two-handed pen and touch input, drawing inspiration from manual handling of physical notebooks. Pfeuffer et al. (2016) presented work on bimanual input on large touchscreens, with the non-dominant hand performing navigation subtasks (i.e. pan and zoom), and the dominant hand operating a stylus.

Symmetric bimanual interaction In contrast to asymmetric cooperation, symmetric bimanual actions describe tasks that assign equivalent roles to each hand. Real life examples include rolling out dough or steering a bike. With regard to computer input, Balakrishnan and Hinckley (2000) have investigated the influence of different visual feedback factors on parallelism in symmetric bimanual tasks. Further, Latulipe et al. (2006, 2005, 2006) presented a range of computer tasks, where symmetric outperforms assymetric bimanual input, such as image alignment and manipulating two points on gradation curves.

Input bandwidth Apart from the rather conceptual work of Krueger et al. (1985), many studies regarding bimanual input involved a setup of two physical input devices. For instance, Latulipe et al. (2005) used a custom mouse driver that allowed her to operate two mice. Buxton and Myers (1986) used a combination of puck and touch pad. With the introduction of multi-touch input, a further complexity has been introduced: while previous experiments assumed two single-point input devices (e.g. mouse, stylus, puck, finger), two points of input are now easily provided by using two-fingers of one hand (e.g., a pinch-to-zoom gesture). In this regard, Moscovich and Hughes (2008) provide insightful details on the relation of visual task perception and adequate indirect touch input mappings. In particular, their results shed new light on one of the tasks used in (Buxton and Myers, 1986): here, rectangle docking was split into the subtasks scaling and positioning and their operation was divided between hands. However, according to Moscovich and Hughes (2008), moving and scaling rectangles are subtasks that are perceived as integral. Hence, a unimanual two-finger control should be better suited for this task than two-handed control.

7.1.2 Experiment – A Two-handed Docking Task

Based on these observations, our experiment explored two-handed multi-finger input. In particular, we were interested in our dexterity to execute established two-finger gestures with both hands, separately and cooperatively. In particular, we were interested in the following questions:

RQ1 How does our ability to integrally control an object's position, scale and orientation using established two-finger gestures compare between our preferred and non-preferred hand?

RQ2 How well can we control two-handed two-finger gesture input and which roles do we assign our hands in a rectangle docking task?

Based on previous research on two-handed input outlined above, we formulated the following hypotheses:

H1 Two-handed multi-touch input will result in the fastest docking task performance.

H2 Two-handed multi-touch input follows the principles of Guiard's kinematic chain model.

Apparatus Our apparatus was based on a pair of tablet devices that resemble the touch input areas suitable for desktop computing proposed by Bi et al. (Bi et al., 2011) and are thought of as a further variant of implementing the *Tool Space* paradigm.

The experimental setup was comprised of an Apple Mac Mini, a 23 inch Dell monitor with full HD resolution (ST2340) and a conventional QWERTY keyboard. Additionally, two 7 inch Samsung Galaxy Tab 2.0 tablet devices running Android 4.0 were used as multi-touch input surfaces (figure 7.3 a). The tablets were placed left and right of the keyboard and allowed participants to perform touch gestures while resting their forearms on the desk. To allow for comfortable operation, participants were allowed to adjust the height of their chair before starting the experiment.

Tasks and Conditions The experiment was based on a simple docking task, a well-established task in research on multi-degree-of-freedom input devices (Zhai and Milgram, 1998). It asked participants to align given turquoise starting squares with varying target squares by means of translation, scaling and rotation. Translation was controlled with one-finger dragging gestures, scaling and rotation with two-finger pinch and rotation gestures.

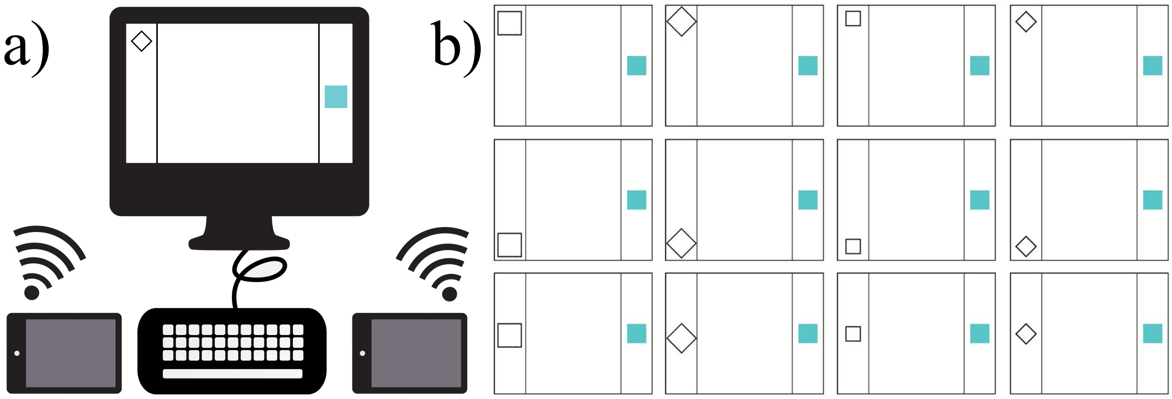


Figure 7.3: a): overview of the experimental setup. b): the 12 different docking tasks for right-handed and two-handed input. For left-handed input, the layout was flipped.

The docking task implementation consisted of two applications: the Android application running on the tablet devices displayed a blank white screen and detected touch gestures, which were sent via Open Sound Control messages to a JavaFX application running on the Mac Mini. This application handled the logic, the visual display, and the data logging for the docking task experiment. In particular, the JavaFX application received and interpreted the touch gesture data from the tablets and accordingly updated the position of both starting and target square.

The docking sequence was repeated across three conditions: using the dominant hand only (*DH*), using the non-dominant hand only (*NDH*), and using both hands cooperatively (*2H*). In the two-handed condition, both starting and target square could be manipulated simultaneously: the non-dominant hand controlled position, scale and orientation of the square shown on one side of the screen, whereas the dominant hand manipulated the square on the other side of the screen. In contrast, in both one-handed conditions only the starting square was manipulated. We used a relative mapping in order to allow participants to clutch. In particular, a linear transfer function with a gain factor of 1 was used, i.e. input signal deltas were used to change square properties in an unaltered fashion.

The starting square always appeared at the same screen location (vertically centered on one side of the display). In contrast, the target square was displayed on the opposite side of the screen and varied throughout the test sequences in position (top, middle, bottom), size (large, small) and rotation (0°, 45°), resulting in 12 distinct target square configurations. Task completion time measurement was based on two vertical lines, confining the starting and target area. Time measurement started only as soon as one rectangle had passed its corresponding line (figure 7.3 b). To support a clear hand-to-square correspondence in the one-handed conditions, i.e. to ensure perceptual compatibility – a decisive factor for the design of bimanual input techniques (Moscovich and Hughes, 2008) – the visual task layout was flipped according to the condition: the starting square was always displayed on the side of the manipulating hand. In order to enforce the use of both hands in the two-handed

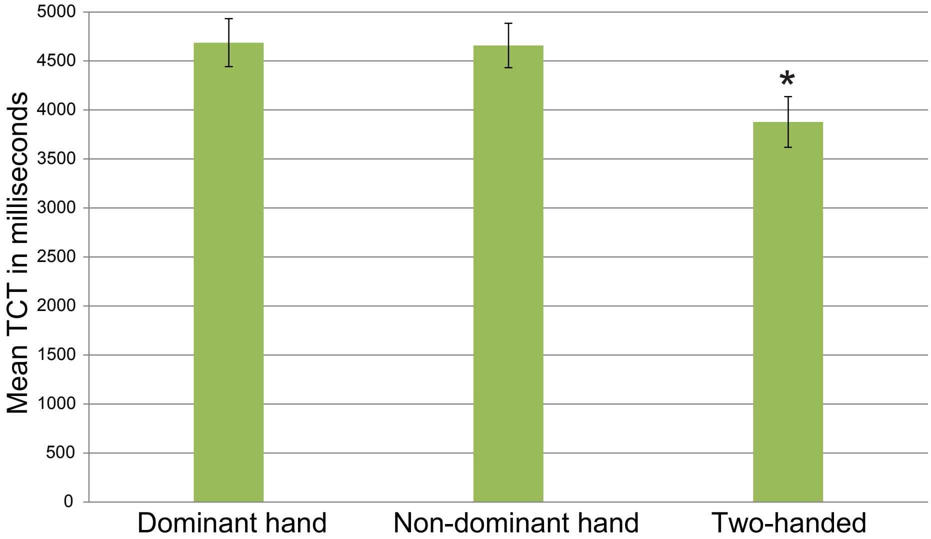


Figure 7.4: Mean task completion times per input condition. Error bars indicate SEM.

condition, the square alignment could only be completed once both rectangles had been dragged across the lines confining their respective area.

Measures We logged task completion time as the time from initial line crossing (either starting or target square in *2H*) until successful square alignment. For the square alignment itself, we used a threshold that accepted an alignment as successful as soon as 95% accuracy was reached for translation, scale and rotation properties of the squares. Further, we continuously logged the squares' screen locations, all square properties and the input source of the successful alignment (*DH* or *NDH*).

Participants From our students' Facebook group, we recruited 20 participants (11 female) aged from 20 to 37 (mean = 23.85, SD = 3.53). All participants were right-handed and used touch input devices on a daily basis. Participation was compensated either with a 5 Euro voucher for an online store or extra study credits.

Procedure The experiment was based on a within-subjects repeated-measures design, with input technique as independent variable with the levels *DH*, *NDH* and *2H*. Each participant had to perform the 12 individual dockings (figure 7.3 b) 4 times per input technique, resulting in 4 blocks of 12 dockings in randomized order for every input condition. For each condition, we regarded the first two blocks as training and used only the last two blocks for data analysis. Therefore, our data analysis is based on 480 individual dockings per input condition (2 blocks x 12 dockings x 20 participants). Further, the input conditions were tested in a fixed order: first, the docking task was executed with the dominant hand, then with the non-dominant hand and eventually using both hands. Because of the simplistic nature of the docking task, the training for each condition and the randomization of task order, we did not expect learning effects between input conditions.

In the beginning of the experiment, demographic data was collected via an online survey. To assess workload, participants filled out a raw (unweighted) NASA TLX questionnaire after each input condition, which is less time consuming than the original NASA TLX and yields comparable results (Byers et al., 1989). Upon completion, participants were asked during a short interview to decide on their preferred input condition, to give an informal performance and demand self-assessment. Overall, the experiment lasted about 30 minutes.

Results

Task Completion Time The mean task completion time per docking was 4686.9 ms (SD = 1096.5) for *DH*, 4658.6 ms (SD = 1014.12) for *NDH* and 3877.27 ms (SD = 1156.67) for *2H* (see figure 7.4). A repeated measures ANOVA with a Greenhouse-Geisser correction determined that task completion times differed significantly between input conditions ($F(1.770,33.63) = 8.930$, $p < 0.005$). Post hoc tests using the Bonferroni correction revealed that *2H* resulted in a significantly faster mean task completion time than both *DH* ($p = 0.002$) and *NDH* ($p = 0.018$). No significant difference was observed between *DH* and *NDH*.

Square alignment In order to investigate Hypthesis 2, we analyzed the square manipulations for *2H* in more detail. From the 480 dockings performed in this condition, 390 were concluded by the dominant and 90 by the non-dominant hand. Further, we observed that most of the participants first moved the target square with the non-dominant hand towards the starting square and subsequently performed further adjustments of position, scale and rotation with the dominant hand. This procedure can also be observed in figure 7.5, where we plotted the screen coordinates of the successful square alignments. It shows that a large part of the distance between the squares is covered by left hand movements and that the final alignment occurs in the center of the display, at a vertical position similar to the starting square's initial position. These findings indicate that participants operated in concordance with our assumptions based on the Kinematic Chain Model: the non-dominant hand's action precedes the dominant hand's action and is responsible for coarse manipulations used to frame the more fine-grained manipulations of the dominant hand.

Workload and subjective data The results of the Raw TLX questionnaire indicate that *2H* causes most workload (3320 points), closely followed by *NDH* (3315 points) (see figure 7.6). *DH* causes least workload (2785 points). These ratings are reflected by the users' subjective assessment of mental demand in the concluding interviews: half of the participants ($n = 10$) stated that *2H*, almost the other half ($n = 9$) that *NDH*, and one that *DH* was most demanding for them. Vice versa, 15 participants stated that *DH* was least demanding ($n = 3$ for *2H*, $n = 2$ for *NDH*).

The question for personal preference (see figure 7.6) revealed that *2H* is preferred by 16 and *DH* by 4 participants. Vice versa, 15 participants indicate that *NDH* as least preferred condition, followed by *DH* ($n = 3$) and *2H* ($n = 2$).

The self-assessment of performance shows that 10 participants think they performed best with *DH*, followed by *2H* ($n = 9$) and *NDH* ($n = 1$). Vice versa, 14 participants thought they performed worst with *NDH*, followed by *2H* ($n = 4$) and *DH* ($n = 2$).

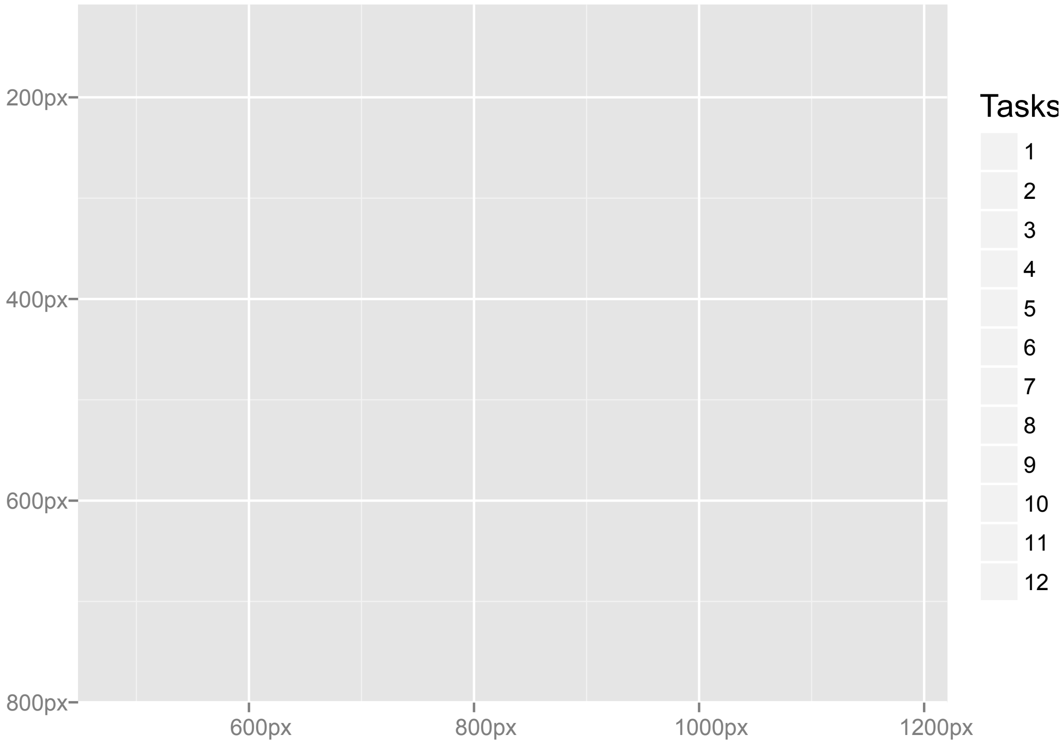


Figure 7.5: Successful square alignment locations in screen pixel coordinates per docking (detail).

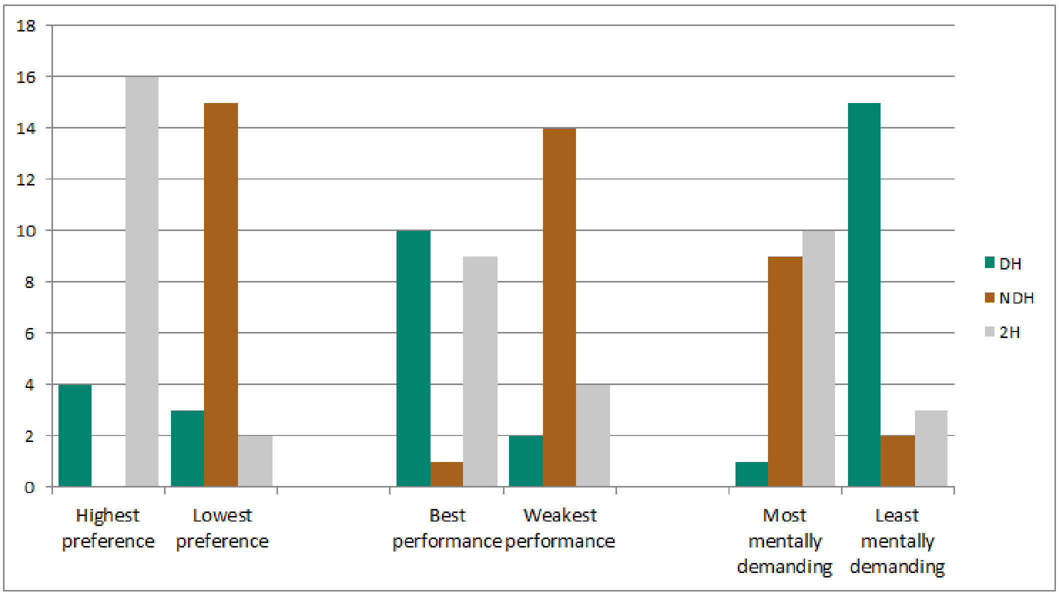


Figure 7.6: Personal preferences and assessments of the three input conditions.

