Enhanced Multi-Touch Gestures for Complex Tasks

by

Jeff Avery

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

Waterloo, Ontario, Canada, 2018

Examining Committee Membership

The following serve on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:

Gilles Bailly
CNRS researcher at the ISIR Laboratory, Sorbonne Université

Supervisor(s):

Edward Lank
Professor, Cheriton School of Computer Science, University of Waterloo

Daniel Vogel
Associate Professor, Cheriton School of Computer Science, University of Waterloo

Internal Member(s):

Edith Law
Assistant Professor, Cheriton School of Computer Science, University of Waterloo

Géry Casiez
Adjunct Professor, Cheriton School of Computer Science, University of Waterloo
Professor of Computer Science, Université de Lille

Internal-External:

Oliver Schneider
Assistant Professor, Management Sciences, University of Waterloo

Statement of Contributions

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

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

The following list serves as a declaration of the works included in this dissertation. This material is expanded and revised from the original publication.

Chapter 3:

> Jeff Avery and Edward Lank. Surveying Expert-Level Gesture Use and Adoption on Multi-Touch Tablets. In DIS '16: Proceedings from the ACM Conference on Designing Interactive Systems, New York, NY, USA, 2016. ACM.
> `http://doi.acm.org/10.1145/2901790.2901895`

Chapters 5 and 6:

> Jeff Avery, Mark Choi, Daniel Vogel, and Edward Lank. Pinch-to-zoom-plus: an enhanced pinch-to-zoom that reduces clutching and panning. In UIST '14: Proceedings from the ACM Symposium on User Interface Software and Technology, New York, NY, USA, 2014. ACM.
> `http://doi.acm.org/10.1145/2642918.2647352`

---

[1] `http://authors.acm.org/main.html`, retrieved: May 25, 2018

Chapter 7:

Jeff Avery, Sylvain Malacria, Mathieu Nancel, Géry Casiez, and Edward Lank. Introducing transient gestures to improve pan and zoom on touch surfaces. In CHI '18: Proceedings from the ACM CHI Conference on Human Factors in Computing Systems, New York, NY, USA, 2018. ACM.
http://doi.acm.org/10.1145/3173574.3173599

Abstract

Recent technological advances have resulted in a major shift, from high-performance notebook and desktop computers – devices that rely on keyboard and mouse for input – towards smaller, personal devices like smartphones, tablets and smartwatches which rely primarily on touch input. Users of these devices typically have a relatively high level of skill in using multi-touch gestures to interact with them, but the multi-touch gesture sets that are supported are often restricted to a small subset of one and two-finger gestures, such as tap, double tap, drag, flick, pinch and spread. This is not due to technical limitations, since modern multi-touch smartphones and tablets are capable of accepting at least ten simultaneous points of contact. Likewise, human movement models suggest that humans are capable of richer and more expressive forms of interaction that utilize multiple fingers. This suggests a gap between the technical capabilities of multi-touch devices, the physical capabilities of end-users, and the gesture sets that have been implemented for these devices.

Our work explores ways in which we can enrich multi-touch interaction on these devices by expanding these common gesture sets. Simple gestures are fine for simple use cases, but if we want to support a wide range of sophisticated behaviours – the types of interactions required by expert users – we need equally sophisticated capabilities from our devices. In this thesis, we refer to these more sophisticated, complex interactions as “enhanced gestures” to distinguish them from common but simple gestures, and to suggest the types of expert scenarios that we are targeting in their design. We do not need to necessarily replace current, familiar gestures, but it makes sense to consider augmenting them as multi-touch becomes more prevalent, and is applied to more sophisticated problems.

This research explores issues of approachability and user acceptance around gesture sets. Using pinch-to-zoom as an example, we establish design guidelines for enhanced gestures, and systematically design, implement and evaluate two different types of expert gestures, illustrative of the type of functionality that we might build into future systems.

Acknowledgements

I'd like to thank my Ph.D supervisors, Edward Lank and Daniel Vogel, for their guidance, mentorship and encouragement during my studies. This dissertation represents both the most challenging and rewarding task of my academic career, and I am grateful for their guidance and support through the process.

I'm also very fortunate to have had an outstanding committee. Géry Casiez and Edith Law provided detailed feedback early in the process, which greatly improved this thesis. I was also very fortunate to have had Oliver Schneider and Gilles Bailly on my defence committee, and I'm grateful for their support.

I'd like to thank the members of the HCI Lab at the University of Waterloo, for helping to create such a positive research community. I've grateful for your friendship and encouragement.

Finally, I'm thankful to my family, Sandra and Austin, and my parents Jeff and Judy. I couldn't have done this without you.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

> My general rule is that everything is best for something and worst for something else. The more diverse the population is, the places and contexts where they interact, and the nature of the information that they are passing back in forth in those interactions, the more there is room for technologies tailored to the idiosyncrasies of those tasks.
>
> - Bill Buxton [Buxton, 2009b]

The launch of the iPhone in 2007 shifted consumer interest towards smartphones and portable devices, and established multi-touch as the standard input modality for portable devices. Consumer demand quickly led to smartphones and tablets replacing desktop and notebook computers, and they often serve as primary computing devices for many users [Müller et al., 2015]. Smartphone ownership in the US grew from 33% in 2011 to 64% in 2015 [Smith, 2015], and is expected to reach 95% of the U.S. population in 2018 [Zenith Media, 2018, Matsa et al., 2018]. For many users, multi-touch is their primary input modality, and tapping and swiping on multi-touch screens has become as second-nature as typing and mousing was for the previous generation. Given this shift, we want to ensure that multi-touch devices are capable of efficiently handling the tasks that users expect them to perform, and that interfaces are sufficiently expressive to support this range of tasks. This requires usable interfaces, effective interaction techniques, and consideration of the efficiency of these techniques [1].

Smartphones have proven to be well-suited to specific, simple tasks that are not hampered by a small screen and restrictive input, such as listening to music or reading [Smith, 2011, Yarow and Goldman, 2012]. To support these types of tasks, interaction designers typically leverage a small set of common gestures (see Figure 1.1), along with common widgets, to handle input. However, it is unclear whether these simple gestures are sufficient to support complex tasks such as document editing or graphics manipulation, which place higher demands on a system, and typically require correspondingly sophisticated interfaces. In practice, when attempting to perform complex tasks, users often eschew multi-touch phones and tablets in favour of a more traditional form-factor computers, with keyboard, mouse, toolbars, menus, keyboard shortcuts and the familiar array of interface elements [Müller et al., 2012, 2015].

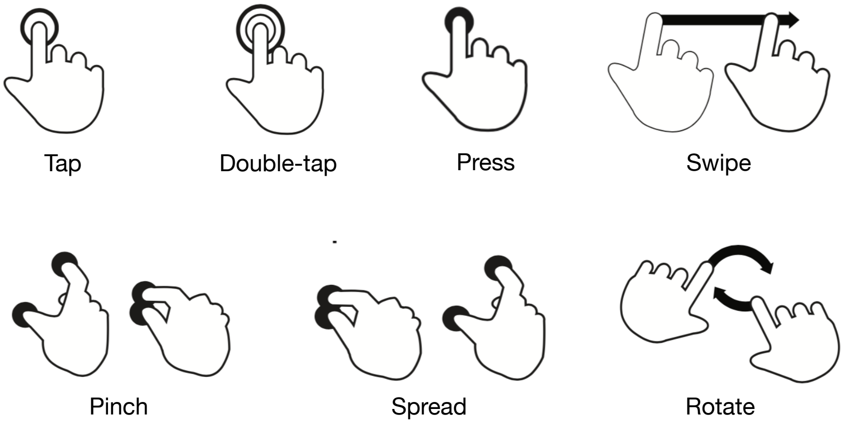


Figure 1.1: Common gestures

---

[1] **Usability** implies that actions or gestures that are supported in an interface are discoverable and easy to use. **Effectiveness** implies that our gesture design needs to be expressive enough to cover the range of required actions for the tasks that we want to perform. **Efficiency** refers to the ability of the user to execute a particular gesture with the minimum amount of wasted effort, energy or time. All of these are desirable characteristics. We discuss these design factors in Chapter 2.

In research, a distinction is sometimes made between novice interaction, targeted a new users, and expert interaction, targeted at more experienced users with more complex requirements. For the purposes of this thesis, we define **novice techniques** as *interaction techniques which are designed to favour simplicity and ease-of-use over efficiency* . By this definition, we might consider standard smartphone gestures to be novice techniques: easy to learn and use, and designed to ease adoption for new users.

By contrast, we can define **expert techniques** as *more efficient or capable versions of interaction techniques, which optimize capabilities and efficiency over ease-of-use, and which may require additional effort to learn to use effectively.* Expert techniques may be optimized versions of existing novice techniques, designed to improve efficiency, or they can offer new functionality in support of more complex workflows or features. A user's choice of interaction technique does not necessarily align with their level of experience. Relatively inexperienced users may dedicate the effort to learn an expert technique, while experienced users may continue to use relatively simple techniques, depending on the requirements of their particular task [2].

Although our devices are capable of supporting both novice and expert multi-touch techniques [Wobbrock et al., 2009], these devices often leverage a small, standardized set of novice gestures for most of their functionality. It is likely that this evolved as vendors sought to ease user adoption when these devices were first introduced, but has led to a relatively small interaction space, lacking more complex or expert gestures. I see this as an opportunity for enhanced interactions: there is a gap between what users might wish to accomplish on these multi-touch devices, and their ability to efficiently accomplish those tasks using existing gestures. If multi-touch devices truly are primary usage devices, we need to ensure that they are sufficiently capable, so their users are not excluded from particular categories of tasks, or forced to seek out alternate hardware.

Prior research has recognized this deficiency, and attempted to address it by introducing expert versions of standard gestures both to achieve greater efficiency over novice gestures, and to introduce new capabilities. For instance, more robust implementations of pinch-to-

---

[2] These definitions break the assumption that experience aligns with a particular choice of interaction technique, and matches previous definitions of these terms ([Lafreniere et al., 2017, Cockburn et al., 2014, Lane et al., 2005, Odell et al., 2004].

zoom have been suggested that improve efficiency, or offer expanded forms of interaction [Albinsson and Zhai, 2003, Bourgeois and Guiard, 2002, Hornbæk et al., 2002, Olwal et al., 2008, Malacria et al., 2010, Käser et al., 2011, Negulescu et al., 2011, Spindler et al., 2014]. Various approaches have also been proposed to ease the adoption of expert gestures by making these replacement gestures easier for users to discover [Bau and Mackay, 2008, Freeman et al., 2009, Tokárová and Weideman, 2013, Nacenta et al., 2013], or easier to learn [Grossman et al., 2009, Krisler and Alterman, 2008].

Despite their performance benefits, and efforts to encourage their use, these techniques have not been widely adopted. Research suggests that users rarely transition from novice to expert levels of technical expertise on their own [Carroll and Rosson, 1987, Lane et al., 2005]. In fact, prior work supports the notion that there is no direct correlation between experience level of a user and the efficiency of their actions [Rosson, 1983], and we cannot expect users to adopt more efficient gestures without encouragement. Standard gestures that work 'well enough' are difficult to replace when users already have established familiar ways of working; this is the paradox of the 'active user' who is too engaged in current tasks to spend the time to learn techniques that may pay off in the future [Carroll and Rosson, 1987].

Considerable research has explored ways to overcome this inertia, and transition users to expert use [Baher and Westerman, 2009, Scarr et al., 2011, Samp, 2013, Cockburn et al., 2014, Lafreniere et al., 2017, Grossman et al., 2007]. Scarr identifies three design goals [Scarr et al., 2011] that are essential in encouraging novice to expert-transitions: (i) maximizing the chance that the user will switch, by lowering barriers to adoption and making expert functionality discoverable; (ii) reducing the cost of the transition to the user, and (iii) enabling a high performance ceiling to reward the user for making the transition (and increasing the likelihood of subsequent transitions).

So, we cannot simply introduce 'newer and better' interaction techniques and expect results, but we instead need to identify and address barriers to adoption that may prevent users from transitioning to new techniques. This is not easy, since there is an obvious cost-benefit decision that the user has to address: will the benefits gained from switching to a new technique overcome the cost of investing time to learn that technique? Our design approach also has to directly address this issue for users.

Fundamentally, we can identify two specific approaches to increasing the capabilities of the gestural input language, without taxing the user:

- We can introduce more gestures, increasing our input vocabulary, and directly increasing the capabilities of our multi-touch systems. However, this puts the burden on the user to learn and adapt to these new gestures, and there are likely limits to the number of gestures we can reasonably introduce.
- Alternatively, we can explore enhancements to current gestures as a means of increasing our input vocabulary, while minimizing the adoption cost of the user.

In this work, we focus on the second option: augmenting and enhancing existing gestures. We will use the term **enhanced-gestures** to describe expert gestures that are *variants of standard gestures, designed to increase the efficiency or expressivity of gestural input languages by modifying the behaviour of familiar gestures in predictable, or learnable ways*. These enhanced gestures are intended to augment rather than replace standard gestures, and are designed to facilitate advanced functionality without fundamentally changing the input vocabulary. As Raskin notes, consistency in input language helps users build automaticity, and promotes trust in the interface [Raskin, 2000].

