

# Extending Touch with Eye Gaze Input

**Ken Pfeuffer**

B.Sc., M.Sc., University of Duisburg-Essen, Germany



School of Computing and Communications

Lancaster University, UK

Thesis submitted for the degree of Doctor of Philosophy

April 2017

# Extending Touch with Eye Gaze Input

Ken Pfeuffer

Submitted for the degree of Doctor of Philosophy

April 2017

## Abstract

Direct touch manipulation with displays has become one of the primary means by which people interact with computers. Exploration of new interaction methods that work in unity with the standard direct manipulation paradigm will be of benefit for the many users of such an input paradigm. In many instances of direct interaction, both the eyes and hands play an integral role in accomplishing the user's interaction goals. The eyes visually select objects, and the hands physically manipulate them. In principle this process includes a two-step selection of the same object: users first look at the target, and then move their hand to it for the actual selection.

This thesis explores human-computer interactions where the principle of direct touch input is fundamentally changed through the use of eye-tracking technology. The change we investigate is a general reduction to a one-step selection process. The need to select using the hands can be eliminated by utilising eye-tracking to enable users to select an object of interest using their eyes only, by simply looking at it. Users then employ their hands for manipulation of the selected object, however they can manipulate it from anywhere as the selection is rendered independent of the hands. When a spatial offset exists between the hands and the object, the user's manual input is indirect. This allows users to manipulate any object they see from any manual input position. This fundamental change can have a substantial effect on the many human-computer interactions that involve user input through direct manipulation, such as temporary touchscreen interactions. However it is unclear if, when, and how it can become beneficial to users of such an interaction method. To approach these questions, our research in this topic is guided by the following two propositions.

The first proposition is that gaze input can transform a direct input modality such as touch to an indirect modality, and with it provide new and powerful interaction capabilities. We develop this proposition in context of our investigation on integrated gaze interactions within direct manipulation user interfaces. We first regard eye gaze for generic multi-touch displays, introducing Gaze-Touch as a technique based on the division of labour: gaze selects and touch manipulates. We investigate this technique with a design space analysis, prototyping of application examples, and an informal user evaluation. The proposition is further developed by an exploration of hybrid eye and hand inputs with a stylus, for precise and cursor based indirect control; with bimanual input, to rapidly issue input from two hands to gaze-selected objects; with tablets, where Gaze-Touch enables one-handed interaction across the whole screen with the same hand that holds the device; and free-hand gesture in virtual reality to interact with any viewed object at

---

a distance located in the virtual scene. Overall, we demonstrate that using eye gaze to enable indirect input yields many interaction benefits, such as whole-screen reachability, occlusion-free manipulation, high precision cursor input, and low physical effort.

Integration of eye gaze with manual input raises new questions about how it can complement, instead of replace, the direct interactions users are familiar with. This is important to allow users the choice between direct and indirect inputs as each affords distinct pros and cons for the usability of human-computer interfaces. These two input forms are normally considered separately from each other, but here we investigate interactions that combine those within the same interface.

In this context, the second proposition is that gaze and touch input enables new and seamless ways of combining direct and indirect forms of interaction. We develop this proposition by regarding multiple interaction tasks that a user usually perform in a sequence, or simultaneously. First, we introduce a method to enable users switching between both input forms by implicitly exploiting visual attention during manual input. Direct input is active when looking at the input, and otherwise users will manipulate the object they look at indirectly. A design application for typical drawing and vector-graphics tasks has been prototyped to illustrate and explore this principle. The application contributes many example use cases, where direct drawing activities are complemented with indirect menu actions, precise cursor inputs, and seamless context switching at a glance.

We further develop the proposition by investigating simultaneous direct and indirect input by bimanual input, where each input is assigned to one hand. We present an empirical study with an in-depth analysis of using indirect navigation in one hand, and direct pen drawing on the other. We extend this input constellation to tablet devices, by designing compound techniques for use in a more naturalistic setting when one hand holds the device. The interactions show that many typical tablet scenarios, such as browsing, map navigation, homescreen selections, or image gallery, can be enhanced through exploiting eye gaze.

# Declaration

This thesis is a presentation of my original research work. No part of this thesis has been submitted for another degree or qualification. The work was done under the guidance of Hans Gellersen and Jason Alexander at Lancaster University.

Author: **Ken Pfeuffer**

# Acknowledgements

A lot of people were involved in the direction that my life took to get to the point that I am now. Many helped me during my way from the start of life in university and start of the computer science program, which I only did because my lovely sister Mariko suggested to me, and further persuasion by my beloved mother Toshiko. I started computer science at the University of Duisburg-Essen, Germany, where I have not only met Enrico Rukzio and Christian Winkler who connected me to Lancaster, but also I met Ayse, Kay, Jens, Elena, Sven, and I can't name all of the people involved in the joint activities around WG21 but one I must, my wife Tata, who I love. All these people and experiences affected my way of thinking and with it the research.

None of the research would be possible if it wouldn't be for the always positive and open-minded supervisors Hans Gellersen and Jason Alexander. I wanted to stay in Lancaster because of the dynamic research team with Janxia, Minki, Melodie, Chris, Pauline, Eduardo, Jayson, Adalberto and so on from the old garde, and from the new, Chris, Argenis, Baiyang, Diako, Bennedict, and there were so many more people I've met in Lancaster, such as Harry, Matt, and Gareth, or those of summer'15 including Martin, Markus, Becky, and the visiting Thomas. But the biggest influence was Hans who shaped the group that I wanted to be in, and shaped my research to what it is now.

I acknowledge all the prior academic literature and the respective authors that have inspired me in my work. Much of the things that I have coded, designed, or written in a paper was to a greater or lesser extent based on previous art. The reference section does partial justice to this, but I got to point out once more that anything remotely good in this thesis is with high probability just a combination of great ideas from others (and anything bad as well, in the unlikely case there is anything bad). Thanks to all.

# Publications

This work has been published in peer-reviewed publications at conferences. Below are the references of these publications. Each publication relates to one chapter in this thesis.

- Chapter 3 Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. 2014. Gaze-touch: combining gaze with multi-touch for interaction on the same surface. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (UIST '14). ACM, New York, NY, USA, 509-518. DOI: <http://dx.doi.org/10.1145/2642918.2647397>
- Chapter 4 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 and Technology* (UIST '15). ACM, New York, NY, USA, 373-383. DOI: <http://dx.doi.org/10.1145/2807442.2807460>
- Chapter 5 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* (CHI '16). ACM, New York, NY, USA, 2845-2856. DOI: <http://dx.doi.org/10.1145/2858036.2858201>
- Chapter 6 Ken Pfeuffer and Hans Gellersen. 2016. Gaze and Touch Interaction on Tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (UIST '16). ACM, New York, NY, USA, 301-311. DOI: <http://dx.doi.org/10.1145/2984511.2984514>

# Contents

Abstract	ii
1 Introduction	1
1.1 Direct Manipulation	1
1.2 Eye-tracking	2
1.3 Extending Touch with Eye Gaze Input	3
1.4 Design Space and Thesis Outline	5
1.5 Methodology	6
1.6 Contributions	7
2 Related Work	9
2.1 Gaze Interaction	9
2.1.1 Explicit Gaze Interaction	10
2.1.2 Implicit Gaze Interaction	14
2.2 Direct and Indirect Interaction	17
2.2.1 Direct Input	18
2.2.2 Indirect Input	19
2.3 Summary	21
3 Gaze-Touch: Indirect Input by Combining Gaze with Multi-touch On the Same Surface	22
3.1 Design Space	24
3.2 Applications	26
3.3 Application: Image Gallery	26
3.3.1 Switching between Gaze-Touch or Direct-Touch	26
3.3.2 Interaction Techniques	27
3.3.3 Implementational Details	28
3.3.4 Study Design	28
3.3.5 Apparatus	29

3.3.6	User Feedback . . . . .	29
3.3.7	Summary . . . . .	30
3.4	Application: Paint . . . . .	30
3.4.1	Interaction Techniques . . . . .	31
3.4.2	Implementational Details . . . . .	32
3.4.3	User Feedback . . . . .	32
3.4.4	Summary . . . . .	33
3.5	Application: Map Navigation . . . . .	33
3.5.1	Implementational Details . . . . .	33
3.5.2	User Feedback . . . . .	34
3.5.3	Summary . . . . .	34
3.6	Application: Multiple Objects . . . . .	35
3.6.1	Implementational Details . . . . .	35
3.6.2	Interaction Techniques . . . . .	35
3.7	Discussion . . . . .	36
3.7.1	Limitations . . . . .	37
3.8	Conclusion . . . . .	38
4	Gaze-Shifting: Direct-indirect Input with Pen and Touch Modulated by Gaze . . . . .	39
4.1	Gaze-Shifting . . . . .	41
4.1.1	Design Opportunities . . . . .	41
4.1.2	System Design Considerations . . . . .	42
4.2	Gaze-Shifting for pen and touch . . . . .	44
4.3	Application . . . . .	44
4.3.1	Toolbars . . . . .	44
4.3.2	System Implementation . . . . .	45
4.4	Direct-indirect Interaction Examples . . . . .	46
4.5	Pen . . . . .	46
4.5.1	Draw Directly, Configure Pen Indirectly . . . . .	46
4.5.2	Trace Line Directly, Edit Line Indirectly . . . . .	47
4.5.3	Trace Line Directly and Indirectly . . . . .	49
4.6	Touch . . . . .	49
4.6.1	Select Colour Directly, Set Colour Indirectly . . . . .	49
4.6.2	Scale Object Directly, Zoom UI Indirectly . . . . .	49

4.7	Pen and touch . . . . .	51
4.7.1	Copy & Paste Directly and Indirectly . . . . .	51
4.7.2	Pen Draws and Edits, Touch Zooms . . . . .	52
4.7.3	Draw Directly, Associate Indirectly (Pen), Select Directly, Transfer and Manipulate Indirectly (Touch) . . . . .	53
4.8	User Study . . . . .	54
4.8.1	Results . . . . .	56
4.9	Discussion . . . . .	57
4.10	Conclusion . . . . .	58
5	Direct-indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction . . . . .	59
5.1	Interaction Techniques . . . . .	60
5.1.1	Analysis of Interaction Properties . . . . .	61
5.2	Experiment . . . . .	64
5.2.1	Research Questions . . . . .	64
5.2.2	Tasks . . . . .	64
5.2.3	Design and Factors . . . . .	65
5.2.4	Participants . . . . .	66
5.2.5	Procedure . . . . .	66
5.2.6	Apparatus . . . . .	66
5.2.7	Statistical Analysis . . . . .	67
5.3	Results . . . . .	67
5.3.1	Task Completion Time . . . . .	67
5.3.2	Accuracy . . . . .	68
5.3.3	Gesture Characteristics . . . . .	69
5.3.4	Parallelism between Pen and Touch . . . . .	70
5.3.5	Spatial Distribution of Input Modalities . . . . .	73
5.3.6	User Feedback . . . . .	74
5.4	Discussion . . . . .	75
5.5	Conclusion . . . . .	77
6	Gaze and Touch Interaction on Tablets . . . . .	78
6.1	User Study . . . . .	79
6.1.1	Study Design and Method . . . . .	79

6.1.2	Results . . . . .	81
6.1.3	Discussion . . . . .	83
6.2	CursorShift . . . . .	83
6.3	Touch, Gaze, and Cursor . . . . .	85
6.4	Browser . . . . .	86
6.4.1	Scrolling and Clicking . . . . .	86
6.4.2	Menu Navigation . . . . .	87
6.4.3	Keyboard . . . . .	88
6.5	Maps . . . . .	89
6.5.1	Zooming . . . . .	89
6.5.2	Minimap . . . . .	91
6.5.3	Bimanual Combinations with Markers . . . . .	91
6.6	Gallery . . . . .	92
6.6.1	Area Scrolling . . . . .	92
6.6.2	Image and Folder Selection . . . . .	93
6.6.3	Bimanual Combinations with Image Dragging . . . . .	93
6.7	Discussion . . . . .	96
6.8	Conclusion . . . . .	97
7	Discussion . . . . .	98
7.1	Summary and Reflection . . . . .	98
7.1.1	Gaze-Touch: Indirect Input by Combining Gaze with Multi-touch On the Same Surface . . . . .	98
7.1.2	Gaze-Shifting: Direct-indirect Input with Pen and Touch Modulated by Gaze . . . . .	99
7.1.3	Direct-indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction . . . . .	100
7.1.4	Gaze and Touch Interaction on Tablets . . . . .	101
7.2	Generalising Across Input Technologies . . . . .	103
7.3	Understanding Direct and Indirect Input . . . . .	104
7.4	Reflection on Eye-tracking HCI Research . . . . .	106
7.5	Generalising to Visual Attention Area Based Techniques . . . . .	108
7.6	Limitations and Challenges . . . . .	109
7.7	Future Work Pointers . . . . .	110

8 Conclusion	112
References	115
Appendix	126
.1 Gaze-Shifting: Direct-indirect Input with Pen and Touch Modulated by Gaze .	127
.2 Direct-indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction . . . . .	130
.3 Gaze and Touch Interaction on Tablets . . . . .	136

## List of Figures

1.1 The direct touch modality becomes indirect by redirecting input to the gaze position. . . . .	4
1.2 Left: eye gaze and touch converge, Right: eye gaze and touch are separate.	4
1.3 Explored design space and outline of the thesis. . . . .	6
2.1 MAGIC pointing: the cursor warps to the gaze area (blue circle), followed by fine manual positioning. Image from Zhai et al. [ZMI99]. . . . .	15
3.1 More expressive input from the same touch position: three examples of users touch on the same touch position, but each time manipulate a dif- ferent target. . . . .	23
3.2 More expressive input to the same target: three examples of manipulating the same target that the user sees, but each time with different touches on the surface. . . . .	23
3.3 Users select by gaze, and manipulate with multi-touch from anywhere (a). This can enable seamless switching between indirect (a) and direct manipulation (b), implicit mode switching during direct-touch tasks (c), zooming into map locations the user looks at (d), and dragging multiple targets that are out of the hand's reach (e). The gray cursor indicates the user's gaze. . . . .	24
3.4 Illustrated differences between Gaze-Touch and direct-touch. . . . .	24

3.5	Indirect-RST: in addition to direct image manipulation (a), users can indirectly manipulate images for easy acquisition of small targets (b), or more precision (c). . . . .	27
3.6	Multi-Image-Dragging: after multiple Gaze-Touch selections, users can drag them out of the pile using a single dragging gesture. Through a dynamic control-display gain, small movements can overcome large distances.	28
3.7	Setup for the user study, consisting of a 27" 1080p multi-touch sensitive surface (a), and the 120hz Eye Follower eye tracking device (b). . . . .	29
3.8	Remote-Colour-Select: a user draws the tree stem directly (a). The user then changes to the 'green' colour by a look at the corresponding menu icon, and a tap (b). The user directly continues drawing (c). The color change takes place the moment users touch down. If users start drawing immediately, users draw with a new color. If users only tap to change color, no line is drawn. . . . .	31
3.9	Copy-Paste-Object: the user can copy an existing object with a single-touch. Usually, a touch on the object leads to dragging. However, when the user looks at the copy icon in the menu (a), and then touches down on the object, the user obtains a copy of the touched object under her finger (b). Then, the user can directly drag the new copy to a desired position (c).	32
3.10	Gaze-Focused-Zooming: users can change their zoom-in position during several zooms without changing the pinching position. . . . .	34
3.11	Multi-Finger-Dragging: users can select the five closest objects to their gaze by touching down five fingers (a). Users can then sort them out at once with a single dragging gesture (b, c). . . . .	36
3.12	Multi-Object-Pinching: when multiple objects were selected (a), a pinching gesture moves the objects to the hand's position (b). . . . .	36
4.1	Gaze-shifting: manual input by the user is modulated by their gaze. Input is direct when the user's gaze is on the touch location (left), but redirected to the gaze location if gaze and touch are separate (right). . . . .	40
4.2	Direct-indirect input enabled by gaze-shifting: a) direct pen input is used to outline an image with straight lines; b) indirect pen input is used for curving of gaze selected lines; c) indirect touch is used to zoom in and out relative to the user's gaze. The grey circles indicate the user's gaze, and the arrows the redirection of input. . . . .	40

4.3	Direct-indirect transitions: successive input can involve four types of transition. In this example, the same task is completed while transitioning across different modes. Alternatively, a designer can associate each type of transition with different semantics. . . . .	42
4.4	Input/output mappings for direct and indirect input . . . . .	43
4.5	Gaze-shifting from a system's perspective . . . . .	43
4.6	The user interface and its toolbars . . . . .	44
4.7	Draw directly, configure pen indirectly: users switch a toolbar's mode by look and indirect touch down (a), and vertical drag (b). Special toolbars are 'brush' that displays one adjustable widget (c-d), 'objects' where a drag down creates the selected shape at the pen's position (e-f), and 'color palette' where 2D dragging switches colours (g-h). . . . .	47
4.8	Trace line directly, edit line indirectly: direct pen input enables drawing straight lines from point to point (a). Transitioning from direct to indirect pen input makes a mouse cursor appear at the user's gaze. Indirect pen hover moves this cursor (a), indirect pen down selects a specific point of a line (b), and pen dragging moves this point which adjusts the roundness of the line (d). . . . .	48
4.9	Trace line directly and indirectly: direct tap sets an outline point (a). The user lifts the pen and looks at the next point, which extends the line to the gaze (b). Indirect pen down takes over (c), and pen dragging refines the point's position (d). Pen up confirms the point and initiates a new point snapped to the user's gaze (e). Indirect pen down (f) and drag refines the next point (g). Pen up and look back shapes the line to the hover position (g), and direct pen down sets the next outline point (i). . . . .	50
4.10	Select colour directly, set colour indirectly: select the colour by looking at the toolbar (a) and directly tapping on the colour (b), and set the colour by looking at the target and an indirect tap (c). . . . .	51
4.11	Scale object directly, zoom UI indirectly: users can globally zoom the UI with indirect touch (a-b), locally zoom objects with direct touch (c-d), and immediately return to global zooming (e-f). . . . .	52

4.12	Five techniques for copy & paste: direct copy & paste (a), direct copy & indirect paste (b), indirect copy & direct paste (c), indirect copy & indirect paste at the same position (d), and indirect copy & indirect paste at another position (e).	53
4.13	Pen writes and edits, touch zooms: A user first directly traces with the pen (a), performs indirect gaze-directed zooming with touch (b-c), and then performs line editing with indirect pen input (d).	54
4.14	Users can directly draw on notes with the pen (a). Then, users can transfer notes by directly touching the note (b), looking at the transfer destination (c), where it is transferred to (d). Next, users can create associations between notes, by looking at a note and indirect pen down (e), looking at the second note (f), and pen up (g). Lastly, users can perform indirect touch manipulation on notes (h).	55
4.15	Used shapes in the user study for tracing and puzzle tasks	55
4.16	Study tasks for pen (a), touch (b), and both (c)	55
4.17	Likert ratings of gaze-shifting across the study tasks	56
5.1	We investigate the indirect gaze-touch and pen-touch techniques in comparison to direct-touch for use in bimanual interfaces.	60
5.2	Gaze-touch: from an overview medical image, users quickly zoom into their gaze position to then use the pen for annotations. The user's gaze position is indicated with the green circle.	61
5.3	Pen-touch: While users are drawing a line, users can zoom into the pen's position and then precisely finish the line drawing.	61
5.4	Simultaneity: each techniques has different feasibility for simultaneous interaction on the same or on separate targets.	63
5.5	Dynamic Targeting: With indirect touch techniques such as gaze-touch, users can change the target during the gesture without lifting fingers.	63
5.6	Substeps of the two study tasks on the example of the direct-touch technique	65
5.7	System setup: pen and touch display (a), user's multi-touch input (b), pen input (c), eye tracker (d).	67
5.8	Users performed comparatively in time, and were more accurate with the indirect techniques.	68

5.9	Zoom-accuracy over time during zoom-in gestures for the sequence task, revealing that indirect techniques (pen-touch, gaze-touch) have constant accuracy during zooming, but direct-touch's accuracy decreases over time.	69
5.10	Zoom-accuracy over time during zoom-in gestures for the parallel task, revealing that indirect techniques (pen-touch, gaze-touch) have constant accuracy during zooming, but direct-touch's accuracy decreases over time.	70
5.11	(a, b) Zoom-in gesture duration and count, indicating different zoom behaviour between direct-touch and the indirect techniques. (c) How parallel users performed with the pen and touch modalities. . . . .	70
5.12	How distant users kept each modality-pair. . . . .	71
5.13	Sequence task parallelism: Parallelism of the pen and touch modalities over the duration of each trial. Parallelism denotes the minimum movement in cm that both modalities share. Trial durations are normalised to 0-100% of time. . . . .	72
5.14	Parallel task parallelism. . . . .	72
5.15	Illustrative heatmaps of the users' average touch (red), pen (blue), and gaze (green) positions across the screen. . . . .	73
5.16	Mean Likert Scale responses from participants . . . . .	75
5.17	User rankings after the tasks indicate a preference toward gaze-touch. . .	75
6.1	We explore gaze and touch input for tablet interaction during grip, e.g. to enable the user to scroll a webpage with direct-touch (a), and temporally use a cursor to click hyperlinks (b) with the same hand. A two-handed grip allows a comfortable two-thumb based zoom, while the zoom location is specified with gaze (c). Our gallery application allows users to open an image simply by looking at it and a tap from the gripping hand (d). Or, to drag images through direct-touch, while users indirectly scroll the view they look at (e). The red circle indicates the user's gaze in all figures. . .	79
6.2	User study: users interacted with the tablet placed on a table (a), and performed a task that resembled homescreens of tablet UIs (b). . . . .	80
6.3	Quantitative study results . . . . .	81
6.4	Gaze-to-target distance over time, averaged across all tapping tasks. As expected, users users initially look far away, then focus the target. With gaze-touch, users focus on the target for longer. . . . .	82
6.5	Likert-scale ratings . . . . .	83

6.6	CursorShift is a method to temporally enable a cursor, without interfering with default direct-touch actions. . . . .	84
6.7	State model of CursorShift. . . . .	84
6.8	Target selection with the three investigated techniques. . . . .	85
6.9	Illustration of bimanual configurations of direct-touch, gaze-touch, and cursor. . . . .	86
6.10	Click an element with direct-touch (a) or with a cursor (b). . . . .	86
6.11	A user scrolls the page (a), toggles the cursor to click a link (b), then continues scrolling the next page (c). The system uses the user's eye gaze direction, that is indicated by the red circle, only to position the cursor in (b). If users accidentally tap and get a cursor, users can simply tap again on empty space to remove it. . . . .	87
6.12	In a two-handed grip, users can divide cursor control and website scrolling. The right hand scrolls (a), the left hand clicks a hyperlink (b). The blue circle indicates the invisible touchpad area. . . . .	87
6.13	Menu navigation using gaze-touch, i.e. indirect touches that can rapidly navigate the tabs when visually focusing on the menu. The red line indicates the indirect handle to the remote point. . . . .	88
6.14	The virtual keyboard allows to use one hand to hold the device, and a second for direct-touch key input (a). Or, users hold the device and with the same hand issue gaze-touch input to select keys (b). . . . .	89
6.15	Our maps application supports three zoom variants: direct pinch-to-zoom (a), two-thumb gaze based pinch-to-zoom (b), and one-thumb variant that uses a 3-part widget for thumb-sliding (c) to zoom at gaze position (d). . . . .	90
6.16	By gaze-touch on a minimap, users pan long distances with a single drag gesture from remote. . . . .	91
6.17	Three bimanual combinations, each demonstrated in their own task: direct marker dragging with indirect minimap control (a), direct panning with cursor based marker control (b), and indirect zooming with cursor based marker input (c). . . . .	92
6.18	In the gallery and single image UIs, scroll gestures affect the view the user is looking at. The scrolled views are highlighted with an orange border. . . . .	93

6.19	A dynamic switching mechanism allows direct- and gaze-touch image navigation. With direct-touch, users physically touch the image and the ‘back’ button (a-b). With gaze-touch, users can comfortably use the same finger to do the same actions without using a second hand (c-d). . . . .	94
6.20	Illustrating the input flexibility with gaze and touch: the same image drag & drop task (rows) can be performed with three different techniques (columns). . . . .	95
7.1	Example input technologies that could support direct and indirect input modes. . . . .	104
7.2	Generalised principle of utilising visual attention area as a technique. . . .	109

## List of Tables

3.1	Summary of the differences of direct-touch to Gaze-Touch. . . . .	24
4.1	Overview of the direct-indirect interactions ( <i>rows indicate the task, columns the integrated modalities</i> ) . . . . .	46
5.1	Summary of differences between techniques . . . . .	62
5.2	The techniques share the overall division of labour, and vary for the ‘select’ sub-task during touch gestures. The ‘manipualate’ part is touch only across all techniques to support all standard touch gestures. . . . .	62
5.3	Study distance factors (for both tasks) in three metrics. . . . .	66
6.1	Tap gestural differences between the techniques. . . . .	83
6.2	Summary of the interaction techniques’ properties. . . . .	85
8.1	Relationship between computing era, input devices, and input type. . . . .	113

# 1

## Introduction

This thesis explores interaction techniques that enhance a user's capability to interact with computers in novel and efficient ways. A multimodal interface is the main component in this exploration, formed by combining the eye gaze modality with manual input such as multi-touch, pen, or hand gesture. The goal is to better understand its usefulness and usability in common human-computer interaction contexts. In particular, this thesis presents novel concepts that allow designers and manufacturers of user interfaces to extend the familiar direct input paradigm, that, e.g., is widely adopted on touchscreen devices, with advanced eye gaze based interaction techniques.

### 1.1 Direct Manipulation

Interacting with computers through direct manipulation is an integral part of our life. For example, multi-touch emerged as the dominant input paradigm over the last decade across a variety of computing devices. The technology allows users to provide direct input to a computer through physical contact on the screen, contrasting previous input devices such as a mouse [SPS92]. The directness of the interaction affords ease of use and is practical for many kinds of interactive surfaces ranging from hand-helds, tablets, booklets, desktops, table-tops, and wall displays [B<sup>+</sup>07, Rek02, RWA96]. Direct gestures are based on analogies from the real world, making them highly intuitive to use. Most of these gestures are covered by tap, drag, and pinch actions, providing users a simple way to accomplish much of their work with touchscreen devices.

Despite its advantages, direct manipulation has conceptual properties that make it less practical in some other situations. By using direct input, users locate their hand or an input device physically on the object of interest. Although this provides the benefit of having proprioceptive feedback from the hand, at the same time it can detract the interaction when the hand occludes parts of the area of interest [VB07, VB10]. This refers to the fat finger problem, where the size of the finger occludes the target, and reduces the precision of touch [BWB06, HB10, CAG12]. Another factor to account for is comfort during direct manipulation. Ease of use from direct input increases comfort,

however, it can similarly decrease comfort in cases involving frequent interactions as extensive hand/arm movements cause physical fatigue [KAB<sup>+</sup>12, KAB<sup>+</sup>14, SZ11].

The pros and cons of direct manipulation should be considered in comparison to *indirect input*, usually enabled with traditional input devices such as a mouse or a touchpad where the user's manual input space is separated from the output display space [FWC84, BL00, HW12]. Direct and indirect input represent fundamental categories of input technology, with contrasting implications on the design of input devices, interaction techniques, and user interfaces. Direct input resembles real-world physical manipulation that lends itself to create intuitive and easy interactions, to an extent not possible with indirect input. However, indirect input provides benefits where direct input is limited (and vice versa). The separation of input/output surface solves the occlusion problem as users don't occlude the content with their hands [VB07, VB10]. In addition, flexible input/output mappings increase selection precision [AZ03, BWB06, PWS88], and there is lower physical effort without the necessity to reach out across the UI for each selection task [KAB<sup>+</sup>12, KAB<sup>+</sup>14, SZ11].

## 1.2 Eye-tracking

For its wide application to a variety of computer interfaces, advances for direct interaction will be highly beneficial for the many users of such an input paradigm. It is therefore important to explore potential new ways of interaction that work in unity with the standard direct manipulation paradigm. A technology that we investigate for the potential to do so is eye-tracking. It provides the capability to sense the user's eye information, and with it infer the position users are looking at on a computer display [Duc07]. This can be taken for user input for pointing at graphical elements in an user interface [Bol81, Jac90]. The core interaction is looking at a target on a screen to select it, which aligns with the natural human eye behaviour of looking at the object we are interested in [V<sup>+</sup>03].

Eye gaze input has been extensively studied in context of human-computer interaction performance [Duc02, Ray98]. Eye movements are one of the fastest movements of the body with minimal physical fatigue. User studies have shown that use of gaze pointing can be faster than manual input devices [ZMI99, SJ00, TJ00]. Pointing techniques with intelligent integration of eye gaze pointing within manual input devices allows for reduction of pointing effort and higher selection speed [ZMI99]. We can use our eyes to interact over a distance [TBG11, SD12b], and in combination with other modalities [ZMI99, Jac90, Bol81].

Beyond the benefits, eye gaze interaction also comes with technical and conceptual challenges that need to be carefully considered in context of a given human-computer interaction task. As the eyes are primarily in use for visual inspection, using them as an input method can overload the user's cognitive abilities [ZMI99, SD12b]. This *double role* issue is also linked to the *Midas Touch* problem [Jac90]: it is ambiguous whether a user's gaze on a target is intended to select it, or whether the user simply looks at it without intention. An additional method of confirmation is needed to solve these ambiguities, such as dwell-time or button-click [Jac90, MMAR06]. Another issue is eye gaze inaccuracy, as eye data quality may be affected from a variety of factors including technical issues of the tracker and physiological characteristics of the eyes [KKP<sup>+</sup>08, FWT<sup>+</sup>17]. However,

current eye-tracking hardware is rapidly advancing toward precise sensing capabilities of the user’s act of looking, pointing us toward a future where any computer has eye-tracking capability integrated by default [Bul16, FWT<sup>+</sup>17]. In that case, a particular interest is how we can take advantage of this technology within the existing user interface paradigms that users are familiar with, such as direct manipulation.

### 1.3 Extending Touch with Eye Gaze Input

To explore how eye gaze can work together with direct touch input, this thesis is concerned with the exploration of interaction possibilities from combining the two modalities. The goal of the exploration is to provide users with usable and efficient capabilities to interact with computers by designing novel interaction techniques based on hybrid gaze and manual inputs. Early work considered gaze pointing with manual input from devices such as controller, keyboard, or mouse [Bol81, Jac90, ZMI99]. Gaze and manual (touch) gesture in particular has received increasing interest in the last decade, with multi-touch devices becoming ubiquitous in everyday life. Prior research showed this input combination allows to bridge the distance between user and display, with eye gaze acting as a pointer to remote displays [SD12b, TAB<sup>+</sup>13].

The research in this thesis focuses on extension of the direct touch paradigm in general, whether the UI is nearby or far away. Consideration of nearby interfaces is particularly important as it aligns with how multi-touch devices are commonly used at present. This leads to different implications on the user interface design — in essence, use of eye gaze on everyday touchscreens raises the question of how eye gaze can work together with the default direct manipulation paradigm. This calls for a consideration of the natural roles of the modalities into the design of interactions with computers. Eye gaze is considered as an ideal indicator of the object of interest [V<sup>+</sup>03], and is thus used as a method of pointing, to select a target. The user’s hands are particularly fit for expressive spatial manipulation [HPGK94], and therefore manual gesture is used as a method to manipulate targets. Taken together, this yields a hybrid technique where users look at a target by gaze, and manipulate this target using their hands. This represents a fundamental change of the default manual way to interact with computers, with unique implications on the design of user interfaces that integrate eye gaze and manual input. To better understand the design of such interfaces, this thesis is guided by the following propositions.

First, **gaze input can transform a direct input modality such as touch to an indirect modality, and with it provide new and powerful interaction capabilities**. Users interact by looking at a target, and issuing touch gestures (Figure 1.1). Using gaze based interaction, users can still perform the same type of manual gesture as always, but gestures now affect the object of visual interest instead of where they physically perform the input gesture. The rapid speed of eye gaze pointing is unified with the rich gestural possibilities of the user’s hands. As a result, the simplicity of gestures remains by supporting the basic set of tap, drag, and pinch that cover most of the relevant interactions that users perform through direct manipulation, for example on touchscreens. This characteristic is crucial for the design of efficient interaction techniques on touchscreens. Potential new eye gaze technique in this context should be designed to support the same simplicity as the status-quo input paradigm to minimise learning effort and maximise ease of use.

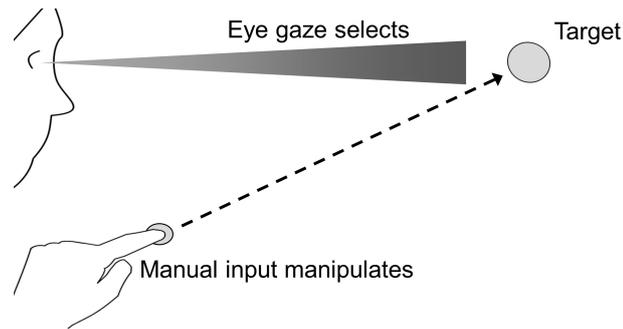


Figure 1.1: The direct touch modality becomes indirect by redirecting input to the gaze position.

We develop this proposition through the design of prototypical systems that allow users to interact with new interactive capabilities. The developed systems cover the extensive design of numerous interaction techniques and fusing of various input modalities on stationary and mobile devices. We also provide user evaluation to support the propositions through a detailed view on usability and user performance with the proposed interactions. The breadth of proposed interaction examples is informally evaluated through user testing and feedback. In depth, we empirically evaluate gaze and hand input in contrast to hand only input through rigorous user study of selected tasks.

Second, **hybrid gaze and touch user interfaces enable new ways to combine direct and indirect forms of interaction**. This is particularly relevant in context of the prior understanding of direct and indirect input in HCI. We normally think of direct and indirect interaction as very different styles, as choices we make. They are normally totally juxtaposed: people promote one or the other. However, by utilising the eyes we can relax this strict relation, and combine direct and indirect input. This is accomplished by eye-tracking during manual interaction in order to implicitly distinguish two different interaction contexts. The first context occurs when the user's visual attention lies close to their touch input, and the second context occurs when it lies somewhere else (Figure 1.2). From a system's perspective, this provides the capability to implement a different system reaction depending on whether the user sees their manual input, or not. For example, to use direct manipulation when the user visually attends to their input, and indirect input otherwise. From a user's perspective, the user can exploit the two contexts to use direct input for nearby objects, indirect input on objects at a distance, and switch between them by a glance. Therefore the interaction technique retains the strength direct manipulation, as users can employ the default way of manipulation but at any time gain advanced eye based interactions.

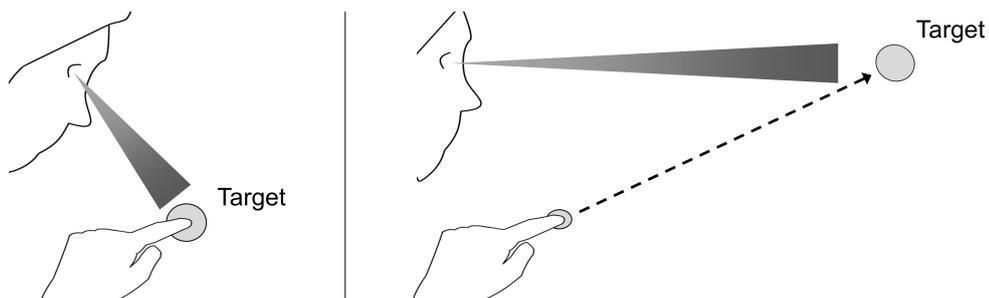


Figure 1.2: Left: eye gaze and touch converge, Right: eye gaze and touch are separate.

This proposition is developed through the design of systems that allow distinction of these two contexts, and with it allow exploration of many examples for this multimodal input as a demonstration of its generality. We first propose Gaze-Shifting as the general mechanism to allow such an input switching, that is then explored in context of a design application. The application supports typical interactions using a stylus, or multi-touch, and users gain the ability to use direct-indirect input with both modalities. We further extend this principle to a tablet device, showing that mobile and commercial devices allow similar application. Here, users can either interact with an application using direct-touch, but at the same time users can take advantage of Gaze-Touch inputs for unimanual interaction with the whole screen using the hand that holds the device. This is shown through the design of prototype map navigation, browser, and image gallery applications on a tablet.

## 1.4 Design Space and Thesis Outline

The main focus of this thesis is the exploration of the propositions as introduced above. Each chapter in the thesis therefore adds to the exploration of both propositions in the development of a new technique, in context of a new scenario, or to demonstrate use cases.

Each chapter also has a primary focus. The first three chapters explore specific points of the design space of direct and indirect input. Here we consider direct input as the default way of interaction with a touchscreen, and indirect input as the technique where manual input is redirected to the user's gaze target. Figure 1.3 illustrates the particular design space. The combinations include using direct as indirect input (a), switching between direct and indirect input (b), and using both at the same time (c). The fourth chapter (d) focuses on application of these concepts to a currently widely successful multi-touch device in form of a mobile tablet. In the following, the chapters are described in more detail.

- **Chapter 3: Gaze-Touch: Indirect Input by Combining Gaze with Multi-touch On the Same Surface** explores the use of indirect touch that affects the target the user looks at. Usually, modalities such as touch are exclusively used for direct input. In this chapter, we regard new interaction possibilities when using indirect touch input that is enabled through redirecting touches to the user's eye gaze position. This allows to increase our understanding of potential benefits and pitfalls when using gaze based indirect touch systems.
- **Chapter 4: Gaze-Shifting: Direct-indirect Input with Pen and Touch Modulated by Gaze** is about the the combination of direct and indirect inputs using the same modality. Normally, users either use only direct input such as with touch, or only indirect input e.g. with a mouse. By enabling users to shift between the two modes, they can take advantage of the best of both worlds. Implicit eye gaze input is used to facilitate this shifting: the system automatically enables indirect input when the user looks at a remote target. The work in this chapter contributes to understanding how direct and indirect inputs can be integrated within the application context, and what interactive capabilities are afforded by using such a system.

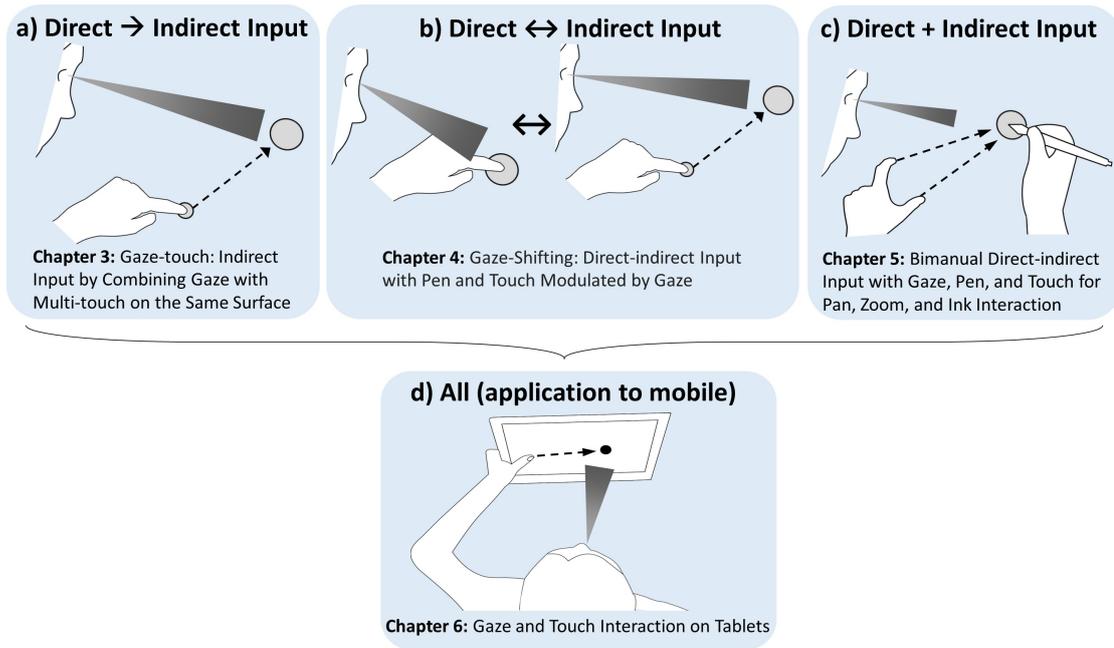


Figure 1.3: Explored design space and outline of the thesis.

- **Chapter 5: Direct-indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction** investigates the simultaneous use of direct and indirect input. Bimanual input is usually either both hands exclusively direct, or indirect. Using eye gaze, we can enable a hybrid variant where one hand employs direct, and the other indirect input. This chapter explores this technique and provides an extensive user study. The work contributes to a better understanding of user performance using hybrid eye gaze and manual interaction for two handed tasks.
- **Chapter 6: Gaze and Touch Interaction on Tablets** takes a step further and explores the application of all direct-indirect relationships on a tablet, a popular personal multi-touch device. Tablet use usually requires two hands, one hand holding the device and the other issuing input. Here gaze and touch can enable single-handed use, by using indirect touch from the hand that holds the device. This interaction capability is studied in a controlled experiment, and explored in applications for uni- and bimanual interaction tasks. The work extends the knowledge on user performance on eye gaze and touch based interaction placed within the context of everyday tablet computers.

At last, **Chapter 7: Discussion** provides a reflection on the conducted research, implications on user interfaces using gaze and touch, and discusses the work in context of prior research. **Chapter 8: Conclusion** presents the concluding remarks and points to future work of eye gaze and manual interaction.

## 1.5 Methodology

In general, the goals and research direction of the propositions are primarily idea-driven, i.e. in most of the cases we consider the given interaction context of eye gaze and manual

input interface and explore various possibilities that fuse the modalities in specific ways to aid the task at hand, with the aim to allow users (more) efficient interaction with the computer.

In particular, the work on the propositions involves the following two specific research methods centered around design and prototyping of human-computer interactions. In addition, we employ a third research method of laboratory study to empirically evaluate proposed interactions.

The first is a design space analysis [MYBM91], a description of the possibilities within the design of a user interface that is based on the respective interaction techniques. Particularly with eye gaze, new possibilities emerge that we systematically analyse regarding interaction properties that are specific to the technique and commonly used in the literature. We refer to Figure 1.3 that summarises the content of the design space, including direct as indirect input (Chapter 3), switch between direct and indirect input (Chapter 4), and bimanual direct vs. indirect input (Chapter 5), and direct and indirect input on mobile devices (Chapter 6).

The second research method is the exploration of the design space through prototyping interaction techniques, user interfaces, and applications for the used system in context. This method is closely coupled with the design space analysis, but is more steered to the technical design of new kinds of interaction artefacts [Fal03]. Artefacts can be basic interaction techniques or whole user interfaces, and in this work also heavily based on application probes [HMW<sup>+</sup>03]. In essence, it is unclear how a new artefact applies to the conventional user interface, and how it integrates within a given application context. Prototyping different examples allows to explore and assess new capabilities quickly, and with it broaden and refine the design space. Chapter 3, 4, and 6 mainly follow this research method, and chapter 6 partly does as well.

The third research method is the laboratory study [DFAB98], i.e. testing an interaction technique with users within a controlled environment, often in comparison to a baseline technique. All chapters involve user studies but with different extent. Chapter 3, 4, and 7 provide informal study and focus on qualitative user feedback of the developed applications. Chapter 5 focuses on a user study of bimanual techniques in a controlled environment for both qualitative and quantitative performance measures. The first part of chapter 6 also provides a controlled study comparing gaze vs. touch on a tablet device, including both qualitative and quantitative performance measures.

## 1.6 Contributions

The contributions in this thesis include the following points.

First, we present and develop a fundamentally new idea of combining eye gaze and manual input. Eye gaze enables use of an additional indirect input mode on normally direct input modalities, and with it provide unique interaction capabilities. The technique supports distinction of two interaction contexts of where the user's visual attention is on the manual input or somewhere else, allowing the system more flexibility to react to a user's specific eye-hand interaction. This contribution is supported by the following sub-contributions.

- Introduction of the Gaze-Touch interaction technique where each single touch input affects the user’s gaze target. We analyse the properties that distinguish this technique to the default direct-touch input paradigm, and provide various prototype applications that showcase beneficial use cases of Gaze-Touch interactions, and evaluate them informally.
- Provision of interaction concepts that combine eye gaze and digital pen interactions for exploiting high precision drawing and pointing for interaction with the area indicated by the user’s visual attention.
- Empirical evaluation of Gaze-Touch based zooming in the context of a bimanual interaction task, showing comparable performance as the direct baseline, and extension to bimanual design/navigation tasks.
- Extension of the Gaze-Touch concept to tablets (that is different as users hold the device), enabling users whole-screen reachability with only the holding hand, supported by an empirical study on Gaze-Touch vs. direct-touch.

Second, we present new ways of combining direct and indirect forms of interaction. Normally direct and indirect inputs as very different styles, isolated from each other through different input devices and techniques. We show that by exploiting eye gaze, direct and indirect input can be mixed together into novel and innovative interaction contexts. We show this on a range of input and output devices.

- Use of direct input as indirect input is demonstrated by the Gaze-Touch work on the example of the multi-touch input paradigm. Following applications show initial design strategies how gaze complements direct-touch.
- Switching between direct and indirect input with eye gaze is introduced as the Gaze-Shifting technique, on a system that supports this mechanism with both pen and touch input modalities for interactions within a design-oriented application environment.
- Simultaneous direct and indirect input is investigated in our work on bimanual interaction in a compound drawing/navigation task. An empirical study is presented showing comparative performance to the baseline of direct input, and further providing insights into interaction benefits for bimanual navigation tasks.
- The application to tablets explores bimanual direct-indirect interactions in a different context, where gaze based touches of the holding hand merge with direct input of the free hand for various interactions involving image organisation of a gallery, map navigation, and web browsing.

Overall, we demonstrate the generality of the conceptual work as presented by the propositions by developing numerous interaction techniques and application demonstrators across the chapters, first on conventional multi-touch surface, and then showing how they apply and extend to stylus and mobile tablets that are distinct for the different interaction context.

# 2

## Related Work

This thesis investigates gaze interaction on touchscreens in the field of human-computer interaction. The gaze modality is considered for explicit input, to select and manipulate targets, but also for implicit input, to indicate whether manual input is direct or indirect. We therefore start with reviewing work in gaze interaction for both categories of implicit and explicit use of the modality. We then review the literature in manual touch interaction, considered from the perspectives of direct and indirect touch input.

### 2.1 Gaze Interaction

Eye-tracking research goes back to before the 19th century, where early tracking devices were used as a tool to better understand the perceptual and cognitive processes involved in human eye behaviour [Ray98]. In HCI, eye-tracking is investigated as a method to interact with computers using their eye movements. The basic eye movements of humans include fixations and saccades [Car77]:

- Fixations occur when a user is fixating on a point, i.e. looking at a point steadily. They typically last between 200-600 ms, and their purpose is to acquire a still image of the scene for neural processing. A fixation has still small, jittery micro-movements [CK08] around the fixated point which is why they are considered an eye movement.
- Saccades are rapid eye movements that happen between fixations, i.e. when looking from one point to another point in the scene. Movement speed can go up to 500 degrees of visual angle per second, within a usual time frame of 20 to 120ms. During a saccade, humans do not acquire visual information of their environment.

These eye movements can be tracked by using eye-tracking hardware. Different systems exist, but most of the current trackers are based on optical tracking. The principle operation of those methods is using infrared light that is projected to the estimated eye area. The light from the eye is reflected back from the retina, and sensed by a video

camera. This information is then used to determine the eye’s rotation, which indicates the eye direction and movement over time. See Hansen et al.’s survey for more details about technical aspects of eye-tracking [HJ10].

Eye based human-computer interaction usually involves an intelligent algorithm that interprets the user’s eye movements received from an eye tracker. Most of the techniques are based on the user’s ability to look at a point (fixation), often called eye gaze, gaze direction, gaze input, or simply gaze. Numerous gaze interaction techniques were designed, implemented, and evaluated in the literature. We review related papers from two perspectives: with regards to how *implicit* vs. *explicit* the user employs their eye gaze for an interactive task.

### 2.1.1 Explicit Gaze Interaction

Eye gaze can be used as a main modality of input, where the user explicitly looks at a target to select it on the computer display. The unique characteristic of explicit gaze interaction is therefore the act of pointing with the eyes that represents the main part within the conducted human-computer task. These include not only interaction techniques where users interact by only gaze, but also where users use their gaze to point, and supplement this with additional modalities.

#### Initial research

The first system that integrated gaze for computer input was *Gaze-orchestrated Dynamic Windows* by Bolt in 1981 [Bol81]. The system presented a user interface that consisted of multiple windows, each gaze-interactive. A window showed a video, with the idea of linking multiple TV channels to one large display. A video was active/running as long as a user was looking at it — the others were automatically paused, reducing visual and cognitive overhead of the multimedia installation, while enabling rapid access to any video. Bolt further discussed challenges of the system, such as zooming in and out of windows. A dwell-time technique, where a user looks at a window for a fixed time (e.g. 300 ms) to trigger an effect, was considered as a method based on the eyes. Further, using manual input (in form of a joystick) or speech (through an speech recognition device) were also considered to allow the user to explicitly trigger what they look at.