Discussion

Firstly, we address a general limitation of the experiment: moving two squares simultaneously is a disputable comparison to one-handed input, however, we kept translation distances equal. The rationale behind this design was that we aimed to find out how parallelism in indirect multi-touch input and the resulting motor and cognitive efforts would affect the task completion time. In particular, we assumed that the observation of increased task completion times would indicate a non-functional concept of two-handed parallelism.

The two-handed input condition resulted in the shortest task completion times This finding was expected and indicates a confirmation of H1. Further, it is in line with the related work, but goes beyond previous research by showing parallelism benefits for two-handed multi-finger input techniques. With regard to Buxton's experiment (Buxton and Myers, 1986), where participants executing a compound task that asked to position an object with one and scale it with the other hand exhibited a high amount of parallelism, we showed that one-handed input for continuous multi-degree-of-freedom tasks (e.g., using two-finger pinch and rotation gestures) can be extended with non-dominant hand input without causing a significant load on the cognitive or motor systems, at least as long as the task's visual feedback exhibits perceptual compatibility.

Two-handed docking is an asymmetric task Also, in previous studies the choice of input devices often reflected the underlying concept of two-handed cooperation: for asymmetric roles of hands in Guiard's sense different input devices were used, whereas identical ones were used for symmetric roles. In contrast, our choice of input device and indirect touch mappings provided each hand with the same interaction capabilities, but nevertheless we observed that participants operated the task in an asymmetric fashion, supporting H2. This finding stresses the importance of understanding the relation between task nature and its implications on the design of two-handed input techniques. Indirect multi-touch seems to be a promising modality for exploring and supporting cooperative input strategies, because it allows to oscillate between symmetric and asymmetric styles both during development and actual usage. This flexibility seems important to account for the multitude of compound tasks and their still not well-understood characteristics, which are decisive for the design of adequate indirect touch mappings (Moscovich and Hughes, 2008).

Novelty effect and subjective assessments The strong preference for the two-handed condition may be attributed to novelty. Further, although input with the non-dominant hand and in the two-handed condition yielded increased mental demand, the results suggest that the increase is not high enough to decrease performance. In particular, this indicates that the operation of simple and well established touch gestures can be transferred to the non-dominant hand and further supports the finding that two-finger gesture input of the dominant hand can be extended with one-finger touch input gestures from the non-dominant hand without exceeding cognitive abilities.

Review Compared to one-handed input, the results of our 2D docking experiment confirm the assumed performance benefits of two-handed cooperation, which exhibited

Guiard's well-known principles of asymmetric bimanual interaction. Yet, the observations on non-dominant input capabilities indicate that designers of two-handed multi-touch input techniques can assume a comparable level of dexterity for both hands. This motivates further work on two-handed tools, such as the *edge-loop scaling tool* presented in section 6.1.4.

7.2 Understanding the Impact of Form Factors and Transfer Functions

In the following sections, I present experimental work targeted at better understanding the impact of form factors and transfer functions on the interaction with virtual tools. In particular, the presented experiments investigate effects of touch indirectness, virtual device size, and non-linear transfer functions on 2D navigation performance.

Besides established metrics to measure navigation performance, such as task completion time or efficiency, I incorporated a measurement of spatial memory performance which was inspired by research aimed at detecting distinct benefits of direct touch input compared to mouse input (Jetter et al., 2012; Tan et al., 2002). This seemed useful to me, since on the one hand, the *Tool Space* rationale envisages a complementary coexistence of different input modalities. A clear understanding of their respective effects on interaction beyond time and accuracy is important to make informed *Tool Space* design decisions. On the other hand, exploring the transfer of direct touch to indirect touch characteristics contributes to a better understanding of indirect touch input techniques in general.

Inspired by (Jetter et al., 2012), I chose a two-dimensional navigation task with a relative input mapping for my experiments. This is a frequent task in both desktop computing environments and touch-based user interfaces (e.g., map navigation), and in order to enable smooth and easy operation of interfaces involving pointing at interface items, it is essential that users remember the location of items on the screen and are able to easily navigate to them.

As related research has indicated that form factors of physical input space seem to have an effect on common tasks (Gilliot et al., 2014b), I was particularly interested in how decreasing the input surface size influences the interaction. With the *Tool Space*, the virtual input devices are smaller than the display, but this also applies to other cases, for instance embedded track pads on laptops.

7.2.1 Background

A distinct line of previous research has inspired the experiments presented in this section: motivated by concepts from the field of *Embodied Cognition*, both Tan et al. (2002) and later

Jetter et al. (2012) observed positive effects of direct touch interaction on spatial memory performance compared to mouse input.

With the emergence of ubiquitous computing and the feasibility of tangible user interfaces (see section 2.1.3), the *Embodied Cognition* theory has become of increasing interest to human-computer interaction. Its core idea is that the mind and the body co-evolve and that proprioception, kinesthetics and the corresponding parts of the brain develop as a unitary system (Kelso, 1997). Based on this assumption, the role of kinesthetic cues, i.e. cues derived from the position of the body's parts relative to itself or to its surrounding environment, has been explored in comparisons between direct touch and mouse input. In particular, Tan et al. (2002) presented an experimental design, which involved two tasks: an interaction task involving a 2D information space, and a memory task testing the recall of spatial features of the information space. The memory task asked participants to recall icon positions within a 11 x 7 grid displayed on a touch-enabled 18 inch LCD monitor, whereas they previously had to position the icons within the grid, either using touch or mouse input. Interestingly, the results indicated a significantly improved memorization performance for direct touch input.

Inspired by these findings, Jetter et al. (2012) conducted a similar experiment with an icon position recall task. In their study, the interaction task consisted of navigating an information space with sets of icons preassigned to grid positions. The information space was displayed on a 30 inch tabletop computer, and the navigation was operated either with 2D touch panning gestures or using a connected mouse. Here, the results not only indicate a significantly increased memorization performance for touch panning compared to mouse input, but also a significantly improved navigation efficiency. Yet, they also indicate that these benefits may only be stable for linear transfer of input signals, as they were not observed when panning was complemented with zooming, which is logarithmic.

In general, the particular effect of the kinesthetic cues provided by touch input remains unclear, as in the above mentioned experiments, not only the *spatial directness* of input modalities varied, but two completely different interaction styles were compared: in contrast to direct touch input, mouse input involves a co-control of click and drag operations and usually is based on a non-linear transfer function. Therefore, I wanted to explore if the spatial memory and navigation performance gains observed in these studies transfer to a *Tool Space* scenario, where touch gestures directly coupled to commands occur spatially separated from the display.

Several studies have explored other effects of indirect touch input: for instance, Schmidt et al. (2009) compared direct and indirect multi-touch input on a large dual-display setup. In contrast to the relative mapping of the panning task in our experiment, their study was based on an absolute mapping for pointing and dragging tasks. Their results imply significant performance benefits for direct touch, yet their indirect condition involved hovering over the horizontal surface to capture and display hand contours on the distant display which led to difficulties in hand-eye coordination.

Also based on an absolute mapping, Gilliot et al. (2014b) have explored the effect of input surface form factors on target selection tasks and observed a positive correlation between decreasing input surface size and target selection accuracy. Further, their results indicate that diverging aspect ratios of input and output areas are detrimental to selection accuracy.

Further, established indirect input with relative touch mappings seems to be not fully understood yet: Nancel et al. (2015) have investigated the effect of clutching on indirect touch pointing tasks using a physical track pad embedded in a laptop computer. They compared different transfer functions and found that while clutching is often regarded as a disadvantage of small input device spaces, clutch-less movements with faster transfer functions are harder to perform and more error-prone.

7.2.2 Experiment 1 - Exploring Touch Input Area Size

The first experiment was inspired by Jetter et al. (2012), and its goal was to test whether the improved navigation and spatial memory performance with direct touch observed in the original study transfers to an indirect touch input condition. Therefore, I formulated the following research questions:

RQ1 How will the spatial separation of visual display and touch input movements affect navigation and spatial memory performance in a 2D panning task compared to direct touch?

RQ2 How will a decreased input area size affect navigation and spatial memory performance when using indirect touch?

Apparatus and Conditions

The testbed was based on a 23 inch touchscreen by Dell (ST2340) with a resolution of 1920 x 1080 pixels, and the implementation of the experimental software including both the panning and recall tasks, study procedure support routines and logging was implemented with Java. A key consideration was to decide on a display orientation, as with regard to the spatial recall test, keeping the visual frame of reference constant was important. We opted for a horizontal screen orientation, since this reflects the *Tool Space* layout and allows participants to rest their arms on the desktop surface, enabling a physically less exhausting interaction than with an upright display (Figure 7.7 left).

The experiment asked participants to navigate a viewport (800 x 600 pixels, 212.2x159.1 mm) using panning operations. In the baseline condition (*DT*), participants used finger-input directly applied to the viewport to navigate its information space. In the alternative conditions the touch gestures were applied on virtual touch pads displayed on the right side of the viewport. In the first indirect touch condition (*IDT*), the virtual touch pad's size and aspect ratio equaled the viewport dimensions. In the second indirect touch condition (*IDT2*), the virtual touch pad was uniformly scaled with a factor of 0.5, resulting in a smaller input area (106.1x79.5 mm) (Figure 7.8). During this experiment, I used a linear transfer function

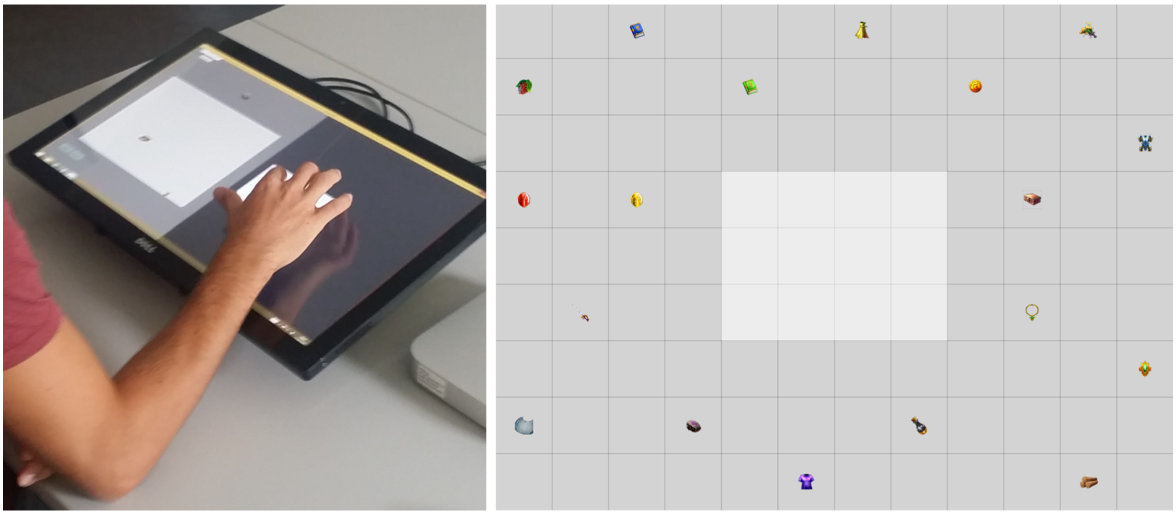


Figure 7.7: Left: A participant of our experiment performing 2D panning with a virtual touch pad. Right: The visual grid with one spatial item configuration. The highlighted empty grid cells in the center represent the home position.

with a gain factor of 1 across all conditions. Further, I disabled scrolling inertia, because there is no evidence on its influence on spatial cognition.

As in the experiments described in related work, I employed a discretized information space: the viewport displayed a part of a 12 by 9 grid, which contained a spatial configuration of 18 graphical items (Figure 7.7 right). Hence, the total grid size was 2400 x 1800 pixels (636.2x477.2 mm) – yet, at any time only a clipping of 800 x 600 pixels (212.2x159.1 mm) was visible.

The 12 central grid cells did not display items and served as a home position, which was displayed by the viewport upon the start of the experiment. The graphical items were samples taken from a coherently designed open source game asset package [1]. I created three spatial item configurations: based on randomly generated configurations, I manually reconfigured the items in order to avoid obvious memorization strategies (e.g., „all weapons are left“). Further, for each of the three configurations, I identified a subset of eight items, which served as search sequence. Each of the search sequences had a comparable optimal path length (between 1856 and 1989 mm) and a similar spatial distribution (same amount of items at borders).

Navigation Task In the navigation task, participants were required to repeatedly navigate through one of these item search sequences. A pop-up window displayed at the top center of the screen always indicated the item currently searched for. In order to successfully find an item, participants first had to click on this pop-up to start the time measurement, then navigate to the item, starting from the current position of the grid, and eventually click on the item. In contrast to Jetter et al. (2012), who initiated each item search from the home position, I employed a consecutive search from item to item, as this better matches real world

[1] `http://opengameart.org/content/basic-rpg-item-icons-free`

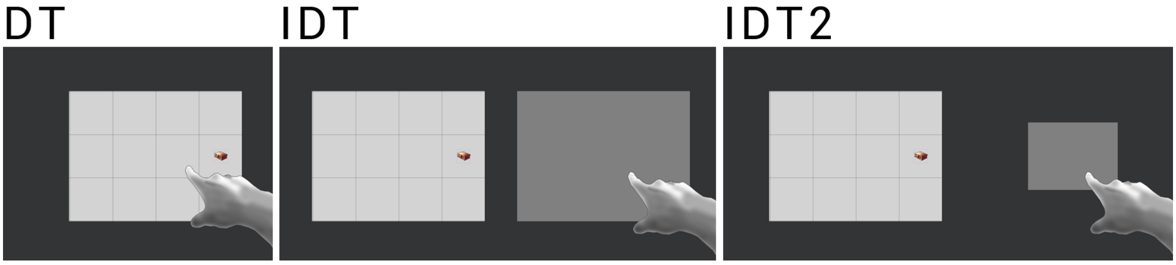


Figure 7.8: The three different input conditions used during our experiment.

tasks. By calculating optimal search paths and designing tasks with comparable path length, I was able to control distances. In particular, the optimal navigation paths were calculated for each search sequence as the minimum amount of grid movement necessary to move each search item's bounding box midpoint into the visible area of the grid.

A consecutive sequence consisting of eight of the 18 items had to be searched and during the task, the sequence was repeated eight times (later referred to as eight blocks of search), resulting in a total of 64 search trials. In the beginning of the navigation task, the search was random, as the spatial configuration of the items was completely unknown to the participants. Yet, through the repetitive search, they were meant to implicitly memorize the item locations and continuously improve their mental representations and navigation paths. By asking the participants to maximize for speed, I tried to prevent intensive memorization strategies and enforce a rather implicit short-term memorization of the item locations.

Based on timestamps and XY-coordinates of the isolated panning operations (excluding clicks on items), efficiency and task completion time were measured. As proposed by Jetter et al. (2012), navigation efficiency was calculated as the ratio between the actual and the optimal panning distance. The ratio is necessary to factor out the (small) differences of search path lengths resulting from the different spatial item configurations. Therefore, an optimal navigation performance would result in a ratio of 1.0, while decreasing performance would increase the ratio.

Spatial Memory Task In the spatial memory task, the sequence of previously navigated search items was shown to the participants in random order and they were asked to position each item at its original location within the grid. The item placement was operated using the arrow keys from a text keyboard in order to prevent unconscious use of motor or kinesthetic memory. The participants did not receive any feedback concerning accuracy upon making an item placement and for each item of the recall sequence, they started over with an empty grid in the home position. Recall performance was measured as placement error, which was calculated as the Euclidian distance in millimeters between the participant's item placement and its original position, whereas participants where asked to be as accurate as possible.

Procedure The experiment was based on a within-subjects design with input modality as counterbalanced independent variable with the levels *DT*, *IDT* and *IDT2*. Therefore, each participant had to perform both the navigation and the recall task three times. In the beginning of the experiment, demographic data was collected with an online questionnaire. Subsequently, there was a short introduction phase, during which I explained both the navigation and the recall task and introduced the different input conditions by demonstrating the application running on the touch display. For every input conditions, there was also a short training phase, during which the participants performed a short version of the navigation task (one block) with a dedicated training item set, different in style and subject. Eventually, the data was collected during the actual navigation task, directly followed by the recall task. Between the conditions, participants watched three-minute cartoon videos to relax. Upon completion of the experiment, participants were asked to informally rate the input techniques according to their subjective preference.