This approach also has the advantage of blurring the distinction between 'novice' and 'expert' users and techniques, and encourages design that reduces both the perception and actual cost of adopting more complex gestures. We recognize that users have *varying levels of expertise*, and even on a single device, a user may have specialized and advanced knowledge in one area of an application, but only basic skills in other areas. In this work, we treat expertise as a continuum, and promoting gestures that complement one another explicitly avoids the temptation to view an advanced gesture as an automatic replacement for a simpler gesture.

1.1 Thesis Statement

My thesis can be stated as follows:

> We can design enhanced gestures to improve the efficiency and expressivity of the input vocabulary for multi-touch interaction, by leveraging a user's familiarity with existing, standard gestures.

In this work, we examine the current multi-touch input space, and consider how to introduce enhanced gestures in a way that facilitates their adoption and use. We focus on *pinch-to-zoom* as a standard gesture that is widely known and commonly used on multi-touch devices, and explore the viability of creating enhanced-versions of pinch-to-zoom that are effective but also relatively easy for users to adopt. Unlike prior work that has focused on replacing standard gestures, our work focuses on seamless integration into the existing design space.

As part of this work, we also examine user attitudes and willingness to adopt enhanced gestures, and determine factors that may impede adoption. We establish design guidelines, which we use guide specific enhancements to pinch-to-zoom to address occlusion, clutching and other issues that impede performance. We introduce two interaction techniques, Pinch-to-Zoom Plus (PZP) and Transient Zoom (TZ), and evaluate their performance and usability compared to standard pinch-to-zoom. Finally, we examine how users learn to use our techniques, and provide insight into training and skill acquisition of these gestures.

1.2 Research Questions

These projects are intended to validate the design, use and performance of enhanced-gestures, specifically enhancements of pinch-to-zoom. In order, we attempt to address the following questions:

1. How can we promote the use of enhanced gestures?

- Do users identify themselves as novice or expert users? Does this identity correlate with use of extended gesture sets that currently exist?
- Is there a correlation between experience with a device, knowledge and use of an expert or enhanced technique?
- Does user awareness of gestures affect their likelihood of being adopted? Does increasing awareness increase the chance of adoption?
- Are there factors preventing the adoption of extended gesture sets and, by extension, enhanced gestures? If so, how can we overcome them?

2. Leveraging the specific example of one canonical multi-touch gesture, pinch-to-zoom, can we design effective enhanced versions of pinch-to-zoom that increase it's efficiency (either by correcting issues, or facilitating a specific workflow)?
    - What problems arise when users perform typical pinch-to-zoom tasks? Do other issues exist apart from precision, occlusion and clutching?
    - Do users exhibit common usage patterns in how they use pan and pinch-to-zoom to solve these tasks?
    - Can we design expert-gestures that overcome some of the issues that we uncovered? Specifically, can we design an expert-gesture that reduces issues with precision, occlusion, clutching and other identifiable issues?
    - Similarly, can we design a technique that reduces the time or effort required to navigate through multi-level documents?
    - Are these techniques effective compared to standard pinch-to-zoom, and would users be willing to adopt and use them?
3. Again, leveraging pinch-to-zoom as a specific case study, can users learn and become proficient with enhanced versions of pinch-to-zoom in a reasonable time-frame?
    - How long does it take to achieve proficiency and mastery with an expert-gesture?
    - How does performance with an expert-gesture change over multiple, repeated training sessions?

1.3 Contributions

We make the following contributions:

- An understanding of user perceptions of enhanced gestures, and identification of barriers to adoption of novel techniques.
- A design framework that facilitates transition to enhanced gestures by decreasing barriers to adoption.
- A foundational understanding of pan and pinch-to-zoom for typical multi-resolution navigation tasks.
- Realization and evaluation of two interaction techniques that offer significant improvements to standard pinch-to-zoom, and demonstrate the viability of designing across a continuum of user expertise.
- An understanding of the costs of gaining proficiency with a novel interaction technique.

1.4 Thesis Outline

The remainder of this thesis is structured as follows:

- **Chapter 2: Literature Review**. We review relevant literature and prior research, identifying issues around expert-use and multi-touch interaction that may impact our design.
- **Chapter 3: User Survey on Enhanced Gesture Use**. We survey experienced multi-touch users to determine their level of familiarity and their attitudes towards expert gestures. This will help us determine barriers to adoption that may need to be considered in gesture design.

- **Chapter 4: Design**. Motivated by our survey, we discuss the design space for pinch-to-zoom interaction, including the nature of pinch-to-zoom tasks, and the factors that we can consider when designing multi-touch gestures to address these tasks. We also discuss the issue of novice to expert transitions as they apply to learning novel multi-touch gestures.
- **Chapter 5: Observational Study of Pinch-to-Zoom**. We observe how pinch-to-zoom is used for navigation, and how users utilize its component actions (i.e. how pan, pinch and spread work together to achieve a task). This provides an opportunity to examine the context in which these gestures are used and identify inefficiencies that we can leverage.
- **Chapter 6: Design, Evaluation, and Longitudinal Study of Pinch-to-Zoom Plus**. Using the insights from the previous chapter, we design and implement an enhanced version of pinch-to-zoom that reduces the need to clutch, and improves occlusion by explicitly managing the area of interest. We evaluate the usability and effectiveness of the technique against baseline pinch-to-zoom in an evaluation study. Finally, we evaluate user performance over five successive days with Pinch-to-Zoom Plus to determine how learning and performance changes over a longer period of time.
- **Chapter 7: Design and Evaluation Study of Transient Gestures**: We examine the most common use for pinch-to-zoom, multi-level navigation, and add features to pinch-to-zoom to address efficiency and effectiveness when navigating through document levels. We explore the design of transient gestures, design a set of enhanced gestures that integrate into the standard gesture set, and evaluate the usability and effectiveness of our technique against baseline contextual zoom and pinch-to-zoom.
- **Chapter 8: Conclusions and Future Work**: Finally, we summarize our results, and recommend specific projects that extend this work.
- **Appendix A: User Survey from Chapter 3**.
- **Appendix B: Architecture and Design of Pinch-to-Zoom Plus and Transient Systems**.

Chapter 2

Literature Review

We are all humans first, beginners or experts second.

- Clifford Naas, as quoted by Jef Raskin [Raskin, 2000]

In this section, we examine the emergence of multi-touch technology, and the creation of standard gestures and gesture sets. We discuss known issues with multi-touch interaction, multi-scale navigation and pinch-to-zoom, and research that has attempted to overcome these concerns. Finally, we explore design issues, including tradeoffs between usability and efficiency, and the challenge of designing cohesive multi-touch gestures across different levels of user expertise.

2.1 Multi-touch Interaction

Consumer technology advances so quickly that it often feels as though new products and innovations appear from thin-air. Multi-touch interaction is often described this way, as if it was invented specifically for modern smartphones. In reality, it has a lengthy and convoluted history, spanning decades of research [Ion, 2013].

A number of early technologies were invented through the 1960s and 1970s that demonstrated the potential of touch interfaces. In 1965, E.A. Johnson invented the first capacitive touch-screen [Johnson, 1965, 1967], used in a commercial radar tracking system in the U.K. In 1972, a touch-screen was developed at the University of Illinois [Smith and Sherwood, 1976] that became part of the Magnavox Plato IV terminals. This system utilized a series of infrared sensors around a plasma display to provide a relatively high-precision touch-screen. A similar touch-screen was used in the HP 150, the first commercial touch-screen computer, released in 1983 [Ion, 2013]. In 1970, G. Samuel Hurst invented the resistive touch-screen at the University of Kentucky, then continued development at the Oak Ridge National Laboratory [Ion, 2013] before filing commercial patents [Hurst and Parks, 1972, Hurst, 1974]. The company that he founded, Elographics, continues to develop five-wire resistive technology, based on these patents.

Many researchers saw the potential to support richer forms of touch interaction that recognized an entire hand instead of a single touch point. To this end, multi-touch was invented by Nimish Mehta at the University of Toronto in 1983 [Kasday, 1984, Lee et al., 1985], using a frosted-glass panel with a rear-mounted camera. In 1985, other researchers in the Input Research Group, including Bill Buxton, extended this work into a capacitive touch tablet. In 1998, Elias and Westerman [Westerman, 1999] built a series of multi-touch keyboards and touch-pads using capacitive sensing technologies, similar to the technologies used in modern devices. Unlike earlier resistive screens, these systems had the ability to distinguish multiple *simultaneous* contact points, making them ideal for rich interaction, and potentially supporting multiple users on large screens.

However, despite these early innovations, commercial multi-touch systems remained uncommon until the early 2000s with the introduction of large-scale multi-touch displays (e.g. Perceptive Pixel's wall-size touch displays), and the introduction of the iPhone in 2007. The iPhone, in particular, helped shift consumer interest towards small, portable, multi-touch based devices [Dvorak, 2008]. In the years since then, consumer demand for traditional form-factor computers has declined, and smartphones and tablets have emerged as primary computing devices [Müller et al., 2015, Poushter, 2016], with smartphone ownership reaching 95% in the U.S. in 2018 [Zenith Media, 2018, Matsa et al., 2018]. For many people, smartphones have become their primary computing device, and multi-touch has become their dominant form of interaction. [West and Mace, 2010].

2.1.1 Standard Gestures

Standardization of gestures across tasks has long been a desirable goal, since it enables users to transfer their skills across different tasks, and provides an environment where they can practice and improve these skills. Around the time the iPhone was rising in popularity, there was significant research effort to design more complete and feature-rich sets of multi-touch gestures [Wobbrock et al., 2009, Morris et al., 2010, Kammer et al., 2010]. However, vendors focused on a more reduced set of simple gestures that were relatively easy to learn and master [Freeman et al., 2009]. This is likely intentional to encourage mass adoption of this technology by relatively inexperienced users [Norman and Tognazzini, 2015].

The standard across devices is currently a set of simple, cross-platform gestures. These gestures often are limited to one or two fingers, and are primarily triggered by taps, slides or chords (i.e. number of fingers on the screen, and possibly their relative position) [Harrison et al., 2014]. Table-2.1 lists some of the most common gestures found in both iOS [Apple Inc., 2015a] and Android [Google Inc., 2015]. These gestures are sufficient for a range of common tasks across different applications. For instance: single-finger tap will select or activate a widget in most applications, drag or swipe to scroll typically pans the underlying view, and pinch and spread zoom in or out of a view.

Given research history in this area, human capability for complex hand movement [Goguey et al., 2016, MacKenzie and Iberall, 1994], and the capabilities of these devices to support more complex interaction models [Apple Inc., 2015a, Google Inc., 2015], we know that we can support more complex gestures, and that users are capable of exploiting them. Applications will sometimes offer specialized gestures for specific, contextual actions (e.g. *Paper by FiftyThree*, a drawing application on iOS, uses rotation gestures to undo or redo an action). However, this level of complexity is uncommon, and to some degree, actively discouraged. For instance, Apple suggests that deviating from standard gestures can lead to ‘confusion and complexity’, but admits that unique multi-touch gestures, when used sparingly, can ‘enrich the experience in some apps, such as games and drawing apps’ [Apple Inc., 2015a].

Table 2.1: Common iOS and Android Gestures

Tap	Activates a control or selects an item.
Double tap	Zooms in and centres content, or zooms out if already zoomed in.
Press	Active submenu or secondary functionality.
Swipe	Moves an element, pans the view, or draws.
Pinch and Spread	Zooms in when pinching outward, zooms out when pinching inward.
Rotate	Rotate view or selected content.

2.1.2 Expert Gestures

Multi-touch interaction models have been proposed that expand gestures beyond simple one-and-two finger gestures [Forlines and Shen, 2005]. Wigdor et al. suggest ways to adopt curved hand positions [Wigdor et al., 2011] to encourage unique gestures. Wagner et al. suggest using hand and device posture to expand the design space for richer gestures [Wagner et al., 2012]. Ghomi et al.'s Arpège [Ghomi et al., 2013] defines a vocabulary of multi-touch chords, in addition to a feedforward technique to teach users chord gestures. Similarly, both Olafsdottir et al. [Olafsdottir and Appert, 2014] and Goguey [Goguey et al., 2018] explore usability and feasibility of expanding the multitouch design to include rich chording gestures. Wagner et al.'s BiPad [Wagner et al., 2012] explores ways in which we can leverage hand grips and device posture, and exploit targetable regions of the screen.

There are many research examples of gestures that expand the design space beyond what is available in commercial devices. Some commercial systems like Perceptive Pixel Multi-touch Collaboration Wall [Han, 2005], or the Microsoft PixelSense (Surface) tabletop system [Microsoft Inc., 2015] support a large number of simultaneous contacts, and are designed for bimanual or multi-person use. However, these complex gestures designed for these systems have not typically been scaled down to tablets or smartphones.

One commercial example of an extended gesture set that is available to a wide number of users is the extended gesture set that Apple has incorporated into their multi-touch input devices. Both iOS-based devices (iPhone, iPad) and macOS devices (multi-touch touchpads) have an extended set of multi-point gestures that serve to accomplish a series of tasks. For example, on iOS, gestures include swipe-up to access the control-center, swipe-left/right to switch between running applications, or pinch to return to the home screen. Similarly, on macOS touchpads, the set of multi-touch gestures includes swipe-left/right to switch applications, or pinch to return to the desktop. This similarity of input vocabulary across platforms is intentional: keeping the gestures consistent makes it easier for users to apply what they learn on one platform to the other [Raskin, 2000]. While these gestures exist and Apple may have data on how widely they are adopted, there is little publicly available research on the adoption rate for these gestures. We address this specific research question in Chapter 3.