The first study of gaze interaction has been presented by Ware and Mikaelian in 1986 [WM87]. In this work, the main concerns were target selection (method to confirm selection of a viewed target), and target size (size suitable for gaze pointing). Experiments on target size are motivated by the fact that eye movements inhere jittery motion even when fixating a point [CK08]. Three target selection techniques were subject of comparison in the study: a button press, dwell-time, and an on-screen button. The on-screen button technique is an interesting alternative: users would look at the target of interest, then at a special screen button, essentially using two successive saccades as the selection (this technique was the an interesting initial variant of gaze gestural input that has been revisited later, e.g. by Drewes et al. [DS07] or Lutteroth et al. [LPW15], that we will also discuss that later). The results of the study showed two aspects that can be considered as relevant for most works in gaze interaction. First, all gaze techniques showed task completion times faster than other modalities as reported in other work. This depicts

the main advantage of gaze: a very fast selection speed. Second, however, gaze input is only fast for large targets — if targets are small, speed declined and more errors happen.

In 1990 Jacob investigated how gaze interaction techniques better integrate in conventional user interfaces, based on the input principle *What You Look At Is What You Get* [Jac90]. He argued that the main barrier is not the technology, but rather the study of techniques that can naturally integrate gaze within the human-computer dialogue. In this paper, he introduced the Midas Touch problem, that describes the ambiguity issue of selecting a target with gaze. When a user looks at an object, it is unclear whether the user wants select the object, or only looks at it without intention. Finding a mechanism to explicitly distinguish these two states has been subject of many research efforts we discuss below. Aligning with Bolt’s explorations, Jacob suggests either using dwell-time or a manual confirmation. This depends on the application context, as Jacob emphasises through his exploration of several application scenarios. In particular, manual input e.g. a button press is faster as users do not need to “wait” for the dwell-time to finish — it’s good for repetitive tasks. However, in other instances, it can be more convenient to use dwell-time as it frees the hand from input — it reduces user effort. Yet for situations where an action is difficult to undo, manual confirmation is definitely preferable. Jacob pioneered many other use cases for gaze input in this paper, such as moving objects (simply look at the destination), text scrolling (automatic scrolling based on the gazed area), menus (where combined gaze and button press is more useful), or window control (always activating the fixated window).

10 years later, Sibert and Jacob revisited the work for a controlled evaluation of this kind of gaze interaction [SJ00]. At the time, the mouse has already become the standard pointing device for computers, hence the interest of using the mouse as a baseline for comparing gaze input (using dwell-time selection). Across two selection tasks in the study, it was found that users performed faster with gaze than mouse, making the point that gaze can provide natural eye based inputs without incurring any performance penalty.

Contrary results were found by Miniotos in his investigation of how Fitts’ Law applies to eye gaze [Min00]. In the study reported, the dwell-time based gaze technique was found slower than mouse input. However, they discuss that it might be affected by their gaze signal smoothing, that makes pointing more reliable but also introduces a lagging effect that slows down performance. They conclude that optimal gaze data filtering, as well as the optimal dwell-time, needs further study for practical use cases. Later work indeed studied gaze signal (c.f. [KKP<sup>+</sup>08, ZRZ08]) and dwell-time configuration (e.g. see [MR02] or [MMAR06]).

Researchers continued to explore how gaze integrates into conventional user interfaces. Lankford provided a holistic system that integrates eye gaze input into the Microsoft Windows operating system, dubbed Eye-gaze Response Interface Computer Aid (ERICA) [Lan00]. It provided the tasks of clicking and typing with gaze-only input, and with it enabled control of a range of tasks within the operating system. Clicking is enabled through dwell-time selection. In order to select small targets, the first dwell on an area was expanded, that could be followed by a second dwell to select targets. The dwell-time technique also enabled typing on a virtual keyboard, by fixating on the keys for a predefined time. Overall, this provided a complete interaction style for interacting with an operating system through enabling dwell-time based clicking and typing.

### Small target selection

Improving small target selection was further investigated by researchers, when interacting with the combination of gaze and manual input. Miniotas et al. investigated how target expansion works for gaze interaction [MvM04]. Target expansion is a method where the size of the targets that users are pointing at gradually increases to ease selection of small targets [MB02]. Applied to gaze, the target would expand as long as the user keeps looking at the target. Across two studies they found that this method reduces gaze selection error, but it also comes with an increase in movement time. They conclude that with gaze with expanding targets can render the eyes a more suitable input modality when the precision of eye-tracking might be improved in future. However, note that target expansion is not suitable for generic UIs, because the space occupied by the expanded area is not available for other interactive objects.

Kumar et al. proposed a variation of the above approach to enable small-target interaction [KPW07]. Rather than expanding the target, they propose to expand the region of the user interface similar to the ERICA system [Lan00], but with manual input. They propose the EyePoint technique that is based on a *look-press-look-release* procedure. The user first looks at the target region and presses a button to magnify this region. Then, the user looks at the target within the magnified region, and releases the button to select it. In the study, they compared this approach to mouse input, where the time performance was similar in both conditions. The use of gaze however led to more errors, that could be accounted to either the technique, or the hardware. They state that the error rate varied significantly across participants, and for those where eye-tracking was poor, error rate increased. Nonetheless, their approach represents a practical technique that enables gaze interaction for both small/big targets within WIMP based interfaces.

Following up on the issues of inaccuracy, researchers developed intelligent methods to improve the gaze signal data [KKP<sup>+</sup>08, ZRZ08]. Signal smoothing algorithms were proposed such as threshold saccade detection, Kalman filter, early trigger correction (Kumar et al. [KKP<sup>+</sup>08]), force field, or speed reduction (Zhang et al. [ZRZ08]). These methods can reduce the noise and thus improve the user's gaze pointing abilities. Kumar et al. also analysed issues of gaze/manual input synchronisation: the problem of users clicking just before or after looking at a target. To approach this issue, they suggest using temporal corrections toward the most recently viewed target.

### Eye gesture interaction

The majority of explicit gaze work focus on two selection techniques used for gaze pointing: dwell-time (e.g. 0.5 seconds), or manual input (e.g. button press). As an alternative, users can issue explicit eye movements to trigger actions. As mentioned earlier in this section, the on-screen button selection technique in Ware and Mikaekian's study [WM87] is a first instance of such a method: a user would first look at a target, then look at a specific selection button on the screen — when done quickly, users perform a directional 'flick'-like movement with their eyes to select the target. A variation of this technique has also been explored by Isokoski, using off-screen targets for text input with the eyes [Iso00].

Drewes and Schmidt took this approach further in their work on gaze gestures [DS07]. Gaze gestures are directional movements with the eyes, i.e. a saccade, or multiple saccades. Compared to the on-screen eye flicks, gaze gestures are relative, and thus independent of the user interface. The idea of gaze gestures is to issue them complementary to the default computer interactions. Users can normally visually inspect the interface, and interact with it manually — but also users can use gaze gestures, eye movement distinct from normal viewing, to quickly issue commands with their eyes. In their study, they showed that users can perform complex gestures intentionally, and that gestures can be designed to not accidentally occur during normal viewing. Gaze gestures have been further investigated in various contexts such as mobile phones [DDLS07, KAR<sup>+</sup>14], games [HIV12], or wearables [BRT09], for use of basic command triggers when other modalities are unavailable.

### Gaze and touch

With the ubiquity of mobile touch-sensitive devices of our time, researchers also explored various ways to combine gaze pointing with manual input from touchscreens. Most of the research in gaze and touch interaction falls into one of the following three categories: *remote display interaction* (Stellmach et al., [SSND11, SD12a, SD12b, SD13]), *across-display interaction* (Turner et al., [TAB<sup>+</sup>13, TBAG13, TBAG14, TABG15]), and *near display interaction* (this thesis). The following sections detail this research with regards to the categories.

*Across-display interaction*: Turner et al. firstly proposed gaze and touch based techniques in a concept paper [TBG11], motivated by the goal to enable interaction with out-of-reach content. The proposed interaction techniques use gaze to select a position, and manual touch to confirm. The investigated tasks were based on drag & drop, where users first select a target on one (local) place, and then move this to a different (remote) place. The proposed interaction technique uses gaze for the pointing task, and touch to confirm the steps. An example operation includes four steps: look at a target (select), touch down (confirm), look at the destination (drag), touch up (drop). The authors conclude that the techniques and variations of it are promising for interaction with remote displays, and for transfer between local and remote systems.

Their ideas were investigated in detail in a few follow-up studies. In Eye Pull, Eye Push [TAB<sup>+</sup>13], the concept for transfer between public and private device is investigated. They compared variations of gaze based drag & drop variations in a user study where users interact across a tablet and a distant, large display. The results indicate that users are able to use gaze and touch for the tasks, and that a technique where users hold their touch input during drag & drop was most positively received. In a subsequent study they compared gaze vs. manual pointing for drag & drop tasks [TBAG14]. Using a gaze snapping mechanism to counter tracking inaccuracy, they found that users performed with as well as with manual input.

Turner et al's Gaze+RST work focused on using multi-touch rotate-scale-translate gestures to manipulate viewed targets on a large display [TABG15]. Four techniques were developed with varying degrees of gaze and manual input — while supporting two-finger rotation and scaling. In a study, the techniques showed that task distance and added

rotate/scale tasks affect performance, and that gaze and touch can be designed integral for rotate-scale-translate tasks.

*Remote display interaction:* Stellmach et al. focused on interaction with a remote display by using a combination of input from the eyes and a handheld touch-enabled device. The first work looked at the scenario of large image collections, and various gaze-supported multimodal techniques including physical keyboard and a tilt-sensitive mobile device [SSND11]. Users preferred the Look-Touch-Tilt technique for zooming, that combines gaze, touch, and device tilt. This technique was later evaluated for a pan and zoom task, where although mouse zooming was preferred, gaze-supported zooming indicated high potential.

They also considered gaze and touch more generally for selection tasks on distant displays in their Look & Touch work [SD12b]. Based on the principle gaze suggests, touch confirms, they developed a set of gaze and touch interaction techniques. In particular, a user point roughly at the area of interest with their gaze, and then uses touch gestures to refine the position, as inspired by Zhai’s MAGIC pointing technique [ZMI99]. In a user study, all gaze and touch based techniques outperformed the gaze-only approach. In a follow-up paper, they investigated combined selection, positioning, and manipulation tasks in the same setup [SD13]. The corresponding study showed that gaze and touch were faster than head based alternatives for the tasks tested.

*Other:* Research conducted by me and my colleagues looked at gaze and touch interactions on diverse setups as early steps into the work of this thesis. The Gaze+Touch vs. Touch study took an empirical look into user performance of gaze and touch interaction [PAG15]. Gaze and touch input on a remote display was compared to default touch input on a close-range display. Results of the study showed first signs of the potential of this modality combination, indicating that gaze and touch is slower for dragging targets, but as fast and more accurate for scaling and rotation tasks. GazeArchers [PAG16] is a two-player collaborative game where the users shoot arrows by gaze pointing (aim at target) and touch tapping (shoot arrow). This work focused on the design space of multiple users that utilises multi-gaze and multi-touch input.

Researchers also considered gaze and touch interaction on tabletop systems. Holman proposed initial ideas for using gaze input on tabletops [Hol07]. Mauderer et al. implemented a technique with gaze pointing and touch flicking to select remote targets [MDK13]. Newn et al. designed a tabletop UI with different techniques, i.e. normal touch input for the near area, gaze and touch input for the further area, and gaze only for far away areas [NVCV16].

### 2.1.2 Implicit Gaze Interaction

A large part of gaze HCI research is concerned with the implicit use of the gaze modality during human-computer interactions, where input from the eyes are in the background of a user’s attention. It is motivated by a different reason. The previous works considered gaze as a potential replacement to manual input in the pointing task. Instead, the following works investigate methods where the user’s manual capabilities remain the main modality, while eye movements are used to improve the manual task.

Initial work was conducted in 1990, where Starker and Bolt proposed their Gaze-responsive Self-disclosing Display [SB90]. A system is presented that operates “interpretively”, i.e. by aggregating fixations of the user to determine an interest level on the displayed objects. Objects of higher interest are shown in more detail, giving the user an impression of subtle gaze highlightings.

### MAGIC pointing

The first work that argued for implicit use of gaze is Zhai et al.’s seminal paper in 1999, where they proposed the Manual And Gaze Input Cascaded (MAGIC) pointing approach [ZMI99]. This approach also investigated gaze and manual input combination as in prior work, but with the main focus on retaining manual input. They argue that *“it is unnatural to overload a perceptual channel such as vision with a motor control task”*. The pointing task should remain a manual task to the user, and the role of gaze is rather in the background of the user’s attention. The MAGIC technique is an instance of this principle. Users perform the majority of the pointing task with manual control, but large parts of the pointing are eliminated by warping the cursor to the user’s gaze area (Figure 2.1). In a pilot study, the technique indicated reduced effort as compared to manual pointing, while having greater accuracy than gaze-only pointing. This showed that the inaccuracy issue found by Ware and Mikaelian can be approached by hybrid gaze+manual interaction techniques, while preserving the speed advantages of gaze as indicated in many other works.

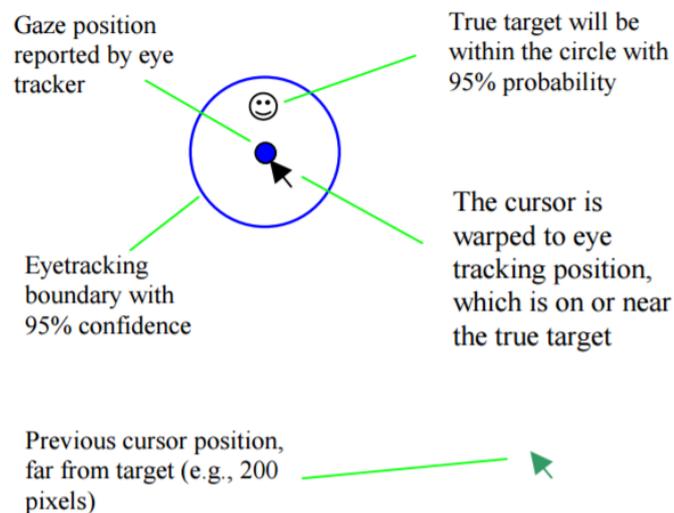


Figure 2.1: MAGIC pointing: the cursor warps to the gaze area (blue circle), followed by fine manual positioning. Image from Zhai et al. [ZMI99].

Note that the idea of using a modality in the background of a user’s attention has been generalised by Vertegaal in his work on Attentive User Interfaces (AUI) [V<sup>+</sup>03]. As devices will ‘bombard’ users with requests for attention, it is necessary that user interfaces better adapt to the limited user attention. The user’s eyes are an ideal indicator of attention to help in this matter. Zhai’s MAGIC work is taken as the prime example, as the user’s visual attention is combined with manual actions to improve the interaction with the UI.

Over the years, The MAGIC pointing technique has been quite influential in the HCI

research community, as researchers extended and studied the principle idea in many more computing contexts. Drewes and Schmidt extended MAGIC pointing to a touch-sensitive mouse in their MAGIC touch project [DS09]. Their work was motivated by a recurring issue of MAGIC: when should the cursor move to the user’s gaze position? Most of the prior work warped the cursor when users moved the mouse, thus requiring small manual effort for each movement. With MAGIC touch, the cursor is warped when the user touches the left mouse button, eliminating the need to initially move the mouse. They compared this technique to mouse input, and found that performance was similar between the techniques, but perceived as faster and more convenient.

Fares et al. explored further variations of the MAGIC technique with a mouse. Instead of warping the mouse cursor, MAGIC-SENSE [FDK12] changes the cursor sensitivity depending on the user’s gaze position. If the cursor is far away from the user’s gaze, it moves rapidly. If the cursor is close to the user’s gaze, sensitivity is reduced for precise pointing. A pilot study showed low error rates and task completion times similar to the mouse. In a follow up paper, they conducted a user study comparing the default MAGIC technique to a mouse, as the original paper used a different pointing device [FFK13]. They also provided a slight design improvement by animating the cursor movement to the gaze to better provide visual feedback. The results of a standard pointing user study showed that MAGIC outperformed the mouse by 8%, and that the amount of hand movement is reduced by half, confirming the initial results of the original paper.

For a further discussion of MAGIC to concepts explored in this thesis, we refer to section

### Gaze-added UI

A notable work that aligns with the direction of implicit gaze interaction is Salvucci and Anderson gaze-added interfaces in 2000 [SA00]. The idea is to keep the conventional manual input as it is, but the user can issue complementary gaze functionality. This gives the benefit of giving users “*more flexibility in choosing when and how to employ gaze input*”. The idea of gaze-added UIs was simple; a conventional WIMP operating system was given that users normally control via the usual mouse and keyboard inputs. However, one button on the keyboard (Control key) was saved for gaze interactions. With this button, users could initiate typical gaze + button interactions as e.g. investigated in Jacob [Jac90] or Bolt’s [Bol81] interfaces. Users point with their gaze at a target, push the button to select it, or hold the button to perform drag & drop actions.

### Gaze as Contextual Information

Researchers have investigated implicit gaze with manual input beyond the MAGIC pointing concept. For example, the LookPoint system uses eye gaze to switch input devices between multiple screens [DHVE06]. The system redirects the input of mouse and keyboard, with regards to which screen the user is visually attending. This provided practical benefits, e.g. the user does not need to reconnect input devices from one to another computer. A study showed that this method is faster than traditional methods (mouse, keys, multiple keyboards), and is preferred by users.

The Rake Cursor interaction technique allows users to facilitate multiple cursors, and gaze selects the desired cursor [BO09]. Multiple cursors in one UI can reduce pointing

time [KI08], but need a method to select a cursor. With the Rake Cursor, the user selects the desired cursor by simply looking at it, and then controls it by standard mouse control. Their experiment showed that it is faster than default mouse or MAGIC pointing.

As an approach for leveraging implicit gaze input, Santella et al. investigated image cropping with eye gaze [SAD<sup>+</sup>06], as an instance of Vertegaal’s implicit AUI [V<sup>+</sup>03]. Their system uses fixation data to identify important image regions, and then crops the images accordingly. A study showed that users prefer this technique over uncropped images, or automatic cropping.

During the course of this thesis, two research papers also involved the use of gaze in consideration with direct and indirect manual input. Voelker et al. investigated input redirection on interactive workspaces, systems that provide users a combined horizontal and vertical touchscreen [VMSB15]. Direct touch input is active when users look at the horizontal surface, enabling default input on the reachable screen. But users can also issue indirect touch input from the same horizontal surfaces, when their gaze lies on the vertical screen. In their study, they compared this method to using direct-touch only, and found that the gaze approach leads to higher performance.

Serim and Jacucci use eye-tracking to support varying degrees of visual feedback during manual input [SJ16]. The system distinguishes two cases: input with visual guidance (user looks where they touch), or without visual guidance (user looks elsewhere). In the latter case, the system provides feedback of the non-visual manual input, to the area the user is looking at. Our direct and indirect input exploration goes in a similar direction, but applied to a different usage context. Rather than redirecting the visual feedback to the user, our focus is on explicit interaction with either direct or indirect input depending on the user’s gaze area.

## 2.2 Direct and Indirect Interaction

The majority of a user’s human-computer interactions are manual, as the hands provide a high fidelity for interaction with computers. Researchers have investigated a diverse set of interaction techniques involving one and two hands of the user, that in their entirety go beyond the coverage of this thesis. We therefore focus on two categories of manual interaction that can characterise many of a user’s manual computer inputs; direct and indirect input. We derive the definition of these input forms based on prior work of Foley et al. [FWC84], Beaudouin-Lafon [BL00], and Hinckley and Wigdor [HW12]. Here the input denotes the actual physically sensed motion that is sensed through an input modality, and the output is the display or the object of interest that the user wants to manipulate.

- **Direct input:** when input and output are spatially equal. Examples include multi-touch input or a stylus.
- **Indirect input:** when input and output are spatially separate. Examples include a computer mouse or a laptop touchpad.

### 2.2.1 Direct Input

Direct input can be described as user input to a computer, where the user directly manipulates the physical position on the computer interface. The most common example is the touchscreen, that senses a user's touch and also displays the digital content on the same screen. The first technical prototypes and research papers of touch interactions date back to the 1960s [Joh65, Hla69], and over the years researchers refined the technology (e.g., [LBS85, SPS92, Wel93, Rek02, Han05]), leading to the current state of the art where multi-touch displays became standard for mobile phones, tablets, and large information displays.

Early explorations of multi-touch possibilities, i.e. the way to interact directly with a screen by tap, pinch, and other gestures, began with Krueger's Videoplace system [KGH85]. The popular pinch-to-zoom gesture has been showcased in Krueger's demonstrations, where a figure's shape, displayed on the screen, was directly manipulated through a thumb and index finger gesture. Among others, a notable work is also Wellner's Digital Desk in 1993 [Wel93]. He explored various new concepts using multi-touch and a digital pen, including paper-like virtual document annotation, drawing/design work, and complementary stylus usage for digital inking operations.

Sears et al. summarised developments of touchscreens in the early 90s [SPS92]. They state one of the main advantages of the technology is the directness of input. Unlike indirect input devices, such as a mouse, users simply point at the desired object, making the input fast, easy to learn, and flexible; all without the necessity of additional input devices. From a system's perspective, the directness makes a hover state obsolete (in contrast to a mouse that needs hover to show visual feedback of the cursor [BHR85]).

With the multi-touch hardware increasingly maturing around the millennium, researchers started exploring direct multi-touch interactions more broadly. The SmartSkin prototype allowed Rekimoto to pioneer various direct touch interaction techniques [Rek02]. For instance, single-touch dragging and pinch-to-zoom for map navigation, or whole-palm gestures for a menu calls. Researchers extensively explored further gestural interaction techniques, e.g. multi-finger and multi-user gestures [WB03], touch on vertical displays [ABCD15], or bimanual touch [BWB06, WPHB16]. Indeed, the expressiveness of touch affords many different techniques and task scenarios.

Considering the current status-quo touch devices, i.e. phones and tablets, a interaction paradigm emerged that is based on single- and two-touch gestures. Basically, single-touch for selection and dragging of objects, and two-touch for continuous pinch-to-zoom actions. The paradigm covers a large range of actions that are offered to the user on contemporary mobile devices. Hence many of our example interactions are based on single touch and two-touch pinch-to-zoom operations.

To enhance interaction while retaining this touch paradigm, researchers have aimed for implicit usage of touch inputs as well [HB16]. One of the approaches is to avoid accidental touches by the palm, that can be detected from the size of the touch area, and hence ignored by the system [SXM<sup>+</sup>14]. Touch detection around the device can provide information of how users are holding a mobile device [YHB<sup>+</sup>15], and with it one can adapt the user interface controls better to the vicinity of the user's hands [WB09, GWP12]. We investigate gaze and touch interaction on a mobile device in chapter 6.

### 2.2.2 Indirect Input

Indirect computer input was considered earlier than direct input, with the mouse introduced by Engelbart in 1963 as a prime example of an established indirect input device [EEB67]. Most indirect input devices are designed for desktop computers, e.g. mouse, keyboard, or a game controller/joystick. As desktop computers have vertical displays, direct-touch input might be too difficult to use over time, but an indirect input device away from the display and closer to the user is comfortable to use. One of the main advantages is that users only need to employ small finger/hand movements to be able to interact with the whole user interface.

Another established input device that in its nature is close to indirect touch input is a touchpad [MO97, ML04], that nowadays comes as a standard component on laptop computers. The touchpad can be regarded as a physically-separate space where users perform touch based tap and drag gestures to control the cursor on the display. This provides the user mouse-like interactions without the necessity to attach an additional device.

Indirect input however is not necessarily tied to a device that is physically separate from the output display. Touchscreens for example, that usually employ direct touch input, can also integrate indirect input. The UI software can simply redirect touch inputs to a target that is offset from the physical location at which the input is sensed [HBPB12], resulting in *indirect touch* input.

Early work explored indirect touch in order to improve the low accuracy of touchscreens. For example, Potter et al. argue that while touchscreens evoke high user satisfaction, they are also prone to high error rates [PWS88]. To improve this, they suggested offset cursor techniques, where users control a cursor with touch. At touch down, the cursor appears that is offset from the finger position, but moves with it to precisely select a target. Evaluation of the technique showed that the technique led indeed to fewer errors than default touch. Further studies by Sears et al. confirm these results [SS91].

With innovations in multi touch sensing hardware around the millenium such as the Portfolio Wall, DiamondTouch [DL01], or SmartSkin [Rek02], it became clearer that direct touch will become the standard input paradigm on interactive surfaces. Thus part of indirect touch research explored how indirect touch can be used complementary to direct touch inputs.

In 2003, Albinsson and Zhai investigated additional interface elements around a direct touch position to improve touchscreen precision [AZ03]. For instance, the Cross-keys technique shows virtual arrow keys around the touch position, that allow post-hoc refinements of the cursor position. The Precision-Handle technique provides a lever that directs to the touch position, that users can indirectly control through touch dragging to manipulate the virtual touch position. Their experiments showed that both techniques are a viable addition to complement existing direct touch interactions.

Benko et al. explored the use of a second finger to enable more precise selection on multi-touch screens in 2006 [BWB06]. The idea was that WIMP interfaces often have small targets such as sliders, that are difficult to select with touch. A second finger can therefore trigger precision methods. These methods include the Dual Finger Midpoint technique, where a mouse cursor appears between two fingers, or Dual Finger X-Menu

where the second touch point calls a menu for variable control-display gains. In a user study, the techniques outperformed a default offset cursor technique in terms of error and user preference.

The offset cursor approach has also been extended to mobile phones with the Shift technique by Vogel and Baudisch in 2007 [VB07]. Shift extends the offset-cursor by also offsetting the UI area under the finger, allowing to interact with the finger-occluded area. A user study confirmed that this approach reduces selection errors, and that it is faster than Potter et al's [PWS88] offset cursor. The authors emphasise that the technique can be used in complement to direct touch: Shift only activates when there are small targets under the user's touch position that would be difficult to reach.

Indirect touch has also been considered to address direct touch's limitation of interaction over distance. With direct touch, users can only interact as far they can reach, albeit touchscreens can extend to much larger surfaces for tabletop or whiteboard like interactions [MIEL99, WJF<sup>+</sup>09, WB03]. A few works have suggested the use of mid-air gestural pointing to select remote targets, that however necessitates additional tracking systems over the touchscreen [BBGV11, PMI05]. Software based indirect touch approaches can provide the same benefit with a simpler system.

For instance, The Vacuum is a technique to reach remote areas on a large display, proposed by Bezerianos and Balakrishnan in 2005 [BB05]. It is a UI widget that spans an arc area from it across the surface. All (remote) objects within this area are mapped to proxy objects that reside close to the widget. These proxies allow indirect control of the original objects. A user study showed that the vacuum overall performed similar to existing techniques.

Forlines and colleagues introduced HybridPointing in 2006, a technique that enables both direct and remote input with with a stylus [FVB06]. The pen is normally used for absolute (direct) input, with a trailing widget following the pen's position. If the user taps on this widget, the system switches to relative (indirect) cursor control using the pen. This is realised by using dragging to move the cursor, and a tap to click. The evaluation of the technique showed that HybridPointing can be beneficial over cursor-only input in close distances, and over absolute dragging for long distances.

The ARC-Pad as introduced by McCallum and Irani in 2009 is an approach to switch between absolute and relative input modes using gestural moding [MI09]. A tap will move the cursor instantly to a position on the screen, while dragging moves the cursor relatively. As this disables the tap for clicking, a dedicated button is used to click, something that is however rare in contemporary touchscreen devices. Nonetheless, a study showed that users were faster with ARC-Pad than a normal touchpad, indicating potential benefits when using more sophisticated indirect touch interaction techniques.

To shed light into differences of direct and indirect input, researchers have compared both techniques in various constellations. Forlines et al. compared direct-touch to a mouse on tabletop surfaces [FWSB07], finding that users benefit from direct-touch for bimanual tasks considering the capability of multi-point input, but mouse input is more appropriate for single point tasks. Kin et al. also provide an empirical evaluation of direct-touch, bimanual, and multifinger vs. mouse input on a multi-touch workstation [KAD09]. The results show that use of direct touch can lead to a 83% reduction of selection time, and multi-finger use can further increase user performance. Schmidt et al. compared direct

vs. indirect touch on a tabletop display [SBG09]. The indirect condition uses a table as input surface to a separate vertical display. The evaluation showed that direct is faster as the necessity to ‘hover’ over the indirect input surface causes fatigue and decreased performance. They suggest designing indirect techniques that allow users to rest their arms on the surface to decrease physical efforts.

## 2.3 Summary

In this chapter we have reviewed work on eye gaze and direct manipulation. In prior eye gaze work, we classified interactions into the groups of implicit and explicit use of eye gaze. The distinction between these groups is not exclusive, but rather can be considered continuous between implicit and explicit. Some techniques such as dwell-time are more explicit, as one is required to stare at a position for a fixed time [Jac90] — but it can also be implicit when the time is used to approximate a user’s interest on an object [TJ00]. Other techniques such as MAGIC are considered more implicit, as the eyes only play a small role, and the hands the major role, to move the cursor on the screen — yet at any time users can decide to explicitly move their gaze to trigger MAGIC as well [ZMI99].

Considering manual manipulation techniques issued through the hands, we reviewed work from the perspective of direct and indirect inputs. Aware of the trade-off between these two modes, researchers have proposed various interaction techniques in either category. The design and implementation of direct input is relatively clear in its interaction technique with a set of default gestures, and its 1:1 mapping between input and output space. Indirect input is unclear, as there is no clear standard how the input maps to the output. Existing input devices such as the touchpad and mouse provide a default interaction technique, however we have also identified a diverse set of indirect (touch) input mappings in the literature that needs to be considered when designing user interfaces at the intersection of eye gaze and direct-indirect manual interaction techniques.

# 3

## Gaze-Touch: Indirect Input by Combining Gaze with Multi-touch On the Same Surface

This chapter explores the utility of gaze input to enable indirect interactions with a direct modality, touch. In contrast to prior work [SD12b, TAB<sup>+</sup>13], this work firstly explores gaze and touch interaction on the same display, allowing us to consider a new, comparative perspective on direct *vs.* indirect (gaze enabled) touch.

*Gaze-Touch* is an interaction technique that integrates the two modalities with a specific division of labour: *gaze selects, touch manipulates*. This emphasises the extensive possibilities of touch gestures that we explore in combination with gaze pointing. In particular, this changes the research focus from considering gaze as a pointer, and touch as a confirmatory role (i.e. Stellmach et al.’s *gaze suggests, touch confirms* [SD12b]), to a perspective where gaze merely indicates targets, and touch provides its whole range of expressive power from multi-touch gestural manipulation.

The particular input steps are best explained in comparison to direct-touch input, which normally involves: (i) moving the hand to the target, (ii) touching down on the target to select it, and (iii) direct manipulation with the fingers on the surface. Gaze-Touch, in contrast, is based on (i) looking at the target, (ii) touching down anywhere on the surface to select it, and (iii) manipulation with the fingers on the surface (but displaced from the target).

The indirection of direct touch input to the gaze target means that it spatially separates the hand from the target. The potential utility of this separation can be considered from two viewpoints:

- More expressive input from the same touch position (Fig. 3.1): finger touches in the same position can resolve to selection of any point on the surface. Without moving their hands out of position, users can reach and select any position on the surface, and rapidly switch context using their gaze.
- More expressive input to the same target (Fig. 3.2): the same target can be manipulated from different positions on the surface. Users can move their hands off an object but continue to manipulate it with their hands “out of the way”. This can

### 3. GAZE-TOUCH: INDIRECT INPUT BY COMBINING GAZE WITH MULTI-TOUCH ON THE SAME SURFACE

---

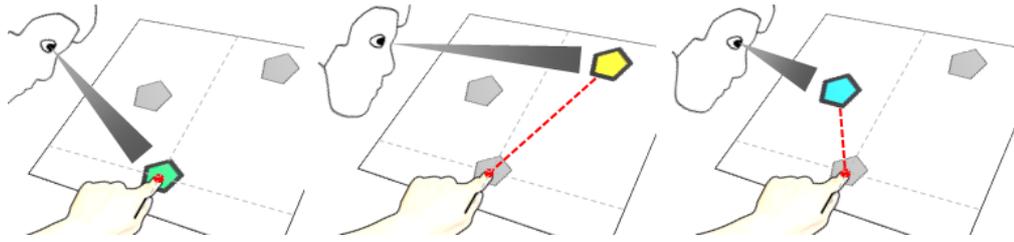


Figure 3.1: More expressive input from the same touch position: three examples of users touch on the same touch position, but each time manipulate a different target.

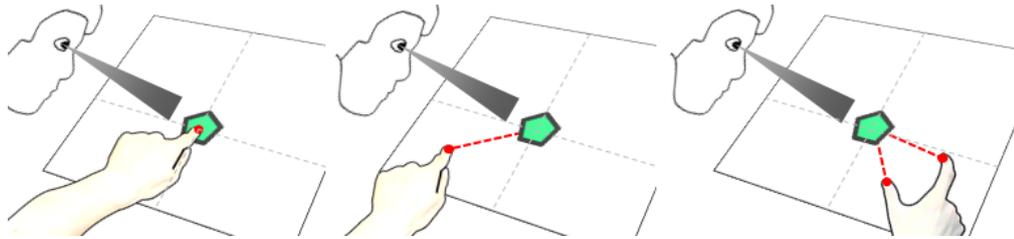


Figure 3.2: More expressive input to the same target: three examples of manipulating the same target that the user sees, but each time with different touches on the surface.

help address occlusion, and also enable novel indirect manipulation techniques, for instance with variable control-display gains to adjust precision of input.

The idea of Gaze-Touch is to complement direct-touch. Our focus in this section is therefore to understand how these two modes of interaction compare, and how Gaze-Touch can be employed alongside direct-touch. For this reason, we first characterize Gaze-Touch in comparison to direct-touch input through an analytical discussion of their interaction properties. The second part presents application prototypes that explore how Gaze-Touch can be used in relation to direct-touch:

1. Gaze-Touch *or* direct-touch. The Image Gallery application allows users to manipulate the same image indirectly by Gaze-Touch (gaze and touch are separate (Figure 3.3a)), or directly with direct-touch (users look and touch at the same target (b)).
2. Gaze-Touch *and* direct-touch. The Paint application allows users to draw and manipulate primitive shapes with direct-touch on the main canvas, and switch e.g. colour mode on the menu through Gaze-Touch (Figure 3.3c).
3. Gaze-Touch *instead of* direct-touch. Two applications demonstrate this class of interactions. The Map Navigation application allows users to zoom into their gaze location instead of where they touch (Figure 3.3d). The Multiple Objects application allows users to quickly select and drag multiple targets anywhere on the surface (Figure 3.3e).

This chapter makes four contributions. First, we introduce Gaze-Touch as a novel mode of interaction to complement direct interaction on the same interactive surface. Second, we analyse the design space of Gaze-Touch in comparison to default direct-touch interaction. Third, we demonstrate how Gaze-Touch complements direct-touch in four application examples. Fourth, we present nine interaction techniques that are based on Gaze-Touch and introduced with the applications.

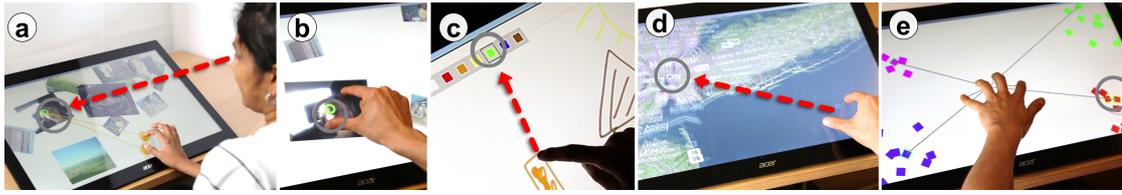


Figure 3.3: Users select by gaze, and manipulate with multi-touch from anywhere (a). This can enable seamless switching between indirect (a) and direct manipulation (b), implicit mode switching during direct-touch tasks (c), zooming into map locations the user looks at (d), and dragging multiple targets that are out of the hand’s reach (e). The gray cursor indicates the user’s gaze.

### 3.1 Design Space

To gain a deeper understanding of the conceptual differences between direct-touch and Gaze-Touch, we analyse the two techniques. We provide a design space analysis under the following headings, without claiming completeness: *similarities*, *occlusion*, *precision of selection*, *precision of manipulation*, *physical aspects*, *multiple object selection*, and *multi-touch to one point*. Table 3.1 provides a summary of the comparison and Figure 3.4 illustrates the conceptual differences.

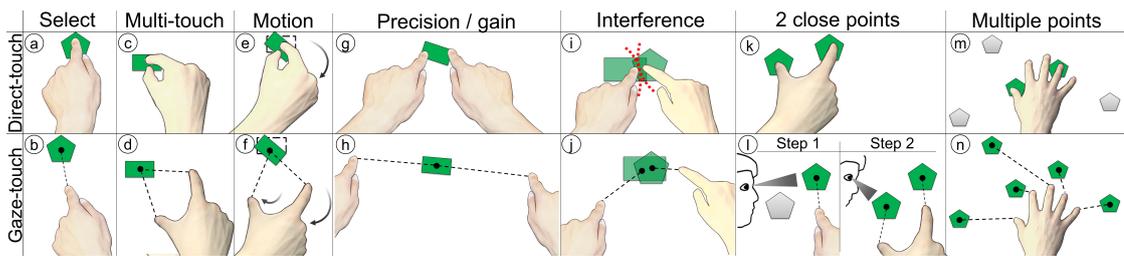


Figure 3.4: Illustrated differences between Gaze-Touch and direct-touch.

Property	Direct-touch	Gaze-Touch
Manipulation start time	<b>Direct</b> (manipulate the moment of touch down)	
Manipulation location	<b>Direct</b> (touch point is point of manipulation)	<b>Indirect</b> (point of manipulation is remote from touch)
Manipulation motion	<b>Similar</b> (manipulate with similar hand motion)	
Remote targets	<b>Low</b> (only targets in physical reach)	<b>High</b> (reach any target by look)
Occlusion	<b>Moderate</b> ('fat-finger') to <b>Large</b> (palm, pinch, hand)	<b>Low</b> (object separate from touch)
Precision of selection	<b>Moderate</b> (precise, but 'fat-finger')	<b>Moderate</b> (no 'fat-finger', but gaze imprecision)
Precision of manipulation	<b>Moderate</b> (usually control-display ratio of 1)	<b>High</b> (control-display ratio through finger distance, that user can adjust)
Physical feedback	<b>High</b> (finger/hand indicate current manipulation)	<b>Low</b> (finger/hand separate from manipulation point)
Physical fatigue	<b>Moderate</b> (move hand / arm)	<b>Low</b> (look, and little hand / arm movement)
Physical interference	<b>High</b> (multiple fingers/hands in same location)	<b>Low</b> (fingers/users can be remote)
Acquisition time	<b>Moderate</b> (move finger to position then touch down)	<b>Low</b> (look and touch down anywhere)
Speed of selection of multiple objects within hand's reach	<b>High</b> (select multiple objects at once)	<b>Low</b> (Must sequentially select each object by gaze & touch)
Selection of multiple objects out of hand's reach	— (impractical, needs two hands or other indirect method)	<b>High</b> (multiple remote targets can be selected by gaze & touch to one hand)
Degrees of freedom per point	<b>Low</b> (1 touch per point)	<b>High</b> (multiple touches map to one gaze point)

Table 3.1: Summary of the differences of direct-touch to Gaze-Touch.

**Similarities.** Both Gaze-Touch and direct-touch are temporally direct, as manipulation of an object starts as users touch the surface. Both techniques accept a single touch point for ‘clicking’ an object (see Figure 3.4a & 3.4b), and two touch points for

manipulating an object (Figure 3.4c & 3.4d). Gaze-Touch uses the same multi-touch gestures for object manipulation; e.g. rotating two touch points to rotate a selected object (see Figure 3.4e & 3.4f), and pinch gestures to scale (Figure 3.4g & 3.4h). These similarities enable ease of learning and preserve consistency, as users can transfer their knowledge of direct-touch for operation of Gaze-Touch.

**Occlusion.** A direct-touch gesture causes occlusion, because users place their hands on top of an object for selection. As users place more fingers on an object, the area of occlusion increases (see Figure 3.4c). Researchers have suggested techniques that avoid occlusion, like creating proxies to the objects [WBP<sup>+</sup>11]; however, it requires additional effort from users, and adds delay to the manipulation task. Gaze-Touch prevents occlusions by enabling spatially-indirect manipulation (Figure 3.4d). Since touch actions are disjoint from the gaze-selected object, users can touch down on any surface location while looking directly at the object.

**Precision of selection.** Using direct-touch for target selection can be problematic when the target’s size is smaller than the user’s finger [HB10]; this is the well-known ‘fat-finger’ problem. Although researchers suggested techniques to alleviate this problem by using multiple touch points (e.g. [AZ03, BWB06]), the use of multiple fingers or hands hinders the selection process. Using gaze for selection in principle can overcome this issue. However, our eyes naturally jitter, and inaccuracy of eye trackers can cause imprecision [ZRZ08]. Touch is still more precise for single-finger taps on large objects, but Gaze-Touch is potentially more suitable when the interaction requires placement of multiple fingers on an object (see Figure 3.4c & 3.4d).

**Precision of manipulation.** The precision of manipulation differs between Gaze-Touch and direct-touch. The standard direct-touch model is based on a 1:1 control-display ratio, so fine-grained manipulations can become difficult as they require tiny and precise movements. In practice, the size of objects has a limit; an object becomes difficult to manipulate if its size is too small to be selected or manipulated with fingers (Figure 3.4g). The standard touch technique could be improved by having users first select a target and then put their fingers elsewhere to manipulate (like the Rock&Rails technique [WBP<sup>+</sup>11]). The necessity to select and deselect the object complicates the interaction and delays the manipulation. In contrast, Gaze-Touch allows users to draw their fingers as far apart as the screen allows, and to immediately start manipulation at the moment of touch down (see Figure 3.4h).

**Physical aspects.** In Gaze-Touch, the finger touch positions are detached from the gaze position. Users only see *feedback* in their focus area, i.e. on the selected object. However, feedback on their touches is out of the users’ focus area. This is in contrast to direct-touch, where users can see feedback for both their touches and the selected objects, because they always occur on the same location. By detaching the touch and gaze, it reduces muscle *fatigue*. Users’ can keep their hands within their comfortable regions and still able to manipulate gaze-selected objects. On the other hand, the active use of gaze to select targets could lead to eye fatigue, as the eyes, a channel to perceive visual content, should not be overloaded with motor tasks [ZMI99]. Another benefit of detaching gaze and touch is that it avoids finger interference. Interference can occur when multiple fingers or hands collide within the same location, which interrupts the task (Figure 3.4i). With Gaze-Touch, the objects can be separate from the finger’s position, so physical collision is prevented (Figure 3.4j).

**Multiple object selection.** Gaze is a single-point input, while multi-touch supports simultaneous input from multiple points (Figure 3.4k). With gaze, users must select multiple targets by looking at each object and placing a touch down (Figure 3.4l). Although conceptually gaze selection of multiple targets is slower than direct-touch, Gaze-Touch yields a benefit that users can select scattered objects on a surface. Selection of multiple objects with direct-touch is limited by the distance that a hand can reach and users can only select multiple objects that are near by each other (Figure 3.4m). Gaze-Touch in contrast eliminates this restriction (Figure 3.4n).

**Multi-touch to one point.** Gaze-Touch can map multiple touch points to a single gaze point (Figure 3.4d). This contrasts with direct-touch where one finger can be physically mapped to one point on the screen (Figure 3.4a & 3.4k). Furthermore, Gaze-Touch is invariant of the hand’s posture. In a rotation gesture with direct-touch, a user fits their hand to the object’s shape to then perform the rotation from this hand posture (Figure 3.4c & 3.4e). Prior work has shown that there are several occasions where rotation or scaling postures and motions can be difficult [HWO<sup>+</sup>13, HNK<sup>+</sup>13]. Using Gaze-Touch, target acquisition is more comfortable as users only look at the object and touch down remotely with any hand posture (Figure 3.4d & 3.4f).

## 3.2 Applications

In the following we describe four applications that each demonstrate a specific use of Gaze-Touch. Each application is described in its own section. Within each application, we describe concept, interaction techniques, and, if relevant, implementational details. The first three applications were also part of a preliminary user study which design and setup are described once, and which task and results are described within each application section. Notably, in all figures the gray circle indicates the user’s current gaze point.

### 3.3 Application: Image Gallery

This application demonstrates that Gaze-Touch *or* direct-touch can be used for the same task. Users can browse through their set of images. They can scale them up for a detailed view, rotate the images to correct the aspect ratio, and drag images across the surface for sorting, grouping, or other object manipulation tasks. In essence, users can perform two types of touch gestures: single-touch dragging, and multi-touch rotate, scale, and translate (RST). Multiples of these gestures can be performed at the same time, when using multi-finger and bimanual input.

#### 3.3.1 Switching between Gaze-Touch or Direct-Touch

The switching between direct-touch and Gaze-Touch is accomplished through using the user’s coordination between gaze and touch position. When a user looks at an image and at the same time touches on it, direct-touch is enabled. This means the touch point is used as input, and not the gaze point (Figure 3.5a). However, when the user looks at a target but touches down somewhere else, Gaze-Touch is enabled (b, c).

### 3.3.2 Interaction Techniques

In addition to standard direct-touch translate, rotate, and scale gestures, the user can perform the following Gaze-Touch techniques:

#### Accelerated-Object-Dragging

When users look at an image and touch down once remotely, they can drag the image with their finger. While the selection is similar to previous techniques for interaction over distance [SD13, TAB<sup>+</sup>13, TBAG14], this technique only uses touch dragging for positioning. This dragging of images uses a dynamic control-display gain. We implemented a dragging acceleration similar to windows XP mouse acceleration, which amplifies the speed of the dragging finger. This enables to overcome larger distances with shorter movement, and be more precise when moving the finger slowly.

#### Indirect-Rotate-Scale-Translate (RST)

This technique is the Gaze-Touch counterpart for the RST gesture. Users touch down two fingers while looking at the same image (similar to [SD13], however without mode-switching). It has some characteristics that are distinct to direct-touch. Users only need the gaze point to be on the image, enabling manipulation of images that are too small to directly lay multiple fingers on it (Figure 3.5b), and when high precision is required (c). The further the user draws apart their fingers at touch down, the more precise is the manipulation. This provides the user with a choice of how precise they want to manipulate the image: users can place their fingers very close for fast manipulation (b), or very far apart for high precision (c).

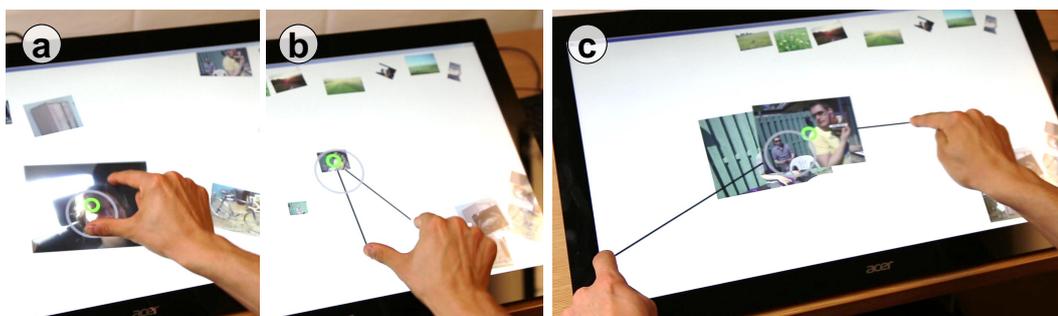


Figure 3.5: Indirect-RST: in addition to direct image manipulation (a), users can indirectly manipulate images for easy acquisition of small targets (b), or more precision (c).

#### Multi-Image-Dragging

While users can sequentially drag multiple images with the *Accelerated-Object-Dragging* technique, they can also drag multiple objects at once (Figure 3.6). The user first selects each image by looking at each image and each time touching down, to then perform one drag gesture. This is particularly interesting as, in contrast to direct-touch, users can simultaneously drag objects that would be out of the hand's reach.

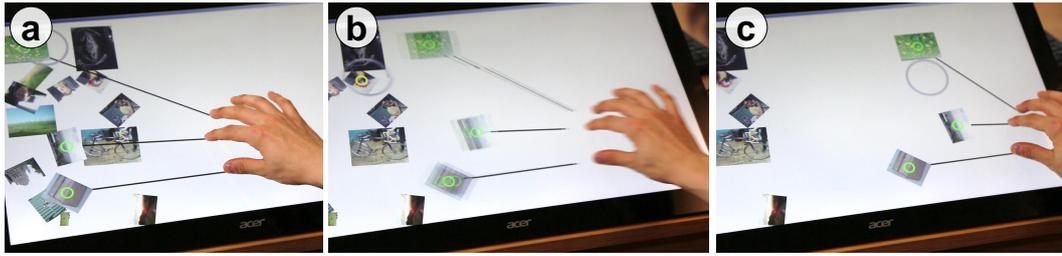


Figure 3.6: Multi-Image-Dragging: after multiple Gaze-Touch selections, users can drag them out of the pile using a single dragging gesture. Through a dynamic control-display gain, small movements can overcome large distances.

### 3.3.3 Implementational Details

The moment the user touched down, the system decides if it is a Gaze-Touch or a direct-touch. If the user touches on an image and does not look at another image, direct-touch is triggered. Else, Gaze-Touch is active. The gaze point is set as the target of manipulation of a touch input session until the user lifts their finger. Intermediately received touch events of this session (`touch_update`) are executed on the point of gaze that was received at the moment of `touch_down` (for Gaze-Touch, respectively). To counter inaccurate gaze data, we used target-assistance. The image is highlighted as ‘looked’, when the system’s gaze estimate is close to the image.