Results

I recruited 18 right-handed participants (ten female) aged 18 to 43 (mean 26.11, SD = 6.63) via a university-wide newsletter service. All of them were using touch input devices (smartphones, tablets) on a daily basis. 15 were students (three from a technical/computer science program), two were researchers (computer science and physics) and one was a librarian. Participants were compensated either with a 10 Euro voucher for an online store or extra credits for their study program.

Results for Spatial Memory Performance

For *DT*, the mean placement error was 104.9 mm (SD = 47.4), for *IDT* it was 101.6 mm (SD = 49.5) and for *IDT2* it was 112.1 mm (SD = 49.1). A repeated measures ANOVA determined that the mean spatial memory performance did not differ statistically significantly between input styles ($F(2, 34) = 0.333$, $P = 0.719$).

Results for Navigation Performance To consider the implicit learning described above, I first compared navigation efficiency and time for all three input conditions with block as factor (figure 7.9 shows the navigation efficiency across blocks). Due to a non-normal distribution of the data (determined with Shapiro-Wilk tests), I performed Friedman tests instead of a repeated-measures ANOVA. Post hoc analysis results (Wilcoxon signed-rank tests, Bonferroni-corrected with the significance level adapted to $p < 0.007$ as we compared 8 consecutive blocks) indicated that time and navigation efficiency improvements were not significant after blocks four (time) and three (efficiency) across all input conditions. Therefore, further analysis was based on data from block four to eight for time, and three to eight for efficiency.

Because also the calculated efficiency ratios were not normally distributed, I again conducted a Friedman test with input condition as factor. Post hoc analysis results (Wilcoxon signed-rank tests, Bonferroni-corrected with the significance level adapted to $p < 0.017$) indicated a significant difference between *IDT2* (1.83, SD = .58) and *IDT* (2.14, SD = .63) ($U = -2.766$, $p = 0.006$) as well as *DT* (2.39, SD = .98)($U = -3.245$, $p = 0.001$), but not between *DT* and *IDT* ($U = -1.263$, $p = 0.207$).

A repeated measures ANOVA with input condition as factor and time as dependent variable (normally distributed, log10(time) because raw data was positively skewed) showed no significant differences ($F(2,34) = 0.896$, $p = 0.417$) between *DT* (30.31 s, SD = 9.98), *IDT* (28.74 s, SD = 8.22) and *IDT2* (27.49 s, SD = 9.17).

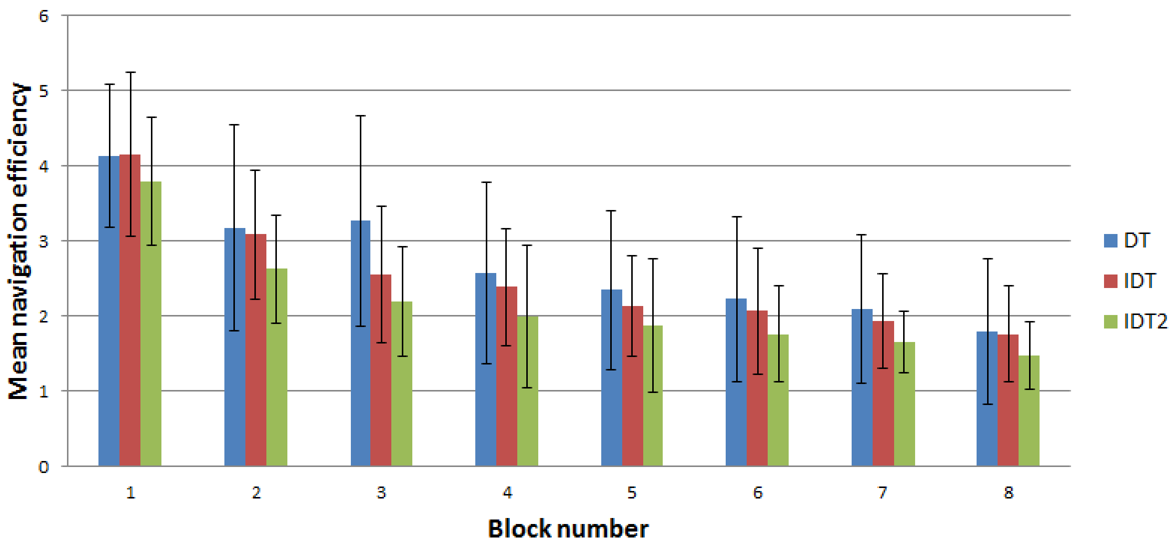


Figure 7.9: Navigation efficiency per input condition based on panning distance (error bars indicate standard deviation).

Participants' Personal Preferences Asked for their subjective preferences, 14 of the 18 participants said that they preferred *IDT*, two preferred *DT* and one *IDT2*. The most important reason given for the preference of *IDT* was the improved overview compared to the direct touch condition due to resolved occlusion (n=12).

Discussion and Limitations

As I did not observe significant differences for spatial recall performance, there is little to say about potential effects of input area form factors on spatial memory performance. At least, the results indicate that a spatial separation of touch input and visual output will not per se result in a decreased spatial memory or navigation performance in a 2D panning task. In particular, they indicate that the effect of haptic and kinesthetic cues provided by touch gestures and spatial memory performance might not depend on a spatial coupling between the input movement and the visual feedback, at least as long as a direct logical coupling between them is maintained.

The positive effect of the smaller input surface size on navigation efficiency was surprising, as using a linear non-gained transfer function required more clutching and constrained the panning movement length. Hence, while Nancel et al. (2015) compared different transfer functions employed with a fixed track pad size, my observations indicate similar trends for a fixed transfer function and varied input area sizes. However, the efficiency gain is not reflected in subjective ratings of the participants, who generally preferred the larger input area size provided by *IDT*.

Also, as Jetter et al. (2012) did not observe spatial memory performance gains for direct touch when zooming was involved in addition to panning, it remains an open question how the introduction of faster and non-linear transfer functions affects recall performance for indirect touch panning. Interestingly, the error rates (error rate as a function of accuracy) observed during the spatial recall task were higher as the ones reported by Jetter et al. (2012), which could be due to the difference in display size.

While trying to replicate the study design of Jetter et al. (2012), who compared only two input conditions, I did not strictly control task difficulty (i.e. the different item configurations), as this would have resulted in too many conditions. Still, the time data indicates that the increased navigation efficiency observed with *IDT2* does not necessarily go along with increased task completion times.

Outlook

While the spatial separation of input and output surfaces was restricted to the display plane in our experiment, future work could systematically explore the effect of specific properties of spatial indirectness, such as angles or distances. This would be especially interesting in the *Tool Space* context, which features differently aligned input and output surfaces. In general, understanding the influence of indirect touch properties and input area form factors on human performance and cognition is crucial to inform the design of both virtual and physical input devices. Therefore, future work could explore indirect control-to-display mappings for 2D panning tasks more systematically, for example by looking at conditions with various input sizes and form factors. Also, the effect of different proprioceptive cues could be explored more systematically, for instance by clearly distinguishing between haptic and kinesthetic cues (e.g., by implementing a hovering condition). A further venue for future work is the observed discrepancy between objective measurements and subjective preferences, which poses further questions: *Why did the participants prefer the larger input area?* and *What are the exact reasons for the observed differences in navigation performance?*

7.2.3 Experiment 2 - Evaluating the Impact of a Non-linear Transfer Function

Based on the experiment described in the previous section, I conducted a second experiment to explore the influence of *non-linearly gained* transfer functions on navigation and spatial memory performance in an indirect and relative touch panning task. In particular, I present observations from a between-subjects user study that – following the methodology presented in Tan et al. (2002), Jetter et al. (2012) and the previous section – involved a 2D navigation and recall task. During the study, participants navigated an abstract 2D information space using a virtual touch pad with relative mapping, either with a linear or with a non-linear transfer function derived from related work.

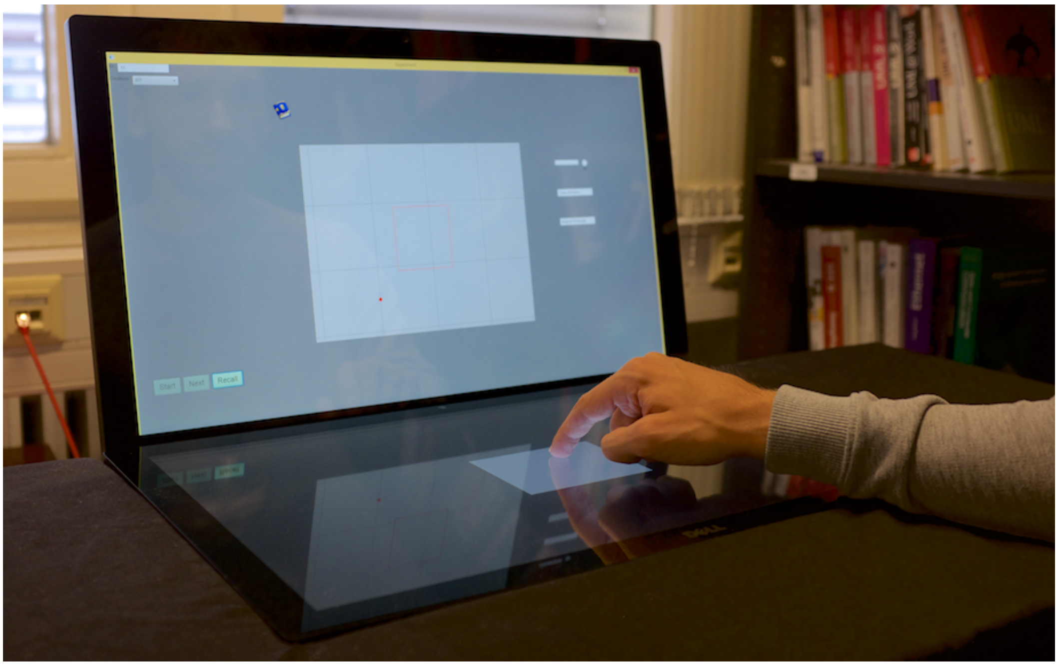


Figure 7.10: Our test setup: On the horizontal display, a virtual touch pad is used to navigate a view shown on the vertical display.

Non-linear transfer functions (TFs) are a powerful tool in indirect input styles: the input sensed in one domain is modified to better serve the requirements of another domain. For instance, mouse movements translate to screen space in a non-linear fashion, allowing to perform both slow accurate and fast coarse cursor movements with fine-graded mouse movements. In the context of spatial activation with indirect touch, TFs have not been explored systematically so far, in particular with respect to the specific requirements that touch movements pose for their design, as well as potential effects beyond task completion time and errors.

I formulated the following research questions:

RQ1 How will a non-linear TF affect navigation performance?

RQ2 How will it affect item recall (i.e. spatial memory) performance?

Apparatus and Conditions

I used a pair of 23" touch screens by Dell (ST2340) with a resolution of 1920 x 1080 pixels, arranged in an angular fashion (see figure 7.10). In order to enable comfortable control with arms resting on the display plane, I prepared a construction with inlets for the horizontal screen, which was placed on a vertically adjustable table in order to provide normal seating conditions.

Again, the experiment asked participants to navigate a viewport (800 x 600 pixels, 212.2x159.1 mm) using one-finger scrolling gestures applied to a virtual touch pad. This time, the viewport was displayed on the vertical, and the touch pad on the horizontal screen. The viewport displayed a portion of a 12 x 9 grid with a total size of 2400 x 1800 pixels (636.2x477.2 mm). The center of the grid (4 x 3 cells) served as home position and did not contain items. Throughout the remaining 96 cells, the 18 graphical items used in the previous experiment were manually distributed (see figure 7.11 left). Based on the findings of the previous experiment, which implied a superior navigation efficiency with the small virtual touch pad, I used an input area size that equaled the viewport in aspect ratio and its size was uniformly scaled with the factor 0.5 (106.1x79.5 mm).

In the first touch input condition (*IDT*), the transfer function of the virtual touch pad was linear with a gain factor of 1, resulting in a one-to-one mapping between input and output movements. In the second condition (*IDTTF*), I introduced a non-linear transfer function that allowed to achieve large output movements with less clutching. Scrolling inertia was disabled in both conditions.

Figure 7.11: Left: the grid with the distribution of 18 graphical items and the highlighted empty home position. Right: detail of the view showing the red visual anchor to indicate the home position and the bounding box.

Deriving a Transfer Function

Indirect relative touch input is widely used through commercial track pads and device drivers that support non-linear transfer functions, however, they are not discussed much in research literature, and also the commercial implementations are not documented well enough to be accessible for the research community. Quinn et al. (2013) specified the resulting problems for researchers, who are faced with a lack of understanding, replication challenges and confound risks, and proposed a reverse-engineering approach to approximate the function definitions.

Further, there seems to be little evidence on the relation between non-linear transfer and spatial memory performance. The observed benefits of direct touch (Jetter et al., 2012; Tan et al., 2002) are ascribed to differences in the perception of body movements, but the introduction of a non-linear TF in turn affects input movements (e.g., less clutching, smaller movements). Also, Jetter et al. (2012) did not observe performance advantages in a task that in addition to panning involved zooming - an interaction technique involving non-linear transfer on the output side.

A further factor are modality-specific demands on TFs. For instance, Rutledge and Selker (1990) identified specific requirements for finger joystick pointing: their TF considered both the low precision and the high effort to keep a position steadily resulting from operating a tiny device with a single finger.

Similar characteristics are often attributed to finger input on touchscreens. Despite being aware of the difference between isometric joystick and touch input, I opted to implement a well-documented transfer function based on the one presented by Rutledge and Selker (1990):

$$v(f) = \begin{cases} gf^2 & \text{if } 0 \leq f \leq \frac{1}{2} \\ g(\frac{1}{2} - (1-f)^2)) & \text{if } \frac{1}{2} \leq f \leq 1 \\ 1 & \text{if } f > 1 \end{cases}$$

While f describes input signal deltas (in our case touch movement delta vectors, normalized with regard to the touch pad size), g is a gain factor. In Rutledge and Selker (1990), g is 2, but in informal tests with four lab members this value resulted in very subtle effects. Therefore, I chose a value of 4, which during the tests delivered a perceptible effect that according to the test subjects felt well-controllable and comparable to commercial implementations of transfer functions.

Navigation and Recall Tasks

Similar to the previous experiment, participants were first asked to repeatedly navigate through a fixed and predetermined sequence consisting of eight of the 18 items. The current search item was permanently displayed above the top left corner of the viewport until the item was found. In this manner, the sequence was repeated across 10 blocks (i.e. 10 times), resulting in 80 search trials. In contrast to the previous study, this time each trial started from the home position, which was marked with a subtle visual anchor, and ended as soon as the currently searched item was fully contained in the bounding box displayed in the center of the viewport (figure 7.11 right). Again, while in the beginning of the navigation task the item search was random, improving mental representations of item locations throughout the repetitions were assumed as a natural side effect. This short-term memorization of the item locations was intended to compare recall performance for item positions across participants. To prevent intensive memorization strategies, we asked to maximize for speed. We recorded input and output movement distances (equal for *IDT*), as well as task completion time per item search. Additionally, we collected unweighted Nasa TLX ratings, to gather a workload perspective on navigation performance.

Recall Task In the spatial recall task, a randomized version of the search sequence from the navigation task was shown to the participants and they had to place each item at its original position within the empty grid. The positioning was done with the arrow keys from a keyboard to prevent unconscious use of motor or kinesthetic memory. The participants did not receive any feedback indicating the placement accuracy and for every item they started over with an empty grid in the home position. We measured the placement error as the Euclidian distance in millimeters between the participants' item placement and its original position and asked for accurateness.

Procedure

In order to avoid learning or carry-over effects as well as differences in spatial item configurations that might confound the results, we designed a between-subjects experiment with input modality as independent variable with the levels *IDT*(with linear gain) and *IDTTF* (with non-linear gain). Therefore, each participant had to perform both the navigation and the recall task only once. In the beginning of the experiment, demographic data was collected. Then, there was a short introduction phase, during which the experimenter explained the navigation task. Subsequently, the data was collected during the actual navigation. The recall task was introduced and performed only upon finishing the navigation task, to prevent the application of learning strategies throughout the navigation task. Upon finishing navigation and recall tasks, the Nasa TLX workload assessments were collected.

Participants and Results

We recruited 25 participants (11 female) aged 20 to 35 (mean 24.52, SD = 4.23), all right-handed and using touch input devices (smart phones, tablets) on a regular basis. 14 were students (10 from a technical/computer science program), the others had various educational and professional backgrounds. 14 participants completed the condition *IDTTF*, eleven completed *IDT*. The unequal distribution of participants are due to an allocation error, yet gender was almost balanced in both groups (4/11 and 7/14).