Designers are often reluctant to introduce more complex or expert gestures, and will often instead use a combination of multi-touch surface gestures and on-screen widgets to support more complex functions. This means that a single multi-touch device may support multiple interaction styles, based on the needs and design of a given application. For instance, some applications choose to replicate WIMP functionality with standard widgets such as buttons, checkboxes and sliders. In this approach, touch interaction is meant to literally replace mouse interaction: dragging resembles a mouse-drag, touching the surface resembles a mouse-down event and so on [Baudisch et al., 2005]. However, this is an incomplete mapping: mouse-based systems support hover state or cursor positioning without actively selecting a target under the cursor. Current generation touch screens cannot support hover state, so touch-based applications often rely on explicit mode switches to mimic this extra state (e.g. selecting a tool from a palette to force a mode switch) [Hancock and Booth, 2004]. Mixing interaction styles can be seen as a compromise, to compensate for the lack of expressivity in multi-touch gestures.

2.1.3 Issues with Multi-Touch

2.1.3.1 Precision and Occlusion

As intuitive as direct, multi-touch input might seem, it suffers from some inherent problems. Fingers are poor input devices, and do not have the accuracy and precision of a mouse pointer or other precision pointing device [Wang and Ren, 2009, Benko et al., 2006]. While a mouse pointer can accurately be tracked to precise screen coordinates, fingers on a capacitive screen cover a relatively large area which needs to be mapped to a single point (typically by inferring the weighted centre of the centroid) [Potter et al., 1988, Westerman, 1999]. However, this is an approximation of the user's intended point of contact, and necessarily imprecise. Also, as Potter et al. point out [Potter et al., 1988], parallax effects related to the position and angle of the finger on the surface can impede accurate perception. For these reasons, this 'fat finger' problem makes fingertip selection and precise pointing difficult [Hinckley et al., 1998, Potter et al., 1988, Wang and Ren, 2009, Benko et al., 2006, Holz and Baudisch, 2011].

Early research by Pickering [Pickering, 1986] and Potter [Potter et al., 1988] identified precision problems across a variety of technologies and input strategies. Sears and Shneiderman [Sears and Shneiderman, 1991] noted that precision problems prevented wide-spread use of touch-screen systems. To address this, they introduced the notion of an Offset-Cursor [Sears and Shneiderman, 1991, Shneiderman, 1991], which places the cursor at a fixed distance from the user's finger. Although this avoids occlusion, the lack of correspondence between the visual feedback and contact point can lead to higher target acquisition times, and decreased precision as users need to iterate to find the proper position. Additionally, moving the cursor to a position relative to the contact point means that some areas of the screen remain effectively unreachable, since the offset cursor cannot be positioned without moving off-screen.

Albinsson and Zhai presented a series of techniques designed to allow pixel-precise selection [Albinsson and Zhai, 2003]. Cross-Lever presented two intersecting lines, which could be moved independently, with the intersection point representing the target. Although this technique afforded high precision, it was time-consuming to use. Another technique, Cross-Keys allowed selection using a targeting cross hair, which also increased precision also at the cost of task time. Additionally, cross hairs were found to be difficult to manipulate close to the screen-edge.

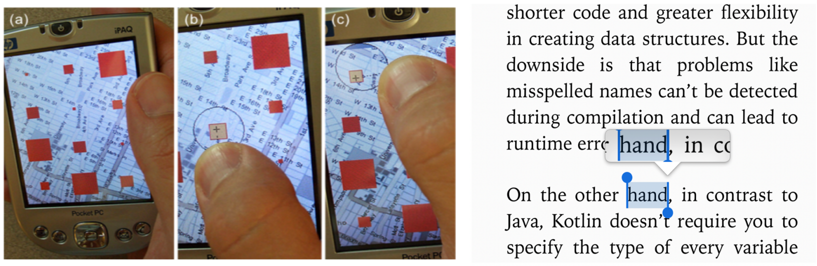
Other techniques attempt to manipulate the display space to make precise selection easier. Olwal and Feiner [Olwal et al., 2008] focused on fluid interaction with existing gestures. They introduced rubbing (Rub-Tapping) and tapping (Zoom-Tapping) to increase the scale of the target. Similarly, Käser designed Fingerglass to zoom a target area and improve selection accuracy [Käser et al., 2011]. These techniques addressed precision issues, but are designed to completely replace standard gestures, and required training to use effectively.

Any direct manipulation technique requires the pointer to be placed over the area of interest during target selection or acquisition. Direct touch interaction causes occlusion, where the act of directly interacting with a region blocks the users view of their target [Shneiderman, 1991, Vogel and Casiez, 2012]. This is a well-studied phenomenon, and there is a great deal of evidence to suggest that occlusion leads to greater user fatigue, inefficient movement and a greater number of interaction errors [Shneiderman, 1991, Hinckley and Wigdor, 2007].

There has been significant research to directly address occlusion issues. Approaches include adding a cursor offset to shift the cursor out from under the occluded area [Potter et al., 1988, Vogel and Balakrishnan, 2010], enlarging the target area to reduce the effect of the occlusion [Pahud et al., 2013], and translating content out from under the point of contact [Negulescu et al., 2011].

Potter and Shneiderman introduced a cursor-offset technique, Take-Off, to address high-level problems with multi-touch interaction [Potter et al., 1988]. Take-Off places a cursor slightly above the users finger, and that activates the target under the cursor when the user lifts their finger from the surface. However, using a fixed offset above the point of contact does not allow targeting near the top of the display. Vogel et al. take a similar approach with Shift [Vogel and Baudisch, 2007], which displays a bubble-like callout of the occluded area. This has the advantage of dynamically placing the callout at an area that is not occluded but still on-screen and visible, and overcomes the deficiencies of the earlier approach. However, intelligent manipulation of this callout requires knowledge of potential targets, which may limit its applicability to interfaces with explicit widgets and selection targets.

These problems and effects overlap, and techniques have been proposed that deal with one or more of these issues simultaneously: alternate techniques have been introduced to improve precision [Albinsson and Zhai, 2003, Baudisch et al., 2005], reduce clutching [Forlines and Shen, 2005, Käser et al., 2011, Malacria et al., 2010], and limit the impact of



shorter code and greater flexibility
in creating data structures. But the
downside is that problems like
misspelled names can't be detected
during compilation and can lead to
runtime errc hand, in cc

On the other hand, in contrast to
Java, Kotlin doesn't require you to
specify the type of every variable

(a) (b)

Figure 2.1: (a) Shift demonstrates an offset cursor that avoids occlusion [Vogel and Baudisch, 2007]. (b) iOS uses a similar mechanism to show highlighted text that would be otherwise occluded when the user selects it.

occlusion [Albinsson and Zhai, 2003, Vogel and Baudisch, 2007, Vogel and Balakrishnan, 2010].

2.1.3.2 Tracking and Hover State

Any attempt to replace mouse-input with touch-input also needs to address the lack of hover state inherent in touch [Buxton, 1990]. Mouse movement allows for separate tracking (i.e. moving the mouse) and activation (i.e. pressing the mouse button). Current generation touch-based systems can track movement, but cannot distinguish hover state from activated state. However, this is more a hardware limitation more than a design limitation, and promising research has enhanced touch-screen capabilities to sense above-surface motion [Mizumata and Sakamoto, 2010, Annett et al., 2011, Hinckley et al., 2016]. Pressure-detection is starting to appear in commercial devices like 3D Touch on Apple's iPhone [Apple Inc., 2018], or Force Touch on their touchpads, though it is rarely integrated into applications [Austin, 2018].

2.2 Multi-Level Navigation

Many tasks require users to examine content at different resolutions. Examples include map navigation, image manipulation, drawing tasks, and reading and annotating text.

To support these tasks effectively, we need to provide users with the ability to view the relevant data at multiple levels, and efficiently navigate between these different resolutions when required.

One common approach for dealing with multi-resolution data is to provide multiple simultaneous views of an information space: a high-level view of the entire space (overview), alongside a more detailed view, presenting a portion of the data at a higher scale (detail). Examples of this include text editors, and image editing applications. This overview+detail approach provides users with the ability to spatially navigate, and explore details while still keeping track of their position in the global information space [Cockburn et al., 2009]. This approach can be effective, but limits the user to a small number of fixed resolutions. Also, users are limited by the size of the view area.

```
    // For the checkbox
    public void onCheckboxClicked(View view) {
        // Is the view now checked?
        boolean checked = ((CheckBox) view).isChecked();

        // Check which checkbox was clicked
        switch(view.getId()) {
            case R.id.checkbox_morning:
                if (checked) {
                    // Add morning session
                } else{
                    // Remove morning session
                }
                break;
            case R.id.checkbox_afternoon:
                if (checked){
                    // Add afternoon session
                } else {
                    // Remove afternoon session
                }
                break;
        }
```

Figure 2.2: Multi-view displays are common. In Visual Studio Code, a document overview on the right-hand side is used for navigation, and mirrors the active code window.

A second approach is to provide a zoomable interface that allows users to change the scale or resolution of a single window to the desired level. Users can then dynamically change resolutions as required to view data at the appropriate level. Navigation in this type of interface requires panning and scaling the main window the navigate content. This approach is very common in devices with limited screen real estate, such as tablets and smartphones, and is commonly used for map navigation, web browsing and most other navigation tasks on those devices [Kurtenbach et al., 1997]. Pinch-to-zoom reflects this approach, where the pinch or spread actions are used to change the target resolution. This ability to quickly change resolutions is a critical factor in the appeal of the pinch-to-zoom gesture when working in navigation tasks.

(a) (b) (c)

Figure 2.3: (a) selecting an image to manipulate, (b) zooming out to an overview, and (c) zooming in to details.

Hornbæk and Bederson [Hornbæk et al., 2002] discuss user preferences for zoomable interfaces over alternatives such as outline views, and determine that users prefer zoomable interfaces for most tasks. Given a user preference for zoomable views, and the suitability of single-view navigation on small screens, the ubiquity of two-finger pinch-to-zoom on multi-touch tablets and smartphones is not surprising. Zoomable interfaces have the advantage of maximizing the use of screen real estate relative to overview+detail interfaces, and also map well to direct manipulation. Both of these factors are critical when designing interaction techniques for small-scale, multi-touch devices, and help explain the importance of pinch-to-zoom in multi-scale interaction.

2.2.1 Pinch-to-Zoom Gestures

Pinch-to-zoom is a two-finger gesture used to change the scale of the viewport, or the on-screen target. Pinch-to-zoom is performed by placing two fingers on the surface, typically thumb and forefinger of the dominant hand, and then pinching them together or spreading them apart. The standard implementation of pinch-to-zoom adjusts the document zoom level according to the change in distance between these two simultaneous touch points [Hinckley et al., 1998, Tran et al., 2013].

Pan is closely associated with pinch-to-zoom, since it is often used to reposition the viewport during, or after zooming. In fact, pan and pinch-to-zoom are so closely related, that it is often difficult to separate them: although users may pan without zooming, it would be unusual to zoom without some level of corrective pan being performed. For this reason, pinch-to-zoom is often implemented to allow simultaneous zoom and pan operations. We will discuss this further in Chapter 5.

Two-finger pinch-to-zoom has long been the standard technique for multi-scale navigation. Buxton [Buxton, 2007], in his essay on multi-touch systems, traces the history of the pinch gesture to the early 1980s [Lee et al., 1985]. Krueger's Videoplace supported the use of a two-finger pinch gesture to scale content as early as 1983 [Krueger et al., 1985]. Wellner's Digital Desk video from 1991 [Wellner, 1991] clearly demonstrates a similar gesture. Kurtenbach et al. demonstrate the use of a pinch gesture to zoom and rotate artwork [Kurtenbach and Buxton, 1991]. Hinckley et al. utilize a similar pinch gesture [Hinckley et al., 1998] in 1998 to zoom and pan around the centre of two contact points for map navigation.

Hoggan et al. [Hoggan et al., 2013a] examine the mechanics of pinch-zoom gestures, identifying some of the factors that contribute to performance, such as direction, distance, angle and position. They provide insight into which hand postures and positions are the easiest for users to achieve, and further provide significant design insights. Similarly, Berard determines the limits of hand spread during rotation, which could impact gesture design [Bérard and Rochet-Capellan, 2012]. One practical problem that emerges with dynamic zooming is that sometimes a target resolution cannot be achieved in a single pinch or spread, and multiple successive actions must be taken. This is similar to a user's need to lift and reposition a mouse when moving the mouse cursor a large distance. In this context, making a series of repeated actions to achieve a target is called **clutching**.