An interesting case is the control-display gain for multi-touch gestures, such as two-finger scaling. In direct-touch, this case is clear as the distance between the two fingers can be mapped to the same distance on the screen, thus an absolute 1:1 control-display gain. RST with Gaze-Touch relates two-touch input to one gaze point, and therefore it is unclear to what display-distance it should be mapped to. In our application instance, the distance between the fingers of a two-touch gesture is mapped to the radius of the target’s size.

### 3.3.4 Study Design

We conducted a preliminary user study to demonstrate the feasibility of and to gather user opinions about the applications. 13 volunteers between 22 and 33 years took part in the study ( $M=27.9$ ,  $SD=3.73$ , 4 female). On a scale between 1 (no experience) to 5 (very experienced), users perceived themselves as well experienced with multi-touch ( $M=4.3$ ,  $SD=0.9$ ), and as less experienced with eye based interaction ( $M=2.5$ ,  $SD=1.4$ ). After a brief introduction, users were once calibrated to the eye tracking system. Users then interacted with the applications (counterbalanced). Each application began with a short training session where the experimenter explained the interaction techniques, and ended with an interview session. Each application test and interview lasted approximately 5-10 minutes. Users were not bound to a specific performance goal of the tasks to keep it to natural usage of the interactions.

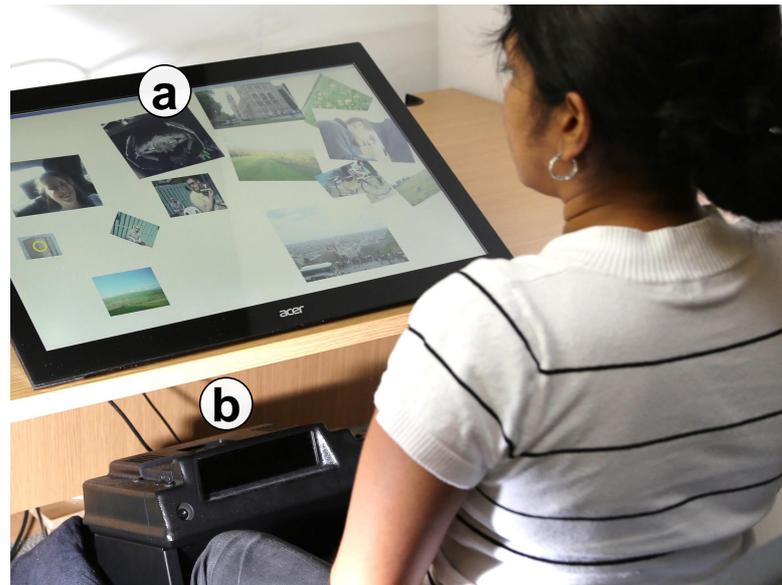


Figure 3.7: Setup for the user study, consisting of a 27" 1080p multi-touch sensitive surface (a), and the 120hz Eye Follower eye tracking device (b).

### 3.3.5 Apparatus

We use an LC Technology Eye Follower with a touchscreen that is tilted  $30^\circ$  toward the user to enable convenient touch reaching (Figure 3.7). The user's eyes were approximately 50cm in front of the screen's center. Occlusion of the eye tracking camera could occur during the use. In practice, however, mostly users bend their arms around the tracking camera's view because of the close proximity of the touchscreen. As touchscreen we used an Acer t272 27" 1080p display that allows up to 10-finger multi-touch input. The system is running at a frame rate of 60hz, on a quadcore i7 2.3GHz CPU computer. The applications are written in Java using the Multitouch For Java library<sup>1</sup>).

### 3.3.6 User Feedback

Users were provided with ten images and were trained using both direct-touch and Gaze-Touch techniques. They performed two tasks of sorting images into groups (e.g. indoor/outdoor), and two tasks of searching for an image with a specific element in it (e.g. a bus). Before each task, the images were randomly placed, rotated, and sized. Users could scale the images between 50 and 750px.

All users got quickly used to the techniques in this applications. Users did not have difficulties to switch between the direct and indirect counterpart. The study showed that most users stick to one technique for each particular task:

**Single-Touch Dragging.** Twelve users kept on using Gaze-Touch after the training. Interviews revealed that their reasons were speed, ease, and less physical effort. This was considered important with multiple images, where moving back and forth for each image is avoided, as one user stated: *"you do not always have to go back with your hand, but [you] keep it [the hand] stationary while your gaze goes back to the imagepool"*. Users

<sup>1</sup>Used library available at <http://www.mt4j.org> (16/04/2014)

emphasized that Gaze-Touch has less physical fatigue (*“You just move your arms, not your whole body”*). Users also liked the speed of dragging (*“It is effortless to move, as you can accomplish more with less movement”*). Some users were also positive about less occlusion through their fingers (*“My fingers sometimes obscure the pictures [with direct-touch].”*).

**Two-Touch RST.** Seven users kept on using direct-touch and four users Gaze-Touch. The user who preferred direct-touch found it to be easier and more intuitive (*“It is more intuitive, the movement”*). They also stated prior knowledge of direct-touch (*“I prefer on the picture [...] based on how I use my phone”*). An interesting case occurred when these users wanted to acquire small images with two fingers. They tried to put their fingers directly on it, yet in a failed attempt they put their fingers only close to the image as it was too small. This triggered Gaze-Touch on the very image (users looked and touched close to it) with which users scaled it up, without being aware of a Gaze-Touch.

**Errors.** Three users stated some difficulties with overlapping images. Inaccurate gaze tracking by the hardware we used lead to false positive image selections (*“When pictures overlapped sometimes, it did not jump at the picture that I wanted”*). Another issue occurred when selecting an image to drag. The user already looked away to the dragging destination during touch down, which lead to a wrong selection (*“I already looked at where I wanted to move it before I touched, so it moved something else”*).

**Specific Findings.** Two users stated they used direct in front (user’s comfort zone), and Gaze-Touch in the remaining area. They intuitively use direct-touch in close proximity, however to avoid reaching out, Gaze-Touch became convenient (*“When it is far from me, then I can drag it from distance. If it is close to me, I can use the picture itself”*). One user emphasized an interesting feature of Gaze-Touch: users can manipulate an image, even though touching on another (*“If I look at a picture, I can go anywhere with my fingers. Even if I have my fingers on another picture”*).

### 3.3.7 Summary

Our evaluation showed that having direct and indirect manipulation within the same application is feasible. The majority of users kept using Gaze-Touch for single-touch dragging, and direct-touch for two-touch scaling and rotation. Users acknowledged the speed, reachability, reduced movement and reduced fatigue of Gaze-Touch in comparison to direct-touch. However, many users preferred using direct-touch for RST gestures. They perceived it easier to perform this gesture directly on the image.

## 3.4 Application: Paint

This application demonstrates how Gaze-Touch *and* direct-touch are used. The user is provided with standard tools of a drawing application. With direct-touch, users can draw on the main canvas of the interface. In the menu, users can create three types of primitive shape (rectangle, circle, triangle), that initially have a size of 100x100px. After creation, they can be dragged and scaled using direct-touch input. Thus the user can create figures based on individually drawn lines and these primitive shapes.

The menu is completely Gaze-Touch enabled (but can also be directly touched). The menu provides the functions select colour, create primitive, and copy existing primitive. To trigger a menu mode, users look at a menu icon, and select it by touching down anywhere on the surface. We believe this can have an advantage for drawing tasks, as users do not need to remove hands from their current target. And after a mode is switched on the menu with Gaze-Touch, users do not need to relocate the previous position of the hand to continue the task. Users can keep their hand at the drawing position, and from there perform Gaze-Touches to the remote menu. This concept can be applied to many applications that involve a main interactive area and remotely positioned menus, such as ribbon menus in office tools, tabs in browsing, etc.

### 3.4.1 Interaction Techniques

#### Remote-Colour-Select

Most actions of the user are around the main canvas, where the figure is drawn directly. From here, users can quickly change the colour through Gaze-Touch (Figure 3.8). The user looks up at the colour (a), and touches down at their current position to confirm (b). Once done, the user can continue the drawing task (c). This technique can be easily extended to multiple finger use. Users can touch down many fingers, and each time look at a different colour, to simultaneously draw with several colours. In direct-touch, the user would have to reach out to the canvas or use a second hand to apply different colours to each fingers.



Figure 3.8: Remote-Colour-Select: a user draws the tree stem directly (a). The user then changes to the ‘green’ colour by a look at the corresponding menu icon, and a tap (b). The user directly continues drawing (c). The color change takes place the moment users touch down. If users start drawing immediately, users draw with a new color. If users only tap to change color, no line is drawn.

#### Create-Object

Contrary to mode changes, this technique creates a new element into the canvas. When users perform a Gaze-Touch on a graphical primitive icon of the menu, the primitive is created at the position of the user’s touch. From here, the user can directly drag it to a more precise position, or perform direct RST manipulation. The operation of this technique is similar to drawing (Figure 3.8), but instead of a colour it adds an object.

### Copy-Paste-Object

Graphical primitives are direct-touch enabled in our application, thus users can drag them with single-touch on it. However, a single-touch can also be used for copy-paste of the primitive. The system switches to this special mode when users touch on the object, while they look at the copy-paste icon in the menu (Figure 3.9). This creates a copy directly under the user’s finger, that can then be dragged elsewhere. This technique is distinct as the user is required to coordinate both the touch and gaze point. This requires more mental effort. However, this technique allows the user to perform two different tasks (dragging or copying) with a single-touch on the object, that are distinguished by where the user looks at. The technique also scales to multi-touch. Users can instead touch down two fingers to create two copies simultaneously.

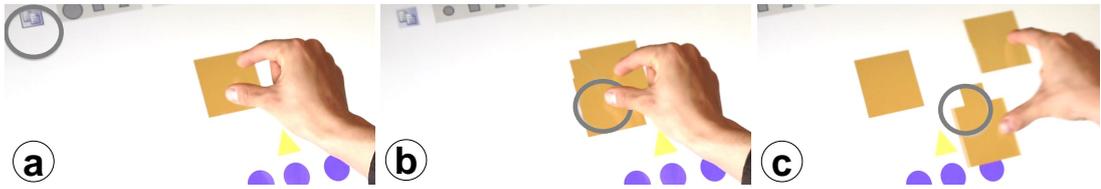


Figure 3.9: Copy-Paste-Object: the user can copy an existing object with a single-touch. Usually, a touch on the object leads to dragging. However, when the user looks at the copy icon in the menu (a), and then touches down on the object, the user obtains a copy of the touched object under her finger (b). Then, the user can directly drag the new copy to a desired position (c).

#### 3.4.2 Implementational Details

The moment the user touched down, the system determines whether the gaze position is on one of the icons of the menu. If true, Gaze-Touch is triggered, otherwise direct-touch is kept. To aid a potential inaccurate gaze position, we used target-assistance for the icons. If the gaze cursor is close to the menu, it attaches to the closest icon. No gaze cursor is shown, but the icons in the menu are highlighted when the user looks at them.

#### 3.4.3 User Feedback

For the purpose of this study that investigates the switching between direct-touch and Gaze-Touch, we limited the interactions to direct-touch drawing on the canvas, and Gaze-Touch selection of colours in the menu. The task of the users was to draw a house, a tree, a car, a sun, and their name with various colours. All users were able to perform the drawing tasks.

The interviews revealed that seven users were positive, three users negative, and the other participants had mixed opinion about the application. Most users commented that the Gaze-Touch menu is easy to use, fast, and convenient (*“It goes quicker to select the colour [...] than by hand”*). Also it was noticed that it helps to focus on the main drawing task (*“It indirectly saves interaction, you can focus on the draw surface”*), and that it reduces mental effort (*“There is less thinking involved”*).

Users also commented on false positive colour selections. This includes two reasons: (1) inaccuracy of eye tracking hardware, and (2) eye-hand coordination of the system.

Often, users looked at a colour, but already moved on before touch down. It occurred that users passed close to other colours when looking back to the canvas, which the target-assistance wrongly interpreted as the colour of choice. This has been reported as ‘Late-Trigger errors’ and can be addressed by delayed selection based on the last detected eye fixation [KKP<sup>+</sup>08].

Two users stated that they disliked the Gaze-Touch menu, because of mental demand (“*I feel like I have to focus*”) and non-normal behaviour (“*Often your eyes move without you knowing [that] they are moving*”).

#### 3.4.4 Summary

The evaluation showed users can use direct-touch in conjunction with Gaze-Touch. Both techniques are used for separate areas on the screen, and therefore give the user a clear separation of input. Users recognized that Gaze-Touch is useful for menus that are often out of reach. They also indicated that it can be easy to use, comfortable, and contribute to better focus on the main drawing task. On the downside, our implementation and some of the design decisions led to false positive colour selections for some users (further discussed by Kumar et al. [KKP<sup>+</sup>08]).

### 3.5 Application: Map Navigation

This application demonstrates where Gaze-Touch can be used *instead of* direct-touch. The application begins with a world map, that the user can then explore with direct single-touch dragging gestures to pan the whole map, and Gaze-Touch based zooming to zoom in locations. To complement previous work that used gaze for interaction on maps [SD12a], we use gaze implicitly as the target of a two-finger zooming gesture.

#### Gaze-Focused-Zooming

To perform zooming, the user looks at the location of interest, and then performs a pinching gesture anywhere on the surface. This triggers zooming into the user’s gaze point. This yields several benefits over the direct counterpart. First, users can keep their hand on the same position for multiple zooms that reduces hand movement, occlusion, and fatigue, as only the user’s gaze is used for target selection (Figure 3.10). Second, the user’s gaze is faster than the hand for the selection of a zooming target. Third, users are able to change the zooming target during the gesture. With direct-touch, the target is fixed to the touch position once touched down. With Gaze-Touch, users can change the position by a glance. This becomes useful for corrective zooming: if a user zoomed into the wrong area, the user can zoom out, look at a different location, and zoom in again; all within a single pinch gesture.

#### 3.5.1 Implementational Details

Within the touch input manager, we changed the zooming target from the touch center position to the gaze position. During zooming gestures, the system receives gaze events



Figure 3.10: Gaze-Focused-Zooming: users can change their zoom-in position during several zooms without changing the pinching position.

on-line to enable dynamic changing of the zooming focus. We also added a gaze cursor for this application. To avoid distracting behavior and gaze jittery, the cursor is set large (width=250px) and we average jittering gaze samples for 150 ms when only short eye movements occur.

### 3.5.2 User Feedback

In this part of the study we let user compare direct-touch against Gaze-Touch zooming. Users performed both conditions (counterbalanced). In each condition, users searched for five capital cities starting from a world view. Users did not have any difficulties finding the cities. Four users stated they had to get used to the gaze-based approach within the first or first two city tasks.

**Preferences.** Nine users favored map navigation with Gaze-Touch, two users thought they were equal, and the remaining two preferred direct-touch zooming. Users preferred Gaze-Touch zooming because of ease, speed, less physical effort, precision, and reachability. Users commented that it is more precise and reliable, as with direct-touch *“You often zoom in a bit too close, [...] and you have to zoom out again to correct”*. Interaction with Gaze-Touch was perceived as easy and intuitive, since users already look where they want to zoom anyway (*“I always look at the area where I expect the city”*). A user mentioned that it is much less fatiguing in comparison to her own touch-enabled device: *“Because sometimes with the iPad you always use your hands, you get tired”*. In addition, users were positive about no occlusion through hands and less body movement (e.g. *“[With direct-touch] I cover what I see with my hand and when the area is further away I have to lean forward to zoom in with the hand”*). Two users favored direct-touch zooming. The first user thought it was more precise with direct-touch (*“It is a little vague with the eyes”*). The other user stated the gaze-cursor that is used is confusing, as it moved constantly with the user’s gaze.

**Gaze-Touch Experience.** While some users did not notice any difference, other users perceived a different map experience with Gaze-Touch. For example, users stated that Gaze-Touch helps map navigation (*“It helps you on what you are searching, you are not distracted”*). Another user mentioned increased zooming awareness (*“I was more aware of where I zoom”*) and another user perceived it as being guided (*“It is like you are guided”*).

### 3.5.3 Summary

The majority of users preferred Gaze-Touch over direct-touch for zooming. Reasons were speed, less effort, precision, and reachability. Further discussions with users showed that

the map navigation experience is altered; users felt it is more helpful, and increases location-awareness.

### 3.6 Application: Multiple Objects

This application demonstrates interactions using Gaze-Touch *instead of* direct-touch interactions, like the previous application, but with a focus on multi-finger and whole-hand touch gestures. The application allows users to manipulate a large number of objects spread across the surface. It is configurable with regards to number, shape, size, and colour of objects. Users can quickly select multiple objects, and reposition them by dragging gestures. Users can touch down with up to ten fingers, that would lead to 10 object selections. This allows us to experiment with Gaze-Touch's capability of fast and simultaneous manipulation of objects. To overcome the physical friction of the screen and gain fluent and reliable multi-touch, we used multi-touch gloves in our demonstrations. Because of its experimental state, this application was not included in the user study. These techniques can be useful, for example, in visual analytics that commonly involve sorting, searching, or grouping of many objects [WTP<sup>+</sup>95, SGL08].

#### 3.6.1 Implementational Details

Our goal was to optimize object dragging. Therefore a touch down will always map to the target that is closest to the user's gaze point on the screen. Further, one touch will only map to a single target. This allows to quickly select multiple objects, e.g. when touching down two fingers at once, the two objects closest to the user's gaze are selected. In addition, the dragging acceleration from the Image Gallery application is integrated.

#### 3.6.2 Interaction Techniques

##### Instant-Multi-Object-Dragging

Users can instantly select up to five objects to a hand (Figure 3.11). When the user touches down, the system binds the closest object to the finger. If multiple fingers are downed, each finger will get one object associated (a). This can be useful, for example, when sorting a large amount of objects. The user can sort out all selected objects at once by a single dragging gesture (b, c). Immediately after this, the user can continue to sort out the next objects as the user only needs to look at the next objects.

##### Multi-Object-Pinching

We implemented a variant of this application where pinching leads to relative movement of objects toward the user's hand. When the user selects multiple objects as explained above, the user can perform a pinching gesture to move all objects to the hand's position (Figure 3.12). The distance between each finger and the center of all fingers is mapped to the distance between the object and the fingers' center. Thus this technique allows continuous movement of objects toward the hand, but moreover, it can also be used for

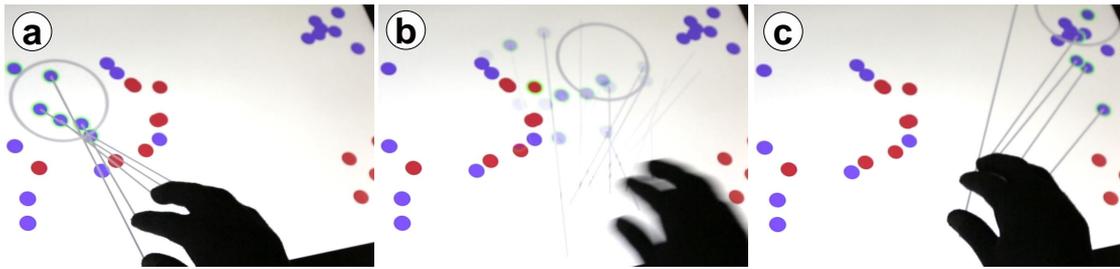


Figure 3.11: Multi-Finger-Dragging: users can select the five closest objects to their gaze by touching down five fingers (a). Users can then sort them out at once with a single dragging gesture (b, c).

positioning anywhere on the screen. To move close objects far apart, the user can start with a small distance between the fingers. By expanding the fingers (pinch-out), the objects would be drawn away (Figure 3.12, from (b) to (a)).

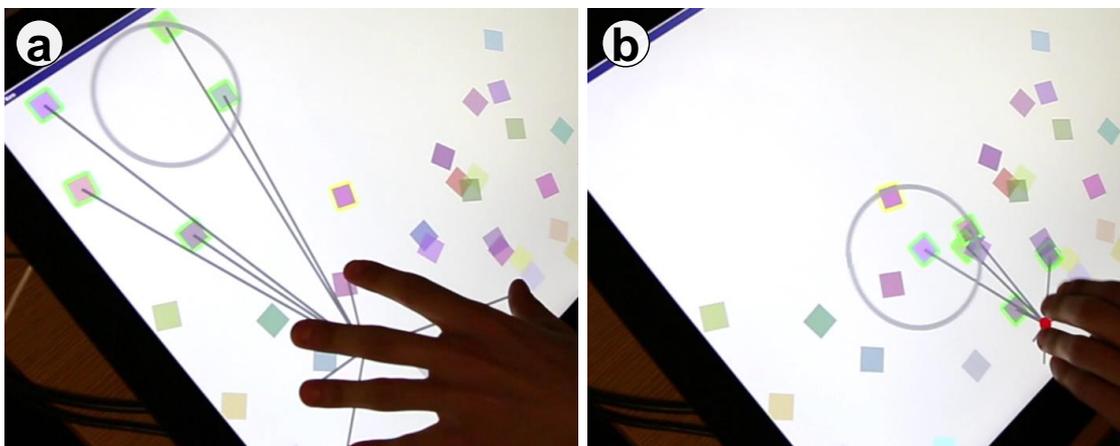


Figure 3.12: Multi-Object-Pinching: when multiple objects were selected (a), a pinching gesture moves the objects to the hand's position (b).

### 3.7 Discussion

Starting from our conceptual analysis we outlined the differences between Gaze-Touch and direct-touch. The beneficial differences that we identified, such as reachability, no occlusion, speed, less fatigue, and less physical movement, were confirmed in our user study. Besides differences, a key characteristic of Gaze-Touch is its similarity to direct-touch. Users can manipulate objects at the moment they place a touch down, they can perform the same multi-touch gestures to manipulate content, and they look at the target before they touch on it. This greatly reduces learning effort as users are already familiar with touch-based interaction, so they can apply the existing knowledge to Gaze-Touch. This was further shown in our user study. Our participants required little training, and were able to get familiar with Gaze-Touch interaction very quickly.

The similarity between Gaze-Touch and direct-touch enables users to switch seamlessly between the techniques. Users can use direct-touch to interact with objects that are within their immediate comfort zone, while they can seamlessly switch to Gaze-Touch for reaching distant objects or modes switching as illustrated in our paint application. Furthermore, direct-touch enables natural single target manipulation by simply touching

an object, and users can employ Gaze-Touch for multi-target operations. Our images application allows the use of both techniques; which led many participants to choose direct for single-target scaling and rotation, and Gaze-Touch for multi-target dragging. Our participants confirmed that these kind of divisions improve the interaction within the applications.

Our work shows potentials and examples of using gaze and touch for interactive surfaces of combined input and output. While we can confirm prior work that this combination allows to efficient reaching of remote targets [SD12b, SD13, TAB<sup>+</sup>13, TBAG14], we discovered additional benefits for surface interaction. A single-touch is now more expressive as it can have many different meanings – users can drag an object like in direct-touch, but also copy, delete, add, or any other task depending on which mode the user looks at. Users can perform the same task either directly or indirectly with Gaze-Touch, in essence providing more expressive input to the same target. Techniques can take advantage of both gaze and touch point, e.g. drag objects to the close touch position, or copy the object that is under the touch. Multiple target manipulations are more efficient. Users look at each target and perform manipulation on the same position, such as zooming into different locations, or manipulate all targets in sight at once, such as sorting of multiple images across the surface.

### 3.7.1 Limitations

#### Eye Tracking

In our setup, the position of the eye tracker is non-trivial because users can occlude the camera’s view. When users positioned their arms in front of the eye tracker, the action can block the tracking of the users’ eyes. Another problem is eye tracking inaccuracy by hardware limits and natural eye jittering, that can increase with a larger surface space [Jac93]. We approached this issue individually for each application: e.g. target assistance when objects were involved (e.g. the menu of Paint application), or by filtering gaze noise (Map Navigation application), however further improvements can allow a smoother Gaze-Touch experience.

#### Inappropriate Tasks

A conceptual limitation of Gaze-Touch is that it requires the user to look at a target of interest. For many tasks the user’s gaze is already at the target of interest, but there are cases where users do not need to look at the target. For example, when users are familiar with the input position, they simply use their muscle memory for input (e.g. PIN entry). This example, however, only applies to input targets that are fixed in location, and in this case Gaze-Touch can simply be disabled. In other cases however, where content is dynamic e.g. image aligning, video editing, or multi-view interfaces, the use of Gaze-Touch might become difficult. In these cases Gaze-Touch is more of benefit when used complementary to direct-touch, e.g. as shown in our Paint application (Gaze-Touch for mode switching, direct-touch for primary task).

### Eye-Hand Coordination

Eye-hand coordination plays a relevant role in Gaze-Touch. Often users already gaze away from the target before acquisition. Known as the ‘Late-Trigger errors’ [KKP<sup>+</sup>08], it can be approached by selection delay or intelligent eye fixation detection, however further investigation is required.

### Multiple Selection and Eye Overload

A Gaze-Touch selection is completely based on the single-channel gaze modality. This principally disallows simultaneous selection of multiple targets. One approach is selecting as many objects close to the user’s gaze as the user touches down fingers (c.f. our ‘Multiple Objects’ application). However, when sequences of tasks require users to visually fixate many points over time, the users’ cognitive or visual abilities might get overloaded. While our principle ‘gaze selects, touch manipulates’, reduces gaze usage to the moment when users touch down, it is yet unknown how much it affects the user’s mental and physical abilities. In this context, it has to be considered that the utility of Gaze-Touch is its complementary nature, in cases direct-touch is limited.

## 3.8 Conclusion

In this paper we introduced Gaze-Touch as a novel interaction technique that facilitates gaze and multi-touch on the same surface. The technique makes existing direct interactions more flexible, as it allows for implicit mode switching by a glance, and manipulation of many targets without directly touching them. This leads to novel application designs where Gaze-Touch can be used complementary or alternately to existing direct manipulation, and even can replace or extend tasks that previously belonged to the territory of direct input. Gaze-Touch enhances touch interactions with seamless and efficient interaction techniques, as reachability, physical movement and fatigue are overcome, while the speed and familiarity with common multi-touch gestures prevail. Gaze-Touch is simple in its core technique, but lends itself to extend surface interactions with dynamic and effortless capabilities.

One of the interesting findings in this work was the possibility to use single touch inputs for both direct and indirect input, depending on the user’s gaze location. We showcased initial potential with our image and drawing application. In the next chapter, we will explore this mechanism in more detail on the example of two direct modalities: pen and touch.

# 4

## Gaze-Shifting: Direct-indirect Input with Pen and Touch Modulated by Gaze

Any direct input device can also be operated in indirect input mode, by redirecting input to a target that is offset from the physical location at which the input is sensed [HW12]. Support of indirect input alongside direct input has been motivated to overcome problems that are inherent with direct input, such as occlusion of the target during manual input [VB07], effort and time required for reaching targets located further from the user [BB05], and precision limitations due to the low “resolution” of fingers [AZ03] and static control-display ratio [MI09]. A range of techniques have been designed to address these issues by extending direct input with indirect input without changing device or modality. For example, multi-touch as well as pen interfaces have been extended for users to be able to dynamically switch between absolute positioning and relative positioning [BWB06, FVB06, WBP<sup>+</sup>11]. The combination of the two modes can enable users to perform the same task interchangeably with direct or indirect input [FVB06], or to switch tasks associated with either mode (e.g., writing/drawing versus command input [GHB<sup>+</sup>06]). A dynamic redirection of input generally increases expressiveness of surface interactions as it allows the same target to be manipulated from different input positions [PACG14].

In this chapter, we explore the combination of direct and indirect modes of input with pen and touch, and propose to use gaze to facilitate the combination. We focus on hybrid *direct-indirect input* techniques that provide an integrated user experience of direct and indirect input. Direct-indirect input requires a switching mechanism, for users to dynamically transition between the two modes, and a redirection mechanism for input to be shifted to an offset target when the user enters indirect mode. Mode switching has been studied extensively for pen interfaces [LHGL05] and a host of sensory mechanisms have been explored [HWP<sup>+</sup>14]. Mode switching of manual touch has been explored in more specific contexts (e.g., to shift input to resolve “fat finger” target ambiguity [VB07]), but can generally be facilitated by dedicated gestures (e.g., [WBP<sup>+</sup>11]). Input redirection is commonly supported by a cursor that appears in indirect input mode, but the redirection can also be context-dependent (e.g., indirect manipulation of a previously selected target) or facilitated by special-purpose widgets (e.g., for target reaching [ALMP09, BB05]).

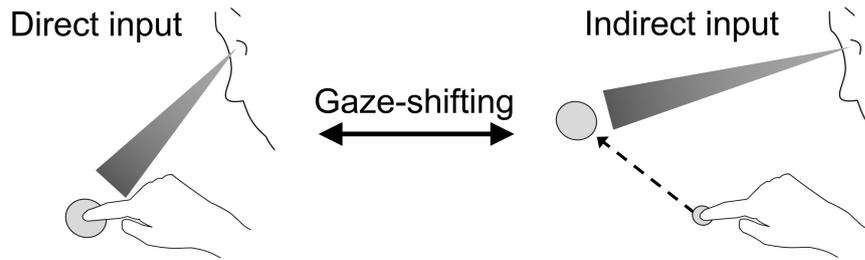


Figure 4.1: Gaze-shifting: manual input by the user is modulated by their gaze. Input is direct when the user’s gaze is on the touch location (left), but redirected to the gaze location if gaze and touch are separate (right).

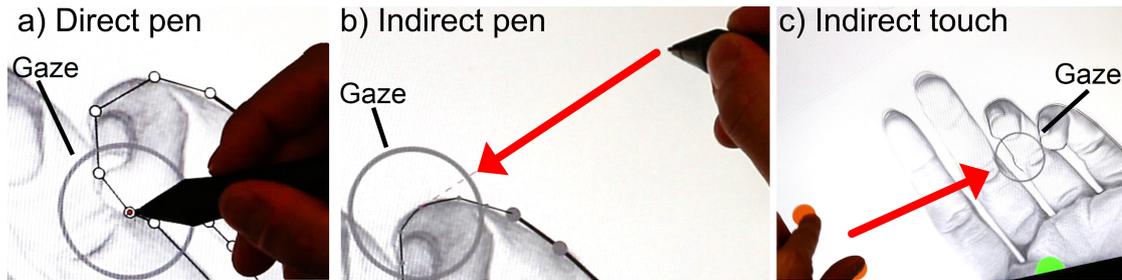


Figure 4.2: Direct-indirect input enabled by gaze-shifting: a) direct pen input is used to outline an image with straight lines; b) indirect pen input is used for curving of gaze selected lines; c) indirect touch is used to zoom in and out relative to the user’s gaze. The grey circles indicate the user’s gaze, and the arrows the redirection of input.

We introduce *gaze-shifting* as a novel technique for direct-indirect input, embracing the user’s gaze to modulate their manual input. Figure 4.1 illustrates the principle of gaze-shifting. When a user initiates manual input in the area under their gaze, then it is processed as direct input at the touch location. When manual input is initiated while the user looks at an offset target, the input is redirected to the target. Gaze-shifting is generic and can be used in the same manner with different input modalities, and specifically pen input, touch input, and combinations of both. The technique provides a natural mechanism for mode switching that is based on the alignment of manual input and visual attention for direct versus indirect input. Implicitly, the technique provides a consistent mechanism for shifting the user’s input from a touch point to a gaze-identified target, and vice versa from an indirectly controlled target back to the touch point.

Gaze-shifting enables the design of novel direct-indirect techniques that combine direct and indirect input with pen and/or touch in cohesive workflows. Figure 4.2 illustrates this with a scenario of tracing an image. A user can mark outline points with direct pen input, and the points are connected by straight lines to create a rough outline. The lines can be edited in detail with indirect pen input for which the user moves the pen off the visual target so not occlude any detail. The user can zoom in and out with indirect touch input relative to the visual target. All three tasks (outlining, curving, zooming) have a distinct mapping to an input mode, and the user can easily transition between these modes.

To explore gaze-shifting, we have integrated a pen and touch display with an eye tracker, and created a design application as playground for the development of direct-indirect interaction techniques. The application provides a meta-interface in which gaze-shifting is used to dynamically configure the four modes of direct pen, indirect pen, direct touch

and indirect touch. We have used this application environment to develop novel direct-indirect techniques that leverage gaze-shifting with touch input, pen input, and a combination of both pen and touch.

Our contributions are as follows. First, we introduce gaze-shifting as a generic concept and technique to facilitate direct-indirect input. Second, we describe a prototype system and application that supports dynamic configuration of pen and touch modalities for use with gaze-shifting. Third, we contribute novel direct-indirect input techniques that leverage pen, touch, and gaze in different configurations and illustrate the design space opened up by gaze-shifting. Finally, we report on first insights on the user experience of direct-indirect interaction with gaze-shifting.

## 4.1 Gaze-Shifting

Gaze-shifting is conceived as a generic mechanism for switching between direct and indirect modes of manual input. The technique enables direct and indirect input to be combined in novel and dynamic ways. We first review the design opportunities afforded by concept, and then discuss system design considerations.

### 4.1.1 Design Opportunities

*Direct-indirect Transitions.* Designers can leverage gaze-shifting to create techniques that leverage not only direct versus indirect modes but also the different transitions between the modes. There are four possible transitions for input to be provided from one position to another: *direct to direct*, *direct to indirect*, *indirect to indirect*, and *indirect to direct*. A designer can assign different behaviours to any of these transitions. Figure 4.3 illustrates the four transition within a line drawing example. A user touch the first and second point directly, and as the user looks at the touches the system interprets touches as direct (a). The user then looks at a point offset from their hands and triggers indirect touch, which extends the line from the direct to the indirect position (b). From this point, the user looks at another remote point and indirectly selects it as well (c). Lastly, the user looks back at their hand, which the system interprets as intention for direct input, and a touch directly selects the point (d). In this example, each transition results in the same effect, i.e. the creation of a line point. However, each transition can also be associated with different behaviour, and we will illustrate this further below with different copy & paste semantics depending on the direct-indirect transitions (Fig. 4.12).

*Input-to-Task Mapping:* A developer can map tasks to direct and indirect input in two variations. First, each direct and indirect modality can map to a different task, e.g. pen inking with direct (Figure 4.2a), and pen line editing with indirect modality (4.3b). This provides the user with the benefit of easily switching between two different operations. Second, both direct and indirect modality can map to the same task, such as line drawing (Figure 4.3). This enables the user to choose whether to perform the same operation directly for easy selection of reachable points (a-b), or indirectly to quickly reach remote points or for avoiding hand occlusion (c).

*Input/output Mapping:* Direct interaction is based on a 1:1 mapping between input and output, but indirect input can have variable mappings that we can consider as interaction

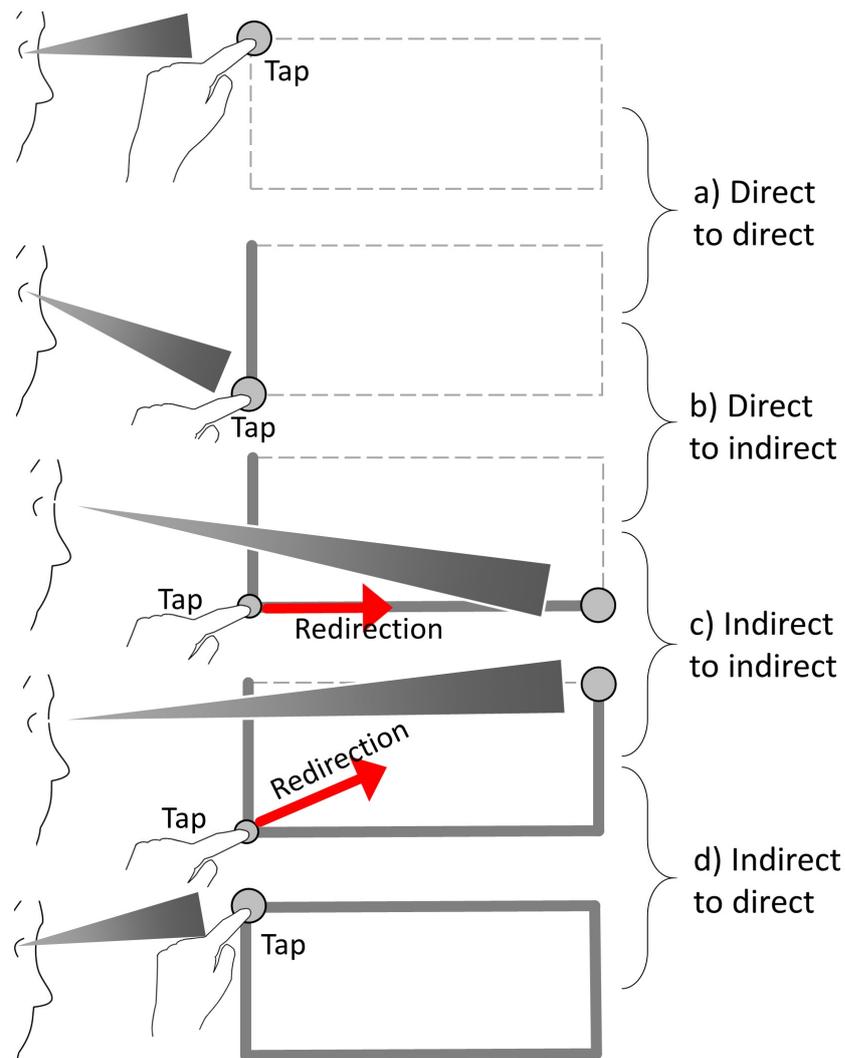


Figure 4.3: Direct-indirect transitions: successive input can involve four types of transition. In this example, the same task is completed while transitioning across different modes. Alternatively, a designer can associate each type of transition with different semantics.

models (Figure 4.4). For example, indirect touch can adopt a model similar to a laptop's touchpad: a cursor becomes visible after shifting to indirect mode, (1) an indirect touch dragging moves the cursor, (2) double tap selects a target, and (3) dragging after double tap manipulates the target. A distinct cursorless indirect model is Gaze-touch [PACG14], where indirect touches issue the same 'direct touch' effect on the gaze-identified target: (1) look to indicate the target, (2) indirectly touch down to select it, and (3) touch drag to manipulate the target. In comparison, cursor based models increase precision as users precisely position the cursor. Cursorless models only use gaze to select which is less precise, but faster as users immediately manipulate without prior cursor dragging. Notably, a hover-enabled device such as a pen can adopt a digitiser or mouse like model: (1) hover drags the cursor, (2) pen down clicks, and (3) pen drag manipulates an object.

#### 4.1.2 System Design Considerations

From a system's perspective, gaze and manual input are two positions on the input surface. As illustrated in Figure 4.5, the system evaluates whether the input is within

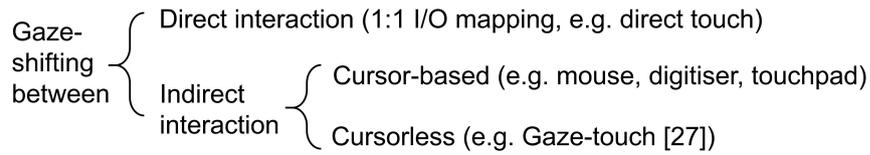


Figure 4.4: Input/output mappings for direct and indirect input

the user’s gaze area, by applying a threshold on the distance between gaze point and manual input position.

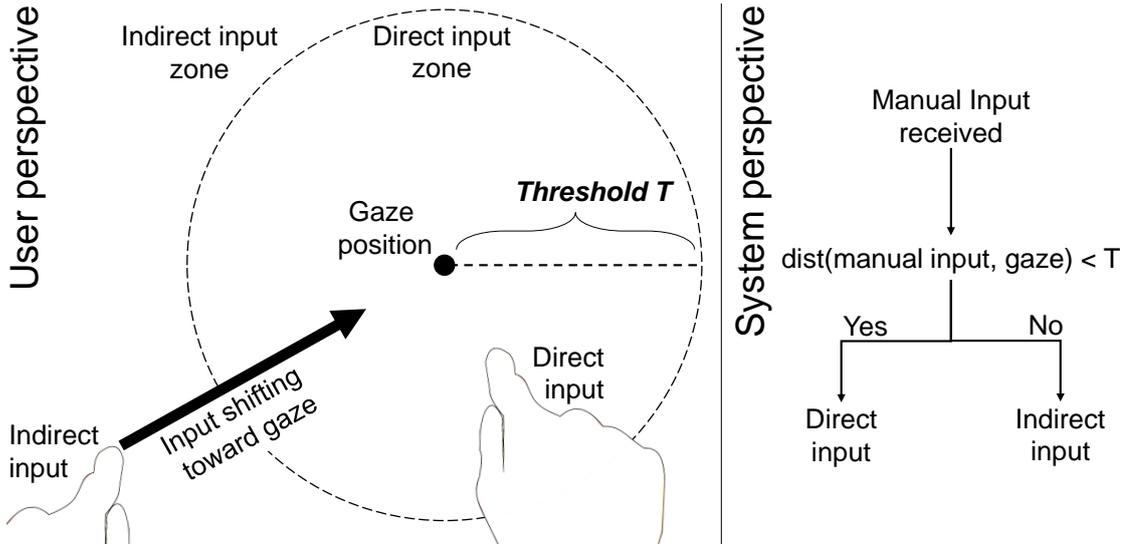


Figure 4.5: Gaze-shifting from a system’s perspective

*Threshold Configuration:* The threshold defines the boundaries of direct and indirect input. Larger thresholds make indirect input difficult as users need to move their hand far away from their gaze, and smaller thresholds make direct input difficult as users need to keep their input close to where they look. As a starting point, the threshold can be based on human vision. The direct input zone includes the foveal vision, an area of higher acuity, and the indirect input zone aligns with the peripheral vision where vision is generally poor.

*Context-awareness:* The threshold should consider the task’s context and the ratio between needed direct and indirect interaction. Principally, activation can trigger at any input event, e.g. at pen down, drag, or up, as users can move their gaze arbitrarily. In practice, the context of the task at hand needs to be carefully considered. For example, gaze-shifting can be disabled when users draw with a pen, to avoid unexpected behaviour when users briefly look away. On the other hand, when users looked at a menu and issue manual input from clearly far off, it is likely that the user intends to trigger indirect input to interact with the menu. When user interaction is mainly based on direct manipulation, a large threshold is recommended to avoid false positive errors (detect indirect instead of direct).

## 4.2 Gaze-Shifting for pen and touch

here show and discuss the conceptual application of gaze shifting to pen and touch; with the sketches from the presentation.

but maybe, its better to just enhance the application section with the sketches. and add the "all" figure at the end of the presentation to the discussion section.

## 4.3 Application

We now describe our prototype design application. The idea is to provide basic drawing, design, and manipulation tools and with them explore gaze-shifting across different contexts. Our application is based on the following components:

- Input devices: pen and touch.
- Input modes: direct/indirect pen, direct/indirect touch.
- Toolbars: 4 input mode toolbars (one for each), 3 pen specific toolbars (brush size, shapes, colour palette).
- Modes: Each toolbar offers a set of modes. A mode enables a specific drawing or object manipulation operation with an input mode.
- Configuration: a state where all input modes are mapped to a mode.

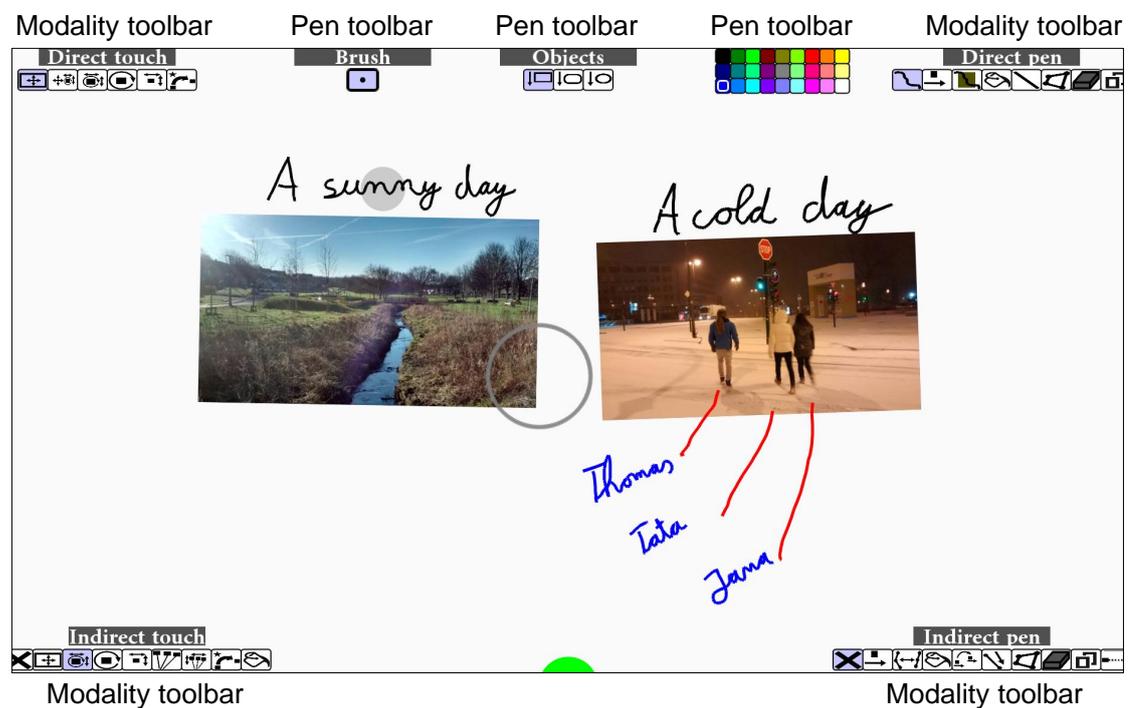


Figure 4.6: The user interface and its toolbars

### 4.3.1 Toolbars

Figure 4.6 shows an instance of the UI and its toolbars. Horizontally, both types of input device are shown (left touch *versus* right pen), and vertically, both types of input mode

are shown (top direct *versus* bottom indirect). In light of pen and touch application principles [BFW<sup>+</sup>08, HYP<sup>+</sup>10], we used inking modes for pen, manipulation modes for touch, and some commonly used modes:

*Pen Toolbars:* Both direct and indirect pen toolbars include conventional operations such as free inking, move object, fill object, draw line, draw shape, eraser, copy & paste of objects, and a default no-mode mode. Modes exclusive for indirect pen are object editing, where users edit vector points of a shape, and object association, which allows users to draw a connection between two objects with a dotted line.

*Touch Toolbars:* Both direct and indirect touch toolbars include Five RST based modes. These enable single-point dragging and two-point pinching gestures. They include one mode for global navigation, where the whole canvas is pan & zoomed, and one mode for single object RST that affects one object. Additional modes are pinch gestures that only affect R, S, or T dimension to support separability of touch gestures [NBBW09]. Lastly, the ‘transfer’ mode allows instant object transfer from a direct touch position to the user’s gaze position; also a default no-mode mode is available.

The additional brush, objects, and colour toolbars allow users to configure the brush size, create primitive shapes, and set a colour for the drawing modes, respectively. Notably, each toolbar dynamically changes its opacity so that the user’s focus is not detracted from the workspace. A toolbar fades in quickly when looked at (.2s), and fades out slowly when looked away (1.6s). ‘Looked’ is detected when the user’s gaze is within 150px around the toolbar. The minimum opacity is 5%, so that users can easily find the toolbars (Figure 4.6 shows all toolbars at 100% for descriptive purpose).

Users can employ direct and indirect input to interact with the toolbars. First, when looking at a toolbar and issuing direct pen/touch input on it, the mode is directly selected by pen/touch. Second, users look at the toolbar, put the pen or finger down on the canvas (Figure 4.7a), and indirectly drag left or right to move the toolbar’s selection mask to the corresponding direction (b).

### 4.3.2 System Implementation

Our system consists of three main parts: a *pen and touch display* (Wacom Cintiq 22HD touch), an *eye tracker* attached to the bottom of the display (Tobii EyeX, 30 hz,  $\approx 1^\circ$  of visual angle accurate), and a *laptop* (Windows 7, 8GB RAM, dualcore i7 2.9GHz CPU). The display is 1080p, 21.5" (475 x 267mm), and supports 10 finger multi-touch and a Wacom Grip Pen. Default Windows 7 settings distinguish pen and touch, and disable touch at pen hover/down. The display is oriented at  $45^\circ$  toward and  $\approx 60$ cm away from the user. Hand occlusion of the tracker’s view to the eyes is mostly avoided as hands are usually held around the view. The software is implemented with Java using MT4J (Multitouch for Java v0.9, <https://code.google.com/p/mt4j/>, 08/04/2015). We use dynamic gaze sampling, using raw data at fast eye movement ( $>75$ px between two samples), and otherwise sampled data for 1s (30 samples) to enable quick travel of distances, but a smooth gaze indication when fixating an area.

Gaze-shifting implementation is based on Figure 4.5 with the following thresholds. For toolbars, indirect is detected when users look  $<150$ px (37mm) close to the menu’s border, and issue input from outside. When no toolbar interaction is detected, on default the

	Figure	Pen	Touch
Draw directly, configure pen indirectly	4.7	✓	
Trace line directly, edit line indirectly	4.8	✓	
Trace line directly and indirectly	4.9	✓	
Select colour directly, set colour indirectly	4.10		✓
Direct object scale, indirect UI zoom	4.11		✓
Direct and Indirect Copy & Paste	4.12	✓	✓
pen edits, touch zooms	4.13	✓	✓
Select directly, transfer indirectly, draw directly, associate indirectly	4.14	✓	✓

Table 4.1: Overview of the direct-indirect interactions (*rows indicate the task, columns the integrated modalities*)

system continuously checks direct or indirect mode based on a 350px (86mm) radius. For touch input, the detection occurs at touch down, drag, and up events. For the pen, hover events are used in addition. When users hover into ‘indirect’ mode, techniques can show a mouse cursor that initially appears at the user’s gaze in order to provide feedback before users issue input.