Results for Navigation Performance

To examine the assumed learning effect and exclude search trials with high randomness, we first compared navigation efficiency and time across the 10 consecutive search blocks. Due to a non-normal distribution of the data (Shapiro-Wilk tests), we performed Friedman tests. Post hoc analysis results (Wilcoxon signed-rank tests, Bonferroni corrected with significance-level set at $p < 0.005$ as we compared 10 consecutive blocks) show that time and navigation efficiency improvements are not significant after block 3 across all input conditions. Therefore, further analysis is based on data from block four to ten. In particular, we used this data and performed independent t-tests to compare navigation distances and times, as well as recall errors between the two conditions.

Distances

Our study results indicate that the mean distance traveled with fingers per item was statistically significantly lower with *IDTTF* (448.53 mm, std = 269.94) compared to *IDT* (659.72

mm, std = 549.46), t(104.613) = -3.092, p = 0.003, corrected for unequal variances (Levene's test).

With regard to cursor distances, our results indicate that the mean distance traveled with the cursor per item was statistically significantly lower with *IDTTF* (497.54 mm, std = 316.812) compared to *IDT* (659.72 mm, std = 549.46), t(114.756) = -2.306, p = 0.023, corrected for unequal variances (Levene's test).

Time Regarding task completion time, our results indicate that the mean search time per item was statistically significantly lower with *IDTTF* (4161.10 ms, std = 2510.73) compared to *IDT* (5733.90 ms, std = 4962.23), t(106.280) = -2.538, p = 0.013, corrected for unequal variances (Levene's test).

Workload The RTLX results indicate a slightly lower workload for *IDTTF* (38.31, std = 12.66) compared to *IDT* (42.00, std = 8.93).

Results for Spatial Memory

The study results indicate that participants had a statistically significantly lower recall error (59.96 mm, std = 17.44) with *IDTTF* compared to *IDT* (120.37 mm, std = 63.11), t(11.295) = -3.077, p = 0.01, corrected for unequal variances (Levene's test).

Discussion

The most surprising finding of our experiment was the significantly lower recall error rate observed with the transfer function, which adds an aspect of scale to the discussion about embodiment, kinesthetics and spatial memory: while improved spatial memory performance for touch input over indirect input has been observed in comparisons involving rather large touch input spaces (Jetter et al., 2012; Tan et al., 2002), this effect might be reversed for small input spaces: here, the transfer function as a classic mediator of indirect input styles not only seems to yield ergonomic benefits, but to free cognitive resources that facilitate the formations of mental representations of 2D information spaces.

While the choice of transfer function was preliminary and certainly leaves room for optimization, the navigation performance results of our experiment confirm its assumed benefits: the significantly lower finger movement distances with *IDTTF* hint at a higher *physical* navigation efficiency, which is also reflected in the lower Nasa TLX workload score. I did not measure the clutching rate, as it is not a well-understood measure for input efficiency (Nancel et al., 2015)).

Further, the transfer function led to significantly lower cursor movements and mean navigation times per item, indicating efficiency benefits beyond physical movement. The improved *virtual* navigation efficiency (i.e. grid movement) may be due to a reduced physical effort necessary to cover large distances, which may have reinforced movement planning strategies.

On the one hand, these findings contribute to the understanding of non-linear transfer functions in general, as their effect on spatial memory has – at least to our best knowledge – not been investigated previously. On the other hand, our findings can inform designers of indirect touch input techniques, who are faced with questions of adequate mappings and form factors.

Limitations and Future Work

The observations are based on a quite basic transfer function derived from a isometric joystick pointing device. While it yielded the desired benefits in terms of increased physical navigation efficiency, a systematic exploration of transfer functions specifically targeted at indirect touch input may unveil more adequate functions.

Further, the choice of the virtual touch pad size was based on findings from the previous experiment. However, as input area size and navigation efficiency seem to interact, I assume that transfer functions should be systematically explored across different touch pad sizes.

An interesting question for future work seems to be the comparison of non-linear transfer functions with alternative scrolling support techniques. In particular, inertia is a touch property supported by many interactive surfaces that is used for similar purposes. While we disabled it, systematically exploring its role in indirect touch input techniques may yield interesting findings. Further, I propose a generalization of non-linear transfer functions to the established touch gesture vocabulary, such as two-finger rotation, pinch-to-zoom, which are an integral part of the *Tool Space* concept.

7.2.4 Chapter Summary

In this section, I have presented experimental work aimed at informing the design of virtual touch tools. The presented experiments explore the effect of *touch indirectness* and *non-linear transfer functions* on a 2D navigation an recall task. Based on related research, which has identified potential benefits of touch input over mouse input with regard to spatial memory performance, I incorporated this aspect into my experiments. Following an experimental procedure proposed in related research, I did not only measure task completion times and navigation efficiency, but also assessed the recall of information space features.

On the one hand, the results suggest that decreasing the size of the input surface in the indirect condition increases the navigation efficiency. On the other hand, I did not observe a decreased spatial memory performance when touch input gestures and visual display were separate. However, more studies – in particular with more participants – are necessary to substantiate this potential effect.

Further, compared to a condition with a linear transfer function, a non-linear function derived from related work led to rather expected effects, such as reduced finger navigation distances, navigation times and workload, but also to less predictable findings, such as an significantly improved spatial memory performance.

Review The work presented in this section taps into an interesting field of future research opportunities aimed at informing the design of virtual tools. Next to proposing an experimental procedure that incorporates spatial memory assessment, the findings imply that for 2D navigation, *increasing the size of virtual touch pads may not yield an increased performance*, and that *employing non-linear transfer functions in virtual input devices may positively influence spatial memory performance*.

In general, the findings of the experiments present important learnings for the design of *Tool Space* instances. On the one hand, the rationale of employing *two-handed* multi-touch input has been substantiated with experimental results, indicating that both hands are equally capable of performing two-handed touch input gestures. On the other hand, the observations on input area form factors and transfer functions in the context of 2D navigation provide useful insights for the conception, design and evaluation of virtual tools.

V

"REFLECTIONS ON THE TOOL SPACE"

8
Conclusion and Outlook

The main goal of this thesis was to gather a better understanding of interaction with computing workspaces incorporating both upright and horizontal interactive surfaces. In the beginning of my work, I was confronted with the *Curve* (Wimmer et al., 2010), a prototype that had been created – I suppose – more out of playfulness and feasibility than with clear objectives in mind. Much of the initial thoughts on the *Curve*, and also on the related prototype *BendDesk* (Weiss et al., 2010), evolved around the effects of a seamless curved connection on basic interaction techniques (e.g., Voelker et al. (2012), Weiss et al. (2010), Hennecke et al. (2013)), and on novel visualization techniques (e.g., (Schwarz et al., 2012)).

In a first phase of my work, I tried to apply these findings in search for suitable application contexts: in various explorations, I created and evaluated diverse interactive prototypes (see section 4), trying to find a *killer application* for this particular display shape. However, during this time, my focus shifted gradually towards the more basic question of establishing adequate interaction paradigms for *dual-surface* interfaces. While there are notions of competing paradigms in the related work, which for instance result in work on *switching between input styles* (see section 2.3.2), I found that little work focused on the question how novel workspaces based on interactive surfaces can embrace existing desktop computing ecosystems and interaction paradigms instead of merely adding an alternative paradigm of *direct interaction*.

Therefore, I developed the idea of a *Tool Space*, which instead of *immediate input* tries to achieve directness through *novel mediators*, i.e. task-specific virtual input devices that allow spatial activation of commands (see section 5.1.2). At a late stage of my work, I found a conceptual precursor to my ideas in Bill Buxton's ideas of partitioning a touch-sensitive surface into several areas, serving specific functions in specific contexts. I implemented and evaluated the *Tool Space* concept in two projects, and the feedback from initial user studies was largely positive.

Eventually, in order to inform the design of the *Tool Space*, I conducted a series of experiments aimed at understanding how to conceive and design the virtual tools. In order to do this, I did not only focus on established performance measures, but tried to incorporate also measures that yield superior results in comparisons of direct touch and WIMP-systems.

In the following, I will sum up my findings, discuss limitations of my conceptual and empirical results, and suggest areas of future research.

8.1 Summary of Contributions

In this section I will recap the contributions of this thesis, which are threefold:

8.1.1 Understanding the Requirements of Personal Dual-Display Interaction

Section 4 presented an excerpt of explorative projects and formative evaluations, which were triggered by the goal to find application contexts that can benefit from the *Curve's* display shape. Each of the projects were characterized by identifying potential contexts (e.g., audio editing, 3D interaction etc.), implementing an interactive prototype, and evaluating the prototype. Also, some of the application prototypes fueled subsequent explorations of specific interaction techniques, for instance targeted at cross-display object movement.

Trade-Offs A key finding of this phase was that the strict application of direct touch principles in this context led to the definition of specific trade-offs (see section 4.3). Direct touch frameworks, such as MT4J, often make it easy to add gesture input to user interface objects. Yet, designing user interfaces for large non-flat surfaces is non-trivial, as ergonomics imply preferences for the locations of manual operations and prolonged visual attention, as well as for the direction of cross-display movements.

The *presentation trade-off* is exemplified with a quiz application built upon the idea of *dragging objects through the curve* (see section 4.1.1): on the one hand, tasks with a clear distinction of input areas and output areas, as well as real-world analogies might imply a certain spatial distribution of user interface elements. On the other hand, the replacement of multi-window ensembles with a juxtaposition of always visible interactive areas (an idea predating the concept of virtual windows/viewports (Engelbart and English, 1968), proposed also in the *Starfire*-video (Tognazzini, 1994)) requires decisions about placement, which are currently not well-understood.

The placement question is directly related to the *input technique trade-off*: throughout the case studies, I explored different touch input techniques. *Literal* techniques that span all display areas, such as direct dragging are easily detected and precise, but they quickly cause fatigue. For spontaneous short-time interaction, such as public installations, these techniques work well, but for prolonged work, *indirect techniques* need to be considered to allow for comfortable interaction. In this regard, I have started to understand the horizontal surface as a *design space for virtual input devices*.

Finally, the *interaction style trade-off* embraces the other two trade-offs and highlights the need to better understand the clash of WIMP and direct touch paradigms that can occur when interactive surface technology is integrated into desktop computing environments. An adequate interaction model for *desktop surface computing* should consider the potential benefits of direct touch interaction, such as its increased input bandwidth, its support for two-handed interaction, and embed them in a matured paradigm.

A notion of virtual input devices During the exploration phase, I iteratively developed the idea of virtual input devices. While in the case of the quiz game, I informally played around with this idea after having experimented with and having evaluated direct techniques before (see figure 8.1), it was also present in the comparison of cross-display interaction techniques, where the implementation of *world in miniature* and *hold-and-point* (see section 4.1.2) were conceptually similar to virtual track pads and yielded superior performance in the experiment.

From a technical point of view, virtual input devices can be understood as proxy-objects which register touch and gesture event listeners, but transmit input signals to arbitrary outlets (see section 5.1.2).

(a) (b)

Figure 8.1: Direct *literal* technique vs. indirect technique for dragging images across the curved display connection: (a) literal techniques are easily detected and understood, but cause arm fatigue. (b) Exploration of an indirect technique: tapping and holding an image for a certain dwell time transforms it into a virtual track pad that allows indirect image movement.

Application contexts A secondary finding from the exploration of various application contexts was that the domain of 3D interaction is well suited to explore novel interaction models for dual-surface workspaces. Not only are tasks related to 3D modeling of growing interest due to advancements in personal fabrication technologies – interestingly, the HP Sprout [1] is a dual-surface device particularly addressing this trend– but 3D interaction is notoriously complex. The mismatch between the degrees of freedom required and provided by established pointing devices, has yielded complex UIs and a variety of specialized input devices.

In a first experiment on virtual input devices, I *transferred an established 3D widget's functionality into the input device space* (see section 4.2) by using both display orientation and visual structures. Study results were encouraging, as the virtual input devices yielded a

[1] http://www8.hp.com/us/en/sprout/home.html

similar performance compared to the mature pointing device, and was preferred by the participants.

Review In summary, the case studies introduce novel techniques suitable for spontaneous interaction, such as the quiz game's answering technique (see section 4.1.1) or the novel angular touch-pad for 3D interaction (see section 4.2). However, they also highlight particular challenges involved in the design for prolonged interaction with more complex applications – a typical scenario in the context of personal desktop computing. The projects indicate that the clash of interaction paradigms (WIMP vs. NUI) will not bear a clear winner. Rather, a thorough reinterpretation of the particular strengths and a careful consolidation will be necessary to create adequate user interfaces.

8.1.2 Conceptualizing a Novel Interaction Paradigm

After the exploration phase has highlighted the need to understand interaction in a dual-display environment on a fundamental level and my first explorations with virtual input devices yielded encouraging results, I formalized the idea of virtual input devices in the *Tool Space* concept. On a theoretical level, the presence of *tools as mediators* and schemes of *spatial activation* and *reification* (see section 5.1.1) have – in addition to practical insights about ergonomics – contributed to the notion of maintaining separate physical spaces for input and output and focusing on novel opportunities for the design of indirect touch tools.

Novel tools Conceptually, the *Tool Space* evolves in the desktop plane. As such, it can be based on interactive tabletops or composed of smaller flat touch-screen devices positioned on the desk. The touchscreen's capabilities are used to dynamically render *task-specific tool sets* that consist of visible areas, suitable for multi-finger and potentially two-handed input. It is important that my conception of tools is not based on *mimicry* of analog tools, as exemplified for instance by *TouchTools* (Harrison et al., 2014). Instead, it is a generalization of the well-established track pad by means of touchscreen technology. This idea has a conceptual precursor in Bill Buxton's *templates for touchscreens* and his subsequent idea of a *window manager for touch tablets* (Buxton et al., 1985), and is now extended with display capability and multi-touch sensing. On the one hand, this allows for a *more complex tool composition*, such as nested touch areas for both integral and separate control of parameters (see section 5.1.2). On the other hand, by employing multi-touch devices, the tools now allow for an *increased input bandwidth*.

Conceptually, the *Tool Space* is meant to extend existing hardware and software ecologies, such as proposed by Bi et al. (2011). In particular, the rationale is neither to replace existing input styles, nor to promote a strict habit of switching (such as proposed by Arai et al. (1995)). Instead, the idea is to provide an *enriched input vocabulary for complex tasks*. I tried to incorporate this notion into my explorations and evaluations by embedding my case studies and experiments in different hardware setups. While much of my early work has been created within the context of the *Curve*, I later used an angular display setup consisting of two identical touchscreens, as well as a set of tablet devices positioned on the sides of a hardware text keyboard.

Interplay between tool design and domain Many *proof of concept implementations* use simple applications, such as the image viewer, which is a popular theme in the context of interactive surfaces. This simplicity has advantages: it elegantly complements the advertised ease-of-use, but it also facilitates comparative studies with regard to both task abstraction and the choice of suitable baselines.

Yet, with regard to the demonstration and evaluation of the *Tool Space* concept, *facilitation outweighs simplicity*: the transfer of GUI widgets into dynamic and directly accessible handles needed to be explored within a domain complex enough to reveal potential benefits. Having explored 3D manipulation in an early case study (see section 4.2), I identified 3D modeling as a promising area, as it is a popular application area notorious for complex user interfaces and steep learning curves.

In order to build a *Tool Space* for 3D modeling, I pondered between developing a plug-in to integrate multi-touch interaction into existing desktop software (an approach exemplified by *sketchupmultitouch* [2]), and developing my own software. Here, the insight that plug-in APIs do not allow to bypass basic application logic catered to single-pointer input was a decisive factor in favor of implementing my own software.

The implementation of the software has been challenging, but also rewarding, both on a personal and factual level as the freedom to reinterpret relations between input signals and domain actions has not only yielded novel high-bandwidth interaction techniques and valuable insights from user feedback sessions, but also led to minor technical contributions to the specific domain (Palleis et al., 2015b).

User feedback However, creating a custom testbed also eludes comparability, as it yields interaction techniques that do not have an established baseline. One approach to allow for comparability is to *identify suitable subtasks* that both provide a certain level of complexity and can be easily supported by different input modalities. In section 6.2, I describe audio editing as a potentially adequate task.

With this *comparability issue* in mind, I opted to gather mostly qualitative feedback. With the 3D modeling *Tool Space*, involving expert users first helped to identify its particular opportunities, resulting in a subsequent multi-session study with novice users.

In general, the feedback was encouraging and confirmed some of my underlying assumptions. On the one hand, the *partitioning character* of the dual-display was well-received and the proposed interaction techniques induced a *hands-on feeling* despite the indirectness of input. On the other hand, especially the multi-session study (see section 6.1.6) revealed positive feedback concerning *unexptectedly good learnability and memorability*.

The feedback sessions also yielded interesting insights on two-handed input: while participants did not struggle with it when it was enforced, they did not adopt it when it was

[2] `https://github.com/nuigroup/sketch-up-multitouch`

optional. Hence, the *Tool Space* allows to employ two-handed input both as means to design *high-bandwidth cooperative input techniques*, and as a way to *progress ones abilities over time*.