While interacting at multiple zoom levels is appealing to users, it can be inefficient due to the need to repeatedly clutch or continually zoom the document in and out [Lank and Phan, 2004, Negulescu et al., 2011]. Also, with repeated zooming, finger occlusion can make it difficult to keep track of the underlying target area, or the location of interest. It can move unpredictably while zooming, even off the screen forcing users to pan to correct the position. DTLens [Forlines and Shen, 2005] and Cyclostar [Malacria et al., 2010] attempt to eliminate this problem by supporting zooming without clutching. However, as inefficient as it can be, clutching is necessary to support the widest range of target resolutions.

2.2.2 Issues with Pinch-to-Zoom

Pinch-to-zoom also inherits well-known precision and occlusion problems. Precision issues are most evident when a user is attempting to select a specific area around which to zoom, typically the area between their two fingers. The lack of precision means that selecting the intended target is difficult, so successive attempts may be required. Also, since scaling operations are centred on the point of contact, the area of interest will be occluded during target selection, and remain occluded even after a zoom is performed. For this reason, users performing a pinch-to-zoom action will often need to zoom, and then perform a corrective pan to reposition the target so that it's visible. Effectively, the cost of precision and occlusion is *compounded* when performing a pinch-to-zoom gesture: the initial zoom target is difficult to accurately target, and remains occluded while the zoom operation is being performed. Fingers that occlude the area of interest also prevent any visual feedback, which prevents users from performing any corrective actions during the zoom.

Some techniques focus on eliminating these issues specifically during zooming and scaling. Albinsson and Zhai introduced Zoom-Pointing [Albinsson and Zhai, 2003] a bimanual technique in which the user draws a bounding box to define a persistent zoom area. Zoom-Pointing thus allows a user to specifically delineate the content they wish to see on-screen, which removes the need to pan after zooming to accurately position content. However, it is designed to work with a fixed resolution, and does not handle seamless dynamic scaling. DTLens [Forlines and Shen, 2005] is similar to Zoom-Pointing, but adds controls for minimizing, closing or annotating the enlarged viewport. It also allows users to save and restore zoom levels. Malacria et al. Malacria et al. [2010] introduced CycloZoom+ to reduce the cost of coordinated panning and zooming. Similarly, Benko et al. [Benko et al., 2006] suggest a dual-finger stretch technique that allows target selection and view scale manipulation with two contact points. However, these techniques require an explicit zoom-out to view context; switching between resolutions is difficult, and clutching is not supported.

Further research attempted to address the overview issue by introducing a transient zoom mode, where completing the gesture returns the scale back to the starting level. Fingerglass [Käser et al., 2011] allows users to specify a zoom-target using a non-dominant hand, then interact with the zoomed-in area using their dominant hand. When the non-dominant hand is released, the view returns to a default zoom level. As a transient technique, Fingerglass supported the idea of zooming-in to make quick changes before snapping-back out to the starting zoom level. The technique was well received, but users found that they were unable to make repeated zoom gestures in succession (they lost the ability to 'clutch', or zoom multiple times to find the right zoom level, something users typically do with Pinch). Negulescu et al. introduced Offset, a pinch-to-zoom replacement, to overcome occlusion issues by including an automatic pan as part of the zoom action [Negulescu et al., 2011]. This has the effect of moving the object of attention out from under an anchor point, and into the user's view. Offset is also a transient technique, so the resolution snaps-back to the original level after the technique ends. However, as with other techniques, the automatic zoom-level adjustment means that clutching is not supported.

2.3 Design Challenges

2.3.1 Usability, Effectiveness and Efficiency

When studying user interaction, it is natural to consider how well a particular interface or interaction works for a user. The term that is often used is *usability*. Neilsen [Nielsen, 1993] characterized usability in terms of issues of learnability, perceived ease-of-use, and efficiency - particularly for new users to a system who had not been trained to overcome its deficiencies. A usable system, by this early definition, should be both easy for a novice user to use to accomplish a particular task, and relatively efficient for accomplishing that task. Interaction techniques are the users methods for interacting with the system,and need to be designed with these simultaneous goals in mind. Both Neilsen and Shneiderman [Shneiderman, 1997b] suggested design factors to help make interfaces more usable: consistency and familiar appearance and controls; consistent and meaningful feedback to users; supporting the reversal of actions to facilitate exploration; including shortcuts or more efficient options for experienced users. Although there are many definitions for the term usability, all emphasize the need for consistent, approachable, easy to identify and activate interaction techniques. We want to support users becoming more proficient over time, so our interaction techniques also need to be memorable, or suggestive of the underlying task or function that they will perform. This aids initial discoverability, learnability, and later recall of the gesture [Nielsen, 1993]. For example, Harrison et al. have demonstrated that designing touch tools in a way that mimics familiar real-world tools aids learning and familiarity [Harrison et al., 2014, Norman, 2002].

We also want our interfaces and interaction to be *effective*. This implies that our interaction design needs to be expressive enough to cover the range of required actions for the tasks that we want to perform on our devices. It's not enough for an interaction technique to be usable, it needs to be useful and allow the user to achieve their specific goals with that interface. When designing multi-touch gestures, we cannot consider them in isolation, but should instead treat them as a set of actions that are used together to complete a task.

Finally, we also need to consider the *efficiency* of our techniques, or the ability of the user to execute a particular gesture with the minimum amount of wasted effort, energy or time. This is challenging, because efficiency can be at odds with features that aid usability. For instance, providing feedback during the activation of an interaction technique can make it easier to discover and learn, but may interfere with rapid performance of a technique. Designing for efficiency may result in a tradeoff, where we want to maximize usability while keeping acceptable performance levels [Norman, 2002].

For multi-touch interaction, we want gestures that meet all of these criteria. They need to be usable, effective for the tasks that we want to perform, and allow the user to perform that task in a reasonably efficient manner. We want to support sophisticated interactions, but retain simplicity, which is often challenging to achieve in a single design.

2.3.2 Discoverability and Feedback

Norman explored discoverability and understanding as two of the most important characteristics of good interface design. He characterized discoverability as the ability of a user to determine how they can interact with a system, while understanding supports their ability to determine the systems intended purpose [Norman, 2002]. Both of these mechanisms rely on providing feedback to users, either to suggest potential actions (affordances), or provide guidance to let the user know the results of their actions.

Discoverability is a challenge for multitouch systems in particular, since there is no standard mechanism to demonstrate or suggest how gestures should be used. Traditional WIMP interfaces support exploration: menu bars reveal available actions, and hovering over a widget frequently provides tooltips or hints on how that widget should be used. Gesture-based systems do not provide any visual cues about the functionality that they support, so users need to have other mechanisms to determine the gesture vocabulary. This makes discovering and learning gestures challenging, requiring users to either experiment to find gestures, use reference material, or ask for help in learning how to use a system. As Scarr et al. point out, this problem is actually worse with expert-gestures, since the tendency is to provide visual cues for novice users at the expense of hiding expert functionality [Scarr et al., 2011].

Feedback to aid in discoverability is difficult, given that gestures that have no visible interface and no simple way to show visual cues. Feed forward mechanisms, which prompt or demonstrate gestures to users, have commonly been suggested. Techniques like Octopocus [Bau and Mackay, 2008] and ShadowGuides [Freeman et al., 2009] have used both feed-forward and feedback techniques to add visualizations suggesting recommended gestures. LightGuide has a similar mechanism for in-air gestures [Sodhi et al., 2012]. Others, like GestureBar [Bragdon et al., 2009], use in-place demonstrations on the desktop to encourage users to practice and learn expert gestures. Anderson and Bischof [Anderson and Bischof, 2013] have suggested an adaptive guide that overcomes learning and retention issues in particular. However, while hints and guidance systems may be helpful for novice users who are learning to use a system, they may be intrusive for users that already know how to perform a gesture. This issue is also complicated by research that suggests that users find predefined gesture sets difficult to learn, and prefer user-defined gesture sets over supplied gestures [Nacenta et al., 2013, North et al., 2009].

2.3.3 Novice vs. Expert Use

Designing interaction techniques for a range of users is challenging: we want to encourage novice users who may be learning to use our system, but also provide more efficient or advanced functionality for expert-users, who may require training to use it effectively.

To address this challenge, we often design for different levels of expertise, offering multiple interaction techniques that provide similar functionality but target different skill levels. It is common to include both *novice interaction techniques* that are easy to discover and perform, and that provide immediate feedback to the user, and *expert interaction techniques* that require up-front training, but which enable higher performance [Lafreniere et al., 2017, Odell et al., 2004, Lane et al., 2005]. For example, commands in a graphical interfaces may be invoked through easy-to-discover menus, or through more efficient but difficult to discover keyboard shortcuts, which may require training to use effectively [Lane et al., 2005, Scarr et al., 2011]. Multi-touch interaction often tackles this problem in much the same way, offering a relatively inefficient gesture which is easy for users to quickly learn, alongside a more complex but efficient technique that requires additional training. This allows us to design a single system that supports users at different skills levels.

Over time, we generally want to encourage and facilitate the transition of users from inefficient, novice interactions toward more efficient expert-level techniques [Cockburn et al., 2014]. As designers, we often make the assumption that if we provide this expert functionality, users may start with the novice technique, but will naturally gravitate towards learning and using the more efficient expert technique over time [Lane et al., 2005, Cockburn et al., 2014]. Unfortunately, research has consistently demonstrated that there is no clear correlation between a user's experience level and the technique that they choose to use. An experienced user may choose to continue using a familiar but inefficient novice technique, despite more advanced techniques being available [Rosson, 1983, Carroll and Rosson, 1987, Lane et al., 2005]. Carroll and Rosson characterize this as the 'Paradox of the Active User', where current techniques are deemed 'fast enough' for the task at hand; the challenge lies in convincing users to invest time *now* for an efficiency tradeoff at some later date [Carroll and Rosson, 1987]. There are numerous examples of this in practice. Rosson et al. studied experienced users of XEDIT, a text editor, and found that there was no correlation between user experience and the likelihood of using keyboard shortcuts, an expert-technique [Rosson, 1983]. For many of their users, the initial way that they learned to perform a task was deemed 'sufficient', or they were 'too busy' to dedicate time to learning a new technique. Lane et al. found similar results with Word users and keyboard shortcuts [Lane et al., 2005], as did Bhavani et al. when studying users of CAD software who chose to not use advanced features [Bhavnani et al., 2001].

Scarr et. al. specifically examine unprompted transitions [Scarr et al., 2011], and suggest three factors inhibiting a user's movement to expert-modes: lack of knowledge about the availability, or the performance benefits of expert functionality; concern about the time or effort required to make the switch; novice performance being considered acceptable; and fears about the drop in performance that can occur when learning a new interface or technique [Scarr et al., 2011]. Scarr et. al. characterize this last factor, the drop in performance, as the 'dip', that inevitably occurs when learning a new technique. They suggest that the size of the performance dip that occurs after switching to a new interface will be influenced by the magnitude of the semantic and syntactic differences between the old and new interfaces [Scarr et al., 2011]. This is not abnormal, and does not indicate any deficiency in the technique: Gray et al. characterize dips and fluctuations as a standard part of user experimentation, and feel that trial-and-error is a necessary part of learning a new technique [Gray and Lindstedt, 2016]. Their observations align with Anderson's model of skill acquisition, which identifies cognitive, associative and autonomous stages of a users skill development as they learn to use, and internalize a particular skill [Anderson, 2000]. This is expanded by Huber [Huber, 2012] in Table-2.2, and is commonly used to frame plateaus in skill acquisition across a variety of domains.

Table 2.2: Three Stages of Motor Learning

Stage	Process	Characteristics	Other Name
Cognitive	Gathering information	Large gains, inconsistent performance	Verbal-motor stage
Associative	Putting actions together	Small gains, disjointed performance, conscious effort	Motor stage
Autonomous	Much time and practice	Performance seems unconscious, automatic and smooth	Automatic stage

Lafreniere et al. explore transitions to expert use in conjunction with user training, where users are aware of the expert techniques but have chosen to not use them. They suggest reasons why a user might deliberately choose to not use an expert technique: reduced performance requirements that suggest that the novice technique is adequate; task compatibility, where the novice technique is deemed to be more suitable for the given task; and risk aversion, where the user may be unwilling to risk errors with a more complex technique. [Lafreniere et al., 2017]. The authors characterize the choice of a particular technique as the result of a *deliberate decision* by the user, where they compare the merits of each technique and choose the one that they feel is suitable for their task or situation (i.e. sometimes leaning towards a familiar novice technique because performance is not important; or favouring the reduced error rate of a familiar novice technique). This suggests that, for users, efficiency is not always the most important factor when deciding on a particular strategy, and that promoting a gesture based on efficiency gains alone may not always be sufficient to encourage its adoption. This also suggests that awareness is important, but probably not sufficient by itself to motivate users to adopt and use expert-techniques.

Scarr et al. suggest three design goals that can facilitate the adoption of expert interaction techniques: first, maximize the likelihood that the user will *initiate* a switch to the expert modality; second, *minimize the cost* of doing so; and third, *enable a high* *performance ceiling* to rapidly reward the migration to the new technique. In their model, *transition cost is minimized by constraining the design of novel gestures to be semantically similar to their standard, non-optimized counterparts.* [Scarr et al., 2011]. In other words, by making an expert-gesture semantically similar to a novice gesture, we can reduce the cognitive cost of learning to use the expert-gesture. This suggests that expert-level gestures should be similar to their novice counterparts, and that subtle changes may be more used more frequently than radical redesigns in encouraging the adoption of a new technique.