#### 4.4 Direct-indirect Interaction Examples

We now describe direct-indirect interaction examples. Each consists of a fixed mode configuration of our application. We categorise them into tasks for pen, touch, and combined input. Table 4.1 overviews the examples and used modalities. Each example demonstrates a specific use of gaze-shifting, and collectively, they depict the flexibility of gaze-shifting’s application across tasks, techniques, and modalities. The figures indicate the user’s gaze with a grey circle.

#### 4.5 Pen

The following examples describe how gaze-shifting aids pen-specific functionality such as inking, shape design, and hover based techniques.

##### 4.5.1 Draw Directly, Configure Pen Indirectly

This example demonstrates rapid pen mode switching. Direct pen input enables standard inking, and indirect pen interaction with toolbars. We added three toolbar variations specifically designed for the pen. In general, the menus are controlled by using gaze to select the overall menu, indirect pen down confirms menu selection, and indirect pen dragging moves the selection mask across the menu.

In particular, we implemented three variations of this interaction, each demonstrated in a specific pen menu type. First, the ‘brush’ toolbar displays only one mode which visual icon changes size according to the dragging. Pen dragging from left (c) to right (d) increases brush size. Second, the ‘objects’ toolbar has extended functionality: dragging left/right changes the type of a shape, but dragging from up (e) to down (f) creates the selected shape at the pen’s direct physical position. This allows users to quickly create new notes of various shapes. Then, the user can further reposition the object (here: a rectangle) with direct pen. Lastly, the ‘colour palette’ toolbar demonstrates how users can easily interact with a menu that involves many small-sized modes. A grid of 9x3 colours is provided (g-h), on which users perform indirect 2D dragging to select a desired colour.

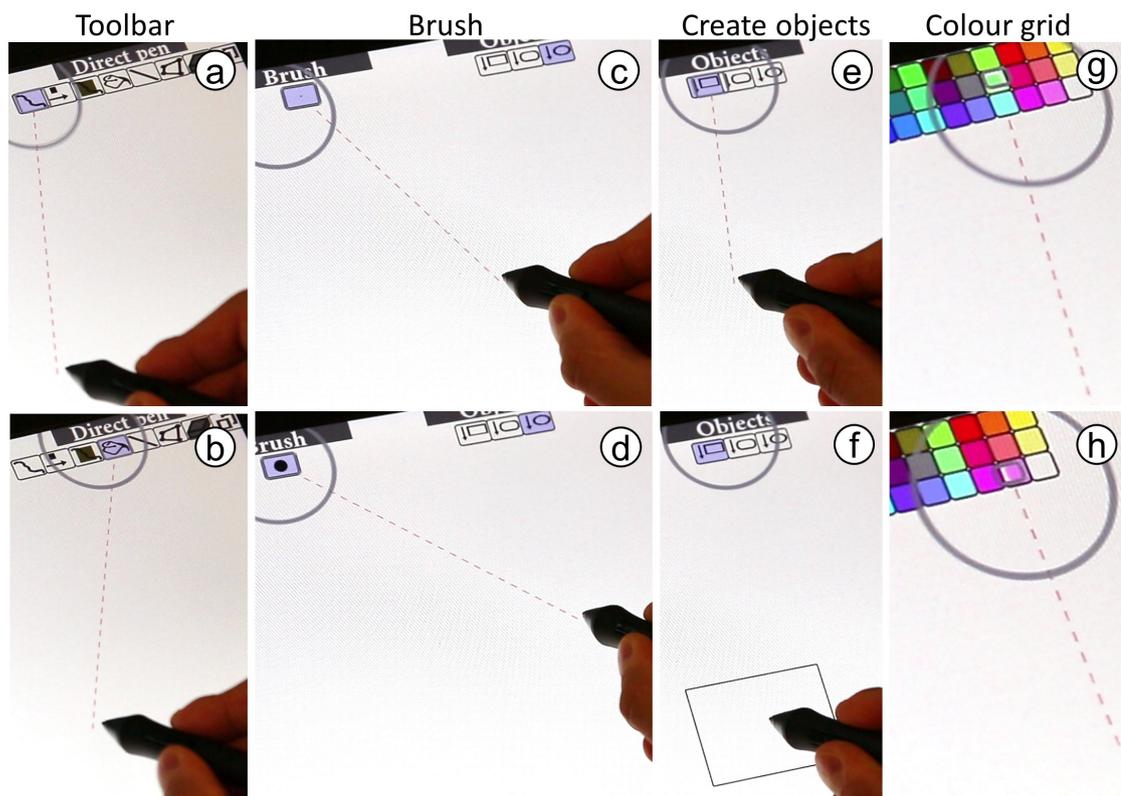


Figure 4.7: Draw directly, configure pen indirectly: users switch a toolbar’s mode by look and indirect touch down (a), and vertical drag (b). Special toolbars are ‘brush’ that displays one adjustable widget (c-d), ‘objects’ where a drag down creates the selected shape at the pen’s position (e-f), and ‘color palette’ where 2D dragging switches colours (g-h).

#### 4.5.2 Trace Line Directly, Edit Line Indirectly

This example demonstrates how users shift between two different modes with a pen. Tracing is a common task, e.g. to digitise hand-drawn figures. A method to accomplish this is drawing straight lines at the outline, and then rounding them off to align them to the figure. These two tasks can be interleaved into pen input: direct pen input allows straight line drawing, and indirect pen input allows rounding off the lines.

For example, when users look at the area of the pen, users draw straight lines (Figure 4.8a). In particular, users perform pen down on each point of a shape, and the

points get interconnected with straight lines. A shape finishes when users connect the current point to an existing point, or perform double click. When pen hovers out of the user's sight, the system switches to line edit mode, where users curve lines with indirect pen input (b).

Hover into the indirect pen mode changes the visualisation of the shapes. Each straight line then shows three points (start, center, end) enlarged to indicate their editability. Hover into indirect also makes a mouse cursor appear at the user's gaze (b). Indirect pen hover moves the cursor to precisely select points. Pen down selects a point (c), and pen drag moves the point 1:1 which adjusts the roundness of the line (d). Pen up releases the selection.

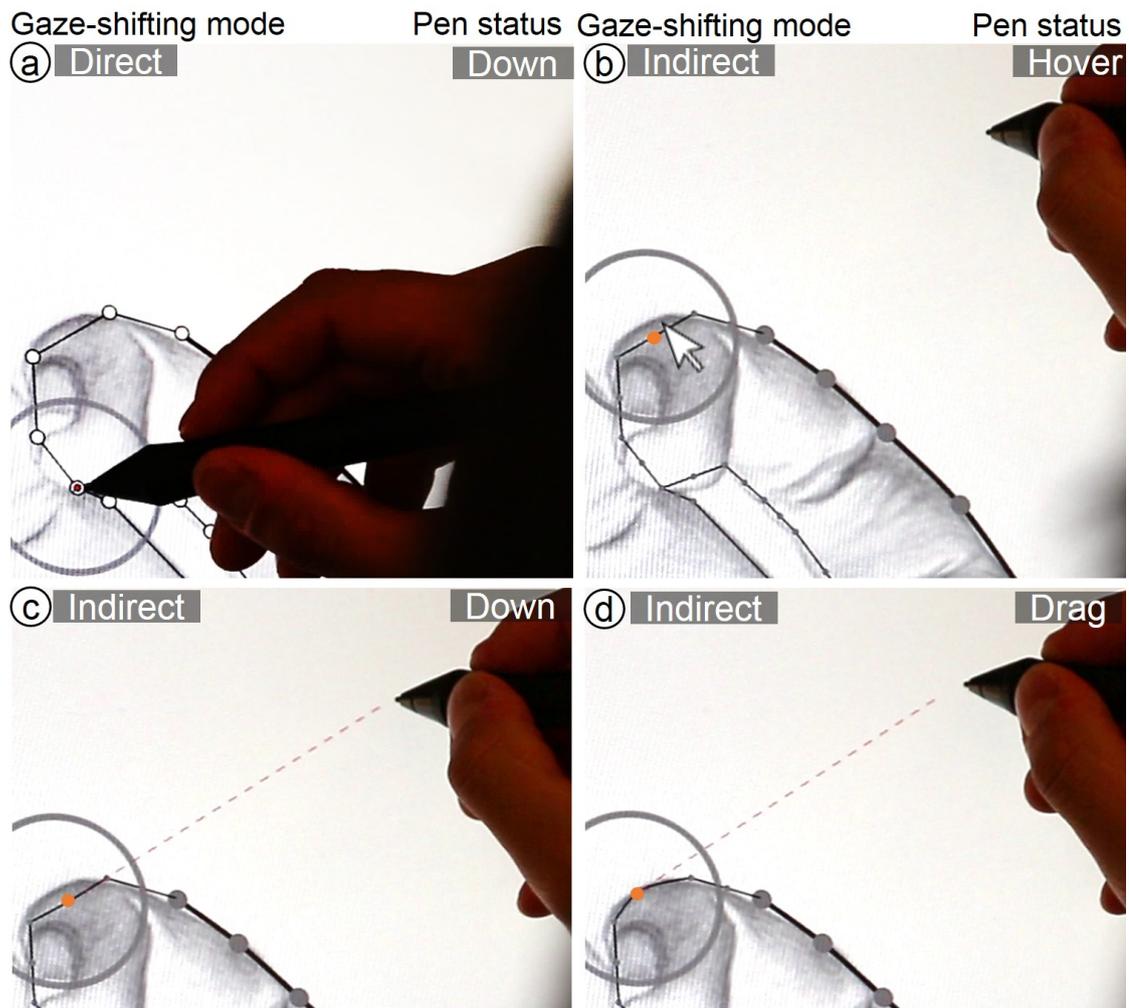


Figure 4.8: Trace line directly, edit line indirectly: direct pen input enables drawing straight lines from point to point (a). Transitioning from direct to indirect pen input makes a mouse cursor appear at the user's gaze. Indirect pen hover moves this cursor (a), indirect pen down selects a specific point of a line (b), and pen dragging moves this point which adjusts the roundness of the line (d).

The cursor appearance is delayed by 300 ms to avoid unwanted cursor movement when users just moved the pen to indirect mode. To avoid pen hover clutching for long distances, the cursor automatically jumps to the gaze position (based on MAGIC [ZMI99]) if the mouse cursor is far away from the gaze position ( $>250\text{px}$  for 300ms).

### 4.5.3 Trace Line Directly and Indirectly

This examples demonstrates how users can facilitate direct and indirect pen input for the same outlining task. It can be useful for tracing shapes of larger size, creating lines that stretch across the display, and for general dot-to-dot designs. The main benefit is that this technique reduces the need to reach across distant points on the display. Tracing involves creating multiple points that become connected to form an outline. Creating points is consistent across direct and indirect input: a pen down denotes trace point creation, pen drag refines its position, and a pen up confirms it.

Figure 4.9 illustrates this technique. A tap of direct pen input creates a new point at its position (a). When looking far away from the pen's position, the next point gets previewed at the gaze position (b). Putting the pen down (indirectly) will confirm the current gaze point as a desired position for the outlining point (c), followed by indirect pen dragging to refine this point's position (d). Pen up confirms the point and begins the next point. Again, this point snaps to the gaze position (e) until pen down (f), followed by pen drag to refine this point's position (g), and a pen up to confirm this point. Users shift back to direct mode by looking close to the pen, where the outlining point snaps to the pen's hover position (h). A direct tap will create the next shape point (i). Notably, users can also indirectly begin the shape in a remote point, or complete a whole shape without using direct pen input.

## 4.6 Touch

The following examples describe gaze-shifting interactions specifically for finger tapping and pinch-to-scale touch gestures.

### 4.6.1 Select Colour Directly, Set Colour Indirectly

Our pen mode switching examples (Fig. 4.7) use direct input for primary manipulations, and indirect input for mode switching. In this example, we want to demonstrate the other direction: direct input for mode switching (colour selection), and indirect input for primary manipulations (colour set). When looking at the toolbar (Fig. 4.10a), users can directly select a colour (b), and when looking at an object in the canvas, the same touch will apply the selected colour to the gaze-identified object (c). This enables a dynamic interplay between selecting and setting colours, with potential application for formatting text, adjusting shapes, or filters on images.

### 4.6.2 Scale Object Directly, Zoom UI Indirectly

This technique demonstrates how users rapidly shift between object and UI pinch-to-zoom operations. To get both operations with direct manipulation, touch on objects is typically used for local object-based RST, and UI zoom is active when touching blank space. This can be difficult with many objects, or in case blank space is hard to acquire. To approach this, we propose object-based RST when interacting with direct touch, and UI canvas zoom when interacting with indirect touch.

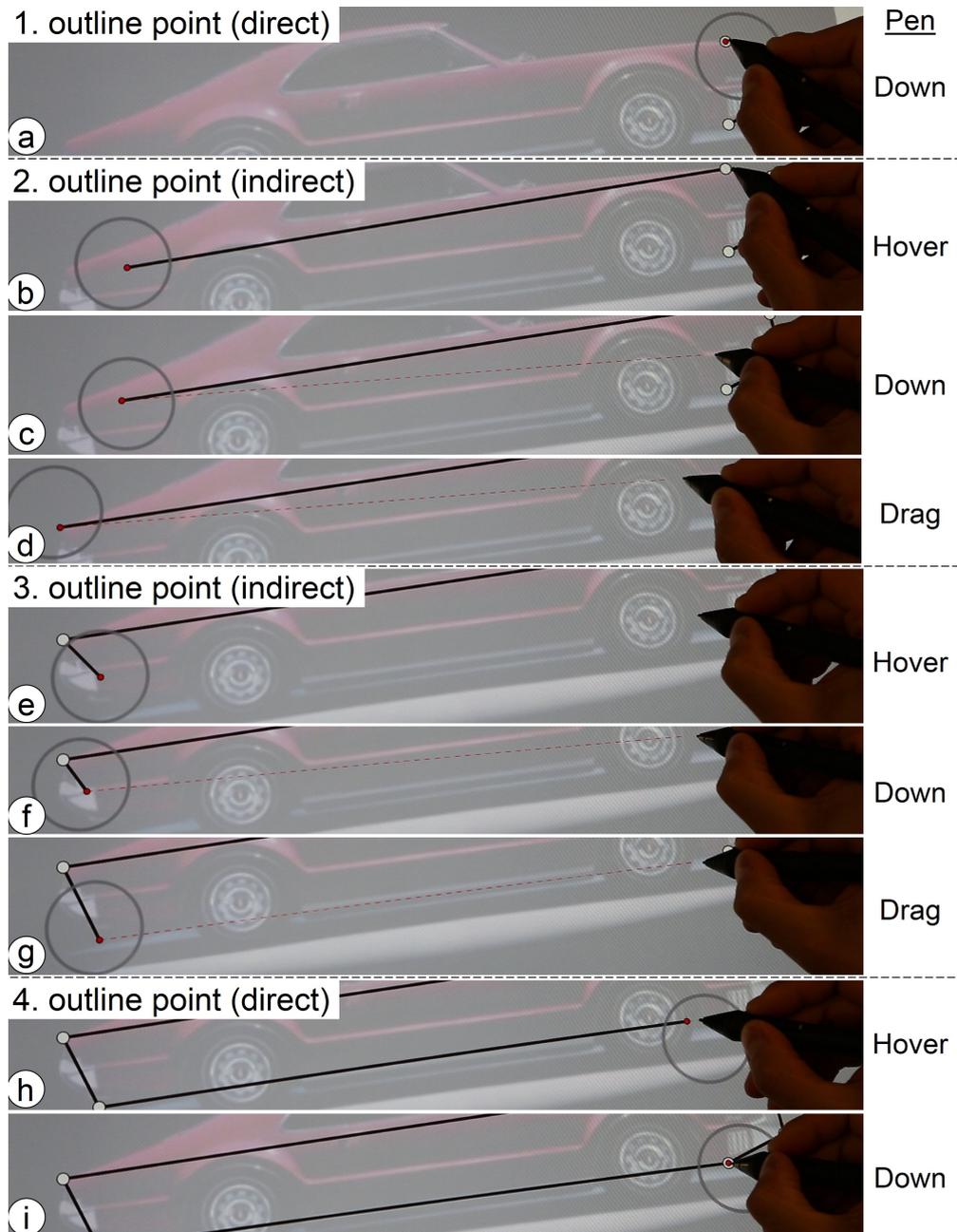


Figure 4.9: Trace line directly and indirectly: direct tap sets an outline point (a). The user lifts the pen and looks at the next point, which extends the line to the gaze (b). Indirect pen down takes over (c), and pen dragging refines the point's position (d). Pen up confirms the point and initiates a new point snapped to the user's gaze (e). Indirect pen down (f) and drag refines the next point (g). Pen up and look back shapes the line to the hover position (g), and direct pen down sets the next outline point (i).

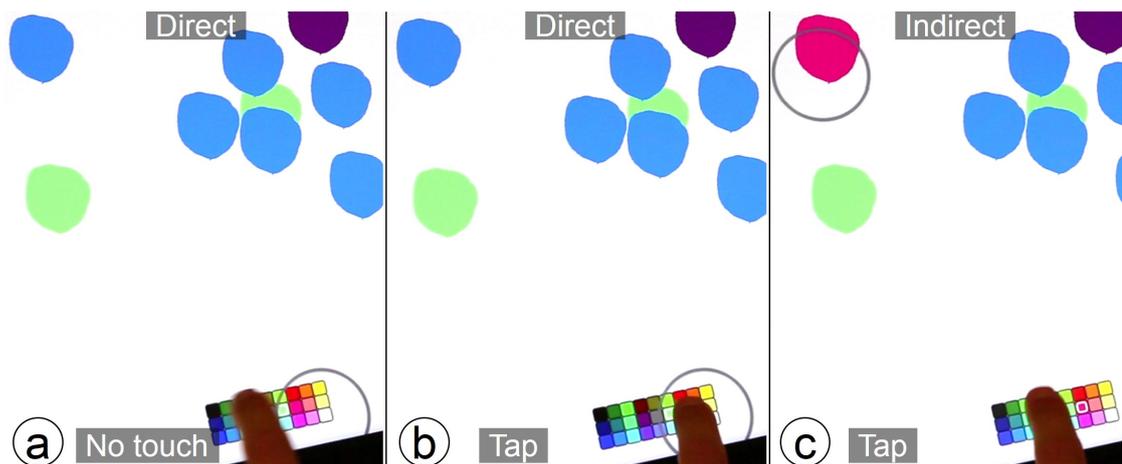


Figure 4.10: Select colour directly, set colour indirectly: select the colour by looking at the toolbar (a) and directly tapping on the colour (b), and set the colour by looking at the target and an indirect tap (c).

Figure 4.11 illustrates this technique. Global manipulation is active when users indirectly pinch far apart from their gaze position (a), which will zoom the canvas into the user’s gaze location (b). When a desired image is found, users directly manipulate this image with direct touch (c-d). Users can immediately return to UI manipulation by moving the hand off the gaze position (e-f).

## 4.7 Pen and touch

We now describe interactions that use both modalities. In the first example, pen or touch is interchangeable, and the other examples divide the labour between the modalities.

### 4.7.1 Copy & Paste Directly and Indirectly

This example demonstrates how the addition of gaze-shifting can principally enable five variations of the same task, on the example of copy & paste. This is possible as each technique is based on a different gaze and manual input behaviour, yet the copy object and paste destination position are consistently located within the user’s gaze. Each copy or paste can be direct or indirect. Pen down selects the copy target, and a pen drag gesture will paste it. Figure 4.12 illustrates the techniques:

- *direct-to-direct*: direct input copies and pastes at the pen’s physical position (Figure 4.12a). This is the standard method as used in conventional applications.
- *Direct-to-indirect*: direct input can copy the object, but after looking at a remote target, the indirect drag gesture pastes the object at the gaze position (b). This is useful when users easily reach the object, but hardly the paste destination.
- *indirect-to-direct*: Indirect pen copies the object, and by looking at the pen’s position, direct drag pastes the object at this position (c). This can be useful to bring remote objects close to the user, to then perform further manipulations directly on it.

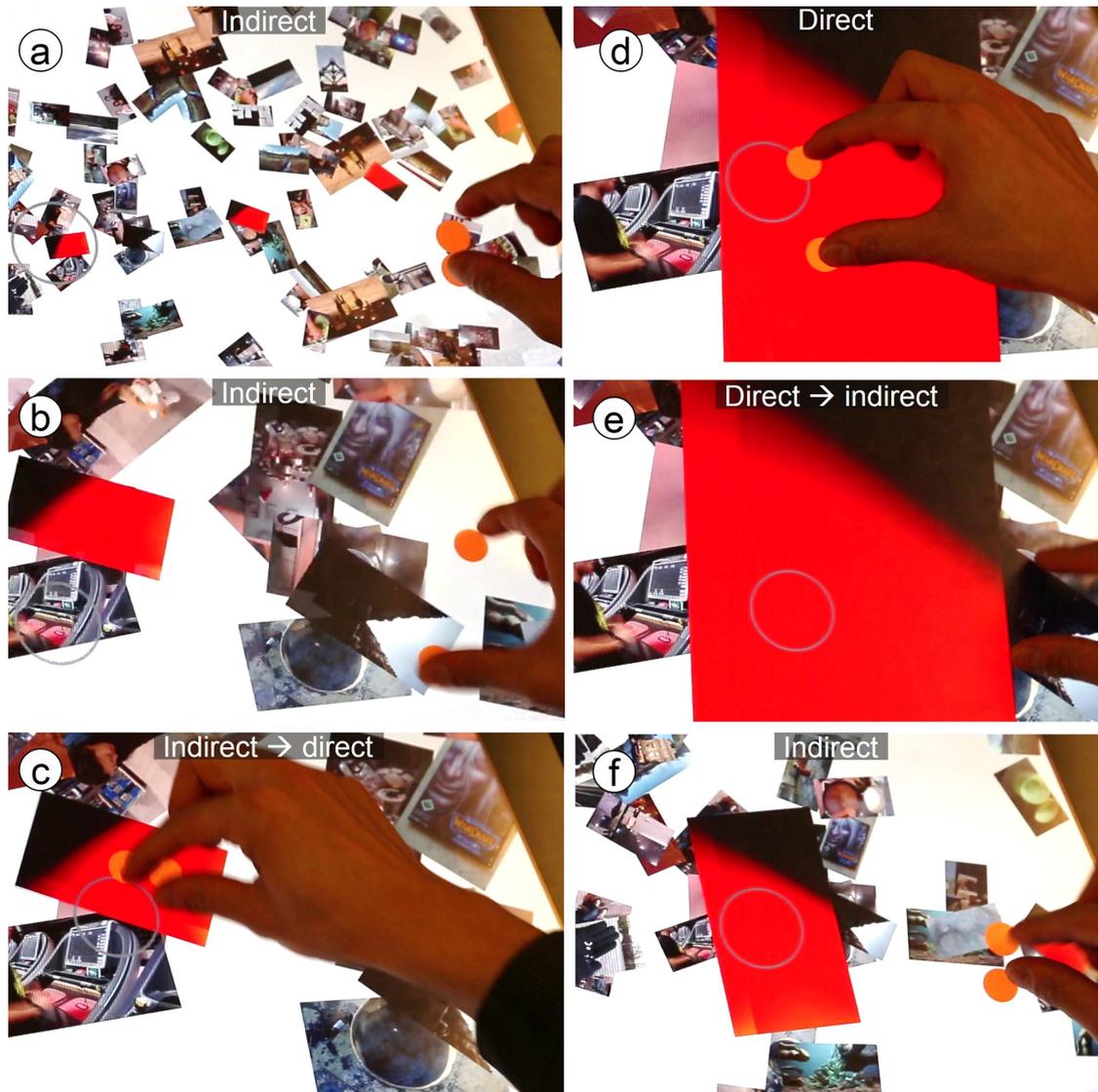


Figure 4.11: Scale object directly, zoom UI indirectly: users can globally zoom the UI with indirect touch (a-b), locally zoom objects with direct touch (c-d), and immediately return to global zooming (e-f).

- *indirect-to-indirect (same)*: indirect pen copies and pastes the object at the same gaze-identified position (d), with the aim to quickly replicate remote objects.
- *indirect-to-indirect (different)*: indirect pen copies the object, and by looking at a remote target, an indirect pen drag gesture pastes it there (e). In this method, users need to look at two remote positions, providing the user with the freedom to copy anywhere located objects to any position.

#### 4.7.2 Pen Draws and Edits, Touch Zooms

This example demonstrates how indirect pan & zoom can be used complementary to pen activities. Figure 4.13 illustrates the interaction. A user directly traces the outline of a figure (a), then lifts the pen to perform zooming. The user indirectly touches down (b) and performs pinch-to-zoom that affects their gaze position (c). Afterwards, the user continues editing the line with indirect pen input (d). This example supports users

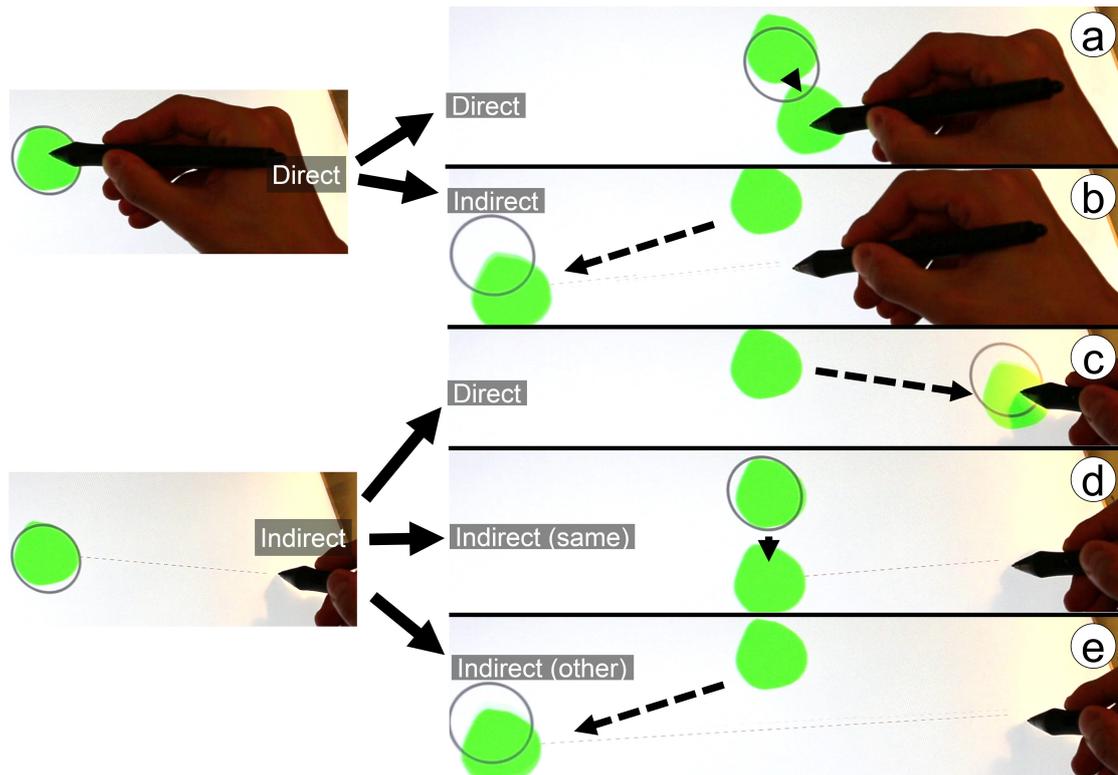


Figure 4.12: Five techniques for copy & paste: direct copy & paste (a), direct copy & indirect paste (b), indirect copy & direct paste (c), indirect copy & indirect paste at the same position (d), and indirect copy & indirect paste at another position (e).

when pen and touch physically interfere on the same target, as both modalities can be temporally used indirectly, from remote.

#### 4.7.3 Draw Directly, Associate Indirectly (Pen), Select Directly, Transfer and Manipulate Indirectly (Touch)

Lastly we present an example where gaze-shifting supports more complex combinations of techniques. A modeling scenario includes the following techniques:

- *Direct pen*: allows free drawing/writing (Figure 4.14a).
- *Indirect pen (on toolbar)*: creates notes at the direct pen's position as illustrated in Figure 4.7e-f.
- *Indirect pen (on note)*: allows to establish associations between notes. A user looks at a note, performs pen down (Fig. 4.14e), then looks at a second note (f), and performs pen up (g). This will create a visual line between both notes. Users can delete it by drawing another association between the same notes.
- *Direct/indirect touch*: direct or indirect touch is set to RST, and thus is used to rescale or move note objects (Figure 4.14h).
- *Direct+indirect touch*: sequentially combined direct and indirect touch input is used to transfer objects. Directly touch a note (Fig. 4.14b), look at a remote position (c), and the note is transferred to this position (d). Vice versa, users can from indirect to direct position. The transfer is delayed by 300 ms, so that the object won't jitter along the user's gaze.

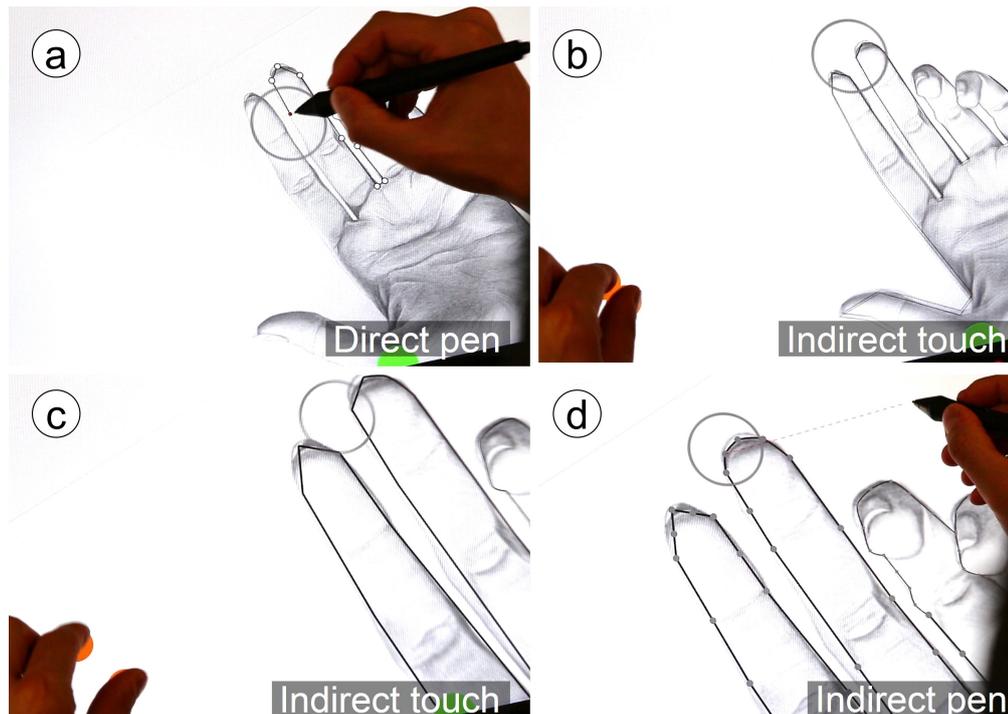


Figure 4.13: Pen writes and edits, touch zooms: A user first directly traces with the pen (a), performs indirect gaze-directed zooming with touch (b-c), and then performs line editing with indirect pen input (d).

Figure 4.14 illustrates a whole scenario: a user first labels the notes with direct pen input (a), then uses touch to transfer the objects into a spatially arranged model (b-d). After this step, the user establishes logical connections between the notes using indirect pen input (e-g), and lastly users adjust the notes' positions with RST manipulation of indirect touch input(h).

## 4.8 User Study

The study goal is to gather insights about the actual act of gaze-shifting, and what potential issues occur during its use. We designed a task for pen, touch, and their combination:

*Pen (Figure 4.16a)*: The pen tracing task investigates shifting between direct line drawing and indirect line editing in an image outlining task (c.f. Figure 4.8). Users traced six shapes sequentially as presented in Figure 4.15. Users were instructed to use few lines, e.g. four lines for the first shape (2 straight lines, 2 lines to round). Each shape appeared at the screen center with an average size of 373x329px. User instructions included what users can do with direct/indirect pen, and how they can switch between them.

*Touch (Figure 4.16b)*: The puzzle task investigates shifting between direct touch for RST and indirect touch for object transfer. The 'indirect to direct' technique as shown in Figure 4.12 was used, but instead of copy/paste, objects are transferred: look at the remote target, touch down locally, look at this touch point and the object snaps to it. Each shape of Figure 4.15 was cut in six pieces, randomly placed and oriented, and users had to solve the puzzle (one shape at a time). The six pieces were scattered at the

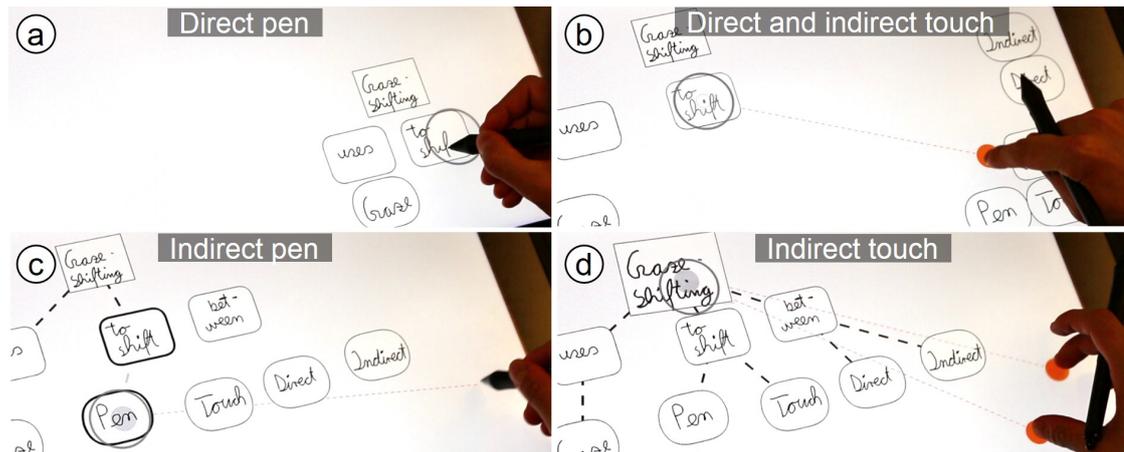


Figure 4.14: Users can directly draw on notes with the pen (a). Then, users can transfer notes by directly touching the note (b), looking at the transfer destination (c), where it is transferred to (d). Next, users can create associations between notes, by looking at a note and indirect pen down (e), looking at the second note (f), and pen up (g). Lastly, users can perform indirect touch manipulation on notes (h).

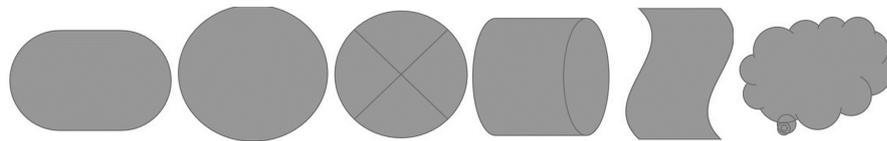


Figure 4.15: Used shapes in the user study for tracing and puzzle tasks

screen's top area, from which users transferred the objects to the screen's bottom area. Then, users solved the puzzle with direct touch RST gestures.

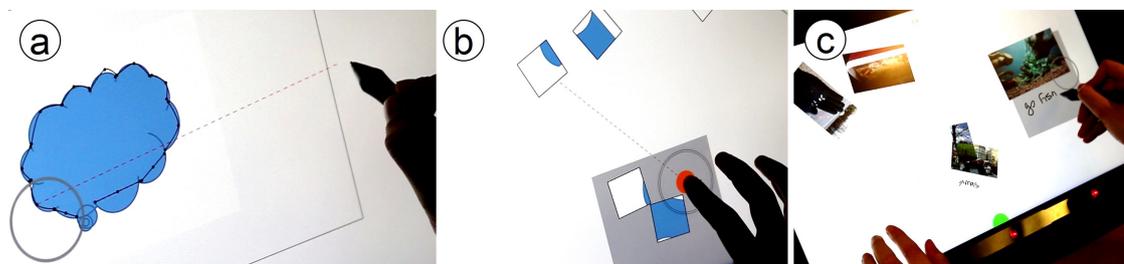


Figure 4.16: Study tasks for pen (a), touch (b), and both (c)

*Pen and Touch (Figure 4.16c):* This image annotation task investigates the shifting between direct pen and indirect touch input. Five images were presented, piled together in the display's top left area. Indirect touches trigger RST image manipulation on the gaze target (like Fig. 4.13, b-c). The task procedure includes: indirect touch to move an image to the pen, direct pen to annotate it, and then indirect touch to drag it back to the pile. Users did this successively for five image, fostering frequent switching between direct pen inking and indirect image movement. Users repeated this task four times.

The study began with a demographic user questionnaire. The study tasks started with the two pen only, or touch only tasks to get familiar with the single modalities, and ended with the combined pen and touch task. After each task, users rated six Likert scale questions (c.f. Fig. 5.16) about the act of shifting between direct and indirect input. A short interview followed up about occurring errors, users' (dis-) likes, and potential

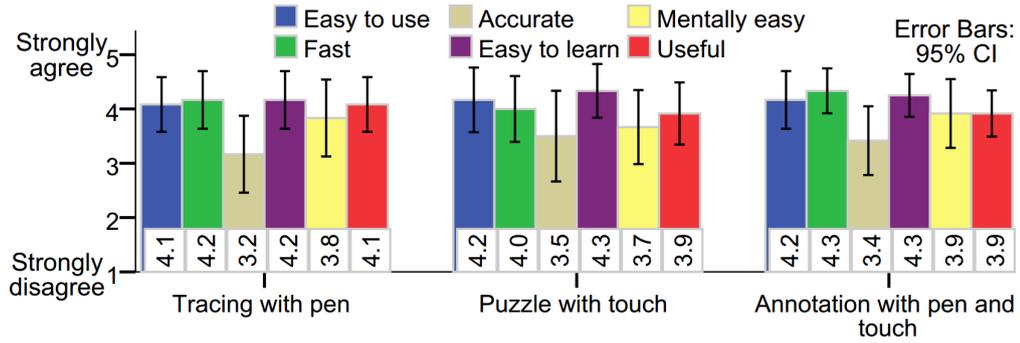


Figure 4.17: Likert ratings of gaze-shifting across the study tasks

applications ideas. Toolbars and non-relevant input modes were disabled. Each user session lasted  $\approx 30$  minutes. The questionnaires are provided in appendix .1.

12 right-handed users from 23 to 34 years old participated ( $M=29.6$ ,  $SD=5.2$ , 3 female). On a scale between 1 (None) to 5 (Expert), on average users were experienced with multi-touch ( $M=4.1$ ,  $SD=1.1$ ), moderately experienced with stylus ( $M=3.2$ ,  $SD=1.2$ ) and gaze interaction ( $M=3.1$ ,  $SD=1.5$ ), and less experienced with combined pen and touch ( $M=2$ ,  $SD=1.3$ ). 4 users wore glasses, and 2 used contact lenses.

#### 4.8.1 Results

The results of our study indicate that despite some accidental gaze-shifting activations users can quickly employ gaze-shifting as they successfully completed all tasks. On average, users rated gaze-shifting as easy to use, fast, moderately accurate, easy to learn, not mentally demanding, and useful across the tested tasks (Figure 5.16). Seven users had eye tracking accuracy problems when leaning forward to ink, for which users had to adjust back.

##### Tracing with the pen

Most users needed 2-5 trials to understand the mechanism, and two users required more trials. Users liked shifting between the two modes (*“its practical, its fast and makes sketching very easy”*), and saw potential for pen mode switches (*“it would be really useful for drawing with graphics, and to change the tool or brush”*) and large displays (*“you can manipulate the points by positioning the stylus at one point, so you don’t have to move the stylus over there”*). Initially, errors occurred when users held the pen just between direct and indirect input, which became less frequent with increasing experience (*“sometimes when I came closer, I was confused, but once I got used to it, it made sense”*).

##### Touch puzzle

Users were used to direct manipulation and initially reached out, but although users reported that they had to *“think different, you look at it and bring it with the eyes”*, all users got quickly used to it within 2-5 trials (*“you can adapt to it easily”*). One user found gaze-shifting counter-intuitive (*“it is disturbing as it is not what you would naturally do”*), seven users did not notice any effort (*“I wasn’t paying a lot of attention on switching, so*

*it was quite fluid*”), and four users found it easier than direct touch (*“It feels like an easy way instead of dragging all across the screen”*). Users suggested use in design tools, e.g. to bring effects or files down to the designs they are working on, or add them to layers.

#### Pen and touch annotation

Similarly, users initially reached out but got quickly used to indirect touch manipulation. 4 users experienced erroneous image selection at overlapping images, and users suggested clear highlighting which image will be gaze-selected. Otherwise, users were positive (*“It’s just easy to use, it’s a kind of way that simplifies such work”*), and would find it useful in applications for browsing, categorising, and archiving of photos, potentially in collaboration with a partner.

Across tasks, users got quickly used to gaze-shifting after a short training, and experienced two types of errors. First, when users attempted to lean forward to use the pen for inking, eye tracking accuracy decreased as the eye tracker positioned at the lower bezel was partly occluded by the arm, and users moved away from the original calibration position. Multiple eye trackers (e.g., another one positioned at the top bezel) covering a larger tracking range can alleviate this problem. Second, ambiguity of the direct and indirect zone can confuse users. We expect improvements by providing users with additional feedback to make users more aware which interaction zone they are currently working with, e.g. subtle icon highlighting when users enter the indirect zone.

## 4.9 Discussion

Gaze-shifting enables users to shift between direct and indirect input by a glance, as demonstrated by the direct-indirect interaction techniques that we developed for validation of the concept. On a pen and touch display, gaze-shifting enhances mode switching, large shape drawings, precise point editing, or pinch-to-zoom operations. These techniques highlight the flexibility of gaze-shifting’s integration across task, technique, modality, and application level. In this work, we have explored gaze-shifting for pen and touch interaction, but the technique is generic and readily extensible to other direct input devices. For example, it could be used in conjunction with tangibles to switch between direct input to manipulate the tangible’s state, and indirect input where the tangible devices become a proxy for remote input.

Exposure of our techniques to users showed that gaze-shifting is easy to understand, and users were able to complete direct-indirect interaction tasks after only a few trials. User feedback also indicated limitations which in part related to general eye tracking issues, but also showed challenges arising from ambiguity of the direct and indirect zone. Users however quickly adapted to these problems, and with this successfully used gaze-shifting and found it easy to use.

In particular, specific conceptual and technical limitations need to be considered in the design of gaze-shifting. Gaze-shifting is highly dynamic, which makes hybrid interactions possible, but also introduces potential false-positive activations. Context-awareness can help, through support of users in selecting the right mode based on their input techniques and task. For instance with static menus, gaze-shifting is only enabled when users look at

the fixed menu position on the UI, and otherwise (when not looking at the menu) gaze-shifting remains inactive (Fig. 4.7). For dynamic objects such as graphics, the system constantly determines direct/indirect input based on proximity of gaze and input position (Fig. 4.8) — which in principle can be used as an overarching direct/indirect mechanism over default UIs. In other cases however, interactions might not be necessarily coupled to the user’s gaze (e.g. file dragging), and then gaze-shifting should be inactive. A technical factor to consider is eye tracking imprecision that affects target acquisition, particularly for overlapping targets. Hover-enabled pens counter this issue: gaze selects the area, pen hover selects the detail position, and pen tap manipulates the target (Fig. 4.8). Hoverless devices such as touch can integrate additional precision techniques [SD12b, ZRZ08].

We have focused on enabling direct input devices to shift to indirect input, however, we also consider generalising gaze-shifting to indirect input devices. While indirect input devices are normally used for interaction with remote displays, the act of looking at the input device can enable additional device-specific interactions. For example, a laptop’s touchpad can become a touchscreen when looked, providing an additional menu to quickly switch between applications, conceptually similar to our colour select & set example (Figure 4.10). For a mouse, when users look at the device, displays like those in the LensMouse [YMM<sup>+</sup>10] can be toggled interactive, and enable mouse-specific configurations. For a virtual or physical keyboards, users can shift between standard key input when looking at the normal screen, and when looking at the keyboard, they interact with on-board displays (e.g., a display like in Touch-Display-Keyboards [BGV10]). Therefore in principle, gaze-shifting can apply beyond direct input devices, and raises potential interaction possibilities on both ends of the direct/indirect input device spectrum.

## 4.10 Conclusion

We presented gaze-shifting as a generic and implicit method to enable combined direct and indirect interaction. With a single manual input (e.g. a tap or pen click), users can perform direct manipulation, indirect control, and transition in between them. This leads to new interactive experiences where direct and indirect inputs are not separate, but continuous within the interaction. Our line tracing example has the closest fit between both inputs: users can directly start a line, extend the line to any remote gaze point, return to continue with direct manipulation, and so on. Yet the closer we bring direct and indirect together, the less is input distinguishable by this categorisation. Eventually, we could arrive at a point where direct or indirect becomes irrelevant, as any device or object is usable for both inputs, and it all coalesces into one interaction driven by our gaze.

Overall, we have unfolded a broad design space for new interaction techniques by applying the gaze-shifting idea to two modalities, pen and touch. In addition to techniques for each individual modality, we were able to look at combined pen and touch interaction techniques that are based on bimanual input. This provides a new class of interactions: those where users can leverage direct and indirect inputs together, using both hands. However, in this work our system did not support simultaneous pen and touch yet — thus no ability to interact with direct and indirect input at the same time. Therefore, in the next chapter, we investigate this new class of simultaneous direct and indirect interaction.

# 5

## Direct-indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction

This chapter explores the use of simultaneous direct and indirect interaction that users can utilise through two-handed control, i.e. one hand issues direct, and the other indirect input. This is investigated by bimanual pen and touch, using the pen in one hand for direct, and touch of the other hand for indirect interactions.

Pen and touch manipulation, where both modalities are direct, is increasingly supported on tablet and large display computers. This efficiently enables asymmetric bimanual input with the pen in the dominant hand and multi-touch of the non-dominant hand [BFW<sup>+</sup>08, Gui87, HYP<sup>+</sup>10]. For instance, *pan and zoom* gestures for UI navigation together with a pen for precise *inking* is useful for sketching [HYP<sup>+</sup>10, WSR<sup>+</sup>06], text editing [HBPB12, YCGS14], or vector graphics work [BFW<sup>+</sup>08, HWP<sup>+</sup>14, Yee04].

In this context, we investigate using the dominant hand for standard *direct pen* input, while the non-dominant hand performs *indirect-touch* input. In essence, this describes a new class of bimanual interaction where direct and indirect inputs are mapped to both hands. With the indirection in one hand, users gain remote, occlusion-free, and precise input (c.f. Table 3.1). As direct and indirect inputs do not physically overlap, users can employ two-handed input in the same space without interference.

The indirection is enabled by the user's gaze input. This is essentially the gaze-touch technique, as introduced two chapters earlier. The operation of the technique involves users looking at a target, and then performing indirect-touch gestures from remote. All this is possible while the other hand, that is holding a pen, can independently perform other tasks with digital ink such as drawing or writing.

However in context of pen and touch, it is also possible to enable indirect touch using an alternative, without gaze. Touch input of the user can be simply redirected toward the pen's position, decoupling touch from its original position and coupling them to the dominant hand's pen input. This can be useful on a pen and touch display, where the user's work is often centered around the pen that is held in the dominant hand. To be consistent with the naming, we call this technique *pen-touch*: a technique where users

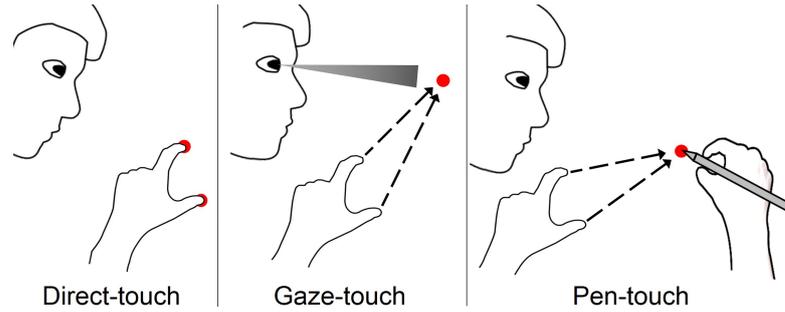


Figure 5.1: We investigate the indirect gaze-touch and pen-touch techniques in comparison to direct-touch for use in bimanual interfaces.

point the pen at the target, and perform indirect-touch gestures from any close or remote position.

Figure 5.1 illustrates the two techniques in comparison to the standard direct pen and touch technique. Direct- and gaze-touch are one-handed — meaning that with the other hand, the user can freely use touch or pen for other, concurrent interactions. Pen-touch is different as it needs the pen as the touch redirection target. However, the user can still use the pen simultaneously, and then both pen and touch would manipulate the same target.