> **Review** In summary, the initial insights are in line with the notions of *natural user interfaces* (see section 2.2.4) and suggest that beneficial novel touch interaction techniques in dual-surface contexts can be based on the concept of separating manual input and visual focus, as well as on the idea of *artificial tools*, which are conceptualized as dynamic and task-specific input areas designed for multi-finger and two-handed input.

8.1.3 Informing the Design of the *Tool Space*

In a last phase of my work, I conducted several experiments to inform the design of virtual tools. On the one hand, I was particularly interested in our ability to perform two-handed multi-touch input techniques. On the other hand, I started to explore the influence of size of input areas and mappings onto 2D navigation – a common task in graphical user interfaces.

Two-handed multi-touch There is a rich history of research on two-handed input. Guiard (1987) contributed an influential model on how our hands cooperate as individual motors integrated into a kinematic chain, and Buxton and Myers (1986) have early argued for and substantiated the use of both hands in input techniques. However, with the increase of input bandwidth through multi-finger input, some of the early conceptions get challenged, because two points of input can now easily be provided by one hand, and sometimes unimanual two-finger input should be preferred over bimanual one-finger input (Moscovich and Hughes, 2008).

As two-handed input is one aspect of the *Tool Space*, I conducted an experiment to explore our abilities to cooperatively use both hands for multi-touch tasks, i.e. tasks that support common one- and two-finger gestures, such as rotation, scaling and translation (RST) (see section 7.1). The study employed a rectangle docking task, which was already used by Buxton in his experiment on two-handed input (Buxton and Myers, 1986). On the one hand, our results indicate that the *established multi-touch gestures involved in the study (i.e. pinch, zoom, rotation) are performed equally well with the dominant and non-dominant hand*. On the other, *two-handed input improves task completion times with multi-touch RST* and the results imply that for rectangle docking, *the cooperation of hands conform to Guiard's model* (Guiard, 1987).

This has implications for the design of virtual tools, which can explicitly involve tool compositions based on two-handed multi-touch input (e.g., the *edge-loop scaling tool* in section 6.1.4), but also employ two-handedness as a natural feature of spatial activation that enables a "virtuoso" (p. 12) experience from novice to expert (Wigdor and Wixon, 2011), such as observed with the *extrusion tool* (see sections 2.2.4 and 6.1.4).

Impact of input area size Reserving touchscreen real estate for rendering abstract input areas raises questions about form factors. In order to start exploring the effect of input area form factors on interaction properties, I conducted an experiment aimed at comparing varying input areas – including a direct touch baseline – during a 2D navigation task performed with one-finger panning operations. Next to traditional performance measures such as navigation time and efficiency, I intended to also capture if touch indirectness preserves some of the known benefits of direct touch. In particular, earlier studies had indicated an increased spatial memory performance for direct touch systems compared to indirect mouse input, which has been explained with the kinesthetic cues provided by touch interaction (Jetter et al., 2012; Tan et al., 2002).

The experiment was inspired by the one presented by Jetter et al. (2012) and the results imply that *decreasing the input area with regard to the size of the navigated information space improves navigation efficiency*. However, the results of this study did not indicate any effects on spatial memory performance, so that further studies are required to substantiate the assumption that indirect touch preserves this distinct advantage over mouse input. In general, the results imply that *bigger is better does not always apply to indirect touch input areas* (see section 7.2.2).

Impact of non-linear transfer Besides form factors, the indirectness of the *Tool Space* also allows to reintroduce non-linear transfer functions. In particular, I explored the use of non-linear functions of the input signals as parameters for application commands. While such functions are a key feature of many pointing devices (e.g., mice, track pads), little is known about their use with spatial activation, which is based on a direct coupling of input gestures and command invocation. An early exploration of non-linear gain was the use of a discrete gain changing function in the *edge-loop scaling* tool (see section 6.1.4).

In an experiment based on the task and methodology described in the previous paragraph, I compared two virtual input areas – one with a linear gain and one with a non-linear transfer function derived from related work. On the one hand, the results indicate the *expectable benefits of a non-linear function* with regard to navigation time and efficiency. However, they also suggest that its use *yields an increased spatial memory performance* (see section 7.2.3). While this was only a preliminary study with a non-optimal transfer function, it shows the potential to advance indirect touch techniques by investigating transfer functions.

> **Review** The studies conducted to inform the design of virtual tools (see chapter 7) outline interesting future research opportunities: the results suggest to actively explore two-handed multi-touch input techniques, and to think about form factors and transfer functions, as they can have significant effects on the performance of standard tasks.

8.2 Limitations

The main limitations of my work are the small amount of participants in the *Tool Space* evaluation, misconceptions resulting from the specific application contexts, and the subsequent limited generalizability of the findings. The following sections provide detailed discussions of these limitations. Finally, the terminology I used throughout this thesis to describe *Tool Space* contexts is briefly reviewed.

Small Amount of Participants

A main limitation of the work presented in this thesis is the small amount of participants involved in user feedback sessions, especially with regard to the case studies and the conceptual work. One reason for this is a lack of formalization that led to quite a lot of small studies which have not been included into this thesis, but nevertheless contributed to my understanding of the domain. However, more importantly, the lack of a baseline or alternative designs at a later stage has provided valuable and mostly positive feedback, but juggling with the conception of a paradigm, implementing complex software and designing novel interaction techniques at the same time has not always provided a solid framework for well-defined formal experiments. I tried to integrate an aspect of formal evaluation by isolating aspects which may inform the design of virtual tools at a late stage of my thesis work, yet I believe that there would have been also ways to better formalize and evaluate the concept itself (see section 8.3).

Designing prototypes for specific application contexts also requires study participants that reflect the respective target audience. Evaluating the concept with domain experts, I identified novices as a suitable target group for the 3D modeling use case, therefore, there is little I can say about its potential for other user groups, for instance domain experts.

Regarding the studies involving spatial recall performance, the experiment design involves a tricky trade-off: while within-subject designs can be based on a smaller total number of participants, they must involve different search tasks to prevent carry-over learning effects. To circumvent this trade-off in a follow-up study, I opted for a between subjects design. However, for studies involving spatial memory performance, these groups need to be homogeneously composed and sufficiently large. For future designs, I would therefore recommend within-subject designs.

Focus on Specific Domains

Designing for specific application contexts also involves the danger to be misunderstood. Experts will focus on missing features during interviews, and reviewers construe the conceptual work as attempts to contribute domain innovations. I have experienced this a lot with my work on the 3D modeling *Tool Space*, and I understand it: the urge to publish in quick cycles has made *system design* a complicated undertaking because it takes time and effort to build, and it turned out to be difficult to communicate the worth of this effort.

Generalizability of Findings

Because of the aforementioned issues, the generalizability of the findings is limited. Furthermore, the feedback sessions and experiments were conducted mainly with young people. In addition, the variety of use cases and tasks was limited: I explored 3D modeling and audio editing, and the abstract tasks include cross-display object movement, rectangle docking, and 2D panning. Nevertheless, I believe that the work presented in this thesis is worth considering when designing indirect touch interaction techniques, and I outline useful future work in the next section.

Tool Space Terminology

During the course of this thesis, I mostly considered my work to fall into the category of *indirect touch input techniques*. Yet, as I have shown in 2.2.1, the property *directness* is overloaded in the context of human-computer-interaction and only insufficiently characterizes the idea of the *Tool Space*. This has made me think of terms like *indirect touch widgets*, or *virtual input devices* – as opposed to the existing term *logical device* (e.g., Beaudouin-Lafon (1998)), which implies a physical counterpart.

A further conceptual difficulty is the relation of the *Tool Space* to physical surfaces. While I have used the terms *dual-surface*, *dual-display*, or *multi-surface* to describe setups that integrate horizontal and upright touchscreens, one *Tool Space* might also be distributed across a pair of tablets (as in section 7.1).

8.3 Future Work

Based on the work presented in this thesis and related research, there are manifold ways to proceed with the work on the *Tool Space*. In addition to the specific *Future Work* sections of the previous chapters, I will discuss suitable application contexts and suggest further follow-up studies. Eventually, I will outline directions for future research that seem promising.

8.3.1 Application Contexts

The presented use cases of the *Tool Space* were situated in the context of *creative desktop applications*, in particular 3D modeling and audio editing. The feedback on the 3D modeling *Tool Space* has indicated its potential to stimulate exploration, which suggests an application in contexts where playfulness and experimentation are more important than precision and time efficiency. This might apply well to educational contexts, such as high school art classes, where the experimentation with different artistic media suggests the engagement with 3D modeling, but teaching curricula do not provide the time necessary to master expert tools.

The focus on specific domains inevitably evokes the question for alternative application contexts. While I argued that the *Tool Space* may be well-suited for complex tasks that involve many degrees of freedom, a variety of modes and lengthy input sequences, my concepts are points in a large design space and can therefore only speak for themselves. A more comprehensive notion of *Tool Space* opportunities would require a systematic task analysis, asking: *What kind of general task complexities can be resolved best with a Tool Space approach?* This is typically the case with many 3D-related tasks, but also with timeline-based interaction, such as key framing or drawing envelops over audio or video tracks.

8.3.2 Suggested Follow-up Studies

A first suggested study is a long-term field evaluation of the *Tool Space* for 3D modeling in a high school art class, including a control group working with an application like Google *SketchUp*, which tries to simplify 3D modeling. Such a study could be based on a joint topic, involve observations and interviews, and modeling results could be assessed by a panel of domain experts. Here, the focus should not primarily be on usability, but on the effects of the system on the creative process.

On the other hand, both tools developed for the 3D modeling *Tool Space*, i.e. the *edge-loop scaling* and the *extrusion* tool could be more formally compared to established input styles. As the source code for the entire application is public [3], adding support for mouse and keyboard as well as direct touch input is feasible. Suitable interaction techniques for these modalities can be informed both by established software and by related research. Such experiments could substantiate our observations that indirect touch input mediated through a *Tool Space* allows for high-precision and fast input.

During all of the studies I observed that despite the lack of haptic feedback, participants were able to operate eyes-free most of time. This observation is also in line with Buxton's informal findings that physical bevels through cardboard templates are not necessary to operate eyes-free with indirect touch tools. Yet, these observations could be substantiated with empirical insights on the relation between form factors, number and location of touch input areas on the one hand, and hand-eye coordination on the other, which could be quantified with traditional performance measures (e.g., task completion time, input errors) and complemented with eye-tracking data. Data could be collected for instance using an extended two-handed docking task (such as in section 7.1), involving a large information space, in order to involve both docking and navigation.

With regard to the studies presented in section 7, I think that especially further explorations of the effect of input area size in combination with non-linear transfer functions is interesting. For instance, little is known about how non-linear transfer functions affect indirect touch input with spatial activation, i.e. a direct coupling of touch input signals and visually offset 2D panning operations: while WIMP systems always require a co-control of click and drag operations for such tasks, direct touch user interfaces typically employ a linear transfer function and use concepts like *inertia* to ease scrolling. As related work on physical track pads indicates that some of our conceptions about high-gained transfer functions might be unsubstantiated (Nancel et al., 2015), and my own experiments indicate that input area sizes affect performance for linearly gained transfer functions, more work is necessary to better

[3] `http://www.medien.ifi.lmu.de/team/henri.palleis/projects/fad/`

understand these relations. The testbed I created for my studies is well suited to explore 2D navigation tasks while easily varying input area form factors and mappings.

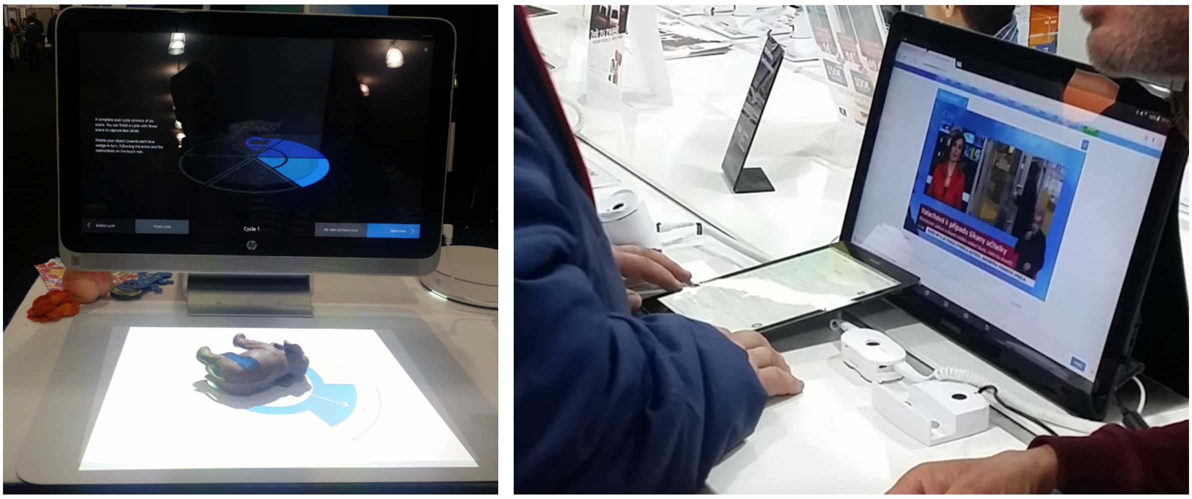
8.3.3 Directions for Future Research

While the ideas presented in this thesis are focused on the *Tool Space* as an interaction paradigm, there is interesting related research proposing gaze-based switching between direct and indirect touch modes (Pfeuffer et al., 2015, 2016; Voelker et al., 2015). The idea of switching had been proposed already by Arai et al. (1995), and I think that the *Tool Space* does not contradict it. In fact, its dynamic character is one of its core features, and a *Tool Space* might be *one of several input styles in indirect touch modes*. With this in mind, a consolidation of rather different interaction paradigms, all having their particular strengths and downsides, seems feasible.

A related opportunity for further research is concerned with the *presentation trade-off* (see section 4.1.1). As I have observed a demonstration of the HP Sprout at a conference coffee break (see figure 8.2(a)), I noticed that on the one hand, many GUI elements are in fact redundantly rendered on both displays, and that the demonstrator mostly used the mouse to control rather classic GUIs on the upright display. I found that observation interesting, because carefully considering the characteristics of both WIMP and direct interaction may result in even more appealing user interfaces. For instance, a better understanding of how to *share domain objects between virtual windows and direct access areas* could enrich our possibilities to interact using our hands. I have started to explore this notion in the *extrusion tool*, where I employed a second interactive view of the domain object as a *polygon selection tool* that allowed finger-drawing selection of polygons. Yet, a more systematic assessment of how to combine different presentation styles will help building adequate interaction techniques for dual-surface workspaces.

A third venue for further research might explore the potential of the *Tool Space* to support co-located collaboration. Existing input devices are tailored to single-person usage. In the same way as interactive surfaces lend themselves to naturally support two-handed input, they may affect both access and control characteristics of people sitting next to each other in front of a computer.

Lastly, thinking about ad-hoc *Tool Spaces* might be a promising endeavor. With an ever increasing amount of touch-enabled devices – even personal touchscreen ecologies often comprise smartphones, tablets, and monitors –, we should ask the question how these interactive surfaces can form novel connections. Not long ago, I observed a random person “connecting” his personal tablet device to a monitor device on sale (see figure 8.2(b)). While such a particular case may not be overstated, it nevertheless emphasizes a potential to use existing hardware in order to support temporary configurations for enriched interaction, such as using personal tablet devices as additional input channels in desktop computing work.



(a) HP Sprout (b) Adhoc dual-surface computing

Figure 8.2: Venues for future research: (a) How should domain objects be presented in a dual-surface workspace? (b) How can existing device ecologies be exploited to form temporary workspaces?

8.4 Closing Remarks

Over the course of this thesis, I presented my experiences with building indirect touch interaction techniques for dual-surface devices, which over time has shaped my interest for thinking about adequate interaction styles for desktop surface computing. By presenting my theoretical, practical and empirical work, I hope to inspire others who are interested in designing interaction techniques in similar contexts.

Growing up with little attention to the world of computing, I became overwhelmed with the empowerment I experienced through learning how to implement interactive computer programs during my university education. Today, I understand prototyping as a sort of sketching, and making meaning with sketches is an important theme of my work. The journey towards my dissertation is also characterized by my hands-on sketches which often helped me to get a *feeling* for specific challenges. I did not hesitate to build rather complex prototypes in order to create testbeds, which then could help me to formalize ideas. Sometimes, this approach of turning intuitions into sketches has challenged advisors and colleagues, as the objectives were not always clearly defined. And while it sometimes caused frustration to spend most of the time on identifying rather than on solving problems, it was worth the effort and resulted in valuable contributions.