A number of expert-level interaction techniques have been specifically designed to encourage this transition. Grossman et al. [Grossman et al., 2007], and Malacria et al. [Malacria et al., 2013b], suggested cues for keyboard shortcuts. Kurtenbach specifically developed Marking Menus to encourage novice to expert transitions [Kurtenbach et al., 1993], where the expert-use of the technique is a faster version of the novice technique. In this case, similarity of action means that users practicing the novice technique are also training for the expert-level version, which makes the transition to the expert-version relatively natural. These techniques appear to ease the adoption of expert-techniques to a point, but Gutwin et al. suggest that there may be additional barriers that limit the effectiveness of rehearsal in encouraging adoption [Gutwin et al., 2015].

It is difficult to determine at this time if the lack of expert-gestures in real-world applications is the due to user-choices, or lack of availability, or some other factors. In the next chapter, we will attempt to determine possible factors that could impede adoption of expert gestures.

Chapter 3

Surveying User Attitudes

Multi-touch interaction on mobile devices typically consists of taps and swipes - simple gestures that don't exploit the full range of technical and human capabilities. As we note in Chapter 2, some devices offer more sophisticated, expert-level gestures; however, it remains unclear both how widely these have been adopted and, if they have not been adopted, what factors might deter users from adopting these gestures. To study adoption of expert-level gestures, we examine the use of multi-tasking features on the iPad as representative of widely available, but optional, expert-level gestures. We survey a group of 106 iPad users about their device habits, awareness of expert-level gestures, and how they are used in everyday tasks. We determine that users have a fairly high awareness of expert-level gestures, and willingness to perform them, but often find them difficult to discover and challenging to learn. We conclude by discussing the implications of this for design.

3.1 Motivation

Users of modern multi-touch devices such as smartphones and tablets have a relatively high level of skill in using multi-touch gestures [Tuncer et al., 2015], but the multi-touch gesture sets used are typically restricted to a small subset of one and two-finger gestures, e.g. tap, double-tap, drag, and pinch-to-zoom. However, modern multi-touch smartphones and tablets are capable of accepting at least ten simultaneous points of contact [Apple Inc., 2015a, Google Inc., 2015], and human physical models suggest that humans are capable of richer and more expressive forms of interaction that utilize multiple fingers [Goguey et al., 2016]. This suggests a gap between the technical capabilities of multi-touch devices, the physical abilities of end-users, and the gesture sets that have been implemented for these devices.

There have been attempts to introduce more sophisticated interactions through the use of expert-level gestures: Apple provides trackpad-swipe gestures on their MacBook Pro [Apple Inc., 2015a], Microsoft has implemented bezel-swipes on the Surface Pro [Microsoft Inc., 2015], and numerous research projects have attempted to improve multi-touch functionality by adding targeted expert-level gestures [Ghomi et al., 2013, Malacria et al., 2010]. These include systems such as BiTouch [Wagner et al., 2012], Cyclostar [Malacria et al., 2010] and ZoomPointing [Albinsson and Zhai, 2003, Negulescu et al., 2011]. The Apple iPad is an example of a commercially successful multi-touch input device with an limited set of extended multitasking gestures [Apple Inc., 2015b]. These gestures can be considered to be expert gestures, since they are designed as replacements for multi-tasking functionality that is usually triggered via the home button. The iPad provides an opportunity to explore user perceptions about these commercially available gestures.

While expert-level gestures may be initially challenging to recall and perform, with appropriate time and training, user performance improves significantly [Lepinski et al., 2010]. However, despite the benefits and the availability of expert gesture sets, in observations of user interaction on past multi-touch platforms [Morris et al., 2010, Tuncer et al., 2015] it appears that these gestures are used infrequently. Users appear reluctant to adopt advanced, expert gestures Scarr et al. [2011], which means that promoting their use is challenging. In this chapter, we examine the use of three expert, direct multi-touch gestures on an iPad, which we consider fully realized expert gestures. We examine how often they are used among experienced users, and identify reasons why they might not be adopted.

3.2 Survey Design

To examine real-world usage, we designed and ran an anonymous survey on SurveyMonkey, where we recruited experienced iPad users and asked them questions about their device usage. First, we attempted to determine if end-users were already aware of the existence of these gesture sets. Second, we explored how frequently they used these advanced gestures. Finally, we explored user perceptions of the expert gesture set, and willingness to adopt them post-survey.

Overall, our goal is to validate the perception of slow adoption (to verify whether users really are slow to adopt such gestures) and, if so, to determine some of the causes for this slow adoption rate. For example, are new awareness mechanisms needed? Are gestures too complicated to learn or use? Do users perceive the current set of gestures as adequate and see no need for enhanced gesture sets?

A number of authors have attempted to describe the population of iPad users, but have typically focused more narrowly on characterizing application use [Coleman et al., 2010, Yarow and Goldman, 2012]. We are not aware of any attempts to determine how users interact with their iPads; this lack of data motivated our survey.

We chose an online survey as a data collection method for the same reasons proposed by Kjeldskov et al. [Kjeldskov and Paay, 2012]: external validity. Our goal is to collect responses related to software features used 'in-the-wild' during day-to-day interactions, which is difficult to accomplish in a controlled lab environment. A similar approach was taken by Snowdon et al. [Snowdon and Kray, 2009] when assessing a mobile map navigation technique.

In our survey, we focus on three specific expert-level direct-touch gestures on the iPad: App-Switching, Multitasking and Pinch-to-Home (see Figure 3.1). App-switching allows a user to switch to the next or previous running application. This is activated by swiping left or right with four fingers, for the previous or next running applications. Multitasking on the iPad allows a user to display a list of all running applications, and either switch to that application or close it. It can be activated by either double-pressing the home button on the iPad, or via an expert-level gesture, swiping-up with four fingers. Finally, Pinch-to-Home exits the currently running application and returns the user to their home screen. This can be accomplished by single-pressing the home button, or performing a four-finger clutch gesture, where the fingers placed down simultaneously, and then brought together to a midpoint.

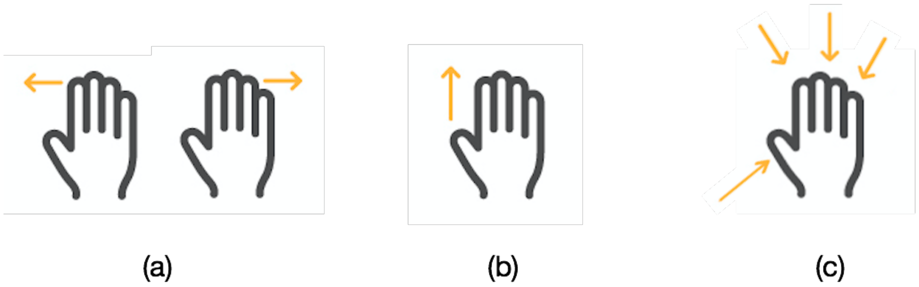


Figure 3.1: Apple expert-gestures: (a) four-finger left or right to switch between running applications; (b) four-finger swipe up to activate the application-switcher; (c) four-finger clutch to return home.

These gestures have the advantage of being currently deployed in commodity hardware. Apple has made these gestures available in the same form, across a large number of devices. These gestures are also intended to provide advanced, but optional, functionality. This implies that users that have adopted these gestures could have avoided them, but instead chose to learn how to use them.

3.2.1 Apparatus and Participants

Our survey was created and hosted on SurveyMonkey [SurveyMonkey Inc., 2015]. Data was collected in two sessions: Dec 1 2015, 3:13 to 4:45 PM EST, and Jan 13, 2016, 11:00 PM to Jan 14, 2016 10:31 AM. Data collection was spaced across two different time periods, to get a broad spectrum of respondents who may be using the devices in different contexts. A target number of participants was specified in each case, and SurveyMonkey automatically closed the survey once the target was reached for a given period. SurveyMonkey advertised the study ('task'), handled recruiting and managing participants, and supplied anonymized results after data collection was complete. We recruited 123 participants; 17 participants stated that they did not own an iPad and their responses were discarded. Our remaining 106 participants (55F, 35M, 16 unreported), had a mean age of 43.1 years (SD 15.4).

3.2.2 Structure

Our survey contained 35 questions across 5 sections. After answering the first question about device usage, participants were allowed to skip any of the remaining questions that they did not wish to answer. Section one asked demographic questions and general questions about iPad usage. The second through fourth sections of the survey asked questions about the three specific expert-level gestures of interest noted earlier: Multitasking, App-Switching and Pinch-to-Home.

For each expert-level gesture, we asked about a participant's familiarity with that gesture, and branched based on their response: if they had used a gesture previously, we asked about frequency of use, preference compared to non-gestural activation methods, and general satisfaction; if they were unfamiliar with a gesture, we explained how to use it, asked them to attempt it on their iPad, and then polled them on ease-of-use and and impressions. Finally, the last section of the survey asked about revised opinions of expert-level gestures.

This branching is a technique suggested by Muller [Müller et al., 2014, Müller and Sedley, 2015]; it is well-suited to online surveys, and allows us to vary questions based on level of expertise. In our analysis, we report both raw Likert scores, and significant results from Pearson's Chi-squared 'goodness of fit' tests performed against various measures. Unanswered questions are excluded from the results.

3.3 Results

We break the results of our survey into five separate categories. While these categories loosely follow the survey, they focus more specifically on our research questions. We first profile our users and then probe gesture awareness, gesture use, impressions of gestures, and potential future adoption rates of these gestures.

3.3.1 Profiling Participants

From the first section of the survey, we were able build a profile of an average iPad user. 72.6% of respondents use full-size iPads with a 9.7" display (77/106), 22.6%, use an iPad Mini with a 7.9" display (24/106), and 4.7%, use an iPad Pro with a 12.9" screen (5/106). 59.4% of participants reported devices from 2013 or later (63/106). All of the iPads reported support the expert-level gestures of interest to us in our survey.

Our users are relatively experienced: 60.8% have owned an iPad for at least 2-3 years (62/102), while 15.7% have owned it for a 'longer period' (16/102). Daily usage was high: 72.6% use it for at least 30 minutes (77/106), and 41.5% use it for more than 60 minutes each day (44/106). As expected, most users use their iPads for media consumption and social media (Figure 3.3). Although there was no correlation between age and experience-level ($\chi^2 = 17.1$, NS), there were differences in usage patterns: younger participants watched more videos ($\chi^2 = 15.9$, $p < 0.01$), while participants age 30-44 used social media features the least ($\chi^2 = 15.8$, $p < 0.01$).

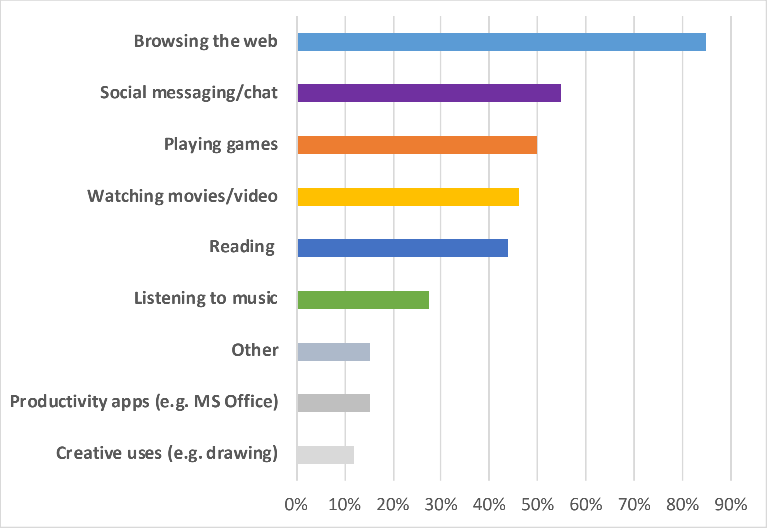


Figure 3.3: User estimates of device usage

We asked users about their satisfaction levels with the device on a 5-point Likert scale: 61.3% 'strongly liked' their device (65/106), 29.2% 'liked'' it (31/106), and 9.4% were neutral (10/106). When asked what Apple could change to improve the iPad, common answers were to make the device cheaper (11/83), more durable (7/83) or improve connectivity (6/83). A number of respondents suggested minor software changes (11/83). The largest category were satisfied with current features (19/83). There was no correlation between age and satisfaction ($\chi^2 = 8.1$, NS) or gender and satisfaction ($\chi^2 = 2.9$, NS).

3.3.2 Prior Knowledge

Multitasking is the ability to bring up a list of running applications, either by double-pressing the home button or swiping-up with four fingers. Participants were asked about their prior experience with the multi-tasking gesture: 6.7% had used only swipe-up (7/104), 32.7% had used the home button (34/104) and 34.6% had used both activation mechanisms (36/104). 10.6% were aware of the swipe-up gesture but hadn't used it previously (11/104), and 15.4% had never heard of the feature (16/104). There was a positive correlation between iPad and multitasking experience ($\chi^2 = 51.8$, $p < 0.01$), suggesting that experienced users are more likely to use this gesture.