To understand the pros and cons of these interactions in detail, we first provide an analytical discussion of the interaction properties of the three techniques. We then conducted a user study comparing the three techniques. In this experiment, users navigate the canvas with pan and zoom by touch, while the pen is used to select and draw objects. Two variations of this task are tested: one where users alternate between pen and touch, and the other one where they use both in parallel. In the context of our study, we define *parallel* interaction as when both pen and touch need to be in contact with the display to accomplish the task.

As a preview to the results of the study, it was found that both indirect touch techniques have (1) comparable time performance to direct manipulation while (2) indirect-touch improves in accuracy of zooming. Further post-hoc analysis of gestural and spatial characteristics showed that (3) users perform zoom gestures faster and more frequent with indirect-touch, (4) indirect-touch can lead to more (pen-touch) or less (gaze-touch) bimanual parallelism than direct touch, and (5) users keep pen and touch modalities spatially further apart with indirect-touch.

Our contributions are (1) the concept and techniques that use the pen with the dominant hand and indirect-touch with the non-dominant hand, (2) a bimanual pen and touch experiment comparing two indirect-touch techniques to direct touch, and (3) novel findings about pinch-to-zoom accuracy, visual behaviour, bimanual parallelism, and direct vs indirect input.

## 5.1 Interaction Techniques

We first describe the investigated interaction techniques and then analyse their interaction properties:

**DT: Direct-touch:** This technique is standard on pen and touch interfaces, where users touch the position they want to manipulate, and the action begins immediately at touch down. Current pen and touch displays employ this technique for multi-touch input by the user, that is combined with the inking mode of the pen.

**GT: Gaze-touch (Figure 5.2):** In graphical context such as pen and touch displays, the user’s visual attention is often correlated with the actual area that users interact in. Researchers have thus suggested to redirect the effect of touch gestures toward the user’s gaze position on the screen [SD12b, TABG15]. This provides benefits such as whole-surface reachability, occlusion-free, and precise input through indirect-touch (more details in [PACG14]). Essentially, gaze-touch consists of a two step interaction: users look at a target to select it, and then touch down and perform a gesture to indirectly manipulate it.

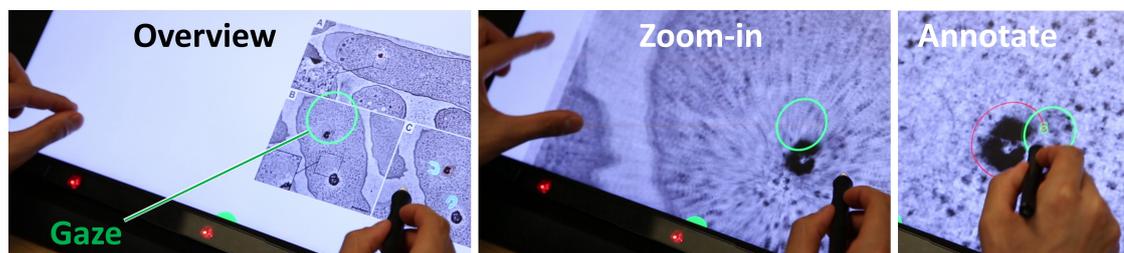


Figure 5.2: Gaze-touch: from an overview medical image, users quickly zoom into their gaze position to then use the pen for annotations. The user’s gaze position is indicated with the green circle.

**PT: Pen-touch (Figure 5.3):** Within a pen and touch interaction context, the user’s focus of interaction is often already located around the pen that is held in the dominant hand. For example, a user draws in a graphical model with the pen, and then drags the same model with touch. Pen-touch is based on this premise as a new technique where the effect of touch gestures is redirected to the pen’s position. This allows users to perform touch gestures on a target that is already occupied by the pen, and focuses the user’s interaction around the pen device. The touch redirection works during pen down and hover events.



Figure 5.3: Pen-touch: While users are drawing a line, users can zoom into the pen’s position and then precisely finish the line drawing.

### 5.1.1 Analysis of Interaction Properties

We now analyse the three techniques with a focus on bimanual interaction properties. We extend our prior comparison of direct-touch vs. gaze-touch (c.f. Table 3.1) with a focus on bimanual pen and touch interaction. The interaction properties are summarised in

	Direct-touch	Gaze-touch	Pen-touch
Gesture target	Touch	Gaze	Pen
Hands needed	1 hand	1 hand	2 hands
No occlusion	-	✓	✓
No interference	-	✓	✓
Same-target Simultaneity	-	✓	✓
Separate-target Simultaneity	✓	✓	-
Dynamic targeting	-	✓	✓

Table 5.1: Summary of differences between techniques

table 5.1. Notably, all techniques still support concurrent pan and zoom with two-touch gestural input.

**Division of Labour (Table 5.2):** In general, all techniques follow Hinckley et al’s division of labour between modalities *pen writes, touch manipulates* [HYP+10]. A further division of labour occurs for the *touch manipulates* part, that has distinct implications on the interaction with the technique.

*Direct-touch* frees the user’s gaze and pen input during touch gestural interactions, but requires moving their hands to where they want to perform the gesture. For instance, it can be appropriate when users want to clearly indicate where they touch to collaborators. *Gaze-touch* does not require relocating either pen or touch to issue gestures, but requires the user to explicitly direct their gaze to a target. Thus it is appropriate for interactions where the hand needs to keep out of the user’s view. *Pen-touch* does not use gaze explicitly nor are users required to move the touch-hand to the gesture target, but requires the user to move the pen to the gesture target. This essentially segments touch gesture selection and manipulation based on Guiard’s proposition that the dominant hand performs precise, and the non-dominant hand performs coarse tasks [Gui87]. For example, the technique is appropriate when precise target selection with the pen tip is required, such as CAD modelling.

	Touch manipulates (non-dominant hand)		Pen inks (dominant hand)
	Select	Manipulate	
<b>Direct-touch</b>	Touch	Touch	Pen
<b>Gaze-touch</b>	Gaze	Touch	Pen
<b>Pen-touch</b>	Pen	Touch	Pen

Table 5.2: The techniques share the overall division of labour, and vary for the ‘select’ sub-task during touch gestures. The ‘manipualate’ part is touch only across all techniques to support all standard touch gestures.

**Occlusion:** Direct-touch naturally induces occlusion when the user’s hand/arm is on the screen [VB10], which increases with two-handed input. Both indirect techniques (gaze-touch / pen-touch) are occlusion-free as the hand is decoupled from the manipulation, only the hand that holds the pen can still cast occlusion.

**Interference:** Direct-touch is prone to interference: when users want to interact with one target with both hands, one hand spatially interferes with the other hand, which requires alternating use of pen and touch modalities. Both indirect techniques enable same-target manipulation with both modalities.

**Same-target Simultaneity (Figure 5.4 top):** Same target interaction occurs when

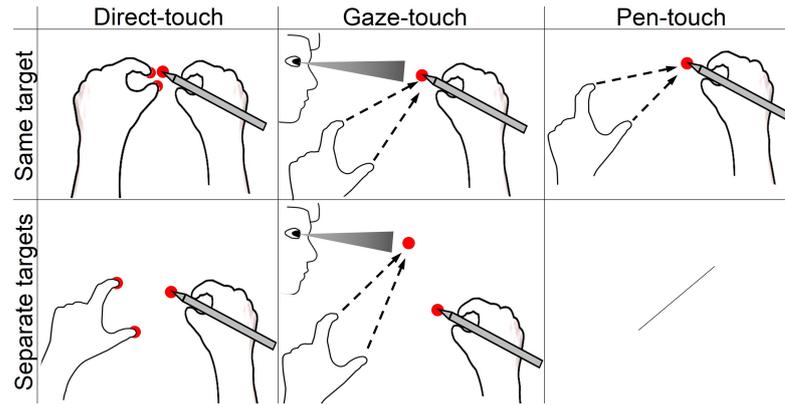


Figure 5.4: Simultaneity: each techniques has different feasibility for simultaneous interaction on the same or on separate targets.

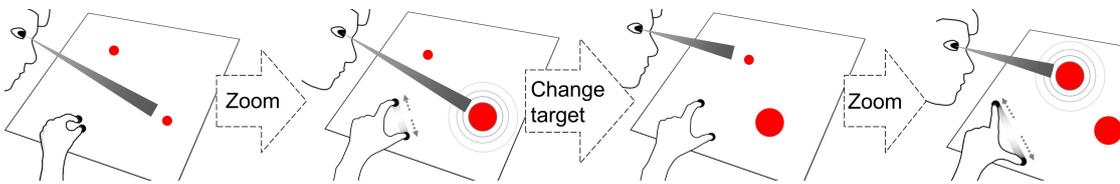


Figure 5.5: Dynamic Targeting: With indirect touch techniques such as gaze-touch, users can change the target during the gesture without lifting fingers.

users perform two modes simultaneously on one target such as drawing a curve while adjusting its roundness. This works with gaze-touch and pen-touch as users can directly ink with the pen, and at the same time indirectly manipulate the same target. At touch down users look at it (gaze-touch), while for pen-touch the target is already at the pen’s position. With direct-touch, users cannot exactly manipulate the same target because of interfering hands, except if the target area is large enough to be manipulated from multiple points.

**Separate-targets Simultaneity (Figure 5.4 bottom):** Users interact with two separate targets simultaneously when for instance dragging an image while opening a folder with the other hand. This works for direct-touch and gaze-touch: users can select a point with the dominant hand (pen), and simultaneously select a different point by touching on it (direct-touch) or looking at a different target (gaze-touch). This does not work with pen-touch, as any touch is redirected to the pen’s position, and the system would have to choose between using either pen only or pen-touch input.

**Dynamic targeting (Figure 5.5):** the established direct-touch paradigm resembles real-world physics, and when users ‘grab’ an object, the touch positions are glued to the object’s local position that users initially touched. To interact with another target, users lift their fingers and move them to the new target.

This is different from the indirect techniques (gaze-touch, pen-touch) where users can *dynamically change the target* during a touch gesture. Without lifting their fingers, users can move the pen or their gaze to a different target. For instance, when performing pinch-to-zoom, users can adjust their zooming pivot while they zoom-in to achieve more precise navigation. Thus Dynamic Targeting can increase the accuracy of touch manipulation. More accuracy can in turn lead to a decrease of the amount of panning and clutching operations that users perform during navigation [ACVL14].

## 5.2 Experiment

To understand how the techniques compare in practice, we evaluate the performance of the three techniques in two tasks: one where pen and touch are used in alternation, and the other one where the modalities are used in parallel.

### 5.2.1 Research Questions

*Task Completion Time:* How does each technique affect the user’s temporal performance in a sequential and simultaneous pen and touch task? The techniques have distinct properties for use in alternating and simultaneous use of pen and touch.

*Accuracy:* How does the Dynamic Targeting feature of indirect techniques come into play? For this we measure the accuracy of zoom gestures (the disparity between positions users zoom in vs. the actual target where users need to zoom).

*Gesture Characteristics:* Does the indirection through pen-touch and gaze-touch affect the users gestures? Across the techniques, users perform the same type of gestures, only the target of the gesture varies with technique.

*Parallelism between Pen and Touch:* Does a technique involve more parallelism between the pen and touch modalities than others? Parallelism can be, but is not necessarily correlated with the efficiency of bimanual interaction [BM86, LZB98].

*Spatial Distribution of Input Modalities:* How do users couple the pen and touch modalities? Users touch at the manipulation point with direct-touch, but it is unclear whether users return to these patterns with the indirect techniques.

*User Feedback:* Do users like the familiar direct manipulation paradigm or come to prefer a new technique?

### 5.2.2 Tasks

We chose touch based pan and zoom with pen based drawing as the underlying task environment, a combination where users benefit from bimanual pen and touch inputs [BFW<sup>+</sup>08, Gui87, HYP<sup>+</sup>10]. We use two tasks, one more suitable for sequential interaction and one more suitable for simultaneous interaction with the two modalities.

**Sequence task (Figure 5.6a):** In this task, users navigate to, and then select three targets. Users first zoom out to get an overview, and then zoom into the target area. When users find the actual target dots, they draw a circle around them to finish the task.

**Parallel task (Figure 5.6b):** In this task, users draw a line while navigating the canvas. Users first select the start point of the line, and then navigate toward the end point. The end point is not visible at the start, and therefore users zoom out to get an overview, and then zoom into the target area. During the navigation, the pen remains pressed on the screen. When the target is visible, users move the pen to the target, and lift up to finish.

Both tasks adapt Guiard et al’s multiscale pointing task [GBL04] for the part where users perform pan and zoom, similar to Nancel et al.’s investigation of pan and zoom

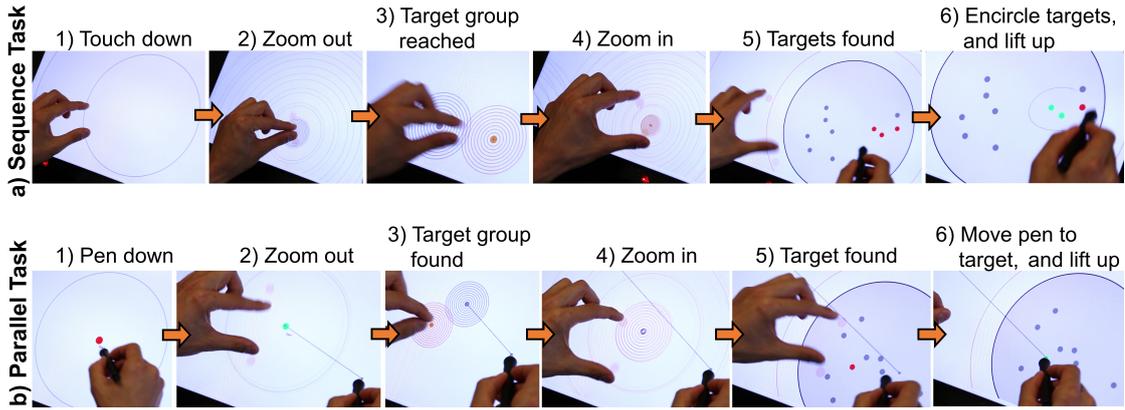


Figure 5.6: Substeps of the two study tasks on the example of the direct-touch technique

techniques [NWP<sup>+</sup>11]. Participants navigate through an abstract interface with two groups of concentric circles (start and target group). The gray start group is where users begin the task and zoom out (Fig. 5.6, first two columns). When zoomed out enough, the orange target group becomes visible (5.6a-3). Users then zoom into the target group. The center of the target group is specifically offset from the center of the start group (Table 5.3). The angle between both circle groups is randomized for each trial. The last circle of the orange target group contains 10 gray dots that are randomly placed within it (5.6a-5). The zoom-in sub-part is finished when the initial zooming level is reached again, and the last circle of the orange target group is within the display’s region (the last circle width=450px, all dots’ width=50px).

The end of the pen task then becomes visible: the relevant dots are highlighted red (three dots for the sequence task, one dot for the parallel task, see Fig. 5.6a-5 and 5.6b-5). The target dots are randomly selected. For the sequence task, the first dot is randomly selected, and then the two closest neighbor dots are additionally selected as target dots.

The sequence task finishes when the user has encircled all three dots (Fig. 5.6a-6), and if not, users can draw additional lines (but need not encircle all three again, only the remaining dots). Each dot is highlighted green when inside of a user’s drawn lines.

For the parallel task, the task begins with users placing the pen at a centered dot before performing the pan & zoom navigation (Fig. 5.6b-2). The task finishes when the pen moved within the ending dot’s area (where it gets highlighted green, Fig. 5.6b-6)), and lifted up. If users lift the pen without being in the dot’s area, the task is voided and will be repeated.

### 5.2.3 Design and Factors

Our experiment used a within subjects design. The task order was counterbalanced for each user, and the technique order was counterbalanced for each user using a balanced latin square. For both tasks, we used the same three distances (Table 5.3). The distance is the length that users navigate from start to end point of the pan and zoom task. The minimum distance was chosen as the minimum index of difficulty where pan and zoom becomes beneficial (ID=8, [GBL04]). The remaining distances are steps of 3 indices of difficulty (using formula  $\log_2(D/W + 1)$  with fixed  $W=50px$ ). Each distance was repeated 15 times. Within each task × technique block, users performed 45 trials (= 15

$\times 3$  distances). The order of the distances was randomised within the block. Overall, this resulted in  $2 \text{ tasks} \times 3 \text{ techniques} \times 3 \text{ distances} \times 15 \text{ repetitions} = 270$  trials per participant.

	Small	Medium	Large
<b>ID</b>	8	11	14
<b>Centimeter</b>	315	2532	20265
<b>Pixel</b>	12751	102351	819151

Table 5.3: Study distance factors (for both tasks) in three metrics.

#### 5.2.4 Participants

18 paid participants took part in the study. On average they were 26.7 years old (SD=6.4, 6 female), and students or employees of the local university with mixed background. Only one user was left-handed, and we mirrored positional data post-hoc for a right-handers dataset. 5 users wore glasses, and 4 contact lenses. On a 1 (no experience) to 5 (expert) scale, users rated themselves as experienced with multi-touch (3.9, SD=1.1), and less experienced with eye gaze interaction (2.6, SD=1.38) and stylus interaction (2.6, SD=1.42).

#### 5.2.5 Procedure

At first, users filled out a demographic questionnaire and conducted the gaze calibration. Then users performed the six task $\times$ technique blocks. Before each block, users performed up to five trials to get used to the technique and were instructed to be as fast as possible. After each block, users filled out a questionnaire with 6 Likert scale questions: ‘*The task with this technique was [easy to use | fast | precise | easy to learn | eye fatiguing | physically fatiguing (hand, arm, shoulder, or neck)]*’. Lastly, users filled out a ranking questionnaire and discussed why they preferred which technique. Overall, the study lasted 60-90 minutes. The questionnaires are provided in appendix .2.

#### 5.2.6 Apparatus

We use a Wacom Cintiq 22HD pen and touch display with an Tobii EyeX eye tracker (30hz, Figure 5.7). The tracker is placed at the bottom border of the display. The display is oriented at a  $45^\circ$  angle to enable comfortable reach, allows 10-finger multi-touch at 120hz, and has a Wacom Grip Pen. The user sits in front of the system with approximately 60cm between the user’s eyes and the eye tracker. Users were calibrated to the tracker at the beginning of the study using the standard EyeX application. We also conducted a 16-point accuracy test after each study session. The average accuracy was  $1.51^\circ$  (SD=.58 $^\circ$ ). The software is implemented in Java and runs on a 64-bit, 16GBRAM, quadcore i7@2.4GHz laptop. Simultaneous pen (WACOM Pen) and touch is detected with the Wacom SDK. Accidental touches that can occur from the pen holding hand are ignored by removing all touches that occur to the right of the pen tip (for right-handers).

The user’s gaze was smoothed during the gaze-touch technique. As smoothing inherently introduces interaction delay, we use a more dynamic method: when users quickly moved



Figure 5.7: System setup: pen and touch display (a), user’s multi-touch input (b), pen input (c), eye tracker (d).

their gaze (above 1050 px/s or  $24^\circ$ /s of visual angle), raw gaze data was used. Otherwise gaze data was averaged for 500 ms (includes 15 gaze samples), which helps to stabilise the jittery gaze cursor during fixations.

When users occluded the eye tracker (e.g. with a hand) or moved their head out of range, an error message was displayed to indicate the user to correct their position. This was explained and tried before study to avoid confusion. We considered gaze data as outliers when the eye tracker reported error (usually when users are out of range or blink).

### 5.2.7 Statistical Analysis

For the quantitative data, a two-factor repeated-measures ANOVA (Greenhouse-Geisser corrected if sphericity violated) was employed, followed by post-hoc pairwise comparisons with Bonferroni corrections. Qualitative data was analysed with a Friedman test and post-hoc Wilcoxon Signed Rank tests with Bonferroni corrections.

## 5.3 Results

We report the results based on the initial research questions. Mean values are reported within each bar in the bar diagrams.

### 5.3.1 Task Completion Time

For task completion time measures, in the sequence task timing starts when users first touch down and ends when users lift the pen after encircling three targets. For the parallel task, timing starts when users pressed the pen at the line start point, and ends when users lift the pen at the line end point.

The results are presented in Figure 5.8a. They indicate that the users performed comparatively across the techniques. In the sequence task, technique had a significant effect on task completion time ( $F_{2,34}=5.5$ ,  $p=.008$ ). Users performed significantly faster with direct-touch than with pen-touch ( $p=.015$ ), no significant differences were found among the remaining comparisons. In the parallel task, technique did not significantly affect task completion time ( $F_{2,34}=1$ ,  $p=.36$ ). Factor distance significantly affected performance in the sequence ( $F_{1.3,22.9}=173.9$ ,  $p<.001$ ) and parallel task ( $F_{1.4,24.8}=110.2$ ,  $p<.001$ , all pairs  $p<.001$ ), though no significant interaction effects between technique and distance were found; neither were any learning effects found across blocks.

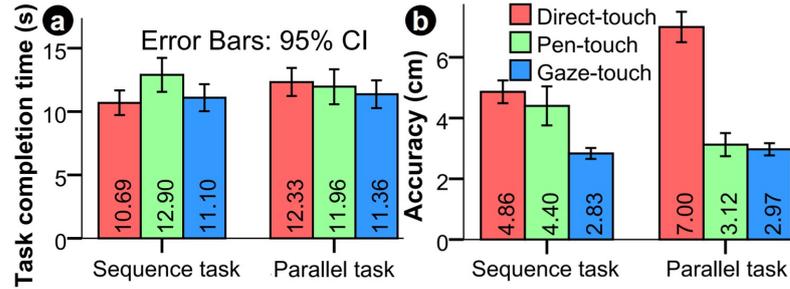


Figure 5.8: Users performed comparatively in time, and were more accurate with the indirect techniques.

### 5.3.2 Accuracy

Zoom-accuracy is how accurate users zoomed during pinch-to-zoom gestures, i.e. the disparity in centimeter between the position users zoom at and the actual target users should zoom at. We only consider zoom-in gestures, as for zoom-out the target did not matter in our task. We measure zoom-accuracy in each frame during zoom-in gestures. These measures were averaged in each trial; providing the same data base as with task completion time.

In both tasks, users were most accurate with gaze-touch, then pen-touch, and lastly direct-touch (Figure 5.8b). This can be accounted to the Dynamic Targeting feature included in both indirect techniques. We found a significant effect of *technique* on accuracy for the sequence ( $F_{2,34}=12.6$ ,  $p<.001$ ) and parallel task ( $F_{2,34}=65.4$ ,  $p<.001$ ). In the sequence task, users were more accurate with gaze-touch than direct-touch ( $p<.001$ ). Also users were more accurate with gaze-touch than pen-touch ( $p=.0021$ ). No significant difference was found between direct-touch and pen-touch ( $p=.813$ ). In the parallel task, users were more accurate with both gaze-touch and pen-touch than direct-touch (both pairs  $p<.001$ ), but no difference was found between pen-touch and gaze-touch ( $p=1.967$ ). Further, no learning effects were found across blocks.

We plotted zooming-accuracy during gestures to see how the Dynamic Targeting feature of indirect techniques behaves over time. For this, we collected the average zooming-accuracy for each frame (120hz), for each zoom-in gesture that users have performed. This results in a list of gestures where each gesture consists of one accuracy value for each frame. We calculated the time for each frame and plotted as presented in Figure 5.8. Each gesture begins at Time=0, but the ending time of a gesture is individual for each gesture (see exact durations in Fig. 5.11a), and we plotted for 1 second. We show the

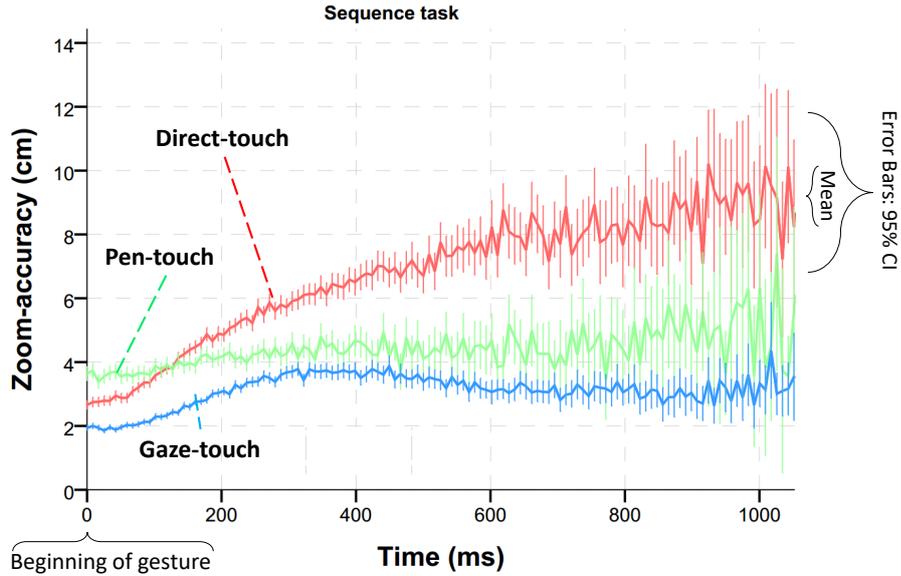


Figure 5.9: Zoom-accuracy over time during zoom-in gestures for the sequence task, revealing that indirect techniques (pen-touch, gaze-touch) have constant accuracy during zooming, but direct-touch’s accuracy decreases over time.

Error Bars (95%) for each to indicate when the data is becoming too ‘spread’.

Figure 5.9 and 5.10 plot the accuracy over time. Overall, The figures show that the indirect techniques have stable accuracy over time. In contrast, with direct-touch the accuracy decreases with increasing time. We think this is because first, when users want to zoom exactly on a target, the target will continuously offset away from the touch positions and become more inaccurate over time. Second, there are cases where users deliberately touch offset from the target, so that the target is still visible, which adds a constant offset to the zoom operation.

### 5.3.3 Gesture Characteristics

We now present results on the different gesture characteristics across the used techniques. For this, we conducted a post-hoc analysis on zoom-in gestures. Priorly, we classify zoom-out, zoom-in, and drag gestures based on Avery et al’s parameters [ACVL14]. We use a minimum factor of 5px movement to classify motion as a gesture. Zoom and drag are distinguished by single and two touch. Zoom is further distinguished to zoom-in/-out by checking initial and ending scale of the gesture.

We measured average gesture count and time for each condition. We conducted an ANOVA to see how they are affected by technique, leading to the following findings:

*Users performed shorter zoom-in gestures with indirect techniques in the parallel task* (Figure 5.11a): In the parallel task, a significant effect of zoom duration on technique ( $F_{1.5,25.2}=18.1, p<.001$ ) showed that users performed shorter zoom-in gestures with gaze-touch ( $p<.001$ ) and pen-touch ( $p=.0095$ ) than with direct-touch. No significant differences among conditions were found for the sequence task.

*Users performed less zoom-in gestures with pen-touch than direct-touch in the sequence task* (Figure 5.11b): a significant effect of zoom duration on technique was found for the

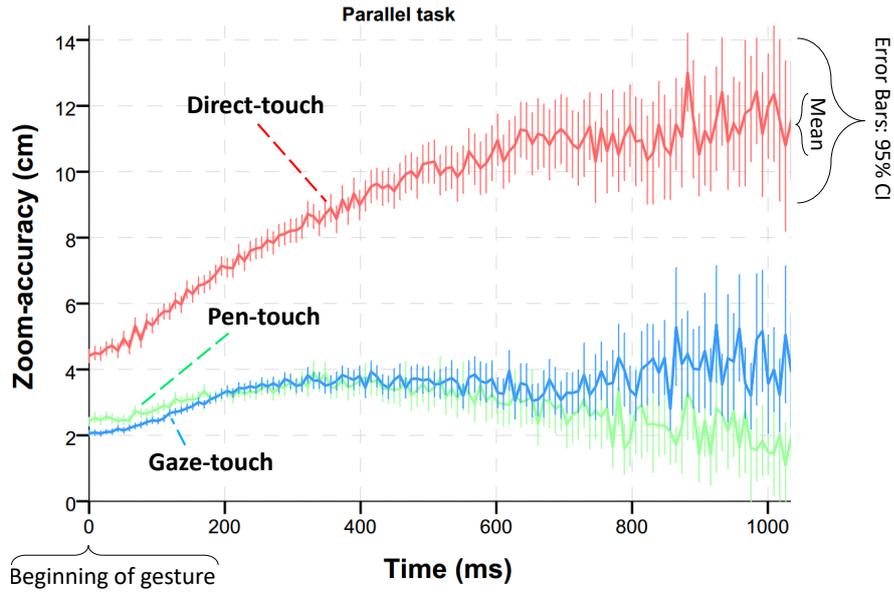


Figure 5.10: Zoom-accuracy over time during zoom-in gestures for the parallel task, revealing that indirect techniques (pen-touch, gaze-touch) have constant accuracy during zooming, but direct-touch’s accuracy decreases over time.

sequence ( $F_{2,34}=7.3, p=.002$ ) and the parallel ( $F_{2,34}=4.5, p=.018$ ). In the sequence task, users performed less zooms with pen-touch than with direct-touch ( $p=.006$ ), while other pairs did not yield significant differences.

These findings correlate with our observations, we often saw users performing indirect-touch gestures faster and therefore more frequent than with direct-touch. We think this can be accounted to the indirection: users do not physically see their fingers touching the surface, and therefore just ‘quickly’ perform a gesture with as least effort as possible. In contrast, with direct-touch users see how their fingers physically touch the screen, making users utilise the pinch gesture with more extensive scales and for longer time.

### 5.3.4 Parallelism between Pen and Touch

Parallelism in bimanual input can improve efficiency (as in our parallel task) but also introduce overhead (e.g. when requiring parallelism in a sequence task) [BM86, LZB98].

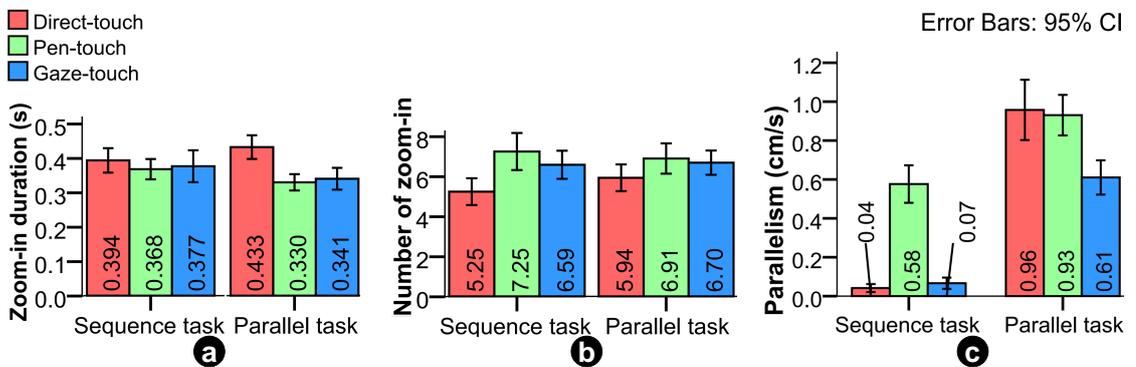


Figure 5.11: (a, b) Zoom-in gesture duration and count, indicating different zoom behaviour between direct-touch and the indirect techniques. (c) How parallel users performed with the pen and touch modalities.

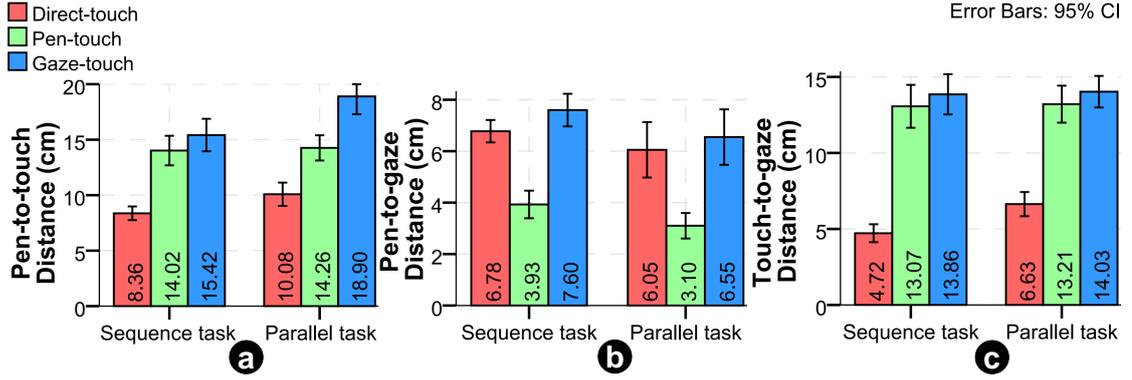


Figure 5.12: How distant users kept each modality-pair.

We now investigate the techniques with regards to their amount of exhibited parallelism between pen and touch modalities. Simultaneous input of both modalities is measured as parallel (i.e. when both pen and touch are in ‘drag’ mode). We did not consider metrics that include efficiency of parallelism [BH00], as our study is based on asymmetric bimanual interaction. It is likely that any movement, whether parallel or not, is intended to successfully finish the task. The particular metric we measure when both pen and touch perform a specific number of cm movement, is the smaller value of both. We computed this value for each condition, and report it as the number of centimeters users would perform in parallel per second.

Figure 5.11c presents the parallelism results. A significant effect on technique ( $F_{1,18}=42.3$ ,  $P<.001$ ) showed an expected result for the sequence task: users performed significantly more parallel movement with pen-touch than with other techniques, as it required bimanual input for zooming (both  $p<.001$ ). No difference was found between direct-touch and gaze-touch ( $p=1.49$ ). In the parallel task, all techniques showed high parallelism as expected by the design of the task. However, we found a significant effect of technique ( $F_{2,34}=5.2$ ,  $p=.011$ ), showing that gaze-touch had less parallelism than pen-touch ( $p=.033$ ). While it is clear why pen-touch has higher parallelism by design, it is unclear why direct-touch has higher parallelism than gaze-touch in the parallel task.

We plotted parallelism over the duration of each trial to gain further insight into the differences between the techniques. As each trial has a different duration, we normalised the task time between 0 and 100%. For this, we divided the task into 300 time segments, and then computed the average parallelism for each time segment, and then distributed the average 300 segments into 100%.

Figure 5.14 shows the parallelism over time. In the sequence task, only pen-touch has high parallelism, as expected because of the bimanual technique design. In the parallel task, all techniques have parallelism corresponding to the task, with three periods: a peak on parallelism at the start when users initiate pen and touch (Fig. 5.14a), a period of low parallelism when users zoom-out as no pen movement is needed (b), and then a period of high parallelism when zooming in (c). The difference of parallelism between direct-touch and gaze-touch can be further reduced to the zoom-in period. In light of this, we hypothesise the difference exists because of the interference problem: users often wanted to zoom where the pen was located. This required direct-touch users to move the pen away in order to make space for touch input, in turn increasing parallelism.

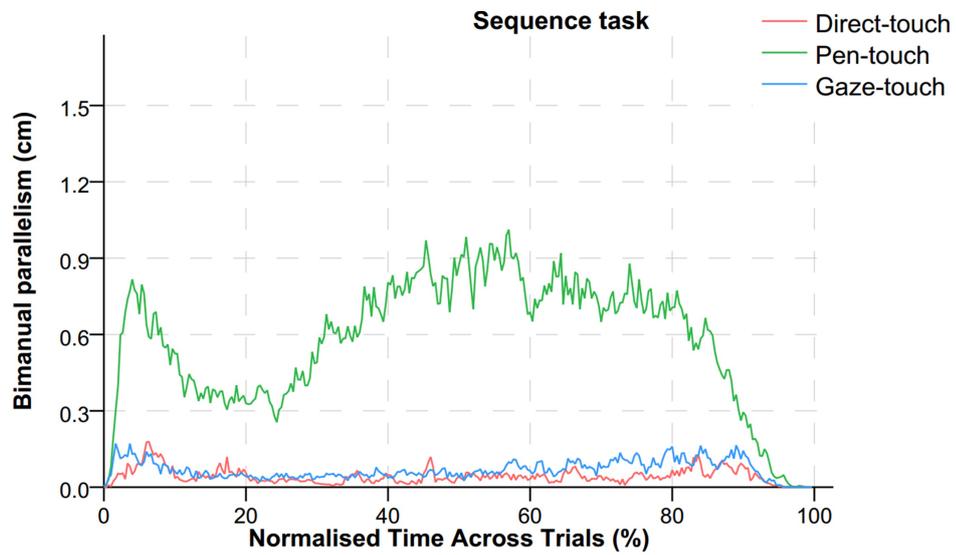


Figure 5.13: Sequence task parallelism: Parallelism of the pen and touch modalities over the duration of each trial. Parallelism denotes the minimum movement in cm that both modalities share. Trial durations are normalised to 0-100% of time.

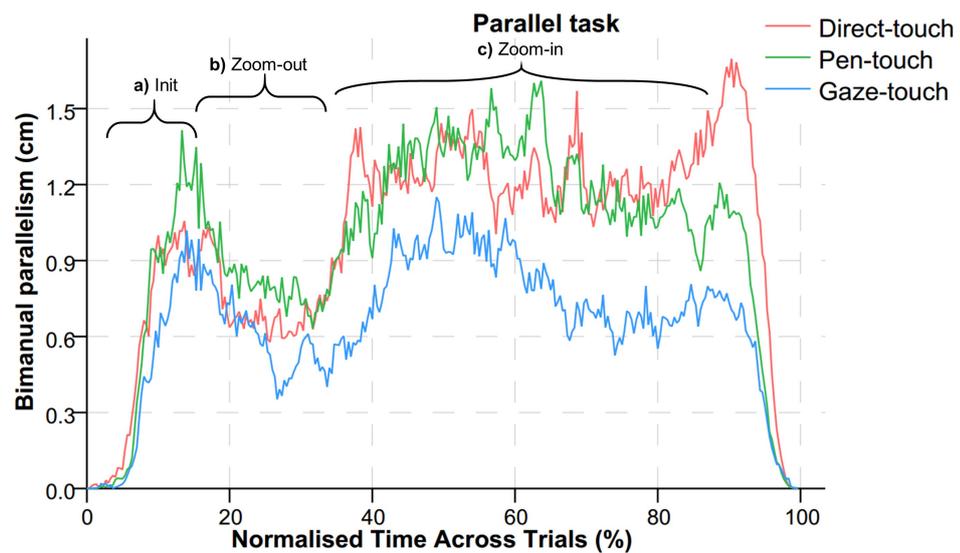


Figure 5.14: Parallel task parallelism.

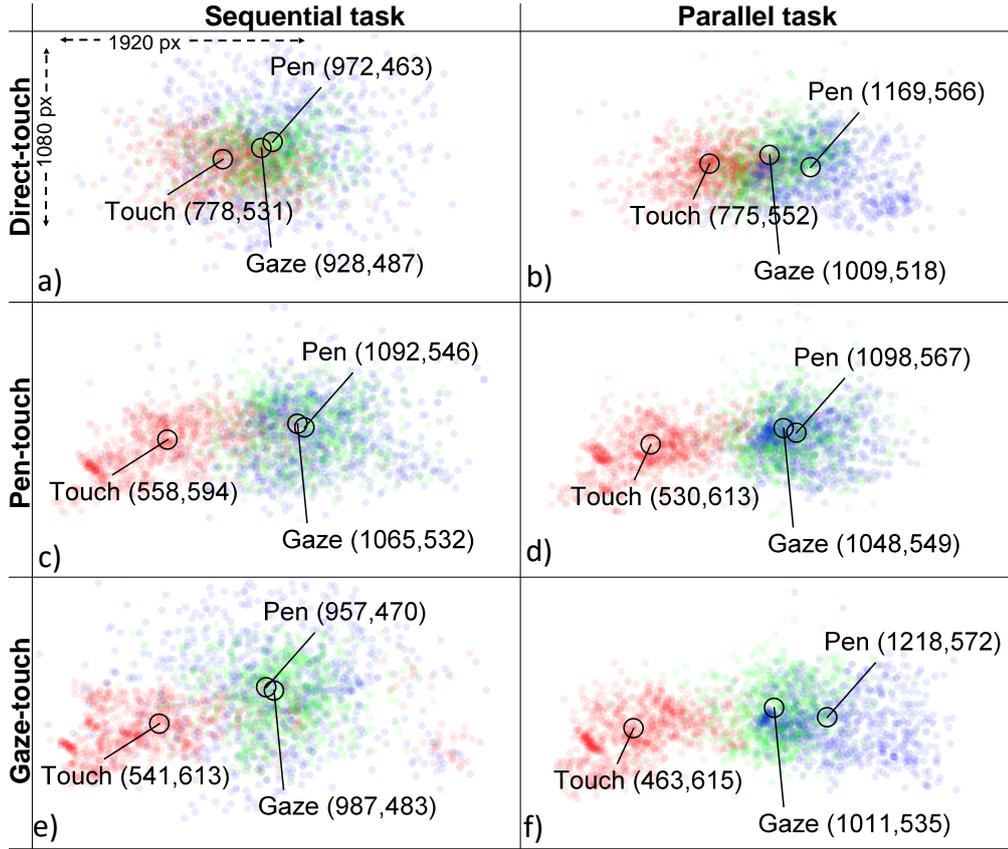


Figure 5.15: Illustrative heatmaps of the users' average touch (red), pen (blue), and gaze (green) positions across the screen.

### 5.3.5 Spatial Distribution of Input Modalities

With direct-touch input it is likely that users look where they touch — but do users return to these patterns with the indirect techniques? To get insights into this we look at how users spatially aligned the pen, touch, and gaze modalities. For this, we analyse the relative distances that users employed during the use of the input modalities. For each trial, we measured the average position of gaze, pen (at both hover/pressed), and touch (zoom-in only, using center of touches). Then we computed the relative distances between the modality positions, for each test condition. Our statistical analysis yielded the following findings (2D heatmap visualisations in Fig. 5.15):

*Users kept pen and touch further apart with the indirect techniques* (Figure 5.12a): Technique significantly affected pen-to-touch distance in the sequence ( $F_{2,34}=18.9$ ,  $p<.001$ ) and the parallel task ( $F_{2,34}=23.8$ ,  $p<.001$ ). In both tasks, users showed a significantly larger pen-to-touch distance with both gaze-touch and pen-touch compared to direct-touch (pair<sub>pen-touch/direct-touch</sub>  $p=.018$ , else pairs  $p<.001$ ). In addition, in the sequence task users kept pen and touch further apart with gaze-touch than pen-touch ( $p=.0181$ ). Overall, this indicates that users utilise the screen space for a relaxed touch placement with the indirect techniques.

*Users look closer to the pen with pen-touch* (Figure 5.12b): Technique significantly affected pen-to-gaze distance in the sequence ( $F_{2,34}=33.2$ ,  $p<.001$ ) and the parallel task ( $F_{2,34}=9$ ,  $p<.001$ ). Across both tasks, pen-touch users looked closer to the pen than with the other techniques ( $p<.05$ ) (sequence task both pairs  $p<.001$ ; parallel task

pair<sub>pen-touch/direct-touch</sub>  $p=.021$ , pair<sub>pen-touch/gaze-touch</sub>  $p=.009$ ). No significant difference was found between direct-touch and gaze-touch (sequence task  $p=.502$ ; parallel task  $p=3$ ). This indicates a distinct behaviour of visual attention and pen usage between pen-touch and the other techniques.

*Users look closer to their touch positions with direct-touch* (Figure 5.12c): Technique significantly affected touch-to-gaze distance in the sequence ( $F_{2,34}=54.2$ ,  $p<.001$ ) and the parallel task ( $F_{2,34}=26.7$ ,  $p<.001$ ). Across both tasks, we found that direct-touch users looked significantly closer to their touch positions (all pairs  $p<.001$ ). No significant difference was found between pen-touch and gaze-touch (both pairs  $p=3$ ). Thus users do indeed deviate from the direct-touch pattern, and use indirect-touch offset from their visual focus.

### 5.3.6 User Feedback

After each task $\times$ technique block, users filled out a Likert Scale questionnaire. The results are presented in Figure 5.16.

In the sequence task, a Friedman test found significant differences between conditions for responses on ease ( $\chi^2(2)=7$ ,  $p=.03$ ) and learnability ( $\chi^2(2)=9.8$ ,  $p=.007$ ). Users perceived direct-touch as easier to use ( $Z=-2.49$ ,  $p=.039$ ) and also as easier to learn ( $Z=-2.65$ ,  $p=.024$ ) than pen-touch. This was expected as users have significant prior experience with direct-touch.

In the parallel task, significant differences between conditions were found for responses on ease ( $\chi^2(2)=11.5$ ,  $p=.003$ ). First, users found pen-touch easier than direct-touch ( $Z=-2.84$ ,  $p=.015$ ). In addition, users found gaze-touch easier to use than direct-touch for this task ( $Z=-2.49$ ,  $p=.015$ ).

Results of the rankings at the end of the experiment are presented in Figure 5.17, indicating more preference for gaze-touch, and least preference for pen-touch interaction.

Users filled out a questionnaire at the end of the study about why they chose their ranking. The results are mixed across the users. Three users stated that direct-touch is intuitive and familiar (*'I don't perceive any difficulty in the task due to the familiarity'*), while two users stated that this technique is too much physical effort over time (*'it requires too much physical effort and hand coordination'*).

One user disliked that pen-touch occludes (*'the pen was distracting my field of view'*), and two users that it is tiring to use (*'it is tiring to have to use both hands at once'*), while three users were positive about the technique for precision (*'my arm movement allowed more precision when zooming into a target circle'*) and little effort.

Three users disliked gaze-touch as *'it tires the eyes on a gradual basis'*, *'Hand operation is more intuitive than gaze'*, and as one user *'found it frustrating at times when the grey circle disappeared off screen'*, yet six users favoured gaze-touch because of less physical effort (*'I did not have to consciously use both hands, which was physically demanding sometimes'*), and for easier zooming (*'I didn't have to think much about where to zoom as I was usually already looking there'*).

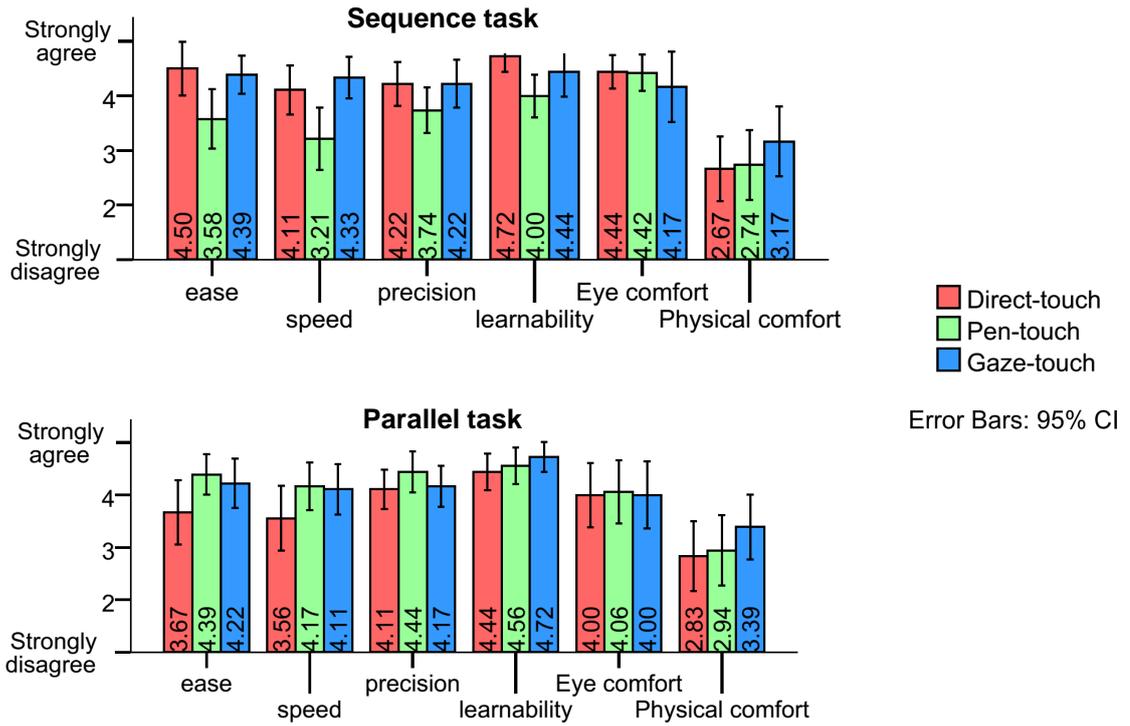


Figure 5.16: Mean Likert Scale responses from participants

## 5.4 Discussion

We discuss the results based on our initial research questions:

**Task Completion Time:** The times indicate comparable performance across the techniques and tasks; however, pen-touch was significantly slower than direct-touch for the sequential task. No differences were found with gaze-touch compared to other techniques. We attribute the lower performance of pen-touch to its necessity of bimanual input. Using two hands for a task that can be accomplished with one hand introduced additional effort. This is partly supported by the qualitative data, where users found pen-touch more difficult than direct-touch in this task.