The *Tool Space* seems inadequate at first glance: it reintroduces artificial input devices and indirect touch input seems to increase visual task load due to the lack of haptic feedback. Yet, initial user feedback sessions indicate that it might have its place and current product trends show related concepts. For instance, the Lenovo *Yoga Book* has a large touch pad with display capability instead of a hardware keyboard: text keyboard and cursor track pad with vibro-tactile feedback are displayed only on demand, otherwise the surface acts as pen-operated graphic tablet. This notion of on-demand input devices could easily be extended to more specific tools that adapt to the application context – a *Tool Space*.

The virtuality of such tools removes some of the constraints that apply to the form factors of physical input devices: physical track pads have a predefined size and location, as well as a fixed mapping. With my studies on informing the *Tool Space* design, I have shown that these variables affect well-established tasks. A better understanding of the impact of form factors and mappings on common tasks bears the potential to both enrich existing input styles with novel tools and inform the design of virtual input devices in general.

BIBLIOGRAPHY

Bengt Ahlström, Sören Lenman, and Thomas Marmolin. 1992. Overcoming Touchscreen User Fatigue by Workplace Design. In *Posters and Short Talks of the 1992 SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 101–102. DOI:http://dx.doi.org/10.1145/1125021.1125103

Pär-Anders Albinsson and Shumin Zhai. 2003. High Precision Touch Screen Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 105–112. DOI:http://dx.doi.org/10.1145/642611.642631

Toshifumi Arai, Kimiyoshi Machii, Soshiro Kuzunuki, and Hiroshi Shojima. 1995. InteractiveDESK: A Computer-augmented Desk Which Responds to Operations on Real Objects. In *Conference Companion on Human Factors in Computing Systems*. ACM, New York, NY, USA, 141–142. DOI:http://dx.doi.org/10.1145/223355.223470

Oscar Kin-Chung Au, Chiew-Lan Tai, and Hongbo Fu. 2012. Multitouch Gestures for Constrained Transformation of 3D Objects. *Comp. Graph. Forum* 31 (May 2012), 651–660. DOI:http://dx.doi.org/10.1111/j.1467-8659.2012.03044.x

Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2008. ILoveSketch: As-natural-as-possible Sketching System for Creating 3D Curve Models. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 151–160. DOI:http://dx.doi.org/10.1145/1449715.1449740

Ravin Balakrishnan and Ken Hinckley. 1999. The Role of Kinesthetic Reference Frames in Two-handed Input Performance. In *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 171–178. DOI:http://dx.doi.org/10.1145/320719.322599

Ravin Balakrishnan and Ken Hinckley. 2000. Symmetric Bimanual Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 33–40. DOI:http://dx.doi.org/10.1145/332040.332404

Ravin Balakrishnan and Gordon Kurtenbach. 1999. Exploring Bimanual Camera Control and Object Manipulation in 3D Graphics Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 56–62. DOI:http://dx.doi.org/10.1145/302979.302991

Tom Bartindale, Chris Harrison, Patrick Olivier, and Scott E. Hudson. 2011. SurfaceMouse: Supplementing Multi-touch Interaction with a Virtual Mouse. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, New York, NY, USA, 293–296. DOI:http://dx.doi.org/10.1145/1935701.1935767

Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. 2010. TeslaTouch: Electrovibration for Touch Surfaces. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 283–292. DOI:http://dx.doi.org/10.1145/1866029.1866074

Michel Beaudouin-Lafon. 1998. Instrumental Interaction: An Interaction Model for Designing post-WIMP User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 446–453. DOI:http://dx.doi.org/10.1145/332040.332473

Hans Belting. 2008. *Florenz und Bagdad. Eine westöstliche Geschichte des Blicks.* C.H. Beck, Munich, Germany.

Hrvoje Benko, Ricardo Jota, and Andrew Wilson. 2012. MirageTable: Freehand Interaction on a Projected Augmented Reality Tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 199–208. DOI:http://dx.doi.org/10.1145/2207676.2207704

Hrvoje Benko, Meredith Ringel Morris, AJ Bernheim Brush, and Andrew D Wilson. 2009. *Insights on interactive tabletops: A survey of researchers and developers*. Technical Report. Microsoft Research.

Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. 2006. Precise Selection Techniques for Multi-touch Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1263–1272. DOI:http://dx.doi.org/10.1145/1124772.1124963

Hrvoje Benko Benko and Steven Feiner. 2007. Balloon Selection: A Multi-Finger Technique for Accurate Low-Fatigue 3D Selection. In *Proceedings of the IEEE Symposium on 3D User Interfaces*. IEEE, Los Alamitos, CA, USA, 85–91. DOI:http://dx.doi.org/10.1109/3DUI.2007.340778

Anastasia Bezerianos and Ravin Balakrishnan. 2005. The Vacuum: Facilitating the Manipulation of Distant Objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 361–370. DOI:http://dx.doi.org/10.1145/1054972.1055023

Xiaojun Bi, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2011. Magic Desk: Bringing Multi-touch Surfaces into Desktop Work. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2511–2520. DOI:http://dx.doi.org/10.1145/1978942.1979309

Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. 1993. Toolglass and Magic Lenses: The See-through Interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, New York, NY, USA, 73–80. DOI:http://dx.doi.org/10.1145/166117.166126

Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. 2010. Touch Projector: Mobile Interaction Through Video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2287–2296. DOI:http://dx.doi.org/10.1145/1753326.1753671

Alan Bränzel, Christian Holz, Daniel Hoffmann, Dominik Schmidt, Marius Knaust, Patrick Lühne, René Meusel, Stephan Richter, and Patrick Baudisch. 2013. GravitySpace: Tracking Users and Their Poses in a Smart Room Using a Pressure-sensing Floor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 725–734. DOI:http://dx.doi.org/10.1145/2470654.2470757

Bill Buxton. 2007. Multi-Touch Systems that I Have Known and Loved. Website. (2007). http://www.billbuxton.com/multitouchOverview.html

Bill Buxton. 2008. The long nose of innovation. *Insight* 11 (2008), 27.

Bill Buxton. 2010. 31.1: Invited Paper: A Touching Story: A Personal Perspective on the History of Touch Interfaces Past and Future. *SID Symposium Digest of Technical Papers* 41, 1 (2010), 444–448. DOI:http://dx.doi.org/10.1889/1.3500488

William Buxton. 1990. A three-state model of graphical input. (1990).

William Buxton, Ralph Hill, and Peter Rowley. 1985. Issues and Techniques in Touch-sensitive Tablet Input. *SIGGRAPH Comput. Graph.* 19, 3 (July 1985), 215–224. DOI:http://dx.doi.org/10.1145/325165.325239

William Buxton and Brad A. Myers. 1986. A Study in Two-handed Input. *SIGCHI Bull.* 17, 4 (April 1986), 321–326. DOI:http://dx.doi.org/10.1145/22339.22390

James C. Byers, Alvah C. Bittner, and Susan G. Hill. 1989. Traditional and raw task load index (TLX) correlations: Are paired comparisons necessary. *Advances in industrial ergonomics and safety I* (1989), 481–485.

Xiang Cao, Anrew D. Wilson, Ravin Balakrishnan, Ken Hinckley, and Scott E. Hudson. 2008. ShapeTouch: Leveraging contact shape on interactive surfaces. In *Proceedings of 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*. IEEE, Washington, DC, USA, 129–136. DOI:http://dx.doi.org/10.1109/TABLETOP.2008.4660195

Stuart K. Card, Allen Newell, and Thomas P. Moran. 1983. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

Johann Cas. 2005. Privacy in pervasive computing environments - a contradiction in terms? *IEEE Technology and Society Magazine* 24, 1 (Spring 2005), 24–33. DOI:http://dx.doi.org/10.1109/MTAS.2005.1407744

Janglin Chen, Wayne Cranton, and Mark Fihn. 2012. *Handbook of visual display technology*. Vol. 131. Springer Berlin, Berlin, Germany.

Michael Chen, S. Joy Mountford, and Abigail Sellen. 1988. A Study in Interactive 3-D Rotation Using 2-D Control Devices. *SIGGRAPH Comput. Graph.* 22, 4 (June 1988), 121–129. DOI:http://dx.doi.org/10.1145/378456.378497

Aurélie Cohé, Fabrice Dècle, and Martin Hachet. 2011. tBox: A 3D Transformation Widget Designed for Touch-screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 3005–3008. DOI:http://dx.doi.org/10.1145/1978942.1979387

Jeremy R. Cooperstock, Koichiro Tanikoshi, Garry Beirne, Tracy Narine, and William A. S. Buxton. 1995. Evolution of a Reactive Environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 170–177. DOI:http://dx.doi.org/10.1145/223904.223926

Paul Dietz and Darren Leigh. 2001. DiamondTouch: A Multi-user Touch Technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 219–226. DOI:http://dx.doi.org/10.1145/502348.502389

Alan Dix, Janet E. Finlay, Gregory D. Abowd, and Russell Beale. 2003. *Human-Computer Interaction (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Tanja Döring, Dagmar Kern, Paul Marshall, Max Pfeiffer, Johannes Schöning, Volker Gruhn, and Albrecht Schmidt. 2011. Gestural Interaction on the Steering Wheel: Reducing the Visual Demand. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 483–492. DOI:http://dx.doi.org/10.1145/1978942.1979010

Economist 2008. Touching the Future. The Econonomist, Technology Quarterly. (September 2008). Case History.

Jörg Edelmann, Andreas Schilling, and Sven Fleck. 2009. The DabR - A multitouch system for intuitive 3D scene navigation. In *Proceedings of 3rd 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*. IEEE, Washington, DC, USA, 1–4. DOI:http://dx.doi.org/10.1109/3DTV.2009.5069671

Douglas C. Engelbart and William K. English. 1968. A Research Center for Augmenting Human Intellect. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I (AFIPS '68 (Fall, part I))*. ACM, New York, NY, USA, 395–410. DOI: http://dx.doi.org/10.1145/1476589.1476645

Steven Feiner, Blair Macintyre, and Dorée Seligmann. 1993. Knowledge-based Augmented Reality. *Commun. ACM* 36, 7 (July 1993), 53–62. DOI:http://dx.doi.org/10.1145/159544.159587

George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. 1995. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 442–449. DOI:http://dx.doi.org/10.1145/223904.223964

Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: Dynamic Physical Affordances and Constraints Through Shape and Object Actuation. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 417–426. DOI:http://dx.doi.org/10.1145/2501988.2502032

Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. 2007. Direct-touch vs. Mouse Input for Tabletop Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 647–656. DOI:http://dx.doi.org/10.1145/1240624.1240726

Mark Foskey, Miguel A. Otaduy, and Ming C. Lin. 2002. ArtNova: touch-enabled 3D model design. In *Proceedings of Virtual Reality*. IEEE, Washington, DC, USA, 119–126. DOI: http://dx.doi.org/10.1109/VR.2002.996514

Anne Friedberg. 2006. *The virtual window. From Alberti to Microsoft.* MIT Press, Cambridge, Massachusetts, USA.

Tinsley A. Galyean and John F. Hughes. 1991. Sculpting: An Interactive Volumetric Modeling Technique. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, New York, NY, USA, 267–274. DOI: http://dx.doi.org/10.1145/122718.122747

Tom Geller. 2006. Interactive Tabletop Exhibits in Museums and Galleries. *IEEE Computer Graphics and Applications* 26, 5 (Sept 2006), 6–11. DOI:http://dx.doi.org/10.1109/MCG.2006.111

Alexander Giesler, Dimitar Valkov, and Klaus Hinrichs. 2014. Void Shadows: Multi-touch Interaction with Stereoscopic Objects on the Tabletop. In *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction*. ACM, New York, NY, USA, 104–112. DOI: http://dx.doi.org/10.1145/2659766.2659779

Jérémie Gilliot, Géry Casiez, and Nicolas Roussel. 2014a. Direct and Indirect Multi-touch Interaction on a Wall Display. In *Proceedings of the 26th Conference on L'Interaction Homme-Machine*. ACM, New York, NY, USA, 147–152. DOI:http://dx.doi.org/10.1145/2670444.2670445

Jérémie Gilliot, Géry Casiez, and Nicolas Roussel. 2014b. Impact of Form Factors and Input Conditions on Absolute Indirect-touch Pointing Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 723–732. DOI:http://dx.doi.org/10.1145/2556288.2556997

Eugen Goldstein. 1876. Vorläufige Mittheilungen über elektrische Entladungen in verdünnten Gasen. *Berlin Akd. Monatsbericht* 279 (1876).

Nan-Wei Gong, Jürgen Steimle, Simon Olberding, Steve Hodges, Nicholas Edward Gillian, Yoshihiro Kawahara, and Joseph A. Paradiso. 2014. PrintSense: A Versatile Sensing Technique to Support Multimodal Flexible Surface Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1407–1410. DOI:http://dx.doi.org/10.1145/2556288.2557173

Gregory Goth. 2011. Brave NUI World. *Commun. ACM* 54, 12 (Dec. 2011), 14–16. DOI:http://dx.doi.org/10.1145/2043174.2043181

Robert Graham and Chris Carter. 2000. Comparison of speech input and manual control of in-car devices while on the move. *Personal Technologies* 4, 2 (2000), 155–164. DOI:http://dx.doi.org/10.1007/BF01324122

Mark Green and Robert Jacob. 1991. SIGGRAPH '90 Workshop Report: Software Architectures and Metaphors for non-WIMP User Interfaces. *SIGGRAPH Comput. Graph.* 25, 3 (July 1991), 229–235. DOI:http://dx.doi.org/10.1145/126640.126677

Tovi Grossman, Ravin Balakrishnan, Gordon Kurtenbach, George Fitzmaurice, Azam Khan, and Bill Buxton. 2001. Interaction Techniques for 3D Modeling on Large Displays. In *Proceedings of Symposium on Interactive 3D Graphics*. ACM, New York, NY, USA., 17–23. DOI:http://dx.doi.org/10.1145/364338.364341

Christian Guererro. 2014. Exploring Direct and Indirect Touch for 3D-Manipultion on a Dual-Screen Setup. (2014).

Yves Guiard. 1987. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior* 19, 4 (1987), 486–517.

Jeremy Gurski and Lee Ming Quach. 2005. Display Technology Overview. (2005).

Martin Hachet, Fabrice Decle, Sebastian Knödel, and Pascal Guitton. 2009. Navidget for 3D Interaction: Camera Positioning and Further Uses. *Int. J. Hum.-Comput. Stud.* 67, 3 (March 2009), 225–236. DOI:http://dx.doi.org/10.1016/j.ijhcs.2008.09.013

Jefferson Y. Han. 2005. Low-cost Multi-touch Sensing Through Frustrated Total Internal Reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 115–118. DOI:http://dx.doi.org/10.1145/1095034.1095054

Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. 2007. Shallow-depth 3D Interaction: Design and Evaluation of One-, Two- and Three-touch Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1147–1156. DOI:http://dx.doi.org/10.1145/1240624.1240798

Mark S. Hancock, Sheelagh Carpendale, Frederic D. Vernier, Daniel Wigdor, and Chia Shen. 2006. Rotation and Translation Mechanisms for Tabletop Interaction. In *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. IEEE Computer Society, Washington, DC, USA, 79–88. DOI:http://dx.doi.org/10.1109/TABLETOP.2006.26

John Hardy. 2012. Experiences: A Year in the Life of an Interactive Desk. In *Proceedings of the Designing Interactive Systems Conference (DIS '12)*. ACM, New York, NY, USA, 679–688. DOI:http://dx.doi.org/10.1145/2317956.2318058

Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011a. OmniTouch: Wearable Multitouch Interaction Everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 441–450. DOI:http://dx.doi.org/10.1145/2047196.2047255

Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011b. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 627–636. DOI:http://dx.doi.org/10.1145/2047196.2047279

Chris Harrison, Robert Xiao, Julia Schwarz, and Scott E. Hudson. 2014. TouchTools: Leveraging Familiarity and Skill with Physical Tools to Augment Touch Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2913–2916. DOI:http://dx.doi.org/10.1145/2556288.2557012

Doris Hausen, Sebastian Boring, and Saul Greenberg. 2013. The Unadorned Desk: Exploiting the Physical Space around a Display as an Input Canvas. In *Proceedings of the 14th IFIP TC 13 International Conference*. Springer Berlin Heidelberg, Berlin, Heidelberg, 140–158. DOI:http://dx.doi.org/10.1007/978-3-642-40483-2_10

Steven Heim. 2007. *The Resonant Interface: HCI Foundations for Interaction Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Fabian Hennecke. 2014. Effekte und Potenziale eines gebogenen interaktiven Displays. (Januar 2014). http://nbn-resolving.de/urn:nbn:de:bvb:19-167824 1.0.

Fabian Hennecke, Alexander De Luca, Ngo Dieu Huong Nguyen, Sebastian Boring, and Andreas Butz. 2013. Investigating Pointing Tasks across Angularly Coupled Display Areas. In *Proceedings of the 14th IFIP TC 13 International Conference*. Springer Berlin Heidelberg, Berlin, Heidelberg, 720–727. DOI:http://dx.doi.org/10.1007/978-3-642-40483-2_50

Fabian Hennecke, Wolfgang Matzke, and Andreas Butz. 2012. How Screen Transitions Influence Touch and Pointer Interaction Across Angled Display Arrangements. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 209–212. DOI:http://dx.doi.org/10.1145/2207676.2207705

Fabian Hennecke, Simon Völker, Maximilian Schenk, Hauke Schaper, and Andreas Butz. 2013. PerspectiveTable: blending physical and virtual collaborative workspaces. (2013).