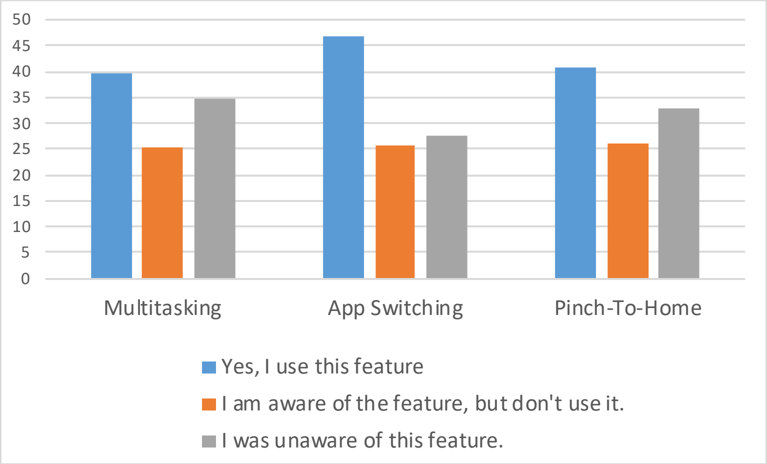


Figure 3.4: Prior experience with expert gestures

App switching is the ability to quickly switch between running applications, without needing to activate multitasking or return to the home screen. This feature can only be activated as an expert gesture, by performing a three-finger swipe left or right from within a running application. 52.0% of participants had used this feature previously (51/98). 24.5% had heard of the feature but never used it (24/98), and 23.5% had never heard of it (23/98). There was no correlation between overall iPad and app-switching experience ($\chi^2 = 17.4$, NS).

Pinch-to-Home is the ability to exit the application and return to the home screen. This can be accomplished by single-pressing the home button, or performing a four-finger clutching motion while an application is running. 43.0% of participants had used this feature (40/93), 25.8% had heard of it but never used it (24/93), and 31.2% were previously unaware of the gesture (29/93). There was no correlation between overall iPad and app-switching experience ($\chi^2 = 17.1$, NS).

3.3.3 Usage

Respondents who had used swipe-up to multitask performed the gesture infrequently: 49.4% rarely used it (38/77), 29.9% occasionally used it (23/77), and only 15.1% used it frequently (16/106). When given the choice, 76.6% preferred to use the home button to activate multitasking (59/77), 11.7% used the swipe-up gesture (9/77), and 11.7% used both mechanisms (9/77). Open-ended comments suggested that the swipe-up was sometimes 'difficult to activate' compared to the home button (16/41). The physical button, for many participants, was easier to activate (14/41) and more familiar (12/41). Participants who did not use multitasking appear to have been mostly unaware of the feature: 65.4% did not know it existed (17/26) and only 23.1% claimed that they do not need this functionality (6/26).

Participants who reported using app-switching, tended to use it infrequently: 39.2% used it rarely (20/51), 41.2% used it occasionally (21/51), and only 19.6% used it frequently (10/51). Participants who did not use it appear to have been unaware of the gesture: 62.2% reported that they did not know that it existed (28/45), 15.6% claim that they do not need this functionality (7/45), and 5.9% found it difficult to use (3/51).

Participants who used pinch-to-home also used it infrequently: 35.0% used it rarely (14/40), 45.0% used it occasionally (18/40) and 20.0% used it frequently (8/40). However, most respondents prefer the home button: 70.0% used the home button exclusively (28/40), 20.0% used both methods (8/40), and only 10.0% (4/40) used the pinch-gesture exclusively. Reasons for this preference included familiarity (9/20), ease-of-use and convenience (8/20).

3.3.4 Impressions

Multitasking was viewed favourably, with 82.5% neutral or higher (31/97 'neutral', 39/97 'like', 10/97 'strong-like'). It was generally considered easy to use, with 84.5% neutral or higher (31/97 'neutral', 39/97 'easy', 12/97 'very easy'). App-switching was also viewed favourably, with 82.0% rating it neutral or higher (26/89 'neutral', 35/89 'like', 12/89 'strong like'). It was generally reported as easy to use, with 85.4% rating it neutral or higher (26/89 'neutral', 34/89 'easy', 16/89 'very easy'). Pinch-to-Home was viewed very favourably: 92.0% of participants rated it neutral or higher (39/88 'neutral', 35/88 'like', 7/88 'strong like'), and 87.5% found it easy to use (27/88 'neutral', 38/88 'easy' and 12/88 'very easy'). Across each of these gestures, there was a positive correlation between experience and user satisfaction with that specific gesture (multitasking $\chi^2 = 41.1$, $p < 0.05$; app-switching $\chi^2 = 64.2$, $p < 0.01$; pinch-to-home $\chi^2 = 102.0$, $p < 0.01$).

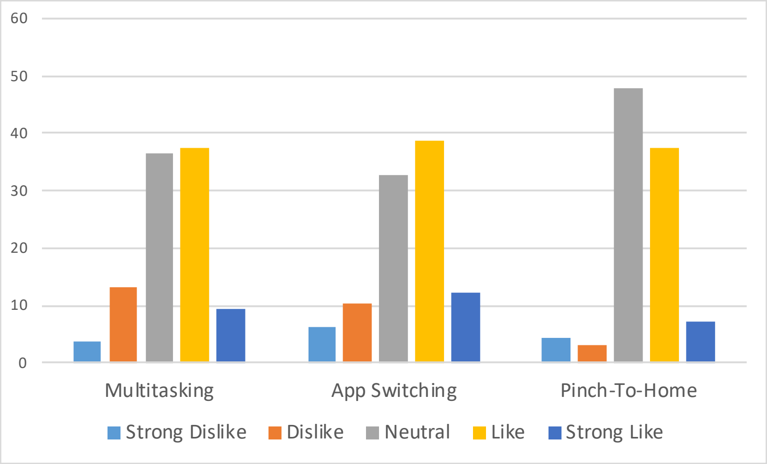


Figure 3.5: User impressions of expert-gestures

Post-survey impressions of expert-level gestures were very favourable: 94.5% were neutral or higher (29/91 "neutral", 42/91 "like", 15/91 "strong like"). This is higher than any single gesture score, suggesting that overall impressions improved as the survey progressed. Open-ended feedback was generally favourable, with many respondents indicating that the gestures required no further changes (14/36 for multitasking, 15/28 for app-switching and 13/31 for pinch-to-home). Complaints were that the gestures were complex and difficult to learn for some (8/41 overall), or difficult to discover (2/36 for multitasking, 6/28 for app-switching). As one respondent stated, "how do you find out about any of these? you have to accidentally run into them or spend hours googling ipad shortcuts" (sic).

3.3.5 Adoption

With multitasking gestures, 15.2% claimed that they are more likely to use them in the future (15/99), with 31.3% (31/99) suggesting that they would be interested to 'try it and see'. There was a significant positive correlation between prior multitasking experience and satisfaction ($\chi^2 = 41.1$, $p < 0.05$), and between satisfaction with the gesture and a willingness to continue using the gesture post-survey ($\chi^2 = 96.4$, $p < 0.01$).

With app-switching gestures, 12.1% (11/91) claimed that they were more likely to use swipe-left and right to app-switch following this survey; 36.3% (33/91) claimed that they would be interested to "try it". There was a significant positive correlation between prior app-switching experience and satisfaction ($\chi^2 = 64.2$, $p < 0.01$), and between satisfaction and a willingness to continue using them post-survey ($\chi^2 = 102.4$, $p < 0.01$).

With pinch-to-home gestures, 19.3% (17/88) of participants suggested that they were more likely to use pinch-to-home after being made aware of it. 34.1% (30/88) suggested that they might "try it and see". There was also a significant positive correlation between pinch-to-home experience and satisfaction ($\chi^2 = 102.0$, $p < 0.01$), and between satisfaction and a willingness to continue using this gesture post-survey ($\chi^2 = 120.4$, $p < 0.01$). When asked afterwards if the survey would impact use of these gestures in the future, 18.6% (17/91) of participants reported that they expected to use them more, and 35.1% (32/91) thought that the survey *might* have some impact (see Figure 3.6).

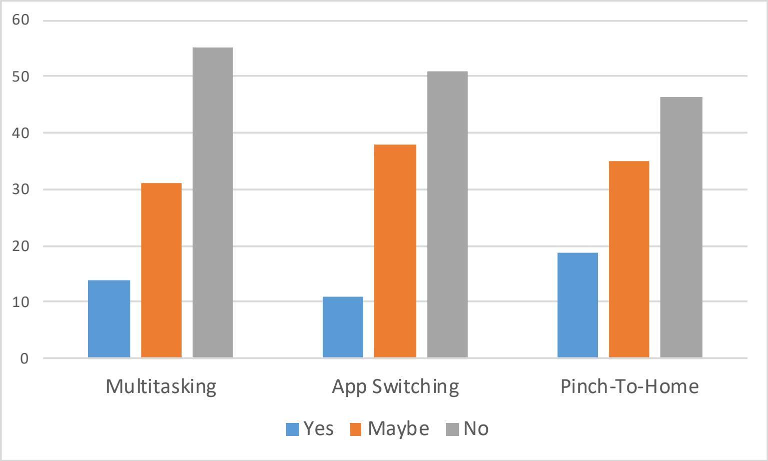


Figure 3.6: User anticipated adoption outcomes

3.4 Discussion

Our respondents are experienced iPad users, using their devices frequently for media consumption, playing games and interacting on social media. This aligns with previous studies of consumer behaviour [Yarow and Goldman, 2012]. However, despite this casual-use profile, our participants reported occasional-to-frequent use of expert-level gestures: 45.0% used swipe-up for multitasking, 60.8% used swipe-left/right for application switching and 65.0% used the pinch-to-home gesture. Familiarity with gestures was high, with awareness of 51.9% for swipe-up, 76.5% for swipe left/right and 68.8% for pinch-to-home. This is surprising, given that these features are not well-advertised.

After exposure during the survey, user impressions of these gestures were relatively high, and actually improved to 94.5% by the end of the survey (novice participants were asked to attempt them, which may have influenced this result). User satisfaction was directly correlated with experience level, suggesting that these gestures may require effort to master, and that users need time to effectively learn to use these gestures. Open feedback suggests that participants initially found these gestures difficult to discover, but that they worked well after some practice.

In the following chapters, we explore specifically the idea of enhancements to one canonical gesture, pinch-to-zoom. The results of the survey are encouraging, suggesting that users are willing to use and learn novel techniques, provided that they add some perceived value. Alongside justifying expert-level gesture sets or extended gesture sets, we believe that this result also adds evidence that enhanced variants of pre-existing canonical gestures – such as pinch-to-zoom – will have value to multi-touch users.

Chapter 4

Gesture Design

In this chapter, we consider factors that can contribute to the design of enhanced gestures, including usability considerations, choice of input factors, and device context. We use these factors to define the scope and design criteria for enhanced gestures as they are used in this thesis.

4.1 Design Factors

4.1.1 Input Modalities

When designing interactive systems, it is common to consider the modality, or information channel being used for a particular technique. According to Bernsen, a modality is simply a "way of representing information in some physical medium" [Bernsen, 2008]. Modalities can be input modalities, for handling user input, or output modalities, used to provide feedback to users. Examples include keyboards, mice or touch for input modalities, or audible or visual feedback for output modalities. Since we are considering human-computer interaction in this context, we typically constrain output modalities to be perceivable by users e.g. audible acoustics, visible graphics.

Nigay and Coutaz treat modalities as the combination of a physical input or output device (d) and a corresponding interaction language (L), which can be formalized as a tuple <d, L >[Coutaz et al., 1995]. In other words, it is the combination of the devices or sensors that are available, and how they are used, that characterizes a particular modality. Similarly, we can view interaction techniques on a smartphone in the same fashion, as a combination of device capabilities and the type of interaction we wish to support. For instance, multitouch implemented solely on capacitive touchscreens is limited to sensing contact points on-screen, and cannot discriminate which finger is being used for input. Adding finger-sensing to the device would expand the fidelity of the input channel and allow for more precise classification of touch input.

In this work, our focus is on the design of current technology, so we only consider sensors and capabilities that are commercially deployed. The current generation of multi-touch devices are fairly uniform in their capabilities: a 'standard' smartphone or tablet will have a high-resolution touch screen capable of discriminating ten or more simultaneous points of contact [Apple Inc., 2015a, Google Inc., 2015], high-resolution front and rear facing cameras, and a suite of sensors, such as an accelerometer to track device movement and a gyroscope to track orientation. These sensors are often used to support specific features, such as reorienting the screen for portrait or landscape mode, but are often under-utilized as input sensors.

Commercial multi-touch implementations rely exclusively on contact or finger placement, movement across the screen, and to a limited degree, contact pressure to infer which gesture is being performed. Commercial hardware cannot discriminate which fingers are being used to perform a touch gesture, and relies strictly on the number, position and movement of fingers or contacts on the screen to infer the intended gesture. Although other factors such as finger identification, have been extensively explored in research, they do not currently exist in commercial devices. A more comprehensive list of factors is shown in Table-4.1. This is a speculative list, meant to both identify the limits of the current approach, and showcase factors that we could consider in future work.