**Accuracy:** Across both tasks, users were most accurate with gaze-touch, and least accurate with direct-touch. We found that direct-touch accuracy degrades over time, while the indirect techniques remained at a stable accuracy. This is clearly accounted to the Dynamic Targeting aspect, and is the main benefit of indirect-touch that we found in our experiment. With increasing accuracy of eye trackers, we see potential of more substantial accuracy improvements, as our post-study accuracy tests showed a gaze inaccuracy of approximately 63px ( $\approx 1.5^\circ$  visual angle). While pen-touch similarly showed this accuracy, the necessity for bimanual input showed a decrease in accuracy in

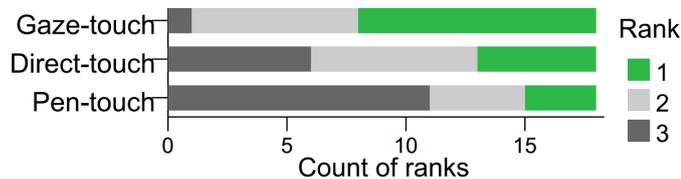


Figure 5.17: User rankings after the tasks indicate a preference toward gaze-touch.

the sequential task, making gaze-touch more suitable for these tasks.

**Gesture Characteristics:** Users perform zoom gestures slightly different when indirect: they can be shorter in duration (parallel task) and less frequent (sequence task). Of course, a shorter duration of gestures naturally increases the frequency of gestures, and it is unclear whether this is a benefit or drawback. It is possible that users performed them faster to minimise finger movement effort. However, in light that there was no best performer in task completion time, further studies are needed to clarify direct/indirect differences and how they affect performance.

**Parallelism between Pen and Touch:** Pen-touch has high parallelism as it requires both modalities, which has potential to introduce overhead when using pen and touch in sequence, but which is acceptable in a simultaneous pen and touch task. Gaze-touch showed less parallelism in the simultaneous task, probably as sub-tasks that are normally associated to two hands, are now offloaded to the gaze modality without introducing significant overhead (qualitative ratings did not show increase in eye fatigue). A decrease of parallelism can indicate a release of manual pen and touch effort, yet further long-term studies are needed to clarify this aspect.

**Spatial Distribution of Input Modalities:** Indirect techniques lead to using pen and touch far apart from each other. We also found that with the pen-touch technique, users look closely at the pen device, and with direct-touch, users look closest to their touch input. These are findings that we expected by the design of the tasks. Interestingly, users held pen and touch further apart during gaze-touch use, thus the addition of gaze decouples both modalities more than other techniques. This can be a problem as users see less physical feedback of their hands but also a benefit as it indicates less occlusion / interference occurrences.

**User Feedback:** While user ratings slightly correlated with the performance data (direct-touch easier in sequential use, but more difficult in parallel use), the commented feedback was mixed. For each technique, some users found it physically demanding while others found it easy, though the overall ranking indicated preference toward gaze-touch.

In summary, our study pointed to the potential for new pan and zoom navigation techniques enabled by the partial indirection of bimanual input. This concept avoids the interference issue of the hands, making simultaneous pen and touch (i.e. when the user wants to issue pen and touch in the same area) easier to use. This can lead to new applications where users e.g. easily navigate the canvas while precisely drawing with the pen; applications that particularly exploit the simultaneity of two modalities while also indicating potential improvements of bimanual UIs in general.

When looking beyond our investigated pan and zoom tasks, interaction designers need to consider limitations such as ambiguous or inaccurate object selection with eye gaze. The gaze-touch technique can be adjusted to only select at touch down, a method that avoids ambiguity and enables new multi-touch gesture possibilities [PACG14]. To improve accuracy and thus interaction with small targets, additional precision mechanism can be integrated [SD12b, ZRZ08]. Another challenge is to integrate these techniques into current applications (e.g. Adobe Photoshop). The new techniques can be implemented as new modes in the existing menus; or use more dynamic context switching mechanisms that leverage the user's visual attention during manual input [PAC<sup>+</sup>15].

Integration of gaze to enable indirect-touch raises more questions about the role of gaze. The distinguishing factor for the indirection is the target of manipulation, which is the gaze target, and which is offset from the touch input position. In here we see similarities to input with direct-touch. For instance, with direct pinch-to-zoom it is unlikely to look at the touch positions because they occlude that part of the UI. We often observed users perform zoom-in offset from the actual target, to see the details appearing while zooming. Our quantitative data aligns with this observation, as touches were approximately 4 to 7 cm (Fig. 5.12d) offset from the gaze position. Then, the characteristics of direct-touch are similar to gaze-touch: the actual target the user wants to manipulate (and sees) is offset from the touch input position. This would suggest to consider gaze-touch as an extended version of direct-touch, where users control any point they see but can manipulate their point of interest more accurately instead of always being slightly offset from what they want.

## 5.5 Conclusion

In this chapter, we explored the simultaneous use of direct and indirect inputs using two hands. In particular, the use of gaze-touch (indirect touch) in one hand together with pen input of the dominant hand. We also introduced the pen-touch technique for indirect-touch interaction without use of gaze, as an alternative. Thus we investigated two configurations where direct pen input is combined with indirect-touch variations: (1) touches redirect to the user's gaze, and (2) touches redirect to the pen's position. These techniques and a control condition (direct pen and touch) were evaluated in a bimanual pen and touch experiment. The results provide new insights into direct versus indirect inputs, how integration indirect-touch retains the dynamics of bimanual pen and touch, and how the gaze modality seamlessly integrates into these contexts.

All together, we covered the design space of direct and indirect interactions using gaze input. Although there were overlapping techniques and examples across the chapters, at the core each chapter explored one dimension of the design space individually. To provide a holistic view across the dimensions, the next chapter explores the whole design space on the example of mobile multi-touch tablets.

# 6

## Gaze and Touch Interaction on Tablets

This chapter explores how all dimensions of the direct and indirect input design space can be applied to tablet devices. It is also relevant as the previous chapters as well as other work in literature focused on the context of interactive multi-touch surfaces [PACG14, PAC<sup>+</sup>15, SJ16, TABG15, VMSB15]. However, a more popular and established multi-touch surface is a tablet, which provides a different challenge because of its portability. The multi-touch capabilities are affected by how users naturally hold the device. While holding, the thumb can theoretically add touch input, but is limited in reach [OC12, TCJD13, WH14]. With eye tracker integration on mobile devices becoming increasingly feasible [BG10, RZ14], we explore how gaze can enhance touch interaction of the hand that holds the device.

We propose combined gaze and touch input on tablets, where touch input of the thumb is redirected to the gaze target. This provides whole-screen reachability with utilising a single hand for both holding and input. For instance, a direct-touch selection task can involve both hands: one hand holds the device, the other hand selects a target. Using gaze and touch, users can simply point their gaze at the target, and use the thumb of the holding hand to select the target. We conducted a user study comparing direct-touch to a previously introduced gaze and touch technique [PACG14] in a homescreen based tapping task. Results show that gaze and touch was only slightly slower, but easily allowed one-handed single-thumb use on the whole screen with less physical effort.

In other applications, however, small targets can be difficult to select as our study also indicated that gaze has lower accuracy than direct-touch input. We aimed at support of precise touchpad-like cursor input, which is difficult because cursor dragging and direct-touch scrolling gestures conflict with each other. We therefore developed *CursorShift*, a method to temporally activate a cursor. The cursor activates at the user's gaze position when issuing a tap from the grip position, and deactivates after another tap which is also used to perform a 'click' on the cursor's target. This technique is particularly useful in browsers. Users can comfortably scroll a webpage using the thumb with the hand that holds the device (Figure 6.1a). Then, if the user wants to select a hyperlink, the user can utilise the same finger to instance a cursor, drag it precisely, and click the link (b).

With the three techniques direct-touch, gaze and touch, and cursor input potentially



Figure 6.1: We explore gaze and touch input for tablet interaction during grip, e.g. to enable the user to scroll a webpage with direct-touch (a), and temporarily use a cursor to click hyperlinks (b) with the same hand. A two-handed grip allows a comfortable two-thumb based zoom, while the zoom location is specified with gaze (c). Our gallery application allows users to open an image simply by looking at it and a tap from the gripping hand (d). Or, to drag images through direct-touch, while users indirectly scroll the view they look at (e). The red circle indicates the user’s gaze in all figures.

available, we explore how these techniques combine and their potential utility in three applications. In our browser application, we show that, in addition to interleaved scrolling and clicking, users can utilise gaze and touch to rapidly interact with the browser menu and the virtual keyboard. In our maps application, we show gaze-enhanced navigation techniques that, for example, allow users to choose the zooming target with their gaze when using a comfortable two-handed grip (Figure 6.1c). In our gallery application, we show how opening and closing images is simplified (d), gaze based scrolling, and how these techniques complement image dragging actions (e). Collectively, the interaction examples demonstrate how tablet interactions can be performed with single-finger, single-hand, and single-grip input, indicating that the introduction of gaze on tablets can make tablet devices easier to use.

Our contributions are as follows. First, we present a user study comparing direct-touch to a previously introduced gaze and touch based method in a tablet tapping task. Second, we introduce CursorShift as a method that enables switching between direct-touch and an indirect cursor. Third, we present three applications that describe how input of the direct-touch, gaze and touch, and cursor combine in the same UI, demonstrate the potential utility of gaze, and with it introduce novel interaction techniques for tablet devices. Our work also shows that by introducing gaze we can reduce physical effort, and combine them for novel bimanual techniques.

## 6.1 User Study

We conducted a controlled study on gaze and touch interaction to investigate the feasibility of the previously introduced gaze-touch technique in the new tablet context. We chose a homescreen based interface for our study design, as they are commonly used in tablets, and as targets are relatively large alleviating potential gaze inaccuracy [KKP<sup>+</sup>08, SD12b]. We compare two techniques: touch-only versus gaze-touch. We chose a fixed setup, although unusual for tablet devices, to avoid potential effects on eye tracking accuracy from user movement. The tablet is placed on a table (its integrated stand fixes a comfortable position), and users sit on a chair in front of it ( $\approx 55\text{cm}$ ) during the study (Figure 6.2a).

### 6.1.1 Study Design and Method

*System:* we use a Microsoft Surface Pro 4 tablet with an attached Tobii EyeX eye tracker. The tablet has a display size of 2736x1824 pixel (27x18 cm), runs on a i5-6300U CPU

(2.4GHz) at 8GB RAM on Windows 10, and supports up to 10 finger touch at 60Hz. The eye tracker provides gaze data at 30 Hz. A 16-point gaze accuracy test for each user, after each study, showed an average accuracy of  $M=1.23^\circ$  ( $\approx 43$  px) of visual angle ( $SD=.88^\circ$ ). The software is implemented with Java and Processing, logging temporal and spatial data of the used modalities at 30 Hz.

*Task Design:* 24 targets were evenly distributed in a 6 x 4 grid on the UI, and designed to resemble a typical homescreen layout with app-like icon targets (Figure 6.2b). All targets were grey, the current target to be selected is red and becomes green when successfully selected. When the wrong target was selected, it briefly appeared blue. Each target has a size of 160x160 px, i.e. 1.57x1.57 cm, and the target's centres are 360 px offset from each other. After an error, the next target was displayed and the missed target was repeated in the same block to ensure equal number of successful trials. Users performed 7 blocks per technique, overall resulting in 2 techniques  $\times$  7 blocks  $\times$  24 targets = 336 successful trials per user.

*Techniques:* With *direct-touch*, users selected a target by directly tapping on it (using their dominant hand). If the user's touch point was within the target's boundaries the task was successful. Users were allowed to use the non-dominant hand to hold the tablet while they interacted with it. With *gaze-touch* [PACG14], users selected a target by looking at it and tapping anywhere on the screen. Here users only used their dominant hand to hold the device, while they performed tap actions with the free thumb of this hand. Due to the inaccuracy of eye trackers we used a target snapping mechanism based on previous work [PACG14, VMSB15]. The target which center was closest to the system's gaze estimate is highlighted with a yellow border, and was selected when users touch the screen. This effectively increases the target's size to 360px ( $\approx 3.6$ cm).

*Procedure:* After an initial briefing and demographic questionnaire, the eye tracker was calibrated to the user with the standard Tobii EyeX tool. Users then conducted the study tasks with each technique (order counterbalanced). Before each technique, users performed a few training trials to become used to the technique ( $\approx 3$ -5 trials) and were instructed to be as fast as possible, while still comfortable. Then, in each technique session, users performed 7 blocks. Targets appeared in random order. An additional target was used at the beginning of the block not included in the data analysis. After

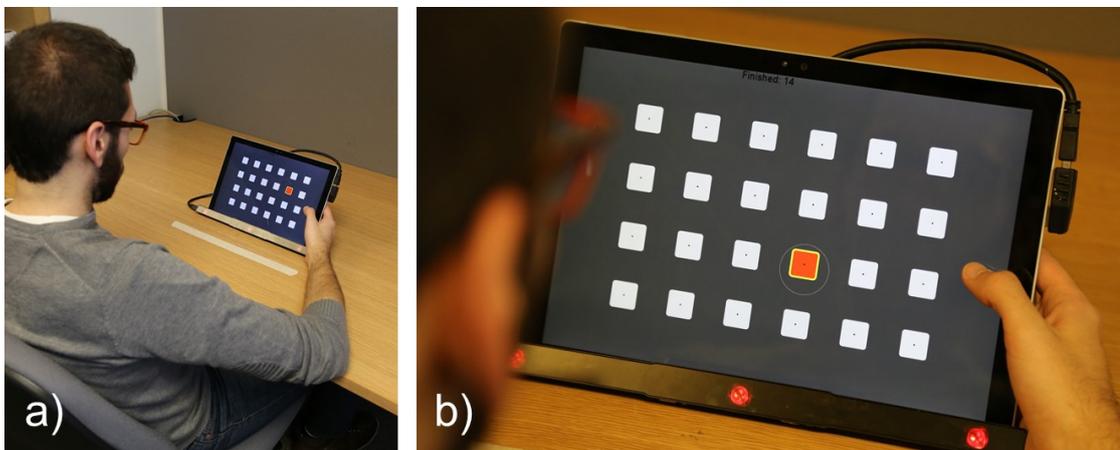


Figure 6.2: User study: users interacted with the tablet placed on a table (a), and performed a task that resembled homescreens of tablet UIs (b).

each block, the user can have a small break (but normally users continued). After each technique, users filled out a questionnaire with six Likert-scale statements from 1 (strongly disagree) to 5 (strongly agree): ‘*The task with this technique was [easy to use | fast | precise | easy to learn | comfortable for my eyes | physically comfortable (hand, arm, shoulder, or neck)]*’. After the study, users ranked the techniques and gave qualitative feedback on the techniques. Overall, the study lasted 15-20 minutes per user. The questionnaires are provided in appendix .3.

*Participants*: 12 volunteers of the local university from 24 to 35 years (M=27.3, SD=4, 4 female) participated in the study. Three users wore glasses, two were left handed. On a scale between 1 (no experience) to 5 (Expert), they rated themselves as well experienced with multi-touch (M=4.1, SD=1.1), and less experienced with eye gaze interaction (M=2.8, SD=1.2).

### 6.1.2 Results

We employed a T-Test for the statistical analysis of the following performance results. For time and accuracy measures, only successful trials were chosen (when the correct target was selected), for non-successful trials see error rate.

*Time (Fig. 6.3a)*: The users were significantly faster with touch (M=725ms, SD=48ms) than with the gaze-touch (M=911ms, SD=227ms) condition ( $t(12)=2.8$ ,  $p=.017$ ).

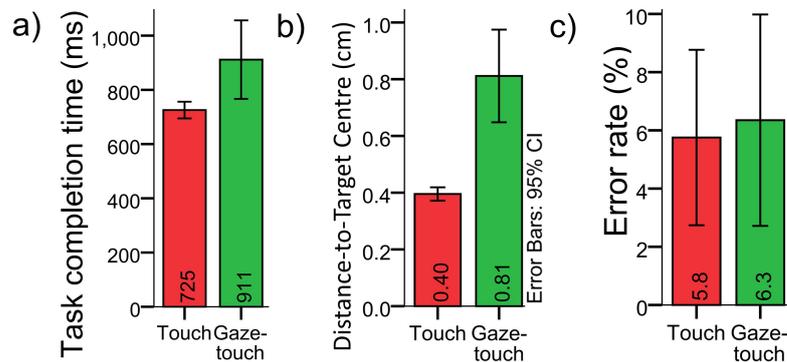


Figure 6.3: Quantitative study results

To get further insights into temporal performance, we measured how long it took users to select the target after they saw it. Our system detects ‘seen’ when the target closest to the user’s gaze equals the task’s selection target. We found that direct-touch users needed 260 ms, and with gaze-touch 388 ms to finish the task after they saw it (significantly different at  $t(13.7)=2.6$ ,  $p=.02$ ). The major performance decrease comes from the point after users visually acquired the target.

We then plotted the spatial relationship between gaze and target position across time in Figure 6.4. The Y-axis shows the distance between gaze and target point, and the X-axis shows the time relative to the point where users touch down ( $t=0$ ). We found users look closer to the target during the task, until they reach a minimum at approximately the moment of touch down. The results indicate that when using gaze-touch users are more visually focused on the target after touch down, whereas they already started to look away with direct-touch. The added attention could be a factor for the increased completion time for gaze-touch.

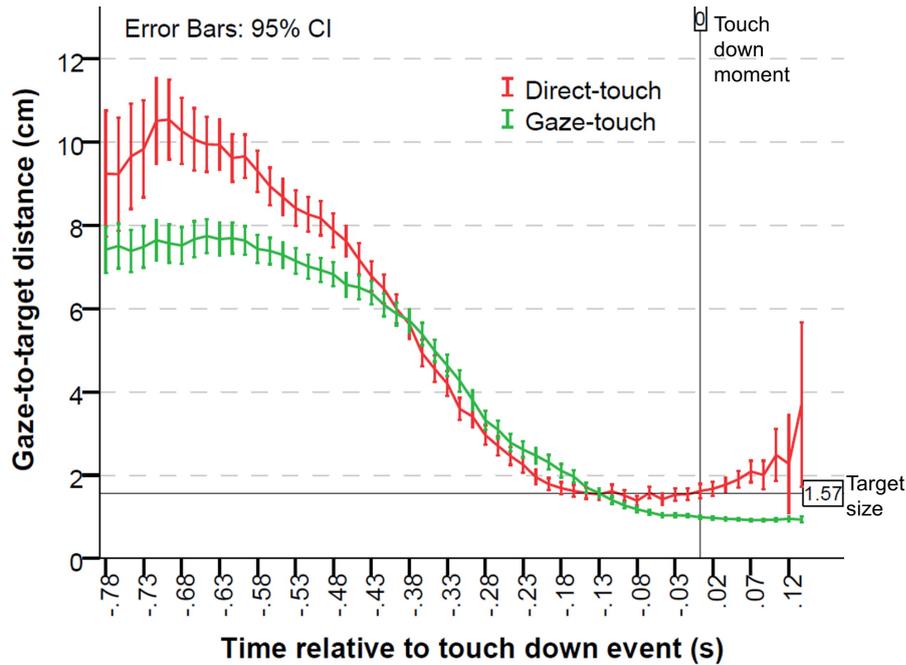


Figure 6.4: Gaze-to-target distance over time, averaged across all tapping tasks. As expected, users initially look far away, then focus the target. With gaze-touch, users focus on the target for longer.

*Distance-to-Target Centre (Fig. 6.3b):* Although users were not instructed to precisely select the centre of the target as it is the norm in homescreens, users selected the target significantly closer to the centre with touch ( $M=40\text{px} / .39\text{cm}$ ,  $SD=4\text{px} / .04\text{cm}$ ) than with the gaze-touch ( $M=82\text{px} / .81\text{cm}$ ,  $SD=26\text{px} / .26\text{cm}$ ) condition ( $t(11.5)=5.5$ ,  $p<.001$ ).

*Error rate (Fig. 6.3c):* No significant difference was found between the touch ( $M=5.6\%$ ,  $SD=4.7\%$ ) and the gaze-touch ( $M=6.3\%$ ,  $SD=5.7\%$ ) condition ( $t(22)=.28$ ,  $p = .79$ ). The error rate describes the probability that users did not select the correct target (i.e. error trials / number of trials).

*Feedback (Fig. 6.5):* We ran a Friedman Test on the Likert-scale categories, and found no significant differences in the categories except for physical comfort ( $\chi^2(1)=11$ ,  $p=.001$ ). Users perceived gaze-touch ( $M=4.2$ ,  $SD=1$ ) as more physically comfortable than direct-touch ( $M=2.1$ ,  $SD=.9$ ). This result correlates with the rankings, where 8 of 12 users favoured gaze-touch instead of direct-touch. While only 2 users stated they dislike gaze-touch for inaccuracy (e.g., ‘the gaze and touch was not always 100% accurate’), 8 users preferred gaze-touch for less physical effort and no screen occlusion (e.g. ‘touch was tiring and putting my hand over the screen obscured’).

*Gestural differences:* We looked at tap path and duration differences between index finger (direct-touch) and thumb (gaze-touch). Results are listed in Table 6.1. For tap path, we found that direct-touch users significantly involve more movement than with gaze-touch ( $t(11.7)=2.7$ ,  $p=.019$ ). This difference is potentially due to the directional use of the index finger on the slightly diagonally oriented tablet. This potentially induces a minor sliding touch effect. For tap duration, we found that gaze-touch users significantly tapped for a longer time than direct-touch users ( $t(13.7)=4.9$ ,  $p<.001$ ). Possibly the thumb provides a larger touchspace and thus could be sensed as a longer touch; second,

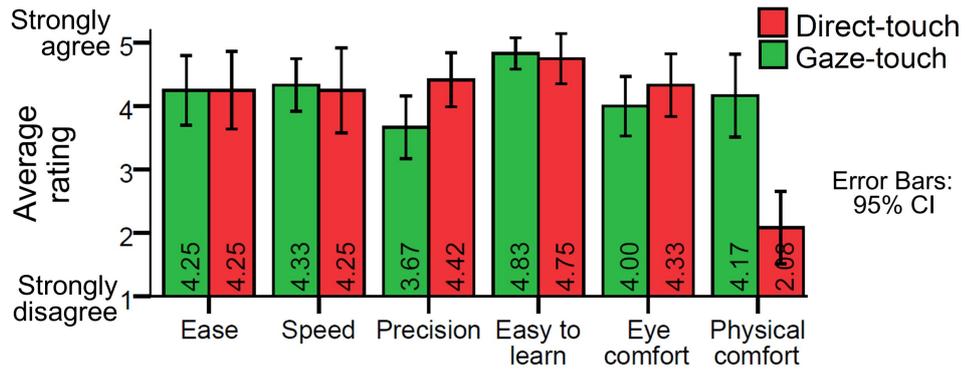


Figure 6.5: Likert-scale ratings

because of above mentioned synchronisation issue, users might touch for longer to ensure reliable selection.

	Direct-touch		Gaze-touch	
	M	SD	M	SD
Tap time	53.2ms	14.3ms	116.8ms	79.3ms
Tap path	14.8px .14cm	6.8px .06cm	4.6px .04cm	2.4px .02cm

Table 6.1: Tap gestural differences between the techniques.

### 6.1.3 Discussion

This study shows that users can easily select targets with just a single hand and use of gaze and touch modalities while holding the tablet. Most users preferred gaze-touch and found it less physically fatiguing than direct-touch selection. These results empirically show the potential of gaze on tablets. Direct-touch users were about 20% faster than with gaze-touch. Yet, users did not perceive this as a hard problem. This result aligns with previous work that compared direct-touch to other indirect techniques (e.g., 35% faster than miniature interaction areas [WH14], or 55% faster than gaze-only techniques [RZ14]). Lastly, we tested with a basic gaze selection in this study as a baseline measure, but in future we want to consider more intelligent selection mechanisms [KKP<sup>+</sup>08, VMSB15].

This study focused on a homescreen task that by default comes with large targets, avoiding issues of gaze inaccuracy. This demonstrates the potential of gaze on tablets, but also that further study is needed to get more insights into its general applicability on tablets. There are numerous applications, e.g. browsing or ports of desktop applications, that require more precise input than eye gaze can offer. In the remainder of the paper, we will focus on how to cope with interfaces that require finer control, and further how gaze input integrates into existing direct-touch user interfaces.

## 6.2 CursorShift

Our study showed that thumb-only input on the whole tablet screen is possible with gaze-touch, but also that gaze selection is potentially inaccurate. We want to support a cursor

technique on tablets for precise input. The cursor interaction can be based on a laptop's touchpad. Its core operation is as follows: touch dragging moves the cursor, and a tap selects the object under the position of the cursor. However, the cursor dragging gesture conflicts with the direct-touch scrolling gesture for which we developed the following technique.

CursorShift is a method to enable switching between direct-touch dragging and cursor interaction. It uses the gaze modality and distinction between tap and drag gestures to accomplish this. The idea is to use an indirect tap to toggle to cursor mode. At tap, the system determines the distance between the gaze and tap position (Figure 6.6a-b). If the distance is lower than a threshold of 5 cm, it is processed as direct-touch. If it is higher, cursor mode starts. The cursor then appears at the user's gaze position (b). Subsequent drag gestures then drag the cursor (c). A tap gesture performs a 'click' at the cursor position, and also toggles back to direct-touch mode (d).

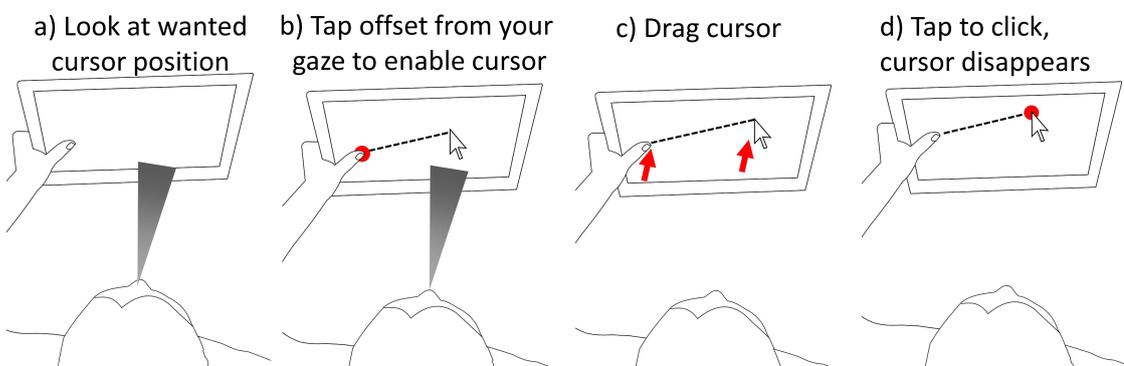


Figure 6.6: CursorShift is a method to temporally enable a cursor, without interfering with default direct-touch actions.

In cursor mode, there are multiple ways to return to direct-touch (Figure 6.7). First, users can tap and perform a click to finish the action; whether the target is an object or blank space. Then, users can use the second hand to touch in the area of the cursor. This is processed as direct-touch input conflicting with the cursor, and the cursor is cancelled.

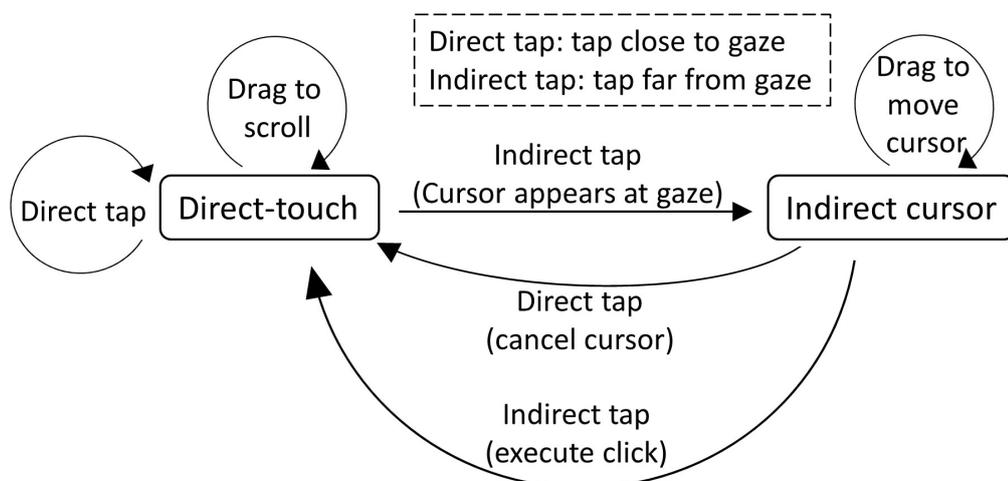


Figure 6.7: State model of CursorShift.

CursorShift is suitable for applications that require both direct-touch input and cursor input. We illustrate its utility in our next section, particularly in the browser application.

PROPERTY	DIRECT-TOUCH	GAZE-TOUCH	CURSOR
Reach and grip	<b>Low</b> (grip constraint, needs two hands)	<b>High</b> (Indirect input; with any grip position)	
Accuracy	<b>Moderate</b> (fat-finger issue)	<b>Low</b> (gaze imprecision)	<b>High</b> (pixel precise)
Speed	<b>High</b> (move hand)	<b>Moderate</b> (look and touch)	<b>Low</b> (Drag cursor before tap)
Fatigue movement /	<b>High</b> (move hand)	<b>Low</b> (direct gaze or drag cursor)	
Occlusion	<b>High</b> (finger/hand/arm)	<b>None</b> (gaze)	<b>Minimal</b> (cursor)
Gesture possibilities	<b>High</b> (e.g. tap, drag, long touch, double tap)		<b>Moderate</b> (drag occupied by cursor control)

Table 6.2: Summary of the interaction techniques' properties.

### 6.3 Touch, Gaze, and Cursor

In summary, we can support three techniques for input on tablets: direct-touch, gaze-touch, and cursor (Figure 6.8). We now explore combinations of the three techniques in three applications: browser, maps, and image gallery. In these applications, CursorShift enables switching between direct-touch and cursor, and gaze-shifting [PAC<sup>+</sup>15] enables switching between direct-touch and gaze-touch.

A short characterisation of the three techniques is given in Table 6.2. Direct-touch allows natural, fast and precise input, but has limited reach during grip, and is affected by occlusion of content and the fat-finger issue [HB10]. Our study showed that gaze-touch is slightly slower, less accurate, but since it requires less movement it also had less physical effort. Complementary, a cursor allows for high-precision input but it is slower as it requires cursor dragging. Also both indirect techniques enable reaching the whole screen from grip.

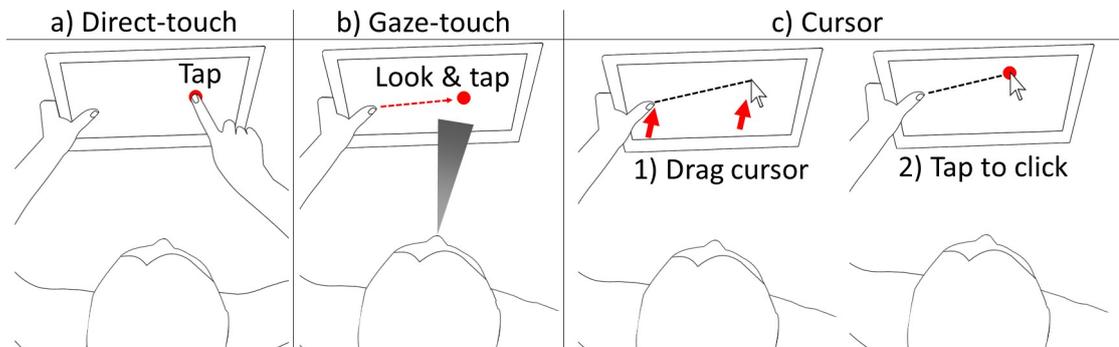


Figure 6.8: Target selection with the three investigated techniques.

In addition to one-handed interaction from the grip position, the techniques also allow for novel bimanual configurations. This occurs when the holding hand maps to a technique, and at the same time the other hand uses a different one. Figure 6.9 illustrates three configurations. Note that the configurations are exclusive; only two techniques can usually be used within the user interface. Which bimanual combination to use depends on the application, which we will also be subject to explorations in the following sections.

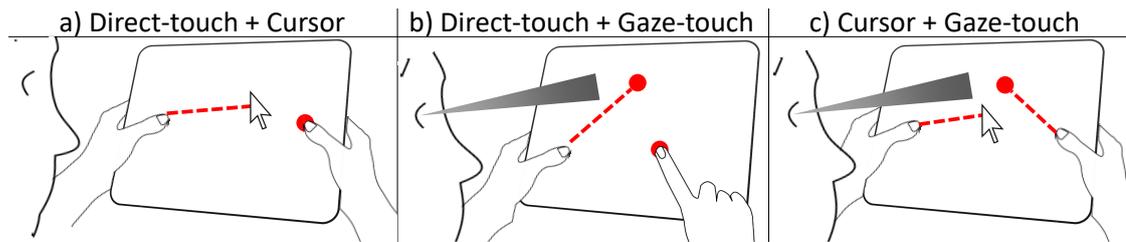


Figure 6.9: Illustration of bimanual configurations of direct-touch, gaze-touch, and cursor.

## 6.4 Browser

Our browser prototype integrates techniques for scrolling and clicking, menu navigation, and virtual keyboards.

### 6.4.1 Scrolling and Clicking

Clicking hyperlinks can be tedious because they are relatively small. With direct-touch, users perform a well-targeted touch, or zoom-in before doing so. With CursorShift, users can temporarily instance a cursor and click the link. For example, Figure 6.10 shows how users do the same task with either cursor or direct-touch. A second available hand allows to physically reach and touch the desired target (a). Alternatively, the holding hand's thumb can select the target as well (b). As the target in this example is a relatively large area, users can just look at this area and perform a quick double-tap that combines cursor activation, click, and cursor deactivation.

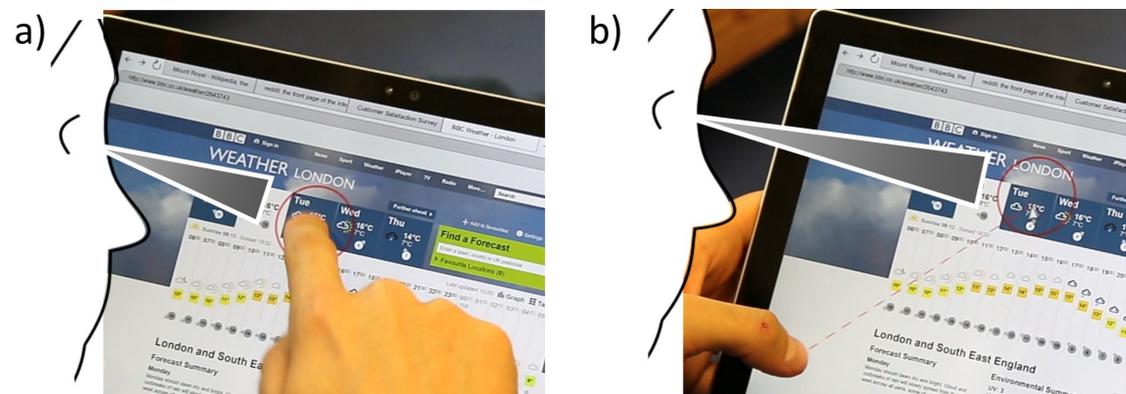


Figure 6.10: Click an element with direct-touch (a) or with a cursor (b).

Figure 6.11 shows how webpage scrolling and hyperlink clicking are interleaved into one flow of interaction. The user is browsing a wiki, and scrolling the information on the website (a). Suddenly, a new item becomes interesting, and the user utilises CursorShift to conveniently click this link (b). Immediately, the user continues scrolling the new site (c).

Figure 6.12 shows how this technique also allows users to perform both tasks bimanually. Users acquire a two-handed grip, and divide the labour between both hands. For example, to scroll the website with the right hand, and issue cursor clicks with the left hand.

This is enabled by including an invisible touchpad area. When users activate the cursor through CursorShift, an invisible area around the corresponding touch position is created.

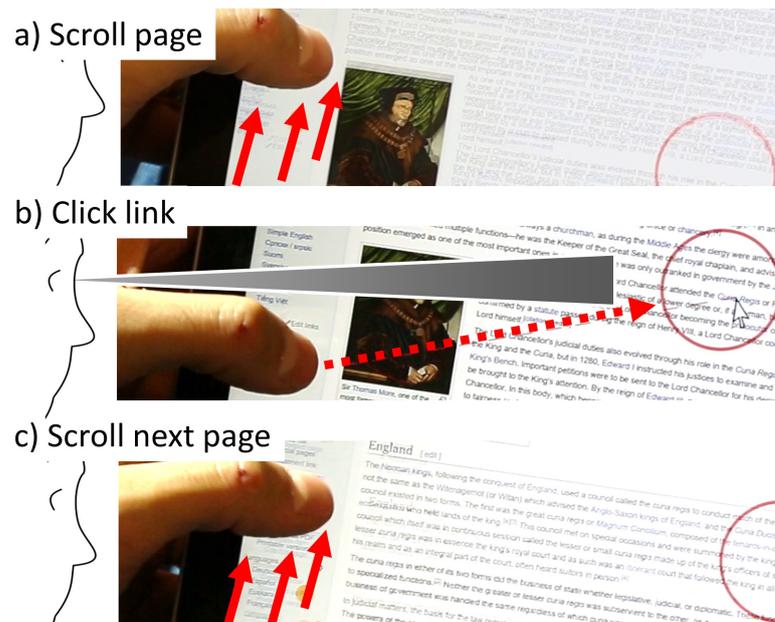


Figure 6.11: A user scrolls the page (a), toggles the cursor to click a link (b), then continues scrolling the next page (c). The system uses the user's eye gaze direction, that is indicated by the red circle, only to position the cursor in (b). If users accidentally tap and get a cursor, users can simply tap again on empty space to remove it.

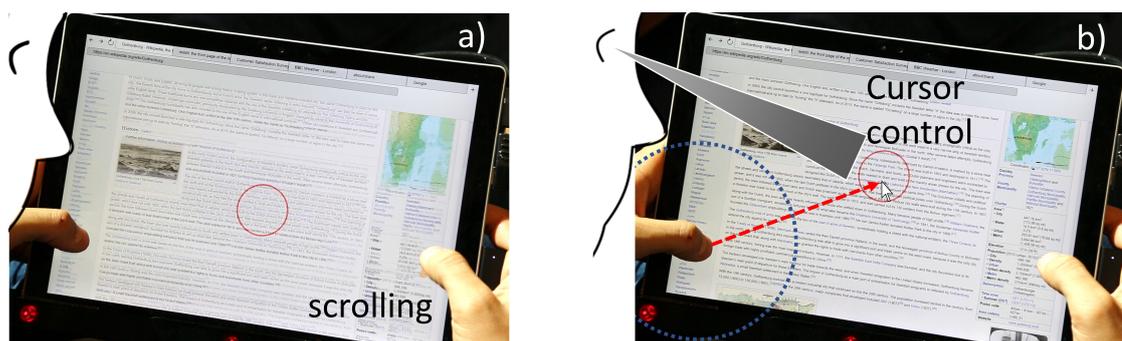


Figure 6.12: In a two-handed grip, users can divide cursor control and website scrolling. The right hand scrolls (a), the left hand clicks a hyperlink (b). The blue circle indicates the invisible touchpad area.

It is essentially a touchpad area, where users drag the cursor (clutch) and click. The area is 7 cm in radius around the initial touch position, and updates its center point with the initial touch position of subsequently arriving drag gestures. It disappears when the cursor disappears. The benefit is that another hand that issues touch input outside this radius can interact with direct-touch, or even instance a second cursor for potential multi-cursor input.

## 6.4.2 Menu Navigation

We also integrated gaze for interaction with the browser menu. We found cursor control for basic menu interactions too slow, and therefore designed a gaze-touch based browser menu. The menu supports default operations, e.g. users can navigate tabs, click 'back' and 'forward' buttons, and open/close tabs. The menu is direct-touch enabled. If users physically touch on one of the browser elements, it is directly triggered. When users look

at any point in the whole menu area, and follow up with touch input anywhere else, users acquire a menu pointer, i.e. an indirect handle toward the menu (Figure 6.13a).

This menu pointer is based on a mechanism that provides rapid selection of large targets, and in case of small targets the option to refine selections. At touch down, the current gaze target becomes indicated as the potential selection by a menu pointer. Users can now optionally perform drag gestures to change the selection to another target (Fig. 6.13b), and then touch up to confirm (c). Or, users can also just touch up after touch down, if confident that the right target is selected at first sight. In contrast to CursorShift, this technique does not support clutching as we assume brief and single-click interactions with the menu.



Figure 6.13: Menu navigation using gaze-touch, i.e. indirect touches that can rapidly navigate the tabs when visually focusing on the menu. The red line indicates the indirect handle to the remote point.

This menu technique extends prior gaze based menus [PACG14, PAC<sup>+</sup>15]. At touch down the menu pointer's x position is equal to the gaze, and the y position lies in the middle of the tabs' height. The upper row with the tabs and navigation elements is therefore the first that users can access, as it is expected to be more frequently used. Overall, the menu integrates the following drag semantic:

- Drag left-right: changes the menu pointer to another tab, or to one of the other buttons in the vertical row.
- Drag up: closes the tab that is selected by the menu pointer.
- Drag down: allows access to the address bar.

The 'back' and 'add tab' buttons are placed at the left and right end of the first row; this allows users to rapidly reach these positions for successive actions.

### 6.4.3 Keyboard

The menu allows rapid access to the address bar, at which the virtual keyboard is triggered as well. This keyboard is then direct-touch enabled. The keyboard supports blind-typing, i.e. as long as users touch on a key on the keyboard, regardless of where they are looking, they will type in keys.

In addition, users can utilise gaze-touch to interact with the keyboard from a remote touch position. This is particularly useful when users have a different grip, e.g. at the top of the tablet, and just want to type a few words. Gaze-touch to the keyboard activates when the user's gaze is within the keyboard, and the user's touch is not on any of the keys. This also allows users to additionally use the empty borders of the keyboard as gaze-touch area to any of the keys.

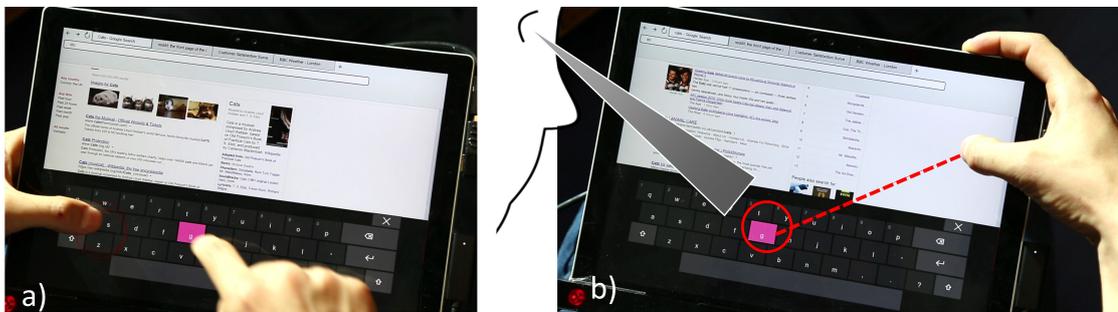


Figure 6.14: The virtual keyboard allows to use one hand to hold the device, and a second for direct-touch key input (a). Or, users hold the device and with the same hand issue gaze-touch input to select keys (b).

The keys are relatively large by design, as in typical virtual keyboard designs, but in other contexts (e.g. tablet in portrait orientation) can be smaller. Thus, the interaction technique involves a similar precision enhancing technique as with the browser menu. Users look at the respective key, and touch down to acquire a handle. Drag gestures move the handle's 2D position, to correct potential erroneous initial gaze selections. Advanced key input, such as a long-touch gesture to access second-level characters such as numbers, is similarly supported: users keep the indirect handle over the key to simulate a long-touch on it.

## 6.5 Maps

Map navigation applications come with the tablet, allowing users to explore locations, or to see how far one place is from another. To support users in these tasks with tablet-specific grips in mind, we built the following techniques.

### 6.5.1 Zooming

In addition to standard direct pinch-to-zoom (Figure 6.15a), we designed two zooming techniques that are specifically useful during grip. They activate when all used touch points were detected as indirect using the gaze-shifting principle (5 cm gaze-to-touch distance); otherwise direct-touch zooming is enabled. Both indirect techniques enable zoom-in at the gaze position.

We fixed the zoom-out position to the screen's center as we found it unnatural to zoom-out at gaze. The reason for this is the user's intent during zoom out. If the user knows the approximate direction they want to go after zooming out, the system would translate toward the inverse direction which is counterintuitive.

*2F-grip-zoom (Fig. 6.15b)*: This technique is used during a two handed grip, where one hand is at each side of the tablet making only two thumb fingers available. By drawing the thumbs apart, users zoom in. In principle users can do the same with direct-touch already — however the distinct benefit is that users can direct their gaze to zoom where they want (instead of only in the center of the thumbs as in direct-touch). We also modified the zooming gain because the two thumbs disallow large movement. The gain depends on the distance between both fingers: with increasing distance, the gain increases and thus allows fast zoom in this grip.

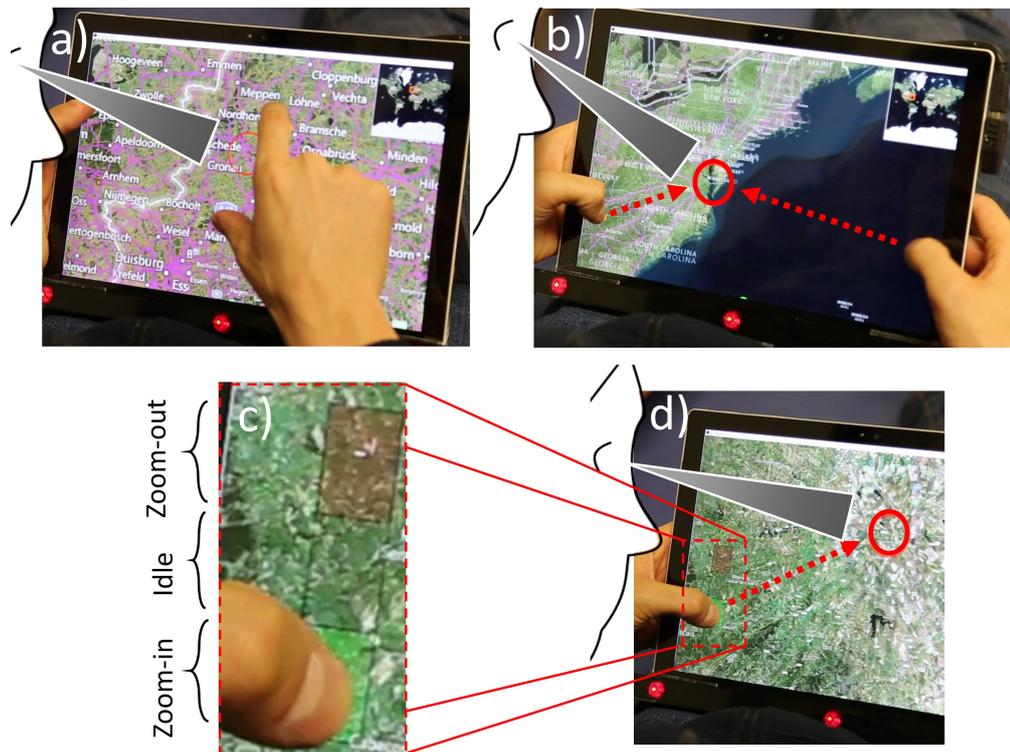


Figure 6.15: Our maps application supports three zoom variants: direct pinch-to-zoom (a), two-thumb gaze based pinch-to-zoom (b), and one-thumb variant that uses a 3-part widget for thumb-sliding (c) to zoom at gaze position (d).

*1F-grip-zoom (Fig. 6.15c)*: Single-finger zoom is particularly useful when users hold the tablet and intend to navigate maps with the same hand. As single-finger drag is used for panning, users perform a double-tap gesture to activate zoom mode. Then a drag up zooms out, and a drag down zooms in. This technique is similar to the current single-finger zoom technique of phones and tablets, but extends it with a *relative, continuous zoom*.

The visual design of the zoom widget is shown in Figure 6.15c. After double-tap touch down, the three elements of the widget appear. Initially, the finger locates at the center of this widget that represents an ‘idle’ mode without functionality. When users move their finger downwards, as in the figure’s example, the map continuously zooms out. Upwards movement continuously zooms in. Zooming occurs at the eye gaze position (d). Returning to the initial touch down point (‘idle’ mode) would stop zooming. It is relative because the further users move their finger away from the initial point, the higher is the zooming gain. To accomplish this, we mapped the distance between both fingers to a zoom gain transfer function; which keeps the zoom gain at about a similar level as when

performing a unimanual pinch gesture. In sum, after double-tap-hold, users can zoom in, out, or remain idle without needing to touch up.

### 6.5.2 Minimap

Some map instances provide users with an additional overview window, e.g. using a minimap of the world to rapidly change locations, while the main window provides standard zooming behaviour. Direct-touch interaction allows control of both by alternating between both available windows.

With the use of gaze-touch, users can easily reach both windows without relocating the physical touch position. When a user looks at the minimap, users can indirectly control the viewport of the minimap. The user touches down with a finger, which renders an indirect handle toward the viewport (Figure 6.16a). Then, indirect touch movement directly translates to movement of the viewport (b). After touch down, users can look elsewhere, as gaze is only used to initially acquire the indirect handle.

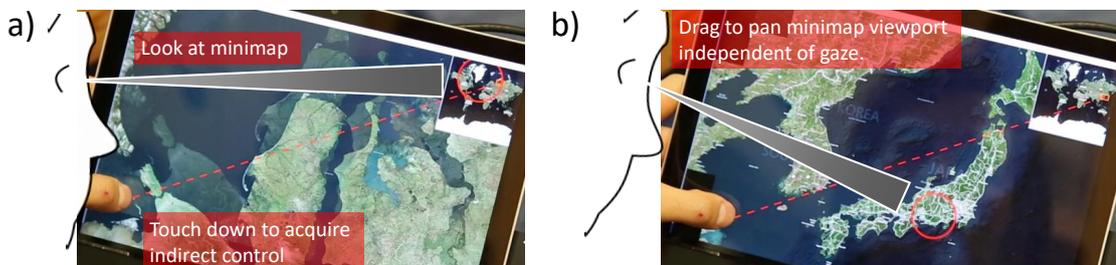


Figure 6.16: By gaze-touch on a minimap, users pan long distances with a single drag gesture from remote.