Fabian Hennecke, Raphael Wimmer, Eduard Vodicka, and Andreas Butz. 2012. Vertibles: Using Vacuum Self-adhesion to Create a Tangible User Interface for Arbitrary Interactive Surfaces. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction (TEI '12)*. ACM, New York, NY, USA, 303–306. DOI:http://dx.doi.org/10.1145/2148131.2148194

Christopher F. Herot and Guy Weinzapfel. 1978. One-point Touch Input of Vector Information for Computer Displays. *SIGGRAPH Comput. Graph.* 12, 3 (Aug. 1978), 210–216. DOI:http://dx.doi.org/10.1145/965139.807392

Otmar Hilliges, Dominikus Baur, and Andreas Butz. 2007. Photohelix: Browsing, Sorting and Sharing Digital Photo Collections. In *Proceedings of the Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. IEEE, Washington, DC, USA, 87–94. DOI:http://dx.doi.org/10.1109/TABLETOP.2007.14

Otmar Hilliges, David Kim, Shahram Izadi, Malte Weiss, and Andrew Wilson. 2012. HoloDesk: Direct 3D Interactions with a Situated See-through Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2421–2430. DOI:http://dx.doi.org/10.1145/2207676.2208405

Ken Hinckley, Randy Pausch, and Dennis Proffitt. 1997. Attention and Visual Feedback: The Bimanual Frame of Reference. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*. ACM, New York, NY, USA, 121–ff. DOI:http://dx.doi.org/10.1145/253284.253318

Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. 2010. Pen + Touch = New Tools. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 27–36. DOI:http://dx.doi.org/10.1145/1866029.1866036

James Hollan, Edwin Hutchins, and David Kirsh. 2000. Distributed Cognition: Toward a New Foundation for Human-computer Interaction Research. *ACM Trans. Comput.-Hum.*

Interact. 7, 2 (June 2000), 174–196. DOI:http://dx.doi.org/10.1145/353485.353487

Christian Holz and Patrick Baudisch. 2011. Understanding Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2501–2510. DOI:http://dx.doi.org/10.1145/1978942.1979308

Stéphane Huot, Olivier Chapuis, and Pierre Dragicevic. 2011. TorusDesktop: Pointing via the Backdoor is Sometimes Shorter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 829–838. DOI:http://dx.doi.org/10.1145/1978942.1979064

Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. 1985. Direct Manipulation Interfaces. *Hum.-Comput. Interact.* 1, 4 (Dec. 1985), 311–338. DOI:http://dx.doi.org/10.1207/s15327051hci0104_2

Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. 2007. Teddy: a sketching interface for 3D freeform design. (2007).

Don Ihde. 1990. *Technology and the lifeworld: From garden to earth*. Indiana University Press, Bloomington, IN, USA.

Hiroshi Ishii and Minoru Kobayashi. 1992. ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 525–532. DOI:http://dx.doi.org/10.1145/142750.142977

Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. 2012. Radical Atoms: Beyond Tangible Bits, Toward Transformable Materials. *interactions* 19, 1 (Jan. 2012), 38–51. DOI:http://dx.doi.org/10.1145/2065327.2065337

Hiroshi Ishii and Brygg Ullmer. 1997. Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 234–241. DOI:http://dx.doi.org/10.1145/258549.258715

Robert J.K. Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. 2008. Reality-based Interaction: A Framework for post-WIMP Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 201–210. DOI:http://dx.doi.org/10.1145/1357054.1357089

Robert J. K. Jacob, Linda E. Sibert, Daniel C. McFarlane, and M. Preston Mullen, Jr. 1994. Integrality and Separability of Input Devices. *ACM Trans. Comput.-Hum. Interact.* 1, 1 (March 1994), 3–26. DOI:http://dx.doi.org/10.1145/174630.174631

Jacek Jankowski and Martin Hachet. 2013. A survey of interaction techniques for interactive 3D environments. In *Proceedings of Eurographics 2013 - State of the Art Reports*. The Eurographics Association, Geneve, Switzerland, 65–93. DOI:http://dx.doi.org/10.2312/conf/EG2013/stars/065-093

Sankar Jayaram, Hugh I. Connacher, and Kevin W. Lyons. 1997. Virtual assembly using virtual reality techniques. *Computer-Aided Design* 29, 8 (1997), 575 – 584. DOI:http://dx.doi.org/10.1016/S0010-4485(96)00094-2

Hans-Christian Jetter, Svenja Leifert, Jens Gerken, Sören Schubert, and Harald Reiterer. 2012. Does (Multi-)Touch Aid Users' Spatial Memory and Navigation in 'Panning' and in 'Zooming & Panning' UIs?. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, New York, NY, USA, 83–90. DOI:http://dx.doi.org/10.1145/2254556.2254575

Eric A. Johnson. 1965. Touch display? a novel input/output device for computers. *Electronics Letters* 8, 1 (1965), 219–220.

Eric A. Johnson. 1967. Touch displays: a programmed man-machine interface. *Ergonomics* 10, 2 (1967), 271–277.

Timothy E. Johnson. 1963. Sketchpad III: A Computer Program for Drawing in Three Dimensions. In *Proceedings of the May 21-23, 1963, Spring Joint Computer Conference (AFIPS '63 (Spring))*. ACM, New York, NY, USA, 347–353. DOI:http://dx.doi.org/10.1145/1461551.1461592

Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. 2007. The reacTable: Exploring the Synergy Between Live Music Performance and Tabletop Tangible Interfaces. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*. ACM, New York, NY, USA, 139–146. DOI:http://dx.doi.org/10.1145/1226969.1226998

Paul Kabbash, William Buxton, and Abigail Sellen. 1994. Two-handed Input in a Compound Task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 417–423. DOI:http://dx.doi.org/10.1145/191666.191808

Paul Kabbash, I. Scott MacKenzie, and William Buxton. 1993. Human Performance Using Computer Input Devices in the Preferred and Non-preferred Hands. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 474–481. DOI:http://dx.doi.org/10.1145/169059.169414

Shaun K. Kane, Daniel Avrahami, Jacob O. Wobbrock, Beverly Harrison, Adam D. Rea, Matthai Philipose, and Anthony LaMarca. 2009. Bonfire: A Nomadic System for Hybrid Laptop-tabletop Interaction. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 129–138. DOI:http://dx.doi.org/10.1145/1622176.1622202

Alan Kay and Adele Goldberg. 1977. Personal Dynamic Media. *Computer* 10, 3 (March 1977), 31–41. DOI:http://dx.doi.org/10.1109/C-M.1977.217672

Alan Curtis Kay. 1969. *The reactive enging*. Ph.D. Dissertation. The University of Utah.

JA Scott Kelso. 1997. *Dynamic patterns: The self-organization of brain and behavior*. MIT press, Cambridge, Massachusetts, USA.

Tijmen Klein, Florimond Guéniat, Luc Pastur, Frederic Vernier, and Tobias Isenberg. 2012. A Design Study of Direct-Touch Interaction for Exploratory 3D Scientific Visualization. *Comp. Graph. Forum* 31, 3pt3 (June 2012), 1225–1234. DOI:http://dx.doi.org/10.1111/j.1467-8659.2012.03115.x

Sebastian Knoedel and Martin Hachet. 2011. Multi-touch RST in 2D and 3D Spaces: Studying the Impact of Directness on User Performance. In *Proceedings of the 2011 IEEE Symposium on 3D User Interfaces*. IEEE Computer Society, Washington, DC, USA, 75–78. http://dl.acm.org/citation.cfm?id=2013881.2014214

Myron W. Krueger, Thomas Gionfriddo, and Katrin Hinrichsen. 1985. VIDEOPLACE&Mdash;an Artificial Reality. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 35–40. DOI:http://dx.doi.org/10.1145/317456.317463

Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. 1997. The Design of a GUI Paradigm Based on Tablets, Two-hands, and Transparency. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 35–42. DOI:http://dx.doi.org/10.1145/258549.258574

Kari Kuutti. 1995. Activity Theory As a Potential Framework for Human-computer Interaction Research. In *Context and Consciousness*, Bonnie A. Nardi (Ed.). Massachusetts Institute of Technology, Cambridge, MA, USA, 17–44. http://dl.acm.org/citation.cfm?id=223826.223828

Celine Latulipe, Ian Bell, Charles L. A. Clarke, and Craig S. Kaplan. 2006. symTone: Two-handed Manipulation of Tone Reproduction Curves. In *Proceedings of Graphics Interface 2006*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 9–16. http://dl.acm.org/citation.cfm?id=1143079.1143082

Celine Latulipe, Craig S. Kaplan, and Charles L. A. Clarke. 2005. Bimanual and Unimanual Image Alignment: An Evaluation of Mouse-based Techniques. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 123–131. DOI:http://dx.doi.org/10.1145/1095034.1095057

Celine Latulipe, Stephen Mann, Craig S. Kaplan, and Charlie L. A. Clarke. 2006. symSpline: Symmetric Two-handed Spline Manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 349–358. DOI:http://dx.doi.org/10.1145/1124772.1124825

Duke Lee. 2011. The State of the Touch-Screen Panel Market in 2011. *Official Monthly Publication of the Society for Information Display* 3, 11 (2011), 12–16.

Jinha Lee, Alex Olwal, Hiroshi Ishii, and Cati Boulanger. 2013. SpaceTop: Integrating 2D and Spatial 3D Interactions in a See-through Desktop Environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 189–192. DOI:http://dx.doi.org/10.1145/2470654.2470680

Sylvanus Y. K. Lee, William Buxton, and Kenneth C. Smith. 1985. A Multi-touch Three Dimensional Touch-sensitive Tablet. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '85)*. ACM, New York, NY, USA, 21–25. DOI: http://dx.doi.org/10.1145/317456.317461

Andrea Leganchuk, Shumin Zhai, and William Buxton. 1998. Manual and Cognitive Benefits of Two-handed Input: An Experimental Study. *ACM Trans. Comput.-Hum. Interact.* 5, 4 (Dec. 1998), 326–359. DOI:http://dx.doi.org/10.1145/300520.300522

Jingbo Liu, Oscar Kin-Chung Au, Hongbo Fu, and Chiew-Lan Tai. 2012a. Two-Finger Gestures for 6DOF Manipulation of 3D Objects. *Comput. Graph. Forum* 31, 7pt1 (Sept. 2012), 2047–2055. DOI:http://dx.doi.org/10.1111/j.1467-8659.2012.03197.x

Jingbo Liu, Oscar Kin-Chung Au, Hongbo Fu, and Chiew-Lan Tai. 2012b. Two-Finger Gestures for 6DOF Manipulation of 3D Objects. *Comput. Graph. Forum* 31, 7pt1 (Sept. 2012), 2047–2055. DOI:http://dx.doi.org/10.1111/j.1467-8659.2012.03197.x

I. Scott MacKenzie. 2013. *Human-Computer Interaction: An Empirical Research Perspective* (1st ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Thor Magnusson. 2009. Of epistemic tools: Musical instruments as cognitive extensions. *Organised Sound* 14, 02 (2009), 168–176.

Shahzad Malik and Joe Laszlo. 2004. Visual Touchpad: A Two-handed Gestural Input Device. In *Proceedings of the 6th International Conference on Multimodal Interfaces*. ACM, New York, NY, USA, 289–296. DOI:http://dx.doi.org/10.1145/1027933.1027980

Anthony Martinet, Gery Casiez, and Laurent Grisoni. 2010a. The Design and Evaluation of 3D Positioning Techniques for Multi-touch Displays. In *Proceedings of the 2010 IEEE Symposium on 3D User Interfaces*. IEEE Computer Society, Washington, DC, USA, 115–118. DOI:http://dx.doi.org/10.1109/3DUI.2010.5444709

Anthony Martinet, Géry Casiez, and Laurent Grisoni. 2010b. The Effect of DOF Separation in 3D Manipulation Tasks with Multi-touch Displays. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*. ACM, New York, NY, USA, 111–118. DOI:http://dx.doi.org/10.1145/1889863.1889888

André Meyer. 1995. Pen Computing: A Technology Overview and a Vision. *SIGCHI Bull.* 27, 3 (July 1995), 46–90. DOI:http://dx.doi.org/10.1145/221296.221308

Margaret R. Minsky. 1984. Manipulating Simulated Objects with Real-world Gestures Using a Force and Position Sensitive Screen. *SIGGRAPH Comput. Graph.* 18, 3 (Jan. 1984), 195–203. DOI:http://dx.doi.org/10.1145/964965.808598

Meredith R. Morris, A. J. Bernheim Brush, and Brian R. Meyers. 2007. Reading Revisited: Evaluating the Usability of Digital Display Surfaces for Active Reading Tasks. In *Proceedings of Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. IEEE, Washington, DC, USA, 79–86. DOI:http://dx.doi.org/10.1109/TABLETOP.2007.12

Meredith R. Morris, A. J. Bernheim Brush, and Brian R. Meyers. 2008. A field study of knowledge workers' use of interactive horizontal displays. In *Proceedings of 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*. IEEE, Washington, DC, USA, 105–112. DOI:http://dx.doi.org/10.1109/TABLETOP.2008.4660192

Tomer Moscovich and John F. Hughes. 2006. Multi-finger Cursor Techniques. In *Proceedings of Graphics Interface 2006*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 1–7. http://dl.acm.org/citation.cfm?id=1143079.1143081

Tomer Moscovich and John F. Hughes. 2008. Indirect Mappings of Multi-touch Input Using One and Two Hands. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1275–1284. DOI:http://dx.doi.org/10.1145/1357054.1357254

Miguel A. Nacenta, Patrick Baudisch, Hrvoje Benko, and Andy Wilson. 2009a. Separability of Spatial Manipulations in Multi-touch Interfaces. In *Proceedings of Graphics Interface 2009*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 175–182. http://dl.acm.org/citation.cfm?id=1555880.1555919

Miguel A. Nacenta, Carl Gutwin, Dzmitry Aliakseyeu, and Sriram Subramanian. 2009b. There and Back Again: Cross-Display Object Movement in Multi-Display Environments. *Human–Computer Interaction* 24, 1-2 (2009), 170–229. DOI:http://dx.doi.org/10.1080/07370020902819882

Mathieu Nancel, Daniel Vogel, and Edward Lank. 2015. Clutching Is Not (Necessarily) the Enemy. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 4199–4202. DOI:http://dx.doi.org/10.1145/2702123.2702134

Donald A. Norman. 2010. Natural User Interfaces Are Not Natural. *interactions* 17, 3 (May 2010), 6–10. DOI:http://dx.doi.org/10.1145/1744161.1744163

Donald A. Norman. 2013. *The design of everyday things: Revised and expanded edition*. Basic books, New York, NY, USA.

Simon Olberding, Michael Wessely, and Jürgen Steimle. 2014. PrintScreen: Fabricating Highly Customizable Thin-film Touch-displays. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 281–290. DOI:http://dx.doi.org/10.1145/2642918.2647413

Dan R. Olsen, Jr. and Travis Nielsen. 2001. Laser Pointer Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 17–22. DOI:http://dx.doi.org/10.1145/365024.365030

Patrick Paczkowski, Julie Dorsey, Holly Rushmeier, and Min H. Kim. 2014. Paper3D: Bringing Casual 3D Modeling to a Multi-touch Interface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 23–32. DOI:http://dx.doi.org/10.1145/2642918.2647416

Henri Palleis. 2014a. *Sound Collage Creation on a Curved Touch Display*. Technical Report. Ludwig-Maximilians-Universität München, Munich. 4 pages.