Finger pressure attempts to accurately determine the pressure that is being applied to a particular contact point. This adds another input channel that can be used to augment other factors. It has been suggested as a means of managing the lack of hover state on multi-touch devices so that systems could, for instance, discriminate between movement

Table 4.1: Input Factors for Multi-Touch Gestures

Category	Enables	Available
Finger position and number	Track number and location of contact points on-screen. Cannot discriminate between contacts.	✓
Finger pressure	Determine the pressure being applied to a specific contact point.	✓
Finger identification	Discriminate individual contacts; allows for persistence of contacts.	
Finger orientation	Determine finger orientation during touch operations.	
Hand orientation	Determine hand orientation above or while touching the device.	

and activation, similar to the way in which a mouse cursor can be tracked, and the mouse button used for activation. Promising research has enhanced touch-screen capabilities to sense above-surface motion [Mizumata and Sakamoto, 2010, Annett et al., 2011, Hinckley et al., 2016]. Pressure-detection is starting to appear in commercial devices like 3D Touch on Apple's iPhone [Apple Inc., 2018], although it is rarely utilized.

Finger identification is the process of associating specific fingers with a given contact point on-screen. This is valuable because it allows input discrimination based on the specific finger being used. For example, index-tap could provide a different response than thumb-tap. It also allows for persistence of a recognized input, where the system can accurately track fingers as they are lifted from, and replaced on-screen. This allows for more complex expressions, and chaining of actions, such as chording or sequences of actions using specific fingers [Goguey et al., 2016]. No commercial systems currently provide this capability, but a number of research projects have investigated the viability of tracking and effectively using this information [Goguey et al., 2014, Masson et al., 2017, Goguey et al., 2017].

Finger orientation refers to the ability to detect finger orientation on the surface, and includes lateral orientation and well as finger pitch and roll at the point of contact. This could facilitate fine-grained gestures, which would greatly increase the input design space, especially for smaller screens or one-handed use [Dang et al., 2009, Roudaut et al., 2009, Dang and André, 2011, Goguey et al., 2018]. Research has demonstrated the viability of some levels of finger posture detection, but current commercial systems do not leverage these capabilities.

Hand orientation is similar to finger orientation, but is more concerned with the orientation of the hand and arm over the device. This could be a proxy for determining the orientation of the user with respect to the device, or possibly handedness i.e. whether the user is using the right or left hand. This could be used, for instance, to improve the accuracy of text entry or other forms of input that could be biased based on the hand used [Goel et al., 2012, 2013, Le et al., 2016].

Adding any of these features greatly expands the design space for multi-touch gestures. Adding finger identification, for instance, enables finger-specific gestures, which operate differently depending on which finger is used for their activation, or chording, the ability to use multiple, specific fingers in combination (see Table-4.2). However, the majority of these features do not exist in commercial devices at this time.

4.1.2 Device Characteristics

Device characteristics, and the context in which a device will be used, can influence which gestures can be performed, and how effective they might be. Although this thesis is focused on portable multi-touch devices, there are drastic differences between models of smartphones, and between smartphones and tablets, watches or other multi-touch devices.

If we limit the discussion to smartphones, the context of how the phone is being used will have an impact on usability and efficiency. For instance, people tend to hold smartphones one-handed, especially while walking or moving [Yarow and Goldman, 2012], to keep a second hand free to hold an item, open doors and so on. In these situations, unimanual (one-handed) hand postures, or cradled positions (holding in one hand while interacting using the other) are likely going to be preferred. Bimanual postures (holding the smartphone in two hands and using both hands to interact) will likely only be considered when

Table 4.2: User Features Enabled by a Richer Input Model

Feature	Description	Enabled By
Chording	Multi-finger gestures, where specific fingers are placed together or in sequence [Westerman, 1999, Ghomi et al., 2013, Goguey et al., 2016].	Finger identification
Finger-specifically gestures	The ability to perform different actions depending on the finger being used to perform the gesture [Zheng and Vogel, 2016, Le, 2018]	Finger identification
Hover-state	Discriminate between movement across, and activation of, on-screen targets [Mizumata and Sakamoto, 2010, Annett et al., 2014].	Finger pressure
Device context	The ability to customize the interaction based on context or situation.	Hand orientation, Device orientation

absolutely necessary, or when they provide clear benefits, such as typing on an on-screen soft keyboard (see Figure 4.1). Tablet use is similarly constrained, and users will often cradle the device in one hand, while using the other hand to interact, or just place the tablet on a flat surface (see Figure 4.2).

Context provides design guidance, since it drives how the device is likely to be actually used in-practice. Pinch-to-zoom for instance, cannot easily be performed on a smartphone one-handed: the user must hold the phone with one hand while using two-fingers of the second hand to interact (or place the phone down on a surface). This makes it tricky to use while walking, and may have been a motivating factor behind the one-handed double-tap to zoom gesture in Google Maps [Google Inc., 2015]. On a device with a very small screen,

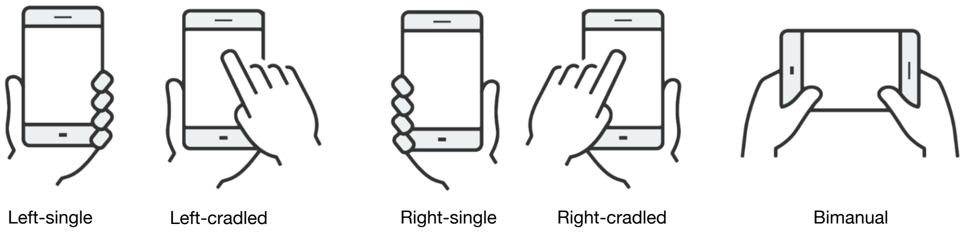


Figure 4.1: Smartphone hand postures

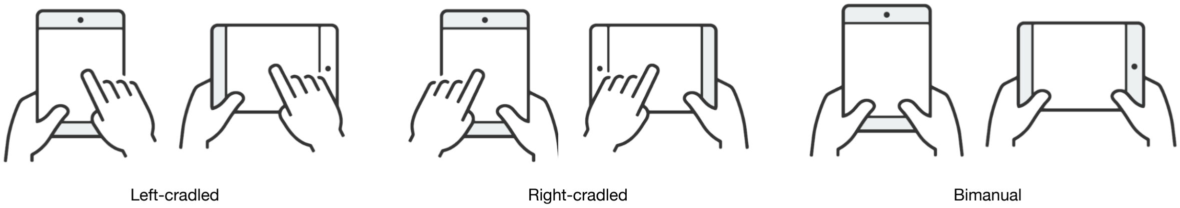


Figure 4.2: Tablet hand postures

like the Apple Watch, pinch-to-zoom is impractical, and zooming is instead handled by a scroll-wheel on the side of the watch. Despite the advantages of bimanual interaction, such as less fatigue and lower error rate, it is rarely used on smartphones [Casalta et al., 1999, Guiard et al., 2004, Ruiz et al., 2008, Huang et al., 2016, Wagner et al., 2012, Malacria et al., 2010]. Practical considerations suggest that in many situations users need to support the device with one-hand, so designs tend to favour unimanual gestures.

4.1.3 Usability Considerations

As discussed in Chapter 1, gesture design needs to consider issues related to usability: how easy a particular gesture is to discover, learn and use. One approach is to consider ways to make a new gesture familiar and similar to something that the user already knows how to use. Norman suggests that designers can make new techniques similar to old familiar techniques, to help users integrate the new technique into their mental model [Norman, 2002].

There are different ways of characterizing and comparing gestures, and it is helpful to distinguish interface semantics from syntax. Interface **semantics** refers to the result of a particular action, or the 'interface and data states that can be achieved with an interface' [Scarr et al., 2011]. In other words, semantics are the results achieved through a particular interaction technique. Interface **syntax** refers to the specific mechanism used to control the interface, which may include one or more interaction techniques. For example, both 'Ctrl-C' and 'Edit-Copy' will copy the selected object in a typical windows application, and are semantically identical, even through they have different syntax, or activation mechanisms. Scarr et al. suggest that making new techniques semantically and syntactically similar to familiar techniques makes it easier for a user to 'make sense' of the new technique, reducing mental effort and time to learn to use it effectively [Scarr et al., 2011]. Further to our example, if we wanted to create a copy that just copied plain-text, 'Ctrl-Alt-C' would be an excellent choice (if available) since it is syntactically similar to 'Ctrl-C', and provides very similar functionality. These similarities would make this technique easier to learn and recall. We will consider this method of minimizing syntactical and semantic differences, when designing enhanced gestures in the next section.

4.2 Enhanced Gesture Design

There has been significant past work on expanding the input space of feature rich applications in a wide variety of ways. In *The Humane Interface*, Raskin Raskin [2000] creates a design space for feature rich input with a focus on gestural interaction. Raskin's design space focuses around the use of modes with three primary categories of input: mode-based interaction; quasi-modes or kinesthetic modes Sellen et al. [1992]; and modeless input. Raskin Raskin [2000] defines a mode as it relates to application state:

> An interface is modal with respect to a given gesture when (1) the current state of the interface is not the user's locus of attention; and (2) the interface will execute one among several different possible responses to a gesture depending on the system's current state. (p. 42.)

So, one way to expand gestural input space is to leverage modes, an approach commonly used in pen-tablet systems Li et al. [2005]. However, the challenge with moded input is that it eliminates many of the advantages of direct manipulation Shneiderman [1997a], requiring users to interact with interface elements rather than directly acting on objects. At the other extreme, elicitation studies Wobbrock et al. [2009] expand gesture sets by adding additional gestures, but this requires mastery of larger command sets.

This thesis aims to design enhancements to one standard gesture, pinch-to-zoom, on commodity hardware, such that the enhanced versions appeal to both novice and expert users. **Enhanced-gestures**, introduced in Chapter 1, aim to provide enhanced capabilities to standard gestures, similar in spirit to many of the advanced multitouch gestures that have appeared in prior work. However, instead of positing two techniques that stand in sharp contrast, we will establish design criteria that allow us to build gestures that are recognizably more similar to standard gestures, either syntactically or semantically (or ideally, both). This serves two purposes: first, as we have suggested, this aids learning and recall for new techniques. Second, it allows us to build a set of cohesive gestures that feel like they are designed together, and 'fit' within the pre-existing gesture set.

Figure 4.3 suggests ways in which we can view pinch-to-zoom gestures along a design continuum.

In this model, we explicitly call out design dimensions that we feel can characterize multitouch gestures, and contribute to making them more approachable and adoptable.

1. **Familiarity**: As discussed, familiarity may be the most critical factor in making gestures approachable. The closer they are to techniques that users already know how to perform, the more likely that they will be attempted and possibly adopted.

2. **Complexity**: Complexity is difficult to characterize, since it is subjective: users will find a gesture simple or complex based on prior experience, and their ability to quickly determine how the gesture is activated, and what it is intended to accomplish. We will avoid attempting to quantify this term, and instead refer to gestures as being more or less complex for a particular user. We expect that familiar gestures will often be interpreted as simpler.

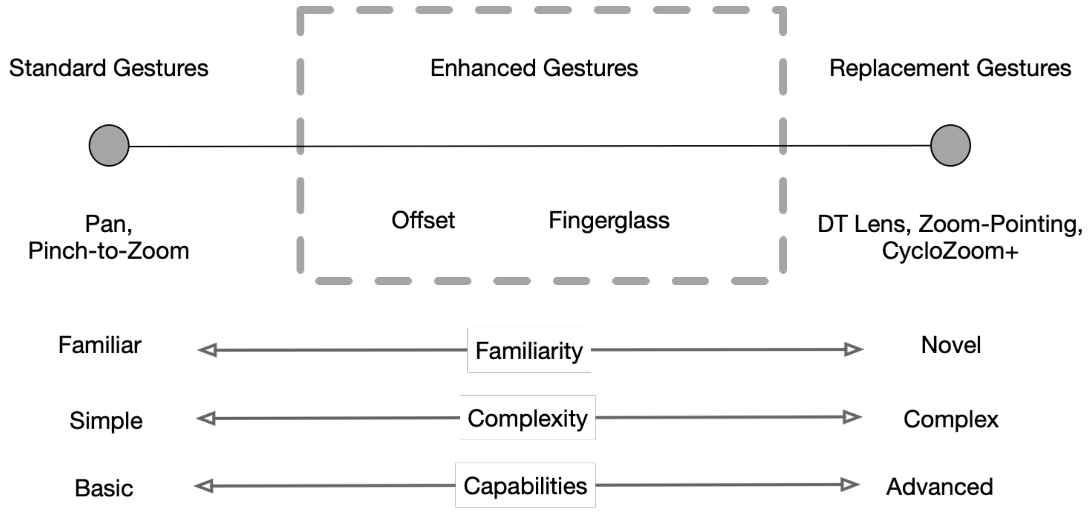


Figure 4.3: Enhanced gestures address the middle-ground between novice, standard gestures and expert replacement gestures.

3. **Capabilities**: This attempts to characterize the idea of gestures being more complicated and/or difficult to use, simply on the basis of the complexity of the task that they perform. Enhanced capabilities often comes at the cost of higher complexity.