### 6.5.3 Bimanual Combinations with Markers

Users can also add markers at locations of the map, as reminders of places, or for measuring the distance between two markers. This can be done with both direct-touch and indirect cursor controls utilising CursorShift. For direct-touch, users reach to a position. For cursor control, users look at the desired position, and tap indirectly to activate a cursor. Then users can perform three actions with the techniques:

- Create marker: long-touch on marker-free space.
- Drag marker: double-tap-drag on an existing marker.
- Delete marker: long-touch on existing marker.

Interaction with the markers can be combined with the other navigation techniques when using both hands. In the following, we describe three example scenarios that we have implemented, that show one example for each bimanual configuration between the three investigated techniques (c.f. Fig. 6.9); the examples are also presented in Figure 6.17.

- Bimanual direct-touch + gaze-touch (a): users can drag a marker with direct-touch (1), while utilising the other hand for gaze-touch on the minimap to quickly travel a long distance (2). This allows to integrate long-distance panning performed with

the non-dominant hand, within the the marker-dragging task performed with the dominant hand.

- Bimanual direct-touch + cursor (b): users can acquire and hold a marker with the cursor (1). While they do this, they can pan the map with the free hand to find the location the marker should be dropped at (2).
- Bimanual gaze-touch + indirect cursor (c): lastly, users can create a marker with the cursor as in the previous example (1). At the same time, gaze-touch is used for single-finger zooming (2).

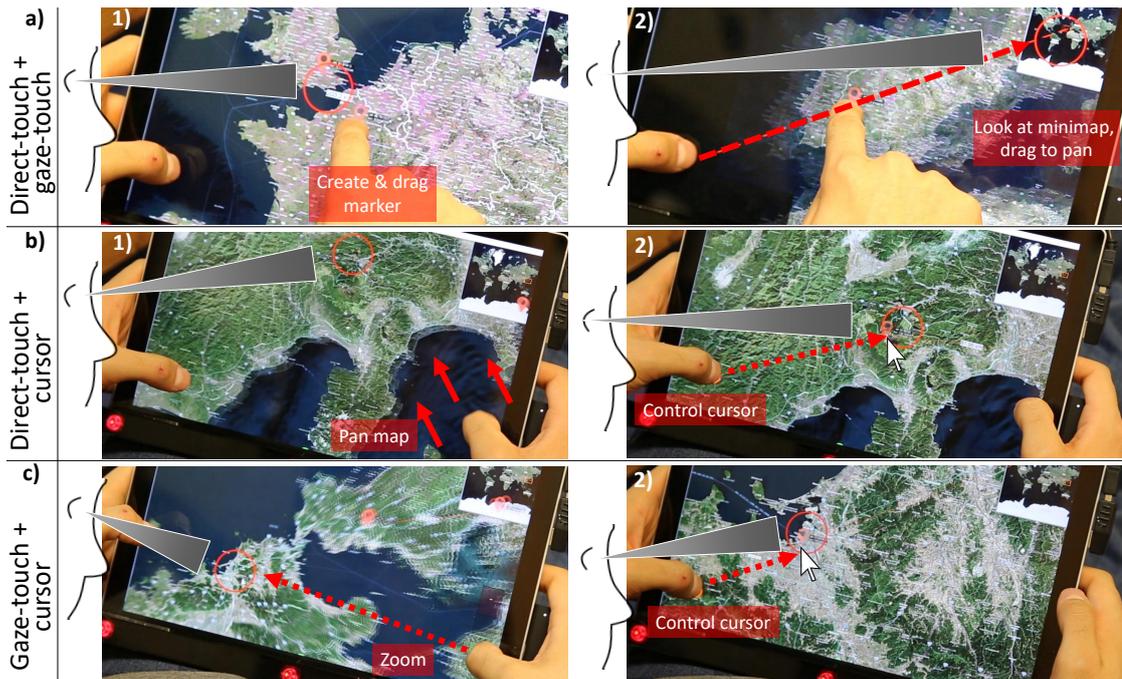


Figure 6.17: Three bimanual combinations, each demonstrated in their own task: direct marker dragging with indirect minimap control (a), direct panning with cursor based marker control (b), and indirect zooming with cursor based marker input (c).

## 6.6 Gallery

An image gallery allows users to browse the images stored in their tablet. The design of our prototype is based on typical gallery applications, enabling users to scroll through image folders and view them in different scales. We describe scrolling, image selection, and image dragging.

### 6.6.1 Area Scrolling

The gallery is based on two user interfaces. The first UI consists of two areas (Fig. 6.18a-b). The first area is a grid of images, and the second area is a smaller panel at the left that lists folders ordered by month. After selecting an image in the first UI, users get to the second UI (Fig. 6.18c-d). This shows the corresponding image enlarged, and in addition provides a horizontal quick list at the bottom with small images.

The drag gestures scroll the area of the UI that users are looking at. At touch events, the system determines which area the user looks at. Drag gestures are then forwarded to the corresponding area. This provides users with the capability to scroll any area in the given UI with a single finger, without the need to leave their current grip. In our first UI, users can look at the list of month folders and scroll it (Figure 6.18a), or look at the image grid to scroll it (b). In the second UI, users can look at the quick list to scroll it (c), or at the enlarged image to flick to the previous/next image (d).

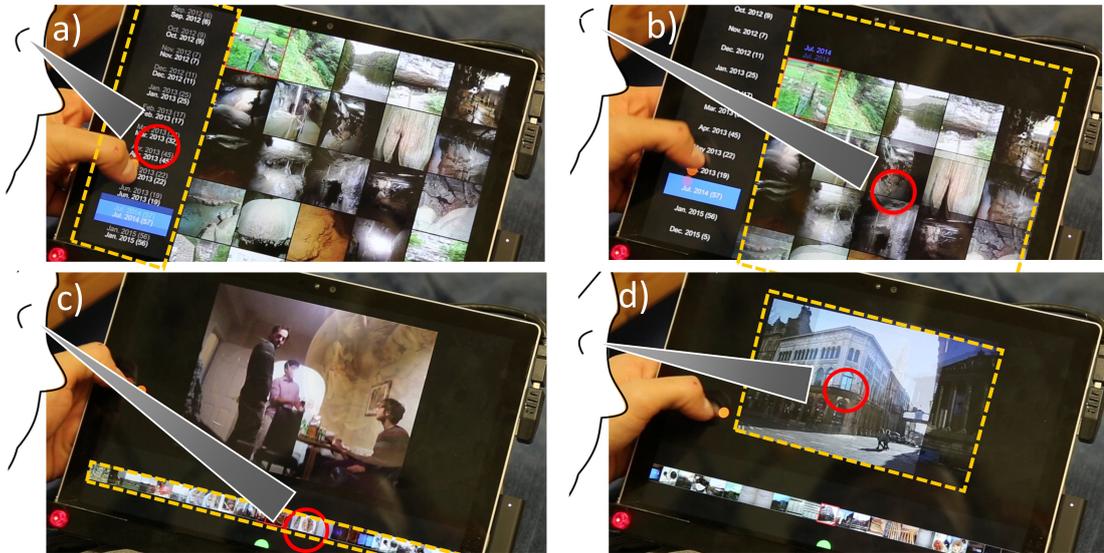


Figure 6.18: In the gallery and single image UIs, scroll gestures affect the view the user is looking at. The scrolled views are highlighted with an orange border.

### 6.6.2 Image and Folder Selection

In addition to the scrolling, the first UI enables users to rapidly select images or folders by a single tap. The images are displayed relatively large in the gallery in order to make them visible to the user. This makes it sufficient to use gaze selection instead of more precise cursor input.

Figure 6.19 shows selection examples with both techniques. Users can directly select the image with the free hand when looking at it (a). The image view shows up, and by touching the ‘back’ button, users can return to the image grid (b). In case the second hand is not available, users can perform the same actions with the tablet-holding hand. To accomplish this, users look at the image and tap with the free thumb (c). This opens the image. To return to the grid view, users look at the back button, and issue a tap from remote (d). Both direct-touch and gaze-touch variant are useful: e.g., direct input clearly shows others which image is selected, while gaze-touch allows users to frequently open/close images with minimal hand movement.

### 6.6.3 Bimanual Combinations with Image Dragging

In the following examples, we show how users bimanually use the two tasks of image dragging and scrolling together. A long-touch on an image acquires it, and then users can drag to a new location.

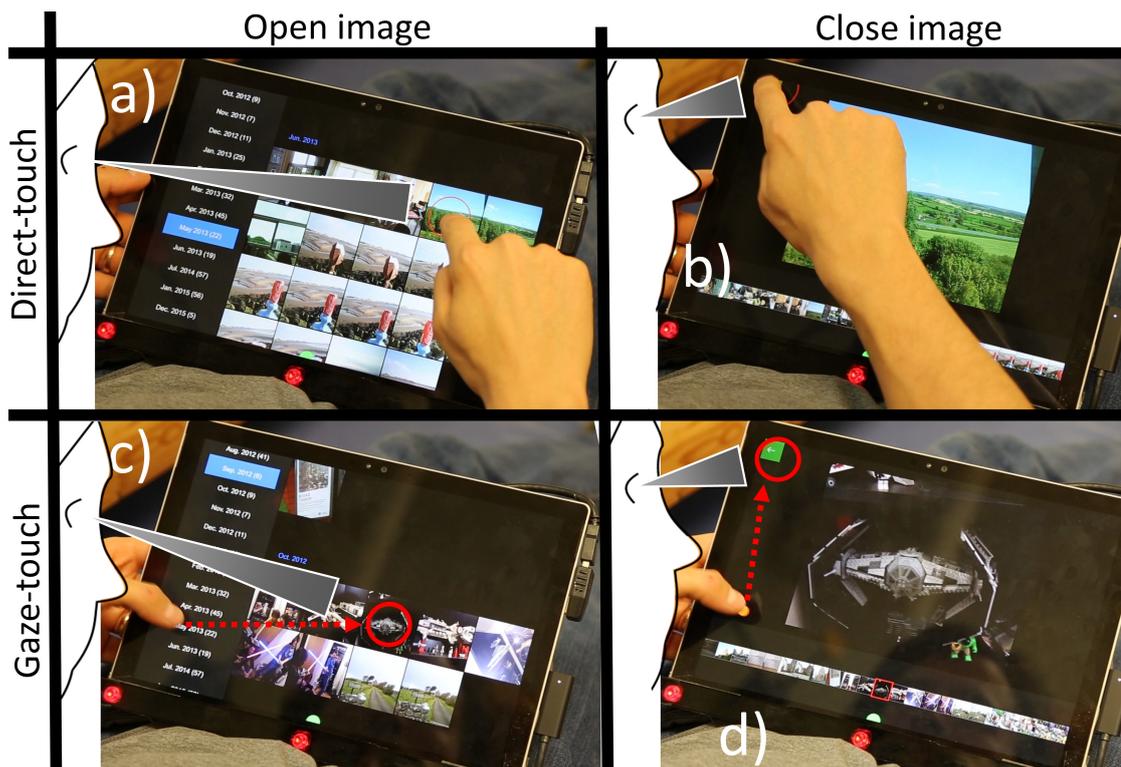


Figure 6.19: A dynamic switching mechanism allows direct- and gaze-touch image navigation. With direct-touch, users physically touch the image and the 'back' button (a-b). With gaze-touch, users can comfortably use the same finger to do the same actions without using a second hand (c-d).

Image dragging is possible with both direct-touch and gaze-touch. Users can directly touch on an image, while looking close to it; or indirectly grab a target by looking at the image, and touching down from a grip position. Notably, with gaze-touch, after users selected the image, gaze has no effect — i.e. there the image will not suddenly jump to the gaze. Instead, touch takes over, meaning dragging will drag the image around.

The image dragging technique fluidly combines with the scrolling techniques (6.20a). For example, users can use direct-touch to grab an image (1), while they use their grip-hand to scroll the month view on the side (2), and then tap to select a month category (3). Naturally, the user looks back to the images that belong to the month. Then, when using the same hand for dragging gestures (4), the images are scrolled — as the system knows the user's attention is on this part of the UI. When the desired location is found, users return to direct-touch and drag the image to the location (5).

In addition, users can perform these actions with unimanual input of the free hand (Figure 6.20b). As in the previous example, the index finger directly grabs an image (1). However, then the thumb of the same hand is used to scroll the month view with indirect touch, as the system detects the user's attention on the month grid (2). Here an indirect thumb tap selects the desired month category (3). The related images are shown in the gallery area, to which the user then visually attends. The system detected this, and the same thumb touches now drag the image area (4). When a desired place is found, users drop the image by releasing the index finger's touch input (5).

Lastly, users can perform all these actions with gaze-touch only, in a two-handed device grip (Figure 6.20c). At look at the desired image, users touch down with the right hand's



Figure 6.20: Illustrating the input flexibility with gaze and touch: the same image drag & drop task (rows) can be performed with three different techniques (columns).

thumb — as the system detects the touch occurred quite offset from the gaze position, the touch is interpreted as indirect, for remote control of the image (1). While holding the image, the user attends to the month grid — and uses the holding hand’s thumb directly, to scroll (2) and select a desired month (3). Then, the user returns to the gallery that displays the images of the just selected month. Again, the holding hand’s thumb touches are used — but this time indirectly, to scroll the image gallery (4). Finally, users drag the image to a position, and release the image (5). Notably, the indirect dragging is implemented with a higher control-display gain, compensating for the small movement that the thumb can perform.

## 6.7 Discussion

In this paper we investigated how gaze integrates into multi-touch based tablet UIs, and presented techniques that specifically enable users to interact with the whole screen using the same hand that grips the tablet. We described three applications that demonstrate the utility of these techniques on tablet computers, and describe how gaze, touch, and cursor input can be designed for combined use on tablet UIs.

Our work shows that tablet interactions can be simplified through the introduction of gaze onto tablets. Our study of a gaze-enabled homescreen demonstrates this best: the simple task of selecting an application that required two hands, is now reduced to simple unimanual input. We also show that diverse interactions from precise (select hyperlink) to coarse (change tab) ranges can be supported with our browser application; in essence, a single touch position can interact with the whole browser. In addition to the examples in the maps and image gallery application, we find that there are numerous basic interactions on tablets that, with gaze and touch input, can become simpler to use with one hand, and require less physical effort.

Our applications highlight novel bimanual interactions that users can employ, whether in a two-handed grip or in a one-handed grip with a second freely available hand. Combinations include use of direct-touch and gaze-touch, direct-touch and cursor, or gaze-touch and cursor inputs. This is enabled by applying previously developed input shifting methods to tablet contexts [PACG14, PAC<sup>+</sup>15], as well as our CursorShift method that provides users with easy switching between cursor and direct-touch input. For each hand, users can switch between input modes, and their combination yields the ability to perform advanced compound tasks, such as image dragging while scrolling the gallery, zooming while holding a marker, or moving a cursor while scrolling a website.

Gaze can change interaction with existing manual interfaces, as our applications demonstrated how interaction with default direct-touch UIs could transform into unimanual, single-grip, thumb-only interaction. It is unclear whether transforming a direct UI to an indirect UI on the tablet is a step forward, or a step back. While in past, users mainly interacted with indirect input on desktop systems, now direct-touch mobile devices are in focus; although desktop computers are still used in productive environments. Rather than increasing the tablet's capabilities with indirect input which can be considered as counterintuitive, we strive for a balance between supporting gaze-enhanced indirect input and direct-touch input through novel techniques and modality combinations for tablets.

Our work comes with limitations regarding technical feasibility and user evaluation. The use of multiple interaction styles and techniques enables a rich extended input vocabulary on tablets. In our initial study, we found that users can successfully employ basic gaze and touch based interaction on a homescreen task in a fixed setting. However, our applications go beyond single-technique approaches and with it can increase complexity, and future evaluation is needed to better understand relevant factors such as performance, affordance, and learnability. Additionally, our techniques are currently limited to stationary tablet usage. Here issues include occlusion of the eye tracker's camera when users reach in with their hand, no support of tablet orientation change, and gaze inaccuracy. Eye trackers are becoming smaller and versatile, which made our dynamic interactions possible. Our study indicated that users can utilise gaze for large targets in a stationary

setup; for more precision users can employ our cursor techniques. Further study of the techniques are needed in order to tease out the right parameters, and to understand their feasibility in more natural tablet scenarios.

## 6.8 Conclusion

We presented an exploration of gaze and touch interaction on tablet devices to increase interaction possibilities with the hand that holds the device. We evaluated the interaction technique against the baseline of direct-touch, and found it is slightly slower but allows unimanual use of the device with less physical effort. We developed CursorShift, a technique that allows users to seamlessly switch between direct-touch input, the current status-quo interaction technique on tablets, and cursor input, the current status-quo interaction technique on desktop systems. We presented three applications that describe gaze and touch based interaction techniques and combined use of touch, gaze, and cursor inputs. Taken together, we show that through the design of compound techniques, we enable users to leverage single-grip, single-hand, and single-finger input for many of the current tablet interactions that require two hands.

# 7

## Discussion

The previous chapters have individually explored the possibilities of using gaze input with direct and indirect user input. Taking a step back, here we will first give a recap on each individual work, connected to the design space of direct and indirect interaction set forth at the onset of the thesis. Then, we reflect on specific design goals that have been found as particularly important for the investigated research space, and connect the work more generally to prior art in HCI.

### 7.1 Summary and Reflection

The following discussions provide a recap on each previous chapter, and reflect on individual findings from the papers considering the bigger picture of extending direct interaction with eye gaze.

#### 7.1.1 Gaze-Touch: Indirect Input by Combining Gaze with Multi-touch On the Same Surface

The first paper focused on enabling indirect input with a direct input device. The user's gaze direction is an ideal indicator of whether touch input should be direct or indirect, and we therefore explored the possibilities of using multi-touch indirectly, redirected toward the looked point on a display. We found that it is not simply using *touch on an offset position*, as there are many qualitative differences that afford interactive possibilities beyond touch. One of the main differences being how multi-finger input affects a target. With direct-touch, multiple fingers allow interaction with multiple targets on the screen, each finger possibly manipulating one target. But with gaze-touch, multiple fingers are mapped to one target that the user's gaze lies upon. This can be considered as a disadvantage of not supporting multi-point interaction, although we also showed variations of the technique to enable multi-point selection in specific cases (c.f. initial discussion in section 3.1, example techniques in section 3.6). But this can yield advantages as well: mapping multiple touch points to the gaze selected target allows multi-point manipulation of small targets that might otherwise be difficult to achieve with direct finger input;

and even if the target would be big enough, visually, users would not occlude the target with the (multiple) fingers or their hands on the target; and lastly, the user can rapidly ‘fire’ multi-touch gestures across the whole screen without requiring to move their hand to each respective screen positions.

These are some of the differences found out through the explorative design of techniques and applications, but it is also imminent to emphasize the similarities between direct touch and indirect gaze-touch. Similarities are as important as differences here, because the more similar gaze-touch is to direct-touch, the easier it is for users to learn the new technique. What is particularly similar here is that users can employ the same gestural input to perform the same actions, e.g. a tap will ‘click’ on a target, or a pinch-to-zoom will still map to the same continuous zoom or scaling task. As a result, the main points users need to learn are (1) to look at what they want to manipulate, and (2) to perform a touch gesture. Both of these points are natural, as obviously users look at the target most of the times anyway, and learning touch gestures is redundant as most users are now proficient with this input paradigm from using touch displays. What would need to be learned is that users do not need to physically reach out with their hand to explicitly select a target. Thus, rather than introducing additional steps, the gaze-touch technique eliminates a step from the direct-touch interaction paradigm. Simply: *gaze-touch = direct-touch - reaching out*.

### 7.1.2 Gaze-Shifting: Direct-indirect Input with Pen and Touch Modulated by Gaze

However, this would be too much of a simplification because, as stated above, there are also some obvious differences, mainly that touch is direct, i.e. intuitive to use and close to realworld interactions, and gaze-touch is indirect, less intuitive but then again enabling interaction across the whole display. Thus, rather than regarding both techniques as separate, it might be better to unify both input paradigms.

The second paper focused on this aspect, asking ‘*What if users can utilise both direct and indirect input?*’, i.e. enabling the best of both worlds. This resulted in the *Gaze-Shifting* technique that uses eye gaze for seamless shifting between direct and indirect input. The basic goal of this technique is to allow users to switch between both techniques with the least possible effort involved to trigger this switch. And to make sense of this technique, one needs to consider how people look during direct and indirect interaction. With direct input, users look closely where they touch — because one wants to see what’s happening with the object that is touched and manipulated. With indirect input, users don’t look at their hands, as the hands (usually) control a remote cursor. The cursor is in the user’s visual focus because it shows what, how, and where something is manipulated — but the hands, or more generally the user’s direct physical input, are not in visual focus as they act from remote. This is how users interact currently, and the implicit suggestion in this work is to take advantage of this pattern for direct and indirect interaction on UIs. Rather than using an input device for each input mode, e.g. using a touchscreen for direct input, and a mouse for indirect input, here it is suggested that any device can be used for both direct and indirect inputs, by using eye tracking. If the user looks at the target and physically reaches out for it, it is automatically processed as direct input. If the user looks at a target remote from their input (hands), it is processed as indirect input.

This is a technique that can work in general for any input device, but in this paper the focus lies on exploring this mechanism for pen and touch displays. Pen and touch are an ideal example, providing two distinct direct input modalities as well as the possibility for bimanual combination of the two. Naturally, the combination of direct and indirect inputs with the two modalities opens up a large design space — to be precise, it extends the existing design space by two — for a variety of pen and touch human-computer interactions. Many of these interactions consist of at least two tasks that are interleaved with each other. For example, task-duos include drawing with a pen *vs.* selecting the ink’s colour, scaling an object *vs.* scaling the whole canvas, drawing a line *vs.* refining the line, copy an object in one position *vs.* pasting it in another, and so on (c.f. Table 4.1 for the complete set of examples we have explored). These task-duos can be efficiently mapped to direct and indirect inputs, and gaze provides a seamless switch between each task. In the paper, many of these task-duos are discussed and implemented in detail, showing the general applicability of the simple concept to various task scenarios.

Gaze-Shifting also has implications on the design of menu interfaces that allow the user to configure their input tools, e.g. in drawing applications users can configure the pen tool (draw line, straight line, select, etc.). In these programs, the user can set the direct pen mode. With Gaze-Shifting included, application designers can consider not only configuration of the direct pen input, but also what mode users can access when they shift to using indirect pen input. Thus part of the paper also explored this topic. A meta interface was developed, where users can configure direct pen, indirect pen, direct touch, and indirect touch modes in the design program — providing full configuration of input, and seamless switching between four different modes. Further, the four modes’ configuration can be conducted by using Gaze-Shifting too. We designed four explicit menus placed at the borders of the screen, each for configuration of one of the input modes — when users look at the menu, any remote pen or touch input will indirectly interact with it.

Of course, this kind of overloaded modal interaction can complicate things up to a point that it might be considered unnatural, and consequently it is surely not the ideal way of introducing Gaze-Shifting interactions to novices. The examples in the paper can be regarded individually, each showing simple additions to direct input tasks, where their indirect counterpart aids the task that users conduct directly, or vice versa — but at the same time they show how all these interactions can be used in one application framework, if desired. Overall, the concluding remark of this work is that Gaze-Shifting can transcend direct and indirect interactions toward a unified task by making the shift between the two modes seamless. Making two one, i.e. from considering one *input = direct xor indirect* toward *input = direct or indirect (depending on eye gaze)*.

### 7.1.3 Direct-indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction

The third paper went beyond the idea of Gaze-Shifting where the user shifts between direct/indirect input. By using pen and touch in two hands, a user can issue simultaneous direct and indirect interactions. However, the system of the previous work did not support simultaneous inputs and we excluded this part. The third paper thus focused on the possibility of performing direct and indirect input at the same time, where one

hand manipulates directly, and the other hand employs gaze-directed indirect input. We leveraged the same technical setup for this work, using pen and touch modalities: users draw directly with the pen in their preferred hand, and zoom indirectly with touch with their nonpreferred hand. This has the potential to enable highly efficient bimanual input, where the user issues precise interactions with their dominant hand, and at the same time can navigate the canvas with relative pinch-to-zoom gestures that rapidly affect any position on the screen the user is looking at.

But equally important is that it solves a critical issue of direct bimanual interaction, that of spatial interference where both hands cannot interact in the same physical space. By making one hand's input indirect, this issue is resolved, as direct and indirect inputs do not physically interfere. One might argue that one could use completely indirect inputs from both hands, however, that would eliminate some of the important affordances of direct input that cannot be used with indirect, such as precise drawing with a pen. In the paper, we have empirically studied this technique in comparison to the baseline of direct pen and touch, and an additional new direct/indirect variant without eye gaze. This new variant made touch indirect by redirecting the input to the user's pen position; thus eliminating the issue of spatial interference (and also alleviating other issues such as occlusion and hand movement fatigue). The disadvantage of this technique is that touch gets coupled to the pen — thus users still need to move the preferred hand to the target of the touch gestures. Nonetheless, it represents a distinct technique that led to further investigations [PHPB17].

Our user study revealed an interesting finding that distinguishes both hybrid techniques (using gaze or pen redirection) to the direct pen and touch variant. The hybrid techniques were more accurate for zooming with pinch-to-zoom gestures. During a single gesture of direct pinch-to-zoom, accuracy degrades over time, i.e. the further one pinches their two fingers out, the more inaccurate it gets. This stands in contrast to the hybrid techniques, where the user can still refine the zooming position while zooming (by moving the pen, or looking with the eyes). Keeping an accurate zoom position is relevant, because when not zoomed at the right position, one might need to zoom back and forth in succession to arrive at their point of interest. It also provides another interactive capability dubbed *Dynamic Targeting*: with the hybrid techniques, users can intentionally change the zooming target during a single pinch gesture — thus reuse the same gesture to zoom in various regions without actually lifting the fingers from the touch display. The paper investigated this and other aspects such as parallelism and eye hand coordination, overall concluding that simultaneous direct and indirect inputs are a new bimanual input combination previously underexplored, but with the possibility to bring new interactive capabilities to pen and touch interactions.

#### 7.1.4 Gaze and Touch Interaction on Tablets

In summary, the previous three parts can be considered as a continuous development of an idea across a design space: using direct input indirectly, switching between direct and indirect inputs, and using direct and indirect inputs simultaneously. We have investigated these works from an input-theoretic perspective independent of the hardware, for direct/indirect touch inputs on interactive surfaces. At present however, most interactive surfaces are of smaller nature, i.e. the majority of multi-touch devices are now

smartphones and tablets. Our previous work showed the advantages in principle, but not yet on systems that are used today. For this reason, the fourth paper focused on the exploration of the direct and indirect input design space on tablets.

A tablet is smaller than the previously investigated surfaces, but big enough to enable gaze tracking through commodity eye trackers. This investigation luckily coincided with the progress of eye trackers, that at the time became very cheap (<\$100) and very small (<2cm width). Thus, with the technical side of it showing potential in near future, the fourth paper explored gaze based direct and indirect interaction on a handheld tablet device. First tests indicated that this new interaction technique can provide a whole new way of interaction with handheld devices unseen before. Or, considered from a different perspective, it solves one of the biggest usability issues of handheld devices: the necessity to use two hands to interact with it. Users need to hold the device with one hand, and use the other to interact with the direct touchscreen. This has not been considered as a significant problem (although it is an active problem in HCI [OC12, TCJD13, WHM12, WH14]), because handheld touchscreens are still relatively new. But, the problem becomes apparent when comparing tablets to other computing devices, such as laptops or desktop computers. Laptops have touchpads used by a single hand, desktop computers have mice used by a single hand — the other hand is free to be used, e.g. to drink a coffee, for scratching one’s head, or for simply not using the hand out of comfort reasons. Being able to interact with one hand is highly important, but tablets (or more generally, handhelds) require two hands: one for holding, one for input.

The issue of one-handed tablet interaction can be approached by using gaze and touch based interactions, by using the following interaction technique. The holding hand uses the thumb to issue touch inputs, that are redirected to the point the user is looking at. Thus, if the user can select any position on the screen with their eye gaze, and issue input with the holding hand, the other hand remains free for other things. In the forth paper, the focus lies on exploring the possibilities of this interaction. First, a user study compared direct-touch (two hands) vs. gaze-touch (one hand) for selecting apps in a homescreen, and although direct was faster, the majority of users preferred gaze-touch because the one-handed technique had significantly less physical effort. With direct touch, not only do users need to move their second hand across the screen, but also users need to hold the tablet tighter to counter the pressure of the interacting hand. Both of these issues were eliminated through gaze-touch interaction.

But of course, while the empirical study provided information about how generic selections might benefit from using gaze, it does not show how all the other beneficial aspects of multi-touch that go far beyond target selection of an app, can be supported by using gaze and touch inputs. We therefore continued our exploration on the tablet, but with a more applied focus by regarding how the variety of conventional multi-touch UIs on tablets can be extended with gaze and touch inputs. It was clear that apps with typical larger targets (to account for the fat finger problem [HB10]) are ideal for gaze selection (as Ware et al. concluded in 1987 [WM87]), but what about applications that provide small targets? With touch, users can perform a pinch-to-zoom gesture to increase the target size, but that is difficult with the holding hand because usually only one of the fingers (the thumb) is available. We therefore searched for a new technique that allows to precisely select targets, without over-complicating thumb input of the user. To accomplish this, an enhanced technique was considered that uses gaze and touch inputs for

interaction with a cursor. A cursor is used in desktop computers and allows for very precise interactions. On a laptop's touchpad, users can control a cursor with a single finger — using drag to move the cursor, and a tap to click — thus it would fit with thumb-only input while holding the tablet device. The issue is, however, that the cursor dragging gesture conflicts with another gesture that is elemental to tablet UIs. Usually, users scroll the UI by dragging (with the thumb, or any finger), a highly relevant technique. Thus, the problem has changed, instead of bringing a cursor to a tablet, the question is how one can enable both cursor and direct touch inputs in the same UI.

With this question in mind, the paper introduces the CursorShift technique that provides the means to enable cursor and direct-touch input. It is based on Gaze-Shifting, but now in combination with cursor input, that is distinct to the typical touch input model that is based on only two input device states (here we consider the three-state model [Bux90]). The technique is simple, though. To activate the cursor, users look at the desired position, and tap (with the thumb of the holding hand) to activate it. Then, the UI's input mode is switched to cursor control, meaning the thumb can now issue drag gestures to move the cursor. And also issue a click through a simple tap. Then, the UI's input mode switches back to direct touch — where the thumb scrolls the input. This simple mechanism can be considered as a hybrid input device, similar to a merged touchpad and touchscreen combination, where users fluidly switch between them on a tablet computer.

With CursorShift available, a tablet UI can principally support direct-touch, touchpad like cursor control, and the Gaze-Touch techniques. Inspired by the support of the three input modes, and the prior investigation on simultaneous direct and indirect input, further applications were developed to explore how they work in conjunction with each other. The user's free hand is ideal for direct touch inputs, and in addition (as pointed out in other work [WHM12]) the holding hand can indirectly support direct touch interactions. For example in map navigation, one can hold a marker using direct touch of the free hand, while using the holding hand's thumb to navigate the canvas. Similarly in an image gallery, users can select and hold an image with the free hand, and navigate the gallery to different folders using gaze + touch interaction with the thumb of the holding hand. These bimanual techniques are for special cases, whereas most typical tablet interactions are simple and a unimanual tap is sufficient. But it is noteworthy that gaze opens up new ways of simultaneous bimanual interactions beyond those investigated in the previous paper, that were hard to do otherwise as one hand is occupied with holding the tablet. Thus, gaze and touch interaction on a tablet not only enables unimanual interactions with the holding hand, but also bimanual interactions using both hands. In essence, the problem of the holding hand is that although it can touch on the screen close to where the hand grips the device, it cannot interact with the rest of the screen — when eye gaze replaces this hand's pointing, the holding hand's interactive capabilities are extended to the whole display.

## 7.2 Generalising Across Input Technologies

Overall, the research of this thesis showed a plethora of possibilities when direct input technologies are extended with indirect input, in short: direct as indirect, between direct and indirect, and direct and indirect at the same time. One category is missing to come

full circle in the design space: using indirect input as direct input. This is a different case, because it is difficult to imagine using an indirect input device, such as a mouse, for direct input. What would that be for? What would be straightforward is how gaze could enable both direct and indirect input modes with a mouse. Indirect is active when using the mouse, and looking at the display. Direct is active when using the mouse, but looking at the mouse (or at the hand that controls it). Then, the question is what the use is of direct mouse input; probably not much as a mouse is now established as indirect input device and it's kind of unimaginable to have significant changes in the future. But one can consider the whole range of input devices, and in essence, all can support both input modes. More examples are illustrated in Figure 7.1.

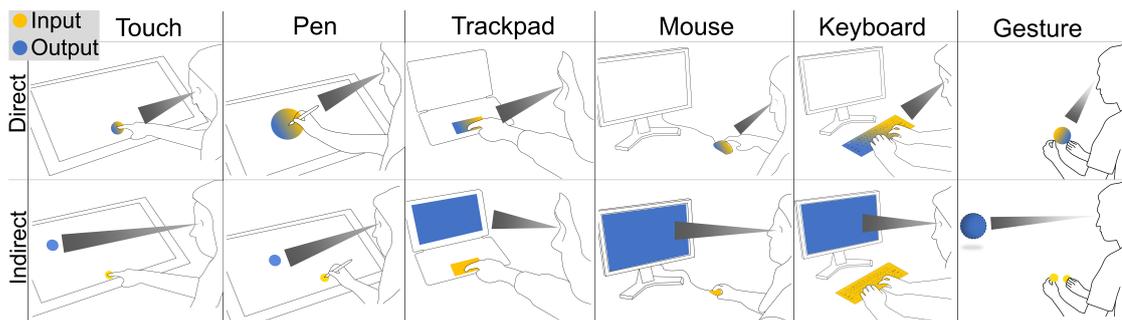


Figure 7.1: Example input technologies that could support direct and indirect input modes.

This raises the question when it makes sense to add an additional direct or indirect input mode to a device that was specifically designed only for one of the modes. It can make sense, as we showed with touch and pen inputs, but it is more difficult to imagine additional capabilities of indirect input devices (also the technological realisation requires more effort). Possibly, new input devices could emerge in future that are designed for the use of both input modes. These thoughts also go back to a more fundamental question of what we consider as direct and indirect inputs.

### 7.3 Understanding Direct and Indirect Input

How the HCI literature [FWC84, HW12] and also the research conducted in this thesis conceives of direct and indirect inputs might not be entirely accurate. During the course of the thesis it became apparent that one can consider a new and potentially more refined description of what these input modes mean. To start off with an example, imagine using your smartphone to watch a video. A typical video application displays the video in the center of the screen, and some buttons around it to control the video. Of particular interest in this example is a slider UI element, the *timeline*, that allows you to go to any time point of a video. Consider two example tasks: jumping to a specific moment, *vs.* scrolling through the video time to find a specific scene.

Example 1: If one wants to get to a specific time of the video, e.g. to 00:23 seconds, one can touch-tap at the (approximate) location on the timeline.

Example 2: If one wants to scroll through the video to find a scene, one would touch down on the slider, and move it along to see the whole video in rapid speed until the scene is found.

Both examples would normally be considered as direct-touch input, because (1) mobile touch displays inherently support the direct manipulation paradigm, (2) in both examples the user touches the slider and manipulates it, i.e. physically the input position equals the output position, resulting in the input being direct by definition in the literature [FWC84, HW12].

However, both examples could equally be considered as indirect touch input. Although direct/indirect inputs can be clearly defined by the spatial relation of input and output, it depends on what is really meant with output. If the output is the slider itself, and the user touches it, it is direct input. But one can also consider the video itself as the output, as the slider is just the means to interact with the video. From the user's perspective, the slider is simply a tool but the actual interaction task is to manipulate the video. If then, the output is the video, but the user is touching the slider that is not physically at the same position as the video, the input is indirect in both examples, according to the input/output definition.

As a consequence, both examples are direct input when considering the actual output as the slider, that makes sense as this is the UI control that the user manipulates directly. But both examples can be indirect input too, when the output is the actual video, thus the user indirectly manipulates the video through the slider. The latter makes sense from a more human-centered approach, as the video is the main interaction element in the examples, and not the slider.

The same can be said about any indirect input task. For instance a tv remote. This device would be classified as an indirect input device, because the remote indirectly manipulates the TV content, e.g. when scrolling through tv channels. However, this changes as well depending on the task. When one wants to press a non-typical button, e.g. to open the menu, the user would focus on the remote to find the button, until the user directly presses the button. This task is completely independent of the TV, and as the user physically presses the button with the corresponding functionality, it can be seen as direct.

As a result, do we conclude that there is no distinction between direct and indirect inputs, that any input could be considered as either of the input modes? Wouldn't it then void much of the research of this thesis, that investigated direct and indirect inputs?

Not when revisiting the definitions of direct and indirect interaction. As pointed out, the spatial relation of input/output alone is ambiguous, as 'output' is subject to interpretation, and with it changing the meaning of direct and indirect inputs. And the former, naive understanding, that 'touchscreens' are direct input, is not correct, but that was mentioned in earlier work too [HW12]. In order to make the input/output definition robust again, the 'output' needs to be clear, without ambiguities. Here we propose the user's visual attention, i.e. eye gaze, to clarify what is the output in the task's context. By using the user's visual attention to indicate the spatial position of interest, we can distinguish whether the current input on an interface element is direct or indirect.

Consider the above two examples. In example 1, the user wants to specifically jump to a time — thus the user's visual attention is on the slider, to find the approximate time position, and then touches on it. As said, the visual attention indicates the output. In the example, the user's attention lies on the slider, thus the output is the slider. If this is the case, and the user touches the slider, it is direct input.

In example 2, the user wants to find a specific scene of the video. Thus, the user keeps looking at the video while fast-forwarding on the slider with touch. The user's object of visual attention is the output: the video. For this reason, this input is indirect, as the user touches the slider, but the output is the spatially-distinct video.

To be exact, example 2, where one is searching a scene via fastforwarding through the video, might have been both direct and indirect input. At the beginning, when the user places their finger on the slider, the user is likely to look at it in order to find and touch it. As a result of the visual acquisition of the slider, the input is direct as the user simultaneously touches the slider. But then, when the actual task of searching the scene begins, the user is likely to look/watch the video to find the scene, and then at this time the touch input on the slider becomes indirect. This means that even within the same interaction session, the user can have multiple sub tasks that each can be either direct or indirect, on the same interface element.

Over time, an input on the same displayed UI element can be both direct and indirect.

At a time, an input is either direct or indirect input depending on what the user is (visually) focusing on.

One issue of this conception of direct and indirect input is when one could consider a user's input confidently as intentionally direct, or not. If the user looks at their fingers? or close to their fingers? How close? How far to be considered 'indirect'? Maybe, it needs to use other modalities to better determine user intent, possibly in combination with brain computer interfaces. Indeed there is some uncertainty in how efficient eye gaze is as the determinant for whether the input from someone was meant as either direct or indirect, revealing that there are cases that the above definition would not be correct. Nonetheless, as much as there is a consequential proneness to false-positive errors, there can be still many true cases especially for computer interfaces where information is constantly visually inspected out of the user's interest, making eye gaze a good indicator of what is the intended 'output' location, and thus allowing to distinguish between direct or indirect input.

In summary, it is a theory based on a small set of subjective observations, and future work is needed to clarify the definition of direct/indirect input, their implications on human-computer interactions, and the utility of this theory.

## 7.4 Reflection on Eye-tracking HCI Research

Eye-tracking hardware is continuously improving, but based on discussions with peers, writing papers and rebuttals, and received reviews, there seems to be unclarity of whether eye-gaze is really usable as an interaction modality. As always, it depends, there are always pros and cons to consider, any modality can be good for something, and bad for something else. While it is true that this technology has to be considered carefully, there are many aspects where one does not need a high level of caution, that can be simply taken for granted. The following points are based on experiences, studies and observations of people using eye gaze systems throughout the course of this thesis. These points are essentially a set of rules of thumb that emerged during the work, and also affected the research direction.

*Consider eye-gaze interaction as just looking.* The most basic component of eye gaze interaction is looking at a target. This is the easiest thing one can do, simply looking at any place in one's view. It seems that there is the perception that eye gaze interactions can be difficult, or straining to the eyes. This is untrue. At its most basic use of the eyes, pointing at a position on a screen, there is no problem for utilising one's eyes for this task. It can even be considered as the oldest interaction technique there is, as humans train 'looking' at objects early on in their life, earlier than use of their manual interactions.

*Consider eyes as highly accurate and precise.* Current eye trackers are susceptible to imprecise eye tracking, however, they are improving, and eventually it's possible to arrive at a point where the limit is the human's eye pointing capabilities. At times the research literature considers  $1^\circ$  of visual angle as the approximate accuracy possible [ADS05, ZMI99], because of the jittery movement of the eyes. Human eye jitter however is so small to remain unnoticed by a user; different fixational eye movements range usually from  $0.001^\circ$  (tremor) to  $0.5^\circ$  (microsaccade) [Rol09], that is clearly under one degree. So, if eye jitter is not the limit, what else? We believe that it is simply how precise one can see. In principle, an eye tracker could become as accurate as the user's capability to point with their eyes. And this capability is much more accurate than one might think: one can almost exactly look at the dot in the letter *i*, for instance. Depending on a screen's resolution, one can almost fixate a single pixel on a screen, which is indeed higher than  $1^\circ$  of visual angle.

*Use gaze only when already using gaze.* The key to using eye-gaze interactions is to take advantage of the cases where the user is already looking at a target, and thus not changing any of the user's visual behaviour. This has been pointed out by Jacob in 1993 [Jac93], but it's worth re-emphasising in context of this work. During manual input, the user often looks at an object and reaches to it manually. These are the cases to exploit — the use of eye tracking can eliminate reaching towards objects, and with it reduce physical fatigue. However, one has to be careful not to do the inverse. Requiring the user to look at a point that is not in the user's interest should be avoided if possible. Even dwell-time can be straining on the eyes, and become unnatural to the user. However, in some cases it is the only way of interaction, and then it's the best possible method. But in principle, one should exploit those interactions where users already naturally look at an object of interest.

*Extend hands with gaze, instead of gaze with hands.* There are many research efforts that explored multimodal gaze and manual input, from early works of Bolt and Jacob up to the current state of the art found in HCI conferences. The majority of such multimodal work put the eye gaze modality as the main focus in their work, and use a second modality to account for eye gaze limitations (e.g., to avoid the Midas Touch problem or to improve accuracy). There are many benefits of going this direction, such as bringing attention of the community to this highly useful modality. Yet, when one considers the other direction, considering the hands as the main modality, and eye gaze as a supplementary input, the research can be seen from a different perspective. The question changes, from improving gaze with manual input, to improving manual input with gaze. Gaze takes a complementary role, while hand based interactions, that are natural to the user, remain the main part of the interaction. This seems to be obvious considering that the hands are the main input modality in most digital and real interfaces, while gaze has not become a standard for the typical interface. However, much of the work sees eye gaze as the

main role in the context of a human-computer dialogue — but the other direction is as important.

*Focus on potential, instead of limitations.* Often, eye tracking research focuses on problems of eye gaze, but only little focuses on what's possible beyond those limitations. You can't be creative when being constantly restricted by multiples of factors. Research on limitations such as inaccuracy, midas touch, or eye fatigue are important to make the technology work. However, they seem to be the main focus, and if not, it seems that one always has to point out all the gaze problems in a paper. We haven't had a review that didn't point out the limitations of how inaccurate eye trackers are, for example. If everyone's thinking about that, it's not an issue if you look at something else. Sometimes we even think that if you don't ignore all the issues, there is no possibility to be creative. We'd to say it's as important to solve existing problems, to answer research questions, as it is to create new things, to raise new questions. Seeing things from a different perspective and creating new interactions can also provide new solutions to old problems. In our eyes, it felt like this is quite an issue in eye tracking research. Don't get us wrong, it is probably more important to solve the existing problems and make eye-tracking hardware working. However, many papers were plagued by discussions of gaze limitations, highly influential work started their papers by summarising all the issues of the technology (e.g. Zhai's MAGIC [ZMI99], or Stellmach and Dachsel's look & touch papers [SD12b, SD13], or Ashmore et al's fisheyes work [ADS05], to name a few). But sometimes it's good to follow up on different directions, independent of its potential for problem-solving.

*Consider eye gaze as meta input device.* Eye gaze can be considered as an input device like other devices (e.g. a mouse), as it allows to accomplish the fundamental task of pointing on displays; thus it makes sense to compare it to other input devices and techniques. However, when thinking independent of the technology, more from a human abilities perspective, then the user's eyes can be part of any hand based interaction of users, regardless of the input device used. While manual input significantly varies across devices (touchscreens, mouse, gesture, etc.), eye tracking can be principally employed side by side with all of the devices. For this reason, it is important to explore techniques that leverage eye-tracking as a meta input device across input devices and techniques with the potential to improve all user's manual interactions.

## 7.5 Generalising to Visual Attention Area Based Techniques

During the course of writing this thesis, we have found that the Gaze-Shifting mechanism shares many characteristics of Zhai et al's MAGIC technique that was introduced in his seminal paper in 1999 [ZMI99], although there are also clear differences.

What is similar is that both use a circular area that defines the user's eye gaze area, and depending on whether input is inside or outside of it, the systems reacts differently (Figure 7.2). Thus, two modes are supported with both MAGIC and Gaze-Shifting. With MAGIC, if the cursor is used within this area, it moves as normally. But if the user's eye gaze jumps to a different area, and the cursor is not within this area, the system would warp the cursor toward this area. Now consider Gaze-Shifting: if the touch happens within the user's gaze area, it is direct. If the touch happens outside the user's gaze area, it is indirect, thus the input is redirected toward the gaze area. In sum,

the basic system operation is the same, using the gaze area to distinguish different input types. This kind of mechanism can be generalised, in techniques that utilise the user’s visual attention area to provide various interpretations of input that are more suited to what the user is attending to, with the defining characteristic being the fact that input is within the user’s sight, or out of it. A notable similar approach that may belong to this class is Serim and Jacucci’s work on using this principle to provide varying degrees of visual feedback [SJ16].

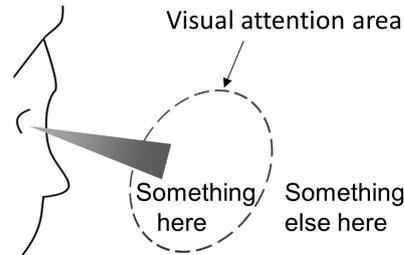


Figure 7.2: Generalised principle of utilising visual attention area as a technique.

There are also differences that distinguish the Gaze-Shifting technique from MAGIC, for example the historical background. MAGIC came at a time where all computer input was based on cursors, to interact with WIMP based user interfaces, i.e. the second era of computing when considering Weiser’s taxonomy. The work of this thesis was conducted during a time where multi-touch devices were predominant. Many differences exist between cursor and multi-touch input, with one of the main differences that multi-touch is direct input. This makes it particularly interesting to regard how one can provide complementary indirect interactions. In contrast, MAGIC was proposed to aid cursor interactions, and as cursor input is already indirect, the technique was mainly focusing on reducing the necessity of cursor movement, that is difficult as usually UIs minimise potential cursor use. However, beyond cursors, there can be more potential application of the principle idea as hopefully this thesis demonstrated to some extent, e.g. for bimanual interaction, context switching, or menu interactions.

## 7.6 Limitations and Challenges

There are some limitations regarding the more general interaction principles of using gaze and manual input, that need to be carefully considered during the design of the proposed user interfaces. Many of the detailed points have been discussed in the individual chapters, and we refer to the corresponding Discussion sections to get a full overview on the issues. Notably, learnability is a significant challenge as many of the proposed interactions introduce a fundamental change on the default manual interaction, i.e. users select by gaze, instead of manual. In our evaluations, we found users learned the techniques quickly with a few trials, however only with having an instructor. The challenge of teaching users how to interact with these novel ways with minimal instructions needs to be addressed in future.

One general issue that cannot be easily solved yet is how one defines how large the ‘visual attention area’ is, and therefore to decide when the mode switches between direct and indirect interaction. Of course, one can further extend this, provide multiple circular

areas for multiple modes, or use a continuous area somehow. In addition, the area should not be defined by a specific threshold in cm as we used, but rather visual angle from the user's view as a normalised method to account for various distances users can hold touchscreen mobile devices from themselves. However, at its core a threshold is needed to distinguish areas. Here it is unclear what is the ideal threshold, even how one could study this kind of parameter. One approach would be to study how users naturally look around during normal direct interactions, i.e. study eye-hand coordination, and then use all the non-direct 'remaining' area for the additional new interactions. However, if the user is aware about the possibility to utilise the technique, would they still be employing 'normal' eye-hand coordination anyway? Would potential interaction benefits lure users to change their ways slightly? How can one even describe these new concepts to a wider average-user audience? These are questions that remain open and subject to study, to be asked when the technical side of eye-tracking improved.