Henri Palleis. 2014b. Towards an Interaction Model for Multi-Surface Personal Computing Spaces. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, New York, NY, USA, 461–466. DOI: http://dx.doi.org/10.1145/2669485.2669488

Henri Palleis, Fabian Hennecke, Axel Hösl, and Heinrich Hussmann. 2013. Designing for Expert Users on a Large Non-standard Display. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*. ACM, New York, NY, USA, 405–408. DOI:http://dx.doi.org/10.1145/2512349.2514915

Henri Palleis and Heinrich Hussmann. 2014. Towards Understanding Spontaneous Interaction on Curved Displays. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2065–2070. DOI:http://dx.doi.org/10.1145/2559206.2581300

Henri Palleis and Heinrich Hussmann. 2016. Indirect 2D Touch Panning: How Does It Affect Spatial Memory and Navigation Performance?. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1947–1951. DOI:http://dx.doi.org/10.1145/2858036.2858334

Henri Palleis, Mirjam Mickisch, and Heinrich Hussmann. 2015a. A Concept for 3D Interaction on a Curved Touch Display. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1163–1168. DOI:http://dx.doi.org/10.1145/2702613.2732856

Henri Palleis and Vanessa Niedermeier. 2016. Exploring Two-handed Indirect Multi-touch Input for a Docking Task. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, New York, NY, USA, 36–39. DOI:http://dx.doi.org/10.1145/2909132.2909285

Henri Palleis, Julie Wagner, and Heinrich Hussmann. 2015b. Perspective-dependent Indirect Touch Input for 3D Polygon Extrusion. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, New York, NY, USA, 89–90. DOI:http://dx.doi.org/10.1145/2815585.2815733

Henri Palleis, Julie Wagner, and Heinrich Hussmann. 2016. Novel Indirect Touch Input Techniques Applied to Finger-Forming 3D Models. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, New York, NY, USA, 228–235. DOI:http://dx.doi.org/10.1145/2909132.2909257

James Patten, Hiroshi Ishii, Jim Hines, and Gian Pangaro. 2001. Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 253–260. DOI:http://dx.doi.org/10.1145/365024.365112

Ken Pfeuffer, Jason Alexander, Ming Ki Chong, Yanxia Zhang, and Hans Gellersen. 2015. Gaze-Shifting: Direct-Indirect Input with Pen and Touch Modulated by Gaze. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, New York, NY, USA, 373–383. DOI:http://dx.doi.org/10.1145/2807442.2807460

Ken Pfeuffer, Jason Alexander, and Hans Gellersen. 2016. Partially-indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2845–2856. DOI:http://dx.doi.org/10.1145/2858036.2858201

Richard L. Potter, Linda J. Weldon, and Ben Shneiderman. 1988. Improving the Accuracy of Touch Screens: An Experimental Evaluation of Three Strategies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '88)*. ACM, New York, NY, USA, 27–32. DOI:http://dx.doi.org/10.1145/57167.57171

Ivan Poupyrev and Shigeaki Maruyama. 2003. Tactile Interfaces for Small Touch Screens. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 217–220. DOI:http://dx.doi.org/10.1145/964696.964721

Philip Quinn, Sylvain Malacria, and Andy Cockburn. 2013. Touch Scrolling Transfer Functions. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 61–70. DOI:http://dx.doi.org/10.1145/2501988.2501995

Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. 2004. Pressure Widgets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 487–494. DOI:http://dx.doi.org/10.1145/985692.985754

Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. 1998. The Office of the Future: A Unified Approach to Image-based Modeling and Spatially Immersive Displays. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, New York, NY, USA, 179–188. DOI: http://dx.doi.org/10.1145/280814.280861

Jef Raskin. 2000. *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.

Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han. 2009. A Screen-space Formulation for 2D and 3D Direct Manipulation. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 69–78. DOI:http://dx.doi.org/10.1145/1622176.1622190

Jun Rekimoto. 2002. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 113–120. DOI:http://dx.doi.org/10.1145/503376.503397

Jun Rekimoto and Masanori Saitoh. 1999. Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 378–385. DOI:http://dx.doi.org/10.1145/302979.303113

Xiangshi Ren and Shinju Moriya. 2000. Improving Selection Performance on Pen-based Systems: A Study of Pen-based Interaction for Selection Tasks. *ACM Trans. Comput.-Hum. Interact.* 7, 3 (Sept. 2000), 384–416. DOI:http://dx.doi.org/10.1145/355324.355328

Simon Rogers, John Williamson, Craig Stewart, and Roderick Murray-Smith. 2011. AnglePose: Robust, Precise Capacitive Touch Tracking via 3D Orientation Estimation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2575–2584. DOI:http://dx.doi.org/10.1145/1978942.1979318

Ilya Rosenberg and Ken Perlin. 2009. The UnMousePad: An Interpolating Multi-touch Force-sensing Input Pad. *ACM Trans. Graph.* 28, 3, Article 65 (July 2009), 9 pages. DOI:http://dx.doi.org/10.1145/1531326.1531371

Mario Russo. 2005. *Polygonal Modeling: Basic and Advanced Techniques*. Jones & Bartlett Learning, Burlington, MA, USA.

Joseph D. Rutledge and Ted Selker. 1990. Force-to-motion Functions for Pointing. In *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction*. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 701–706. http://dl.acm.org/citation.cfm?id=647402.725310

Kathy Ryall, Meredith R. Morris, Katherine Everitt, Clifton Forlines, and Chia Shen. 2006. Experiences with and observations of direct-touch tabletops. In *Proceedings of First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. IEEE, Washington, DC, USA, 89 – 96. DOI:http://dx.doi.org/10.1109/TABLETOP.2006.12

Lawrence Sasaki, Guy Fedorkow, William Buxton, Chris Retterath, and Kenneth C. Smith. 1981. A Touch-Sensitive Input Device. (1981).

Steven Schkolne, Michael Pruett, and Peter Schröder. 2001. Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 261–268. DOI:http://dx.doi.org/10.1145/365024.365114

Dominik Schmidt, Florian Block, and Hans Gellersen. 2009. A Comparison of Direct and Indirect Multi-touch Input for Large Surfaces. In *Proceedings of the 12th IFIP TC 13 International Conference*, Tom Gross, Jan Gulliksen, Paula Kotzé, Lars Oestreicher, Philippe Palanque, Raquel Oliveira Prates, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 582–594. DOI:http://dx.doi.org/10.1007/978-3-642-03655-2_65

Ryan Schmidt, Karan Singh, and Ravin Balakrishnan. 2008. Sketching and Composing Widgets for 3D Manipulation. *Computer Graphics Forum* 27, 2 (2008), 301–310. DOI:http://dx.doi.org/10.1111/j.1467-8659.2008.01127.x

Johannes Schöning, Peter Brandl, Florian Daiber, Florian Echtler, Otmar Hilliges, Jonathan Hook, Markus Löchtefeld, Nima Motamedi, Laurence Muller, Patrick Olivier, Tim Roth, and Ulrich von Zadow. 2008. *Multi-Touch Surfaces: A Technical Guide*. techreport. Technische Universität München. 19 pages. http://drehscheibe.in.tum.de/forschung/pub/reports/2008/TUM-I0833.pdf.gz

Tobias Schwarz, Fabian Hennecke, Felix Lauber, and Harald Reiterer. 2012. Perspective+Detail: A Visualization Technique for Vertically Curved Displays. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, New York, NY, USA, 485–488. DOI:http://dx.doi.org/10.1145/2254556.2254648

Stacey D. Scott, M. Sheelagh T. Carpendale, and Kori M. Inkpen. 2004. Territoriality in Collaborative Tabletop Workspaces. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. ACM, New York, NY, USA, 294–303. DOI:http://dx.doi.org/10.1145/1031607.1031655

Andrew Sears, Catherine Plaisant, and Ben Shneiderman. 1992. A New Era for High Precision Touchscreens. In *Advances in Human-computer Interaction (Vol. 3)*, H. Rex Hartson and Deborah Hix (Eds.). Ablex Publishing Corp., Norwood, NJ, USA, 1–33. http://dl.acm.org/citation.cfm?id=159433.159436

Andrew Sears and Ben Shneiderman. 1991. High precision touchscreens: design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies* 34, 4 (1991), 593 – 613. DOI:http://dx.doi.org/10.1016/0020-7373(91)90037-8

Patricia M. Shields and Nandhini Rangarajan. 2013. *A playbook for research methods: Integrating conceptual frameworks and project management*. New Forums Press, Stillwater, OK, USA.

Ben Shneiderman. 1993. *Touchscreens now offer compelling uses*. Ablex Publishing Corporation, New York, NY, USA, 187–195.

Ben Shneiderman. 1997. Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces. In *Proceedings of the 2Nd International Conference on Intelligent User Interfaces*. ACM, New York, NY, USA, 33–39. DOI:http://dx.doi.org/10.1145/238218.238281

Ken Shoemake. 1992. ARCBALL: A User Interface for Specifying Three-dimensional Orientation Using a Mouse. In *Proceedings of the Conference on Graphics Interface '92*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 151–156. http://dl.acm.org/citation.cfm?id=155294.155312

Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. 2007. Shadow Reaching: A New Perspective on Interaction for Large Displays. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 53–56. DOI:http://dx.doi.org/10.1145/1294211.1294221

Adalberto L. Simeone and Hans Gellersen. 2015. Comparing indirect and direct touch in a stereoscopic interaction task. In *In Proceedings of IEEE Symposium on 3D User Interfaces*. IEEE, Washington, DC, USA, 105–108. DOI:http://dx.doi.org/10.1109/3DUI.2015.7131733

Sophie Stellmach and Raimund Dachselt. 2012. Look & Touch: Gaze-supported Target Acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2981–2990. DOI:http://dx.doi.org/10.1145/2207676.2208709

Norbert A Streitz, Peter Tandler, Christian Müller-Tomfelde, and Shin'ichi Konomi. 2001. *Roomware: Towards the Next Generation of Human- Computer Interaction based on an Integrated Design of Real and Virtual Worlds*. Addison-Wesley, Boston, MA, USA, 553 –578.

R. M. Strong and D. E. Troxel. 1970. An Electrotactile Display. *IEEE Transactions on Man-Machine Systems* 11, 1 (March 1970), 72–79. DOI:http://dx.doi.org/10.1109/TMMS.1970.299965

Sven Strothoff, Dimitar Valkov, and Klaus Hinrichs. 2011. Triangle Cursor: Interactions with Objects Above the Tabletop. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ACM, New York, NY, USA, 111–119. DOI:http://dx.doi.org/10.1145/2076354.2076377

Rahul Sukthankar. 2005. Towards Ambient Projection for Intelligent Environments. In *Proceedings of Computer Vision for Interactive and Intelligent Environment*. IEEE, Washington, DC, USA, 162–172. DOI:http://dx.doi.org/10.1109/CVIIE.2005.20

Ivan E. Sutherland. 1963a. *Sketchpad, a Man-Machine Graphical Communication System*. Ph.D. Dissertation. Massachusetts Institute of Technology.

Ivan E. Sutherland. 1963b. Sketchpad: A Man-machine Graphical Communication System. In *Proceedings of the May 21-23, 1963, Spring Joint Computer Conference (AFIPS '63 (Spring))*. ACM, New York, NY, USA, 329–346. DOI:http://dx.doi.org/10.1145/1461551.1461591

Desney S. Tan, Randy Pausch, Jeanine K. Stefanucci, and Dennis R. Proffitt. 2002. Kinesthetic Cues Aid Spatial Memory. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, NY, USA, 806–807. DOI:http://dx.doi.org/10.1145/506443.506607

Bruce Tognazzini. 1994. The &Ldquo;Starfire&Rdquo; Video Prototype Project: A Case History. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 99–105. DOI:http://dx.doi.org/10.1145/191666.191712

Philip Tuddenham, David Kirk, and Shahram Izadi. 2010. Graspables Revisited: Multi-touch vs. Tangible Input for Tabletop Displays in Acquisition and Manipulation Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2223–2232. DOI:http://dx.doi.org/10.1145/1753326.1753662

Brygg Ullmer and Hiroshi Ishii. 1997. The metaDESK: Models and Prototypes for Tangible User Interfaces. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 223–232. DOI:http://dx.doi.org/10.1145/263407.263551

John Underkoffler and Hiroshi Ishii. 1999. Urp: A Luminous-tangible Workbench for Urban Planning and Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 386–393. DOI:http://dx.doi.org/10.1145/302979.303114

John Underkoffler, Brygg Ullmer, and Hiroshi Ishii. 1999. Emancipated Pixels: Real-world Graphics in the Luminous Room. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 385–392. DOI:http://dx.doi.org/10.1145/311535.311593

Andries van Dam. 1997. Post-WIMP User Interfaces. *Commun. ACM* 40, 2 (Feb. 1997), 63–67. DOI:http://dx.doi.org/10.1145/253671.253708

Simon Voelker, Andrii Matviienko, Johannes Schöning, and Jan Borchers. 2015. Combining Direct and Indirect Touch Input for Interactive Workspaces Using Gaze Input. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*. ACM, New York, NY, USA, 79–88. DOI:http://dx.doi.org/10.1145/2788940.2788949

Simon Voelker, Christine Sutter, Lei Wang, and Jan Borchers. 2012. Understanding Flicking on Curved Surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 189–198. DOI:http://dx.doi.org/10.1145/2207676.2207703

Simon Voelker, Chat Wacharamanotham, and Jan Borchers. 2013. An Evaluation of State Switching Methods for Indirect Touch Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 745–754. DOI:http://dx.doi.org/10.1145/2470654.2470759

Simon Voelker, Malte Weiss, Chat Wacharamanotham, and Jan Borchers. 2011. Dynamic Portals: A Lightweight Metaphor for Fast Object Transfer on Interactive Surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ACM, New York, NY, USA, 158–161. DOI:http://dx.doi.org/10.1145/2076354.2076384

Daniel Vogel and Patrick Baudisch. 2007. Shift: A Technique for Operating Pen-based Interfaces Using Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 657–666. DOI:http://dx.doi.org/10.1145/1240624.1240727

John Walker. 1990. *Through the looking glass*. Addison-Wesley, New York, NY, USA.

Benjamin Walther-Franks, Marc Herrlich, and Rainer Malaka. 2011. A Multi-touch System for 3D Modelling and Animation. In *Proceedings of Smart Graphics*. Springer-Verlag, Berlin, Germany, 48–59. http://dl.acm.org/citation.cfm?id=2032567.2032574

Feng Wang and Xiangshi Ren. 2009. Empirical Evaluation for Finger Input Properties in Multi-touch Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1063–1072. DOI:http://dx.doi.org/10.1145/1518701.1518864

Sidney W. Wang and Arie E. Kaufman. 1995. Volume Sculpting. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics*. ACM, New York, NY, USA, 151–ff. DOI: http://dx.doi.org/10.1145/199404.199430

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

Malte Weiss and Simon Voelker. 2009. BendDesk: seamless integration of horizontal and vertical multi-touch surfaces in desk environments. (2009).

Malte Weiss, Simon Voelker, Christine Sutter, and Jan Borchers. 2010. BendDesk: Dragging Across the Curve. In *ACM International Conference on Interactive Tabletops and Surfaces*. ACM, New York, NY, USA, 1–10. DOI:http://dx.doi.org/10.1145/1936652.1936654

Malte Weiss, Julie Wagner, Yvonne Jansen, Roger Jennings, Ramsin Khoshabeh, James D. Hollan, and Jan Borchers. 2009. SLAP Widgets: Bridging the Gap Between Virtual and Physical Controls on Tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 481–490. DOI:http://dx.doi.org/10.1145/1518701.1518779

Pierre Wellner. 1993. Interacting with Paper on the DigitalDesk. *Commun. ACM* 36, 7 (July 1993), 87–96. DOI:http://dx.doi.org/10.1145/159544.159630

Pierre Wellner, Wendy Mackay, and Rich Gold. 1993. Back to the Real World. *Commun. ACM* 36, 7 (July 1993), 24–26. DOI:http://dx.doi.org/10.1145/159544.159555

Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. 2007a. Lucid Touch: A See-through Mobile Device. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 269–278. DOI:http://dx.doi.org/10.1145/1294211.1294259

Daniel Wigdor, Gerald Penn, Kathy Ryall, Alan Esenther, and Chia Shen. 2007b. Living with a Tabletop: Analysis and Observations of Long Term Office Use of a Multi-Touch Table. In *Proceedings of Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. IEEE, Washington, DC, USA, 60–67. DOI: http://dx.doi.org/10.1109/TABLETOP.2007.33

Daniel Wigdor and Dennis Wixon. 2011. *Brave NUI world: designing natural user interfaces for touch and gesture*. Morgan Kaufmann, Burlington, MA, USA.

Andrew D. Wilson. 2005. PlayAnywhere: A Compact Interactive Tabletop Projection-vision System. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 83–92. DOI:http://dx.doi.org/10.1145/1095034.1095047

Raphael Wimmer, Fabian Hennecke, Florian Schulz, Sebastian Boring, Andreas Butz, and Heinrich Hußmann. 2010. Curve: Revisiting the Digital Desk. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. ACM, New York, NY, USA, 561–570. DOI:http://dx.doi.org/10.1145/1868914.1868977

Andy Wu, Derek Reilly, Anthony Tang, and Ali Mazalek. 2011. Tangible Navigation and Object Manipulation in Virtual Environments. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, New York, NY, USA, 37–44. DOI:http://dx.doi.org/10.1145/1935701.1935710

Peter Yu. 2015. *Exploring Cross-Display Interaction Techniques in Novel Interactive Display Setups*. Master's thesis. Ludwig-Maxmilians-Universität München.

Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. 2007. SKETCH: an interface for sketching 3D scenes. (2007).

Shumin Zhai and Paul Milgram. 1998. Quantifying Coordination in Multiple DOF Movement and Its Application to Evaluating 6 DOF Input Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 320–327. DOI:http://dx.doi.org/10.1145/274644.274689

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig und ohne unerlaubte Beihilfe angefertigt wurde.

München, den 21. Februar 2017

Henri Palleis