A challenge in the interaction design of techniques such as Gaze-Touch or Gaze-Shifting is the possibility to use clutching. Default touch interactions allow users a flexible use of clutching. Users only need to look at it once, and when the spatial location is known, they can issue touches eyes-free. Similar with cursor based input, users only need to place the cursor once to a position, and can issue multiple clicks or clutching eyes-free. This is different with gaze based touches. The technique requires users to look at the target at each selection, and to manipulate a target multiple times users need to remain looking at it. Right now, we consider this a compromise, the drawback of clutching stands against benefits of being able to select anything that users can see. As the user's eyes move at rapid speeds much higher than the hands, it is a significant benefit to interact with any target users look at — at the cost of the necessity of looking at the target each selection. Here we conclude that every technique is good for something, but also bad in some other cases. In the thesis, we pointed to many instances where the gaze based inputs can have advantages.

## 7.7 Future Work Pointers

The thesis research led to the development of a diversity of interaction techniques and use cases. While this shows the rich possibilities of the proposed interactions, it focused less on concrete, general directions on the implementation of a system to support such interactions. How can we enable a consistent interaction technique for users? As a starting point, we can recommend the usage of two principle interaction techniques as outlined initially in the propositions in the Introduction section; we also highlight two further ideas focus on particular UI artifacts:

- To cover our first proposition, users can be provided with an option in their system to use a direct modality as indirect modality — simply redirection of all touches toward the user's gaze (i.e., the Gaze-Touch technique, c.f. Section 3.3.3). This allows first user exploration and learning of the basic interaction idea, and is in principle easy to implement by a simple option in the operating system. It is also a feature that can work with any direct modality.
- Considering our second proposition, the seamless switching between direct and indirect input can be similarly implemented as an optional feature, independent of

the device and modality, because it only considers the distance of the user's manual touch and eye gaze position (c.f. Section 4.5).

- Menus are a ubiquitous UI element involved in most user interfaces. They are often located in remote areas of the UI. An optional indirect access toward those using gaze and touch can become useful for users as it eliminates the step of reaching toward it (c.f. Section 4.5.1 for pen devices, or section 6.4.2 for browser menus).
- A mouse cursor is the main input method to interact with WIMP style user interfaces. The general CursorShift technique (c.f. Section 6.2) can be integrated into operating systems that need to support both mobile (touch) and stationary (mouse cursor) interactions, by allowing users shifting between direct touch or cursor input.

# 8

## Conclusion

Current human-computer interactions are *either* direct or indirect, and eye gaze can relax this mutual exclusiveness and enable both direct and indirect input modes for any manual interaction. That a user's interaction is exclusively either direct or indirect simplifies the interaction, as it clarifies to the developer and user what does what, but at the same time it limits human-computer interactions to one or the other. What if all user input is guided by eye gaze so that both input modes are indeed supported?

This thesis explored this question for a small subset of interactions, those of multi-touch, pen, and gesture, but of course history shows that technology is not bound to specific inputs. As Mark Weiser pointed out, computers evolved from mainframes to the PC to the Ubiquitous Computing era, and each era is characterised by a distinct paradigm of interaction. Input to mainframes are provided through keyboard typing, input to the PC (based on WIMP) is provided by cursor pointing devices such as the mouse, and lastly the UbiComp era is characterised by pads, tabs, and boards, representing today's touch-enabled smartphones, tablets, and large displays [WB97]. While the keyboard might always remain for efficient typing tasks, the latter two input paradigms are indeed interesting to consider in light of our exploration of direct and indirect interaction. The cursor in WIMP UIs is usually controlled indirectly, most often by a mouse (or, on laptops, by a touchpad). The UbiComp interfaces are most often touchscreens, that merge the input and output spaces and thus enable direct input (Table 8.1). For this reason, exploring direct and indirect inputs can go beyond our investigated setup of touchscreens and gesture, toward connecting two major input paradigms of cursor control and direct touch, or connecting two UI types of WIMP and direct manipulation, or even connecting the two eras of computing.

It is not in the scope of this thesis to bring two eras of computing together, however, considering current computing devices, it does seem to be relevant to think about how to best bring WIMP and direct manipulation UIs together. For example laptops, nowadays supporting a plethora of input devices: a touchpad, touchscreen, keyboard, mouse, and there are also early prototypes that include eye tracking. Traditional operating systems such as Microsoft Windows have had difficulties over the years of how their system, optimised for cursor control by touchpad/mouse, can be used for multi-touch input that

	<b>User:computer relation</b>	<b>Interaction paradigm</b>	<b>Input device</b>	<b>Input type</b>
<b>Mainframe</b>	Many people share a computer	Command-line	Keyboard	Indirect (typing)
<b>PC</b>	One computer, one person	WIMP	Mouse and keyboard, touchpad	Indirect (pointing)
<b>Ubicomp</b>	Many computers share each of us	Direct manipulation	Touchscreen, stylus, gesture	Direct (pointing)

Table 8.1: Relationship between computing era, input devices, and input type.

was increasingly gaining traction. Laptops (and other devices) separate input; direct touch on the display, and indirect touchpad/mice as external device. Software-wise, the operating system needs to integrate both direct-touch inputs and indirect cursor based controls.

On the example of Microsoft Windows, many improvements were recently added to render the UI more touch-friendly. Windows 10 provides users with a mode switch, where users can switch the UI to a touch version that, e.g., provides larger UI elements to account for the fat finger problem. However, triggering explicit switches is usually less desirable as it comes with additional transaction costs. However, it is necessary to support touch input in order to bring the power of long-established WIMP based operating system to novel smartphone and tablet devices. However, some applications are ideal for cursor use, and others are better with direct touch. The problem is that those ideal for cursor input are hard to use on touchscreen devices, and those ideal for direct manipulation are hard (or feel unfit) to use on desktop devices. But it is clearly desirable to use an application on any device, whether it is on a desktop PC, or on a tablet. It would be nice to work on a task on a desktop PC using mouse and keyboard, and when travelling, one can continue the work on mobile devices.

For this reason, many applications also offer a direct touch adapted counterpart, to use on a touch device. However, as stated earlier, there are qualitative differences of using cursor and touch inputs, and simply translating an application to touch UIs comes with some kind of penalty. It would thus be desirable to use cursor input, i.e. enable WIMP-style interaction, on a mobile device as well — but at the same time need not be bothered about having to explicitly press a button to switch between both input modes. Here our CursorShift method of using eye gaze to switch between touchpad-like control and direct touch on a tablet is an ideal way to enable the support of both WIMP and direct touch applications on touch-enabled mobile devices. But, when taking a step back, it is not about enabling cursor control on touch devices, but more giving the user the choice for either direct or indirect input, at any time, with the least amount of transaction cost from switching. The user should be able to use direct touch to quickly navigate a menu, but then immediately use a cursor to precisely interact with and to reach any remote UI element, if needed. A future implementation of the Gaze-Shifting technique can allow such an interaction, if we would look beyond the near future that will be busy with making eye tracking usable and solving its limitations, toward a far-looking view into the future where the eyes are tracked during any human-computer interaction and provide users with seamless support of any manual direct and indirect control of digital devices.

If one would ever reach a point where eye tracking would be supported that way, and computer UIs allow for such seamless direct and indirect interactions as we explored along

the work in this thesis, it would be then interesting to see how these computer interactions would apply to real-world interaction. Real-world interactions are mainly direct, by using the hands to interact with physical objects. However, despite the technical challenges, is real-world direct manipulation also potentially extendible with an additional indirect input mode? For example, if you grab a cup of tea, it is quite likely that you look at it as well, thus the ‘system’ would clearly allow these kind of direct interaction. But, if you would intentionally look at a remote cup of tea, and perform an indirect grab gesture with your hand, would it be natural to then be in control of the remote cup? Now that seems far fetched for now and is not possible to study with current technology, but we started investigating gaze interaction in virtual reality showing that the combination of eye gaze and hand gestures can indeed provide new capabilities for freespace interactions [PMMG17]. In future, we aim to investigate technologies that can enable real-world interactions where the user’s hands are really extended by eye gaze. The realisation of such input technology, whether in digital devices or the real world, will require rethinking of many concepts of how we interact with computers. Current interactions are physically connected; i.e. users need to use the according input device to control a computer. With eye-tracking, in principle any user can point at any user interface at their sight and manipulate content, necessitating further development of methods of user access control, collaboration, and understanding of indirect object and UI manipulation beyond what’s currently possible.

## Bibliography

- [ABCD15] Carmelo Ardito, Paolo Buono, Maria Francesca Costabile, and Giuseppe Desolda. Interaction with large displays: A survey. *ACM Comput. Surv.*, 47(3):46:1–46:38, February 2015. [Cited on page 18]
- [ACVL14] Jeff Avery, Mark Choi, Daniel Vogel, and Edward Lank. Pinch-to-zoom-plus: An enhanced pinch-to-zoom that reduces clutching and panning. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 595–604, New York, NY, USA, 2014. ACM. [Cited on pages 63 and 69]
- [ADS05] Michael Ashmore, Andrew T. Duchowski, and Garth Shoemaker. Efficient eye pointing with a fisheye lens. In *Proceedings of Graphics Interface 2005*, GI '05, pages 203–210, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society. [Cited on pages 107 and 108]
- [ALMP09] Martha Abednego, Joong-Ho Lee, Won Moon, and Ji-Hyung Park. I-grabber: Expanding physical reach in a large-display tabletop environment through the use of a virtual grabber. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 61–64, New York, NY, USA, 2009. ACM. [Cited on page 39]
- [AZ03] Pär-Anders Albinsson and Shumin Zhai. High precision touch screen interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 105–112, New York, NY, USA, 2003. ACM. [Cited on pages 2, 19, 25, and 39]
- [B<sup>+</sup>07] Bill Buxton et al. Multi-touch systems that i have known and loved. *Microsoft Research*, 56:1–11, 2007. [Cited on page 1]
- [BB05] Anastasia Bezerianos and Ravin Balakrishnan. The vacuum: Facilitating the manipulation of distant objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 361–370, New York, NY, USA, 2005. ACM. [Cited on pages 20 and 39]
- [BBGV11] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. Pointable: An in-air pointing technique to manipulate out-of-reach targets on tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, pages 11–20, New York, NY, USA, 2011. ACM. [Cited on page 20]
- [BFW<sup>+</sup>08] Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, pages 154–161, New York, NY, USA, 2008. ACM. [Cited on pages 45, 59, and 64]

- [BG10] Andreas Bulling and Hans Gellersen. Toward Mobile Eye-Based Human-Computer Interaction. *IEEE Pervasive Computing*, 9(4):8–12, oct 2010. [Cited on page 78]
- [BGV10] Florian Block, Hans Gellersen, and Nicolas Villar. Touch-display Keyboards: Transforming Keyboards into Interactive Surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1145–1154, New York, NY, USA, 2010. ACM. [Cited on page 58]
- [BH00] Ravin Balakrishnan and Ken Hinckley. Symmetric Bimanual Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 33–40. ACM, 2000. [Cited on page 71]
- [BHR85] William Buxton, Ralph Hill, and Peter Rowley. Issues and techniques in touch-sensitive tablet input. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, pages 215–224, New York, NY, USA, 1985. ACM. [Cited on page 18]
- [BL00] Michel Beaudouin-Lafon. Instrumental interaction: An interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 446–453, New York, NY, USA, 2000. ACM. [Cited on pages 2 and 17]
- [BM86] W. Buxton and B. Myers. A study in two-handed input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '86, pages 321–326, New York, NY, USA, 1986. ACM. [Cited on pages 64 and 70]
- [BO09] Renaud Blanch and Michaël Ortega. Rake Cursor: Improving Pointing Performance with Concurrent Input Channels. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1415–1418, New York, NY, USA, 2009. ACM. [Cited on page 16]
- [Bol81] Richard A. Bolt. Gaze-orchestrated dynamic windows. In *Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '81, pages 109–119, New York, NY, USA, 1981. ACM. [Cited on pages 2, 3, 10, and 16]
- [BRT09] Andreas Bulling, Daniel Roggen, and Gerhard Tröster. Wearable eog goggles: Eye-based interaction in everyday environments. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 3259–3264, New York, NY, USA, 2009. ACM. [Cited on page 13]
- [Bul16] Andreas Bulling. Pervasive attentive user interfaces. *IEEE Computer*, 49(1):94–98, 2016. [Cited on page 3]
- [Bux90] William Buxton. A three-state model of graphical input. In *Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interaction*, INTERACT '90, pages 449–456, Amsterdam, The Netherlands, The Netherlands, 1990. North-Holland Publishing Co. [Cited on page 103]
- [BWB06] Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 1263–1272, New York, NY, USA, 2006. ACM. [Cited on pages 1, 2, 18, 19, 25, and 39]
- [CAG12] A. Cockburn, D. Ahlström, and C. Gutwin. Understanding performance in touch selections: Tap, drag and radial pointing drag with finger, stylus and mouse. *Int. J. Hum.-Comput. Stud.*, 70(3):218–233, March 2012. [Cited on page 1]
- [Car77] R. H. S. Carpenter. *Movements of the eyes*. Pion London, 1977. [Cited on page 9]

- [CK08] Han Collewyn and Eileen Kowler. The significance of microsaccades for vision and oculomotor control. *Journal of Vision*, 8(14):20–20, 2008. [Cited on pages 9 and 10]
- [DDLS07] Heiko Drewes, Alexander De Luca, and Albrecht Schmidt. Eye-gaze interaction for mobile phones. In *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology*, Mobility '07, pages 364–371, New York, NY, USA, 2007. ACM. [Cited on page 13]
- [DFAB98] Alan Dix, Janet Finley, Gregory Abowd, and Russell Beale. *Human-computer Interaction (2Nd Ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998. [Cited on page 7]
- [DHVE06] Connor Dickie, Jamie Hart, Roel Vertegaal, and Alex Eiser. LookPoint: An Evaluation of Eye Input for Hands-free Switching of Input Devices Between Multiple Computers. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*, OZCHI '06, pages 119–126, New York, NY, USA, 2006. ACM. [Cited on page 16]
- [DL01] Paul Dietz and Darren Leigh. Diamondtouch: A multi-user touch technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 219–226, New York, NY, USA, 2001. ACM. [Cited on page 19]
- [DS07] Heiko Drewes and Albrecht Schmidt. Interacting with the computer using gaze gestures. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II*, INTERACT'07, pages 475–488, Berlin, Heidelberg, 2007. Springer-Verlag. [Cited on pages 10 and 13]
- [DS09] Heiko Drewes and Albrecht Schmidt. The MAGIC Touch: Combining MAGIC-Pointing with a Touch-Sensitive Mouse. In *Human-Computer Interaction – INTERACT 2009: 12th IFIP TC 13 International Conference, Uppsala, Sweden, August 24-28, 2009, Proceedings, Part II*, volume 5727 of *INTERACT'09*, pages 415–428. Springer Berlin Heidelberg, 2009. [Cited on page 16]
- [Duc02] Andrew T. Duchowski. A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers*, 34(4):455–470, 2002. [Cited on page 2]
- [Duc07] Andrew Duchowski. *Eye tracking methodology: Theory and practice*, volume 373. Springer Science & Business Media, 2007. [Cited on page 2]
- [EEB67] William K English, Douglas C Engelbart, and Melvyn L Berman. Display-selection techniques for text manipulation. *IEEE Transactions on Human Factors in Electronics*, (1):5–15, 1967. [Cited on page 19]
- [Fal03] Daniel Fallman. Design-oriented human-computer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 225–232, New York, NY, USA, 2003. ACM. [Cited on page 7]
- [FDK12] Ribel Fares, Dustin Downing, and Oleg Komogortsev. Magic-sense: Dynamic cursor sensitivity-based magic pointing. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 2489–2494, New York, NY, USA, 2012. ACM. [Cited on page 16]
- [FFK13] Ribel Fares, Shaomin Fang, and Oleg Komogortsev. Can we beat the mouse with magic? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1387–1390, New York, NY, USA, 2013. ACM. [Cited on page 16]

- [FVB06] Clifton Forlines, Daniel Vogel, and Ravin Balakrishnan. HybridPointing: Fluid Switching Between Absolute and Relative Pointing with a Direct Input Device. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 211–220, New York, NY, USA, 2006. ACM. [Cited on pages 20 and 39]
- [FWC84] James D. Foley, Victor L. Wallace, and Peggy Chan. The human factors of computer graphics interaction techniques. *IEEE Comput. Graph. Appl.*, 4(11):13–48, November 1984. [Cited on pages 2, 17, 104, and 105]
- [FWSB07] Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 647–656, New York, NY, USA, 2007. ACM. [Cited on page 20]
- [FWT<sup>+</sup>17] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish S. Kulkarni, Shaun Kane, and Meredith Ringel Morris. Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '17, page To appear, New York, NY, USA, 2017. ACM. [Cited on pages 2 and 3]
- [GBL04] Yves Guiard and Michel Beaudouin-Lafon. Target Acquisition in Multiscale Electronic Worlds. *Int. J. Hum.-Comput. Stud.*, 61(6):875–905, dec 2004. [Cited on pages 64 and 65]
- [GHB<sup>+</sup>06] Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. Hover widgets: Using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 861–870, New York, NY, USA, 2006. ACM. [Cited on page 39]
- [Gui87] Yves Guiard. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. In *Journal of Motor Behavior*, volume 19, pages 486–517. Taylor & Francis, 1987. [Cited on pages 59, 62, and 64]
- [GWP12] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. Gripsense: Using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 545–554, New York, NY, USA, 2012. ACM. [Cited on page 18]
- [Han05] Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 115–118, New York, NY, USA, 2005. ACM. [Cited on page 18]
- [HB10] Christian Holz and Patrick Baudisch. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 581–590, New York, NY, USA, 2010. ACM. [Cited on pages 1, 25, 85, and 102]
- [HB16] Ken Hinckley and Bill Buxton. *Inking Outside the Box: How Context Sensing Affords More Natural Pen (and Touch) Computing*, pages 27–47. Springer International Publishing, Cham, 2016. [Cited on page 18]
- [HBPB12] Ken Hinckley, Xiaojun Bi, Michel Pahud, and Bill Buxton. Informal information gathering techniques for active reading. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1893–1896, New York, NY, USA, 2012. ACM. [Cited on pages 19 and 59]

- [HIV12] Aulikki Hyrskykari, Howell Istance, and Stephen Vickers. Gaze gestures or dwell-based interaction? In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 229–232, New York, NY, USA, 2012. ACM. [Cited on page 13]
- [HJ10] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(3):478–500, March 2010. [Cited on page 10]
- [Hla69] A. M. Hlady. A touch sensitive x-y position encoder for computer input. In *Proceedings of the November 18-20, 1969, Fall Joint Computer Conference*, AFIPS '69 (Fall), pages 545–551, New York, NY, USA, 1969. ACM. [Cited on page 18]
- [HMW<sup>+</sup>03] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Björn Eiderbäck. Technology probes: Inspiring design for and with families. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 17–24, New York, NY, USA, 2003. ACM. [Cited on page 7]
- [HNK<sup>+</sup>13] Eve Hoggan, Miguel Nacenta, Per Ola Kristensson, John Williamson, Antti Oulasvirta, and Anu Lehtiö. Multi-touch pinch gestures: Performance and ergonomics. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 219–222, New York, NY, USA, 2013. ACM. [Cited on page 26]
- [Hol07] David Holman. Gazetop: Interaction techniques for gaze-aware tabletops. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '07, pages 1657–1660, New York, NY, USA, 2007. ACM. [Cited on page 14]
- [HPGK94] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. A survey of design issues in spatial input. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST '94, pages 213–222, New York, NY, USA, 1994. ACM. [Cited on page 3]
- [HW12] Ken Hinckley and Daniel Wigdor. The human-computer interaction handbook. chapter Input Technologies and Techniques, pages 151–168. Taylor & Francis, 3 edition, 2012. [Cited on pages 2, 17, 39, 104, and 105]
- [HWO<sup>+</sup>13] Eve Hoggan, John Williamson, Antti Oulasvirta, Miguel Nacenta, Per Ola Kristensson, and Anu Lehtiö. Multi-touch rotation gestures: Performance and ergonomics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3047–3050, New York, NY, USA, 2013. ACM. [Cited on page 26]
- [HWP<sup>+</sup>14] Ken Hinckley, Andrew Wilson, Michel Pahud, Hrvoje Benko, Pourang Irani, François Guimbretière, Marcel Gavrilu, Xiang 'Anthony' Chen, Fabrice Matulic, and William Buxton. Sensing techniques for tablet+stylus interaction. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, UIST '14, pages 605–614, 2014. [Cited on pages 39 and 59]
- [HYP<sup>+</sup>10] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. Pen + touch = new tools. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 27–36, New York, NY, USA, 2010. ACM. [Cited on pages 45, 59, 62, and 64]
- [Iso00] Poika Isokoski. Text input methods for eye trackers using off-screen targets. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, pages 15–21, New York, NY, USA, 2000. ACM. [Cited on page 12]

- [Jac90] Robert J. K. Jacob. What you look at is what you get: Eye movement-based interaction techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 11–18, New York, NY, USA, 1990. ACM. [Cited on pages 2, 3, 11, 16, and 21]
- [Jac93] Robert J K Jacob. Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces. In *Advances in Human-Computer Interaction*, pages 151–190. Vol. 4, Ablex Publishing, 1993. [Cited on pages 37 and 107]
- [Joh65] E. A. Johnson. Touch display - a novel input/output device for computers. *Electronics Letters*, 1(8):219–220, October 1965. [Cited on page 18]
- [KAB<sup>+</sup>12] Jeong Ho Kim, Lovenoor Aulck, Michael C Bartha, Christy A Harper, and Peter W Johnson. Are there differences in force exposures and typing productivity between touchscreen and conventional keyboard? *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 56(1):1104–1108, 2012. [Cited on page 2]
- [KAB<sup>+</sup>14] Jeong Ho Kim, Lovenoor Aulck, Michael C Bartha, Christy A Harper, and Peter W Johnson. Differences in typing forces, muscle activity, comfort, and typing performance among virtual, notebook, and desktop keyboards. *Applied ergonomics*, 45(6):1406–1413, 2014. [Cited on page 2]
- [KAD09] Kenrick Kin, Maneesh Agrawala, and Tony DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proceedings of Graphics Interface 2009*, GI '09, pages 119–124, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society. [Cited on page 20]
- [KAR<sup>+</sup>14] Jari Kangas, Deepak Akkil, Jussi Rantala, Poika Isokoski, Päivi Majaranta, and Roope Raisamo. Gaze gestures and haptic feedback in mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 435–438, New York, NY, USA, 2014. ACM. [Cited on page 13]
- [KGH85] Myron W. Krueger, Thomas Gionfriddo, and Katrin Hinrichsen. Videoplace - an artificial reality. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '85, pages 35–40, New York, NY, USA, 1985. ACM. [Cited on page 18]
- [KI08] Masatomo Kobayashi and Takeo Igarashi. Ninja Cursors: Using Multiple Cursors to Assist Target Acquisition on Large Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 949–958, New York, NY, USA, 2008. ACM. [Cited on page 17]
- [KKP<sup>+</sup>08] Manu Kumar, Jeff Klingner, Rohan Puranik, Terry Winograd, and Andreas Paepcke. Improving the accuracy of gaze input for interaction. In *Proceedings of the 2008 Symposium on Eye Tracking Research and Applications*, ETRA '08, pages 65–68, New York, NY, USA, 2008. ACM. [Cited on pages 2, 11, 12, 33, 38, 79, and 83]
- [KPW07] Manu Kumar, Andreas Paepcke, and Terry Winograd. Eyepoint: Practical pointing and selection using gaze and keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 421–430, New York, NY, USA, 2007. ACM. [Cited on page 12]
- [Lan00] Chris Lankford. Effective eye-gaze input into windows. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, pages 23–27, New York, NY, USA, 2000. ACM. [Cited on pages 11 and 12]
- [LBS85] SK Lee, William Buxton, and K. C. Smith. A multi-touch three dimensional touch-sensitive tablet. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '85, pages 21–25, New York, NY, USA, 1985. ACM. [Cited on page 18]

- [LHGL05] Yang Li, Ken Hinckley, Zhiwei Guan, and James A Landay. Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 461–470, New York, NY, USA, 2005. ACM. [Cited on page 39]
- [LPW15] Christof Lutteroth, Moiz Penkar, and Gerald Weber. Gaze vs. mouse: A fast and accurate gaze-only click alternative. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 385–394, New York, NY, USA, 2015. ACM. [Cited on page 10]
- [LZB98] Andrea Leganchuk, Shumin Zhai, and William Buxton. Manual and cognitive benefits of two-handed input: An experimental study. *ACM Trans. Comput.-Hum. Interact.*, 5(4):326–359, December 1998. [Cited on pages 64 and 70]
- [MB02] Michael McGuffin and Ravin Balakrishnan. Acquisition of expanding targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 57–64, New York, NY, USA, 2002. ACM. [Cited on page 12]
- [MDK13] Michael Mauderer, Florian Daiber, and Antonio Krüger. Combining touch and gaze for distant selection in a tabletop setting. In *CHI 2013: Workshop on Gaze Interaction in the Post-WIMP World*. ACM, 2013. [Cited on page 14]
- [MI09] David C. McCallum and Pourang Irani. Arc-pad: Absolute+relative cursor positioning for large displays with a mobile touchscreen. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, pages 153–156, New York, NY, USA, 2009. ACM. [Cited on pages 20 and 39]
- [MIEL99] Elizabeth D. Mynatt, Takeo Igarashi, W. Keith Edwards, and Anthony LaMarca. Flatland: New dimensions in office whiteboards. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 346–353, New York, NY, USA, 1999. ACM. [Cited on page 20]
- [Min00] Darius Miniotas. Application of fitts' law to eye gaze interaction. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '00, pages 339–340, New York, NY, USA, 2000. ACM. [Cited on page 11]
- [ML04] Shahzad Malik and Joe Laszlo. Visual touchpad: A two-handed gestural input device. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, ICMI '04, pages 289–296, New York, NY, USA, 2004. ACM. [Cited on page 19]
- [MMAR06] Päivi Majaranta, Scott MacKenzie, Anne Aula, and Kari-Jouko Rähkä. Effects of feedback and dwell time on eye typing speed and accuracy. *Univers. Access Inf. Soc.*, 5(2):199–208, July 2006. [Cited on pages 2 and 11]
- [MO97] I. Scott MacKenzie and Aleks Oniszczak. The tactile touchpad. In *CHI '97 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '97, pages 309–310, New York, NY, USA, 1997. ACM. [Cited on page 19]
- [MR02] Päivi Majaranta and Kari-Jouko Rähkä. Twenty years of eye typing: Systems and design issues. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, ETRA '02, pages 15–22, New York, NY, USA, 2002. ACM. [Cited on page 11]
- [MvM04] Darius Miniotas, Oleg Špakov, and I. Scott MacKenzie. Eye gaze interaction with expanding targets. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 1255–1258, New York, NY, USA, 2004. ACM. [Cited on page 12]
- [MYBM91] Allan MacLean, Richard M. Young, Victoria M. E. Bellotti, and Thomas P. Moran. Questions, options, and criteria: Elements of design space analysis. *Hum.-Comput. Interact.*, 6(3):201–250, September 1991. [Cited on page 7]

- [NBBW09] Miguel A Nacenta, Patrick Baudisch, Hrvoje Benko, and Andy Wilson. Separability of Spatial Manipulations in Multi-touch Interfaces. In *GI '09*, pages 175–182. CIPS, 2009. [Cited on page 45]
- [NVCV16] Joshua Newn, Eduardo Velloso, Marcus Carter, and Frank Vetere. Multimodal segmentation on a large interactive tabletop: Extending interaction on horizontal surfaces with gaze. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*, ISS '16, pages 251–260, New York, NY, USA, 2016. ACM. [Cited on page 14]
- [NWP<sup>+</sup>11] Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 177–186, New York, NY, USA, 2011. ACM. [Cited on page 65]
- [OC12] Dan Odell and Vasudha Chandrasekaran. Enabling comfortable thumb interaction in tablet computers: A windows 8 case study. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 56, pages 1907–1911. SAGE Publications, 2012. [Cited on pages 78 and 102]
- [PAC<sup>+</sup>15] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, Yanxia Zhang, and Hans Gellersen. Gaze-Shifting: Direct-Indirect Input with Pen and Touch Modulated by Gaze. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, UIST '15, pages 373–383, New York, NY, USA, 2015. ACM. [Cited on pages 76, 78, 85, 88, and 96]
- [PACG14] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. Gaze-touch: Combining gaze with multi-touch for interaction on the same surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 509–518, New York, NY, USA, 2014. ACM. [Cited on pages 39, 42, 61, 76, 78, 80, 88, and 96]
- [PAG15] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. Gaze+touch vs. Touch: What's the Trade-off When Using Gaze to Extend Touch to Remote Displays? In *Human-Computer Interaction – INTERACT 2015: 15th IFIP TC 13 International Conference, Bamberg, Germany, September 14-18, 2015, Proceedings, Part II*, volume 9297 of *INTERACT'15*, pages 349–367. Springer, 2015. [Cited on page 14]
- [PAG16] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. Gazearchers: Playing with individual and shared attention in a two-player look&shoot tabletop game. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia*, MUM '16, pages 213–216, New York, NY, USA, 2016. ACM. [Cited on page 14]
- [PHPB17] Ken Pfeuffer, Ken Hinckley, Michel Pahud, and Bill Buxton. Thumb and Pen Interaction on Tablets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page to appear, New York, NY, USA, 2017. ACM. [Cited on page 101]
- [PMI05] J. Karen Parker, Regan L. Mandryk, and Kori M. Inkpen. Tractorbeam: Seamless integration of local and remote pointing for tabletop displays. In *Proceedings of Graphics Interface 2005*, GI '05, pages 33–40, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society. [Cited on page 20]
- [PMMG17] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. Gaze + pinch interaction in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*, SUI '17, pages 99–108, New York, NY, USA, 2017. ACM. [Cited on page 114]

- [PWS88] R. L. Potter, L. J. Weldon, and B. Shneiderman. 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, pages 27–32, New York, NY, USA, 1988. ACM. [Cited on pages 2, 19, and 20]
- [Ray98] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3):p372 – 422, 1998. [Cited on pages 2 and 9]
- [Rek02] Jun Rekimoto. Smartskin: An infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 113–120, New York, NY, USA, 2002. ACM. [Cited on pages 1, 18, and 19]
- [Rol09] Martin Rolfs. Microsaccades: Small steps on a long way. *Vision Research*, 49(20):2415 – 2441, 2009. [Cited on page 107]
- [RWA96] Scott Robertson, Cathleen Wharton, Catherine Ashworth, and Marita Franzke. Dual device user interface design: Pdas and interactive television. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '96, pages 79–86, New York, NY, USA, 1996. ACM. [Cited on page 1]
- [RZ14] Milena Roetting and Olaf Zukunft. Don't touch that tablet: An Evaluation of gaze-based interfaces for tablet computers. IEEE, 2014. [Cited on pages 78 and 83]
- [SA00] Dario D. Salvucci and John R. Anderson. Intelligent gaze-added interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 273–280, New York, NY, USA, 2000. ACM. [Cited on page 16]
- [SAD<sup>+</sup>06] Anthony Santella, Maneesh Agrawala, Doug DeCarlo, David Salesin, and Michael Cohen. Gaze-based interaction for semi-automatic photo cropping. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 771–780, New York, NY, USA, 2006. ACM. [Cited on page 17]
- [SB90] India Starker and Richard A. Bolt. A gaze-responsive self-disclosing display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 3–10, New York, NY, USA, 1990. ACM. [Cited on page 15]
- [SBG09] Dominik Schmidt, Florian Block, and Hans Gellersen. A comparison of direct and indirect multi-touch input for large surfaces. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part I*, INTERACT '09, pages 582–594, Berlin, Heidelberg, 2009. Springer-Verlag. [Cited on page 21]
- [SD12a] Sophie Stellmach and Raimund Dachsel. Investigating gaze-supported multimodal pan and zoom. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 357–360, New York, NY, USA, 2012. ACM. [Cited on pages 13 and 33]
- [SD12b] Sophie Stellmach and Raimund Dachsel. Look & touch: Gaze-supported target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2981–2990, New York, NY, USA, 2012. ACM. [Cited on pages 2, 3, 13, 14, 22, 37, 58, 61, 76, 79, and 108]
- [SD13] Sophie Stellmach and Raimund Dachsel. Still looking: Investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 285–294, New York, NY, USA, 2013. ACM. [Cited on pages 13, 14, 27, 37, and 108]
- [SGL08] John Stasko, Carsten Görg, and Zhicheng Liu. Jigsaw: Supporting Investigative Analysis Through Interactive Visualization. *Information Visualization*, 7(2):118–132, apr 2008. [Cited on page 35]

- [SJ00] Linda E. Sibert and Robert J. K. Jacob. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 281–288, New York, NY, USA, 2000. ACM. [Cited on pages 2 and 11]
- [SJ16] Baris Serim and Giulio Jacucci. Pointing while looking elsewhere: Designing for varying degrees of visual guidance during manual input. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5789–5800, New York, NY, USA, 2016. ACM. [Cited on pages 17, 78, and 109]
- [SPS92] Andrew Sears, Catherine Plaisant, and Ben Shneiderman. Advances in human-computer interaction (vol. 3). chapter A New Era for High Precision Touchscreens, pages 1–33. Ablex Publishing Corp., Norwood, NJ, USA, 1992. [Cited on pages 1 and 18]
- [SS91] Andrew Sears and Ben Shneiderman. High precision touchscreens: Design strategies and comparisons with a mouse. *Int. J. Man-Mach. Stud.*, 34(4):593–613, April 1991. [Cited on page 19]
- [SSND11] Sophie Stellmach, Sebastian Stober, Andreas Nürnberger, and Raimund Dachsel. Designing gaze-supported multimodal interactions for the exploration of large image collections. In *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*, NGCA '11, pages 1:1–1:8, New York, NY, USA, 2011. ACM. [Cited on pages 13 and 14]
- [SXM<sup>+</sup>14] Julia Schwarz, Robert Xiao, Jennifer Mankoff, Scott E. Hudson, and Chris Harrison. Probabilistic palm rejection using spatiotemporal touch features and iterative classification. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 2009–2012, New York, NY, USA, 2014. ACM. [Cited on page 18]
- [SZ11] Gwanseob Shin and Xinhui Zhu. User discomfort, work posture and muscle activity while using a touchscreen in a desktop pc setting. *Ergonomics*, 54(8):733–744, 2011. [Cited on page 2]
- [TAB<sup>+</sup>13] Jayson Turner, Jason Alexander, Andreas Bulling, Dominik Schmidt, and Hans Gellersen. Eye Pull, Eye Push: Moving Objects between Large Screens and Personal Devices with Gaze & Touch. In *INTERACT'13*, pages 170–186. Springer, 2013. [Cited on pages 3, 13, 22, 27, and 37]
- [TABG15] Jayson Turner, Jason Alexander, Andreas Bulling, and Hans Gellersen. Gaze+rst: Integrating gaze and multitouch for remote rotate-scale-translate tasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 4179–4188. ACM, New York, NY, USA, 2015. [Cited on pages 13, 61, and 78]
- [TBAG13] Jayson Turner, Andreas Bulling, Jason Alexander, and Hans Gellersen. Eye drop: An interaction concept for gaze-supported point-to-point content transfer. In *Proc. of the 12th International Conference on Mobile and Ubiquitous Multimedia (MUM)*, December 2013. [Cited on page 13]
- [TBAG14] Jayson Turner, Andreas Bulling, Jason Alexander, and Hans Gellersen. Cross-device gaze-supported point-to-point content transfer. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14, pages 19–26, New York, NY, USA, 2014. ACM. [Cited on pages 13, 27, and 37]
- [TBG11] Jayson Turner, Andreas Bulling, and Hans Gellersen. Combining gaze with manual interaction to extend physical reach. In *Proceedings of the 1st International Workshop on Pervasive Eye Tracking & Mobile Eye-based Interaction*, PETMEI '11, pages 33–36, New York, NY, USA, 2011. ACM. [Cited on pages 2 and 13]

- [TCJD13] Matthieu B Trudeau, Paul J Catalano, Devin L Jindrich, and Jack T Dennerlein. Tablet keyboard configuration affects performance, discomfort and task difficulty for thumb typing in a two-handed grip. *PLoS one*, 8(6):e67525, 2013. [Cited on pages 78 and 102]
- [TJ00] Vildan Tanriverdi and Robert J. K. Jacob. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 265–272, New York, NY, USA, 2000. ACM. [Cited on pages 2 and 21]
- [V<sup>+</sup>03] Roel Vertegaal et al. Attentive user interfaces. *Communications of the ACM*, 46(3):30–33, 2003. [Cited on pages 2, 3, 15, and 17]
- [VB07] Daniel Vogel and Patrick Baudisch. Shift: A technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 657–666, New York, NY, USA, 2007. ACM. [Cited on pages 1, 2, 20, and 39]
- [VB10] Daniel Vogel and Ravin Balakrishnan. Occlusion-aware interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 263–272, New York, NY, USA, 2010. ACM. [Cited on pages 1, 2, and 62]
- [VMSB15] Simon Voelker, Andrii Matviienko, Johannes Schöning, and Jan Borchers. Combining direct and indirect touch input for interactive workspaces using gaze input. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, SUI '15, pages 79–88, New York, NY, USA, 2015. ACM. [Cited on pages 17, 78, 80, and 83]
- [WB97] Mark Weiser and John Seely Brown. Beyond calculation. chapter The Coming Age of Calm Technology, pages 75–85. Copernicus, New York, NY, USA, 1997. [Cited on page 112]
- [WB03] Mike Wu and Ravin Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 193–202, New York, NY, USA, 2003. ACM. [Cited on pages 18 and 20]
- [WB09] Raphael Wimmer and Sebastian Boring. Handsense: Discriminating different ways of grasping and holding a tangible user interface. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI '09, pages 359–362, New York, NY, USA, 2009. ACM. [Cited on page 18]
- [WBP<sup>+</sup>11] Daniel Wigdor, Hrvoje Benko, John Pella, Jarrod Lombardo, and Sarah Williams. Rock & rails: Extending multi-touch interactions with shape gestures to enable precise spatial manipulations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1581–1590, New York, NY, USA, 2011. ACM. [Cited on pages 25 and 39]
- [Wel93] Pierre Wellner. Interacting with paper on the digital desk. *Commun. ACM*, 36(7):87–96, July 1993. [Cited on page 18]
- [WH14] Katrin Wolf and Niels Henze. Comparing pointing techniques for grasping hands on tablets. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '14, pages 53–62, New York, NY, USA, 2014. ACM. [Cited on pages 78, 83, and 102]
- [WHM12] Julie Wagner, Stéphane Huot, and Wendy Mackay. Bitouch and bipad: Designing bimanual interaction for hand-held tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2317–2326, New York, NY, USA, 2012. ACM. [Cited on pages 102 and 103]

- [WJF<sup>+</sup>09] Daniel Wigdor, Hao Jiang, Clifton Forlines, Michelle Borkin, and Chia Shen. Wespace: The design development and deployment of a walk-up and share multi-surface visual collaboration system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1237–1246, New York, NY, USA, 2009. ACM. [Cited on page 20]
- [WM87] Colin Ware and Harutune H. Mikaelian. An evaluation of an eye tracker as a device for computer input. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, CHI '87, pages 183–188, New York, NY, USA, 1987. ACM. [Cited on pages 10, 12, and 102]
- [WPHB16] Andrew M. Webb, Michel Pahud, Ken Hinckley, and Bill Buxton. Wearables as context for guard-abiding bimanual touch. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 287–300, New York, NY, USA, 2016. ACM. [Cited on page 18]
- [WSR<sup>+</sup>06] Mike Wu, Chia Shen, Kathy Ryall, Clifton Forlines, and Ravin Balakrishnan. Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces. In *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, TABLETOP '06, pages 185–192, Washington, DC, USA, 2006. IEEE Computer Society. [Cited on page 59]
- [WTP<sup>+</sup>95] J A Wise, J J Thomas, K Pennock, D Lantrip, M Pottier, A Schur, and V Crow. Visualizing the Non-visual: Spatial Analysis and Interaction with Information from Text Documents. In *Proceedings of the 1995 IEEE Symposium on Information Visualization*, INFOVIS '95, page 51, Washington, DC, USA, 1995. IEEE Computer Society. [Cited on page 35]
- [YCGS14] Dongwook Yoon, Nicholas Chen, François Guimbretière, and Abigail Sellen. RichReview: Blending Ink, Speech, and Gesture to Support Collaborative Document Review. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 481–490, New York, NY, USA, 2014. ACM. [Cited on page 59]
- [Yee04] Ka-Ping Yee. Two-handed Interaction on a Tablet Display. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 1493–1496. ACM, 2004. [Cited on page 59]
- [YHB<sup>+</sup>15] Dongwook Yoon, Ken Hinckley, Hrvoje Benko, François Guimbretière, Pourang Irani, Michel Pahud, and Marcel Gavrilu. Sensing tablet grasp + micro-mobility for active reading. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 477–487, New York, NY, USA, 2015. ACM. [Cited on page 18]
- [YMM<sup>+</sup>10] Xing-Dong Yang, Edward Mak, David McCallum, Pourang Irani, Xiang Cao, and Shahram Izadi. LensMouse: Augmenting the Mouse with an Interactive Touch Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2431–2440, New York, NY, USA, 2010. ACM. [Cited on page 58]
- [ZMI99] Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (magic) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 246–253, New York, NY, USA, 1999. ACM. [Cited on pages xi, 2, 3, 14, 15, 21, 25, 48, 107, and 108]
- [ZRZ08] Xinyong Zhang, Xiangshi Ren, and Hongbin Zha. Improving eye cursor's stability for eye pointing tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 525–534, New York, NY, USA, 2008. ACM. [Cited on pages 11, 12, 25, 58, and 76]

### .1 Gaze-Shifting: Direct-indirect Input with Pen and Touch Modulated by Gaze

This appendix provides additional documents that helped to conduct user studies of this thesis. Three chapters of this thesis included studies that use additional documents in form of questionnaires.

This part of the appendix shows the questionnaires that aided the study of Chapter 4. The study was informal, and the questionnaires were used to gather qualitative user feedback. In particular, the study consisted of two questionnaires.

1. The first questionnaire allows the collection of the user's demographic information, and the user's experience with parts the tested system.
2. The second questionnaire specifically asks about the user's performance in the three tasks of the study. This involved six Likert scale questions, and then additional open-ended questions.

---

# DEMOGRAPHIC INFO

---

Gender: Male / Female

Age: \_\_\_\_\_

Eyeglasses: Yes / No / Contact Lenses

Hand: left-handed / right-handed

	None	Little experience	Moderate	More than adequate	Expert
Experience with multi-touch interaction	1	2	3	4	5
Experience with digital pens (stylus)	1	2	3	4	5
Experience with combined stylus+touch	1	2	3	4	5
Experience with eye gaze interaction	1	2	3	4	5

Please specifically rate the **switching between direct and indirect** in this task.

*Direct*: manipulate what you touch.

*Indirect*: manipulate what you look, from remote.

		Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	<b>Ease</b> : The switching was easy to use.	1	2	3	4	5
2	<b>Speed</b> : The switching was quick.	1	2	3	4	5
3	<b>Accuracy</b> : The switching was accurate.	1	2	3	4	5
4	<b>Learnability</b> : The switching is easy to learn.	1	2	3	4	5
5	<b>Mental ease</b> : The switching requires <b>no</b> mental effort.	1	2	3	4	5
6	<b>Usefulness</b> : The switching is useful.	1	2	3	4	5

1 How did you do the switching? Please explain the way you performed the switching.

2 Did you switch often? For what reason(s)?

3 What problems did you encounter when performing this activity? Please explain your strategy to solve the problems.

4 In general, was there something you particularly liked or disliked?

## .2 Direct-indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction

This part of the appendix provides the questionnaires used in the study of Chapter 5: a comparison of three bimanual zooming techniques. Three questionnaires are given, that were made by using Google Forms.

1. The first questionnaire gathers the user's demographic information.
2. The second questionnaire gathers information about the user's performance with each technique, and thus was handed to the user after each technique.
3. The third questionnaire was handed out after the study, and provided overall feedback from the user across the tasks and techniques tested.

# Demographics

\*Required

## 1. Participant number \*

---

## 2. Gender \*

Mark only one oval.

- Male  
 Female

## 3. Age \*

---

## 4. Eye sightedness \*

Mark only one oval.

- Glasses  
 Contact lenses  
 None

## 5. Preferred hand \*

Mark only one oval.

- Right  
 Left

## 6. Experience with multi-touch interaction \*

between 1 (no experience) and 5 (very experienced)

Mark only one oval.

	1	2	3	4	5	
No experience	<input type="radio"/>	Very experienced				

## 7. Experience with digital pen interaction (stylus) \*

between 1 (no experience) and 5 (very experienced)

Mark only one oval.

	1	2	3	4	5	
No experience	<input type="radio"/>	Very experienced				

**8. Experience with eye gaze interaction \***

between 1 (no experience) and 5 (very experienced)

*Mark only one oval.*

	1	2	3	4	5	
No experience	<input type="radio"/>	Very experienced				

---

Powered by  
 Google Forms

# Performance

\*Required

## 1. Participant number \*

---

## 2. Which task did you perform? \*

Mark only one oval.

Find 3 targets

Draw line

## 3. Which technique did you use? \*

Mark only one oval.

Zoom at touch

Zoom at pen

Zoom at gaze

Please rate the following categories.

## 4. EASE \*

The task was easy to perform with this technique.

Mark only one oval.

	1	2	3	4	5	
strongly disagree	<input type="radio"/>	strongly agree				

## 5. SPEED \*

I finished the task quickly with this technique.

Mark only one oval.

	1	2	3	4	5	
strongly disagree	<input type="radio"/>	strongly agree				

## 6. PRECISION \*

I finished the task precisely with this technique.

Mark only one oval.

	1	2	3	4	5	
strongly disagree	<input type="radio"/>	strongly agree				

**7. LEARNABILITY \***

The technique is easy to learn.  
*Mark only one oval.*

	1	2	3	4	5	
strongly disagree	<input type="radio"/>	strongly agree				

**8. NO EYE TIREDNESS \***

The task with this technique did not tire my eyes.  
*Mark only one oval.*

	1	2	3	4	5	
strongly disagree	<input type="radio"/>	strongly agree				

**9. NO PHYSICAL EFFORT \***

The task with this technique had no physical effort (hand, arm, shoulder, or neck).  
*Mark only one oval.*

	1	2	3	4	5	
strongly disagree	<input type="radio"/>	strongly agree				

# Ranking

\*Required

**1. Participant number \***

---

**2. Please rank the three techniques. \***

*Mark only one oval per row.*

	Rank 1 (favorite)	Rank 2	Rank 3 (least favorite)
Zoom in touch	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Zoom in pen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Zoom in gaze	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**3. Why did you choose this ranking?**

---

---

---

---

---



### 3 Gaze and Touch Interaction on Tablets

In this part, three questionnaires are presented that were used in the study reported in Chapter 6: comparing gaze and touch *vs.* touch only interaction on a tablet. The questionnaires were designed with Google Forms.

1. The first questionnaire collects the user's demographic and experience information at the beginning of the study.
2. The second questionnaire is given after each technique to assess the user's perceived performance with it.
3. The third questionnaire is post-hoc, where users rank the techniques and provide their rationale for the ranking.

# Demographics

\*Required

**Participant number \***

**Gender \***

- Male  
 Female

**Age \***

**Eye sightedness \***

- Glasses  
 Contact lenses  
 None

**Preferred hand \***

- Right  
 Left

**Experience with multi-touch interaction \***

between 1 (no experience) and 5 (very experienced)

1 2 3 4 5

No experience      Very experienced

**Experience with digital pen interaction (stylus) \***

between 1 (no experience) and 5 (very experienced)

1 2 3 4 5

No experience      Very experienced

**Experience with eye gaze interaction \***

between 1 (no experience) and 5 (very experienced)

1 2 3 4 5

No experience      Very experienced

Submit

*Never submit passwords through Google Forms.*

Powered by

This content is neither created nor endorsed by Google.

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

# Performance

\*Required

**Participant number \***

**Which technique did you use? \***

- Touch only  
 Gaze+touch

Please rate the following categories.

**EASE \***

The task was easy to perform with this technique.

1 2 3 4 5

strongly disagree      strongly agree

**SPEED \***

I finished the task quickly with this technique.

1 2 3 4 5

strongly disagree      strongly agree

**PRECISION \***

I finished the task precisely with this technique.

1 2 3 4 5

strongly disagree      strongly agree

**LEARNABILITY \***

The technique is easy to learn.

1 2 3 4 5

strongly disagree      strongly agree

**NO EYE TIREDNESS \***

The task with this technique did not tire my eyes.

1 2 3 4 5

strongly disagree      strongly agree

139

**NO PHYSICAL EFFORT \***

The task with this technique had no physical effort (hand, arm, shoulder, or neck).

1 2 3 4 5

strongly disagree      strongly agree

Submit

*Never submit passwords through Google Forms.*

Powered by

This content is neither created nor endorsed by Google.

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

# Ranking

\*Required

Participant number \*

Please rank the techniques. \*

	Rank 1 (favorite)	Rank 2 (least favorite)
Touch only	<input type="radio"/>	<input type="radio"/>
Gaze+touch	<input type="radio"/>	<input type="radio"/>

Why did you choose this ranking?

Never submit passwords through Google Forms.

Powered by

This content is neither created nor endorsed by Google.

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)