Designing along this scale suggests a range of options when designing enhancements for standard multi-touch gestures: we can allow users to use the original standard gesture (left), or introduce enhanced gestures that provide subtle enhancements (towards the centre of the figure), or opt for a complete replacement (far right). Enhanced gestures are syntactically similar to standard gestures, making them easier to learn, but are more constrained in terms of their behaviour compared to replacements. Complete replacements provide the most design flexibility, at the cost of being syntactically dissimilar to the original technique, and harder to learn [Scarr et al., 2011]. Our hope is that smaller, incremental changes, keeping the replacement as close to the standard gesture as possible, will encourage adoption. Also, training invested in an easier-to-learn novice technique can be leveraged in the more complex enhanced gesture [Rosson, 1983], which further eases the adoption.

As an example, pinch-to-zoom on the left represents a standard gesture for view navigation and manipulating scale. We would consider it to be a novice gesture, since it is relatively simple to use and demonstrate, and very familiar to most users. On the opposite end of the spectrum, we can consider other gestures which attempt to completely replace pinch-to-zoom. These are designs that are syntactically disimilar from the original gestures, and use new and unfamiliar control mechanisms, which will make them more challenging for users to learn. These replacement gestures are semantically similar, but not identical to pinch-to-zoom. Cyclostar for instance, supports zoom but not pan operations, and doesn't handle clutching. It operates very differently from pinch-to-zoom, which will make it more challenging for users to learn. Enhanced techniques are the middle-ground, where we attempt to minimize these differences. Offset and Fingerglass, for instance, also attempt to replace pinch-to-zoom, but use the same control mechanisms - adjusting the spread between two contact points to change the scale magnification factor. There are semantic differences of course, which provide additional value, but the 'gap' is confined to those small differences.

We have found this a useful framework for gesture design, and in the remainder of this thesis, we will use it to design enhanced gestures, with the following caveats.

As stated earlier, we wish to extend the capabilities of standard gestures on *existing* multi-touch systems; we are looking for enhancements to gestures that *users already know how to use*. For this reason, we will limit the design of enhanced gestures to those capabilities that exist in standard, consumer devices such as standard contact or finger sensing on commodity smartphones and tablets. Since other factors, like finger pressure or finger identification are not leveraged in current generation devices, we will not consider them at this time. However, we are confident that these guidelines will continue to be relevant, and can be adapted to those features as they are introduced.

Secondly, we focus on pinch-to-zoom as an example of a standard gesture to be optimized. In the following chapters, we design and implement Pinch-to-Zoom Plus and Transient Zoom, two enhanced versions of pinch-to-zoom, using these criteria. Although both are enhanced gestures, they solve different problems and represent drastically different implementations:

1. **Pinch-to-Zoom Plus (PZP)** is intended to be a more efficient verison of standard pinch-to-zoom. It uses the same activation mechanism as pinch-to-zoom (two-finger

pinch and spread), and can be described as an **implicit** technique compared to the standard gesture. It also appears in our design language as "closer to" the standard gestures on the left.

2. **Transient Zoom** adds new capabilities to pinch-to-zoom. It does not attempt to replace the standard gestures, but rather adds more capabilities that co-exist with the previous functionality. The addition of new capabilities, activated as a quasi-mode makes this an **explicit** gesture.

In the following sections, we discuss the utility and design of these two gestures, and evaluate their use compared to standard pinch-to-zoom. Although these examples focus specifically on bimanual interaction on tablets, we believe our results are extendable to other gestures, on different multitouch devices, across different contexts.

Chapter 5

Analyzing Pinch-to-Zoom

Pan and pinch-to-zoom are standard interaction techniques, commonly used for multi-level navigation and content manipulation. However, despite their popularity, there have been few formative studies examining pinch-to-zoom behaviour. Prior research has identified numerous drawbacks to both techniques, including the need to clutch and provide corrective pans to the region of interest.

In an attempt to identify opportunities to optimize these gestures, we conducted three formative studies of pinch-to-zoom, where we profiled how users performed pan and zoom with a simple target selection tasks, and determined kinematic and clutching characteristics of standard pinch-to-zoom. We identified patterns in pan and zoom behaviour: zooming actions follow a predictable ballistic velocity curve, and the tendency of users to reposition the point-of-interest towards the centre of the screen to avoid the risk of content moving offscreen. These conclusions will be used to motivate design in later chapters.

5.1 Motivation

Buxton [Buxton, 2007] traces the history of pinch-to-zoom to the early 1980s based on a demonstration of two-finger pinch and pan interaction in Wellnerś Digital Desk video [Wellner, 1991]. Hinckley et al. also describe the Pinch gesture [Hinckley et al., 1998] which allows users to zoom and pan around the centre of two contact points. Pinch-to-zoom has since become the standard zooming gesture on multi-touch devices [Spindler et al., 2014], and is represented in every mainstream multi-touch implementation Apple Inc. [2015a], Google Inc. [2015].

Despite the popularity of these gestures, there is little prior work that has examined how users actually use these techniques. Hoggan et al. [Hoggan et al., 2013a] examine the ergonomics of pinch-to-zoom, identifying factors that contribute to performance, such as direction, distance, angle and position. Tran et al. [Tran et al., 2013] build a quantitative model of pinch-to-zoom performance. This research seeks to broaden our understanding of pinch-to-zoom behaviour, specifically with the goal of identifying trends that we can exploit, to improve the efficiency of the technique.

5.2 Observational Study

We designed and conducted an observational study to determine how participants combined panning and zooming to solve specific, standard, zooming tasks.

5.2.1 Participants

We recruited 6 participants, which included students and computer professionals (2 female, 6 right-handed, mean age 42 years). The study took approximately 20 minutes to complete, which included training time on each technique. Participants filled out a short questionnaire afterwards.

5.2.2 Apparatus

This study was performed on a Toshiba AT200 1.2 GHz Dual Core tablet with 1 GB RAM and a 1280 × 800 pixel display (218 × 136 mm, 5.88 px-per-mm). Custom software was implemented in Android 4.03, and tuned to duplicate standard Android pinch-to-zoom behaviour. Observed lag was minimal, and events were logged at a device sample rate of

60Hz. The tablet was placed flat on a standard desk in landscape mode, and anchored to the surface using rubber pads to prevent movement. An overhead video camera recorded the sessions.

5.2.3 Design

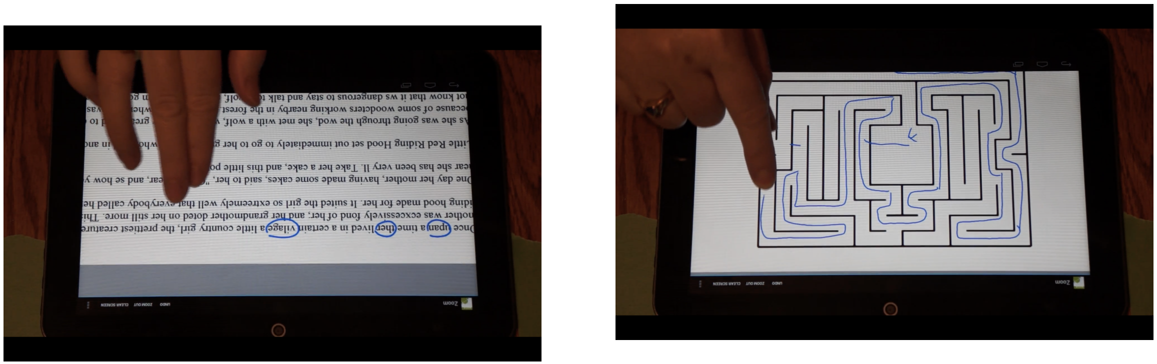
Participants were presented with 4 blocks, with each block presenting one of the following tasks. At the start of each block, the participant was trained on the technique, given time to practice, and then asked to perform the task described below. The tasks were designed to be progressively more difficult; users could complete the first task with minimal zooming, but needed to pan and zoom to successfully complete later tasks.

The following tasks were presented:

1. A page of text containing approximately 20 spelling mistakes. Users were asked to zoom-in and circle each mistake as they read the passage. This represents a common annotation task (see Figure 5.1a).
2. A maze, where they had to draw a line to navigate the maze without touching the walls. This is similar to a precision drawing task (see Figure 5.1b).
3. A grid containing 10 randomly distributed donut-shapes. Users were asked to zoom-in to each shape, in order, and draw a line between the inner and outer circles. This represents a random navigation task (see Figure 5.1c).
4. A map, where they had to locate and circle 10 items. The map was larger than the screen resolution, so users had to pan and zoom to perform the operation. This represents a typical map navigation task (see Figure 5.1d).

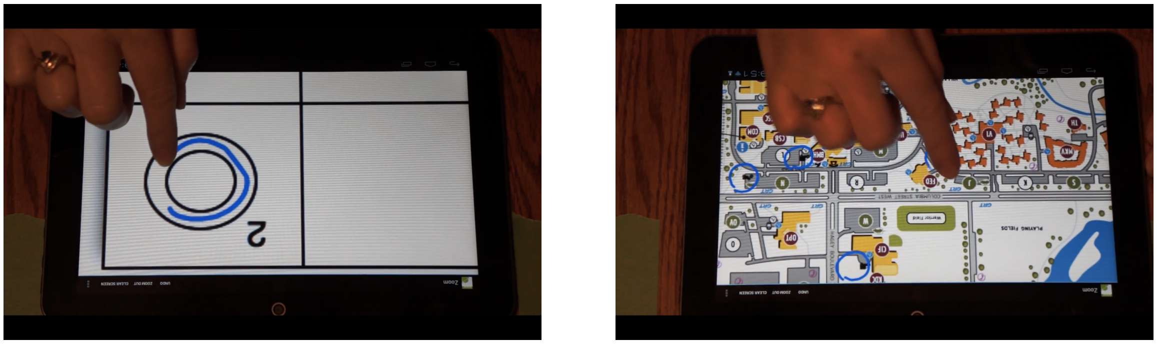
5.2.4 Results

There appeared to be some variation in how participants approached these tasks: 2 appeared to prefer bimanual operation (i.e. using a finger from each hand to zoom), while



(a) Circling spelling errors in a block of text.

(b) Navigating through a maze, while not touching the sides.



(c) Drawing inside the border of a circle.

(d) Finding waypoints on a map.

Figure 5.1: Observational study tasks

others appeared to prefer unimanual operation. Participants were not observed to approach or perform these tasks with any particular strategy.

We observed the following elapsed times for each block:

- Spelling: 312s (SD 64s)
- Maze: 134s (SD 47s)
- Grid: 181s (SD 34s)
- Map: 154s (SD 68s)

Task times for each task, by participant, are shown below in Figure 5.2. The tasks are significantly different, so we cannot derive meaningful conclusions by directly comparing time for each task. However, the task time variance within a task is significant, indicating a large amount of individual performance variation. This suggests that further experiments should continue to be designed as within-subject to account for individual variation.

Also, performance in one task was a poor indicator of performance in other tasks. P3, for instance, struggled with the Spelling and Grid tasks compared to the other participants, but excelled in the Map navigation task. This suggests that we cannot easily generalize across tasks, but need to focus on a particular category of related tasks when measuring performance.

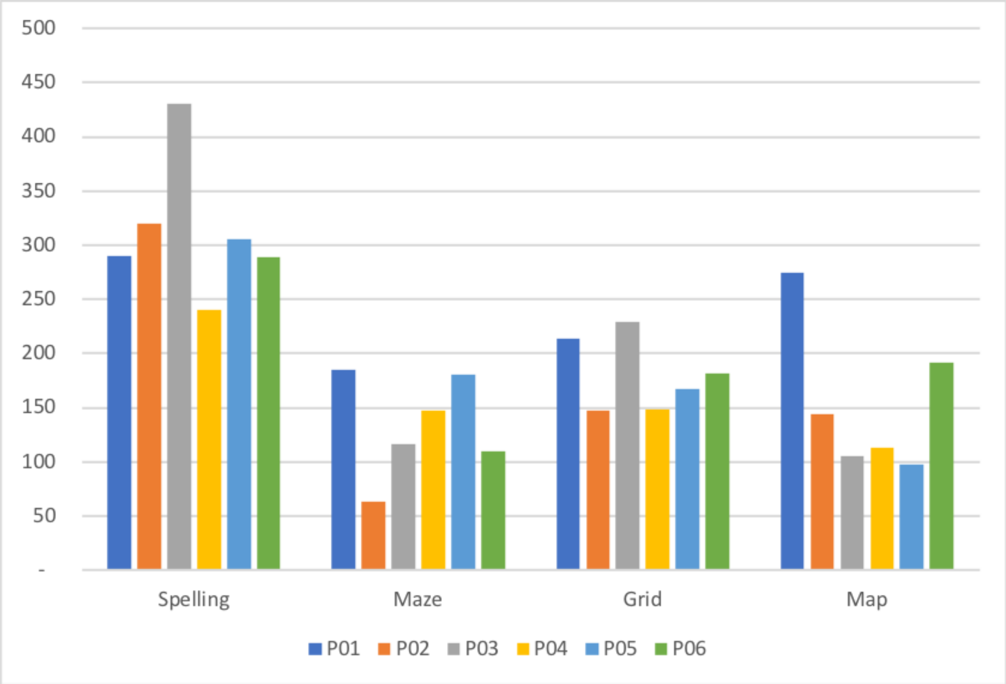


Figure 5.2: Pilot task times by participant

We also interviewed participants after the tasks were complete, and asked them to assign score from 1-5 (Likert scale) on ease-of-use, speed and learnability of pan and pinch-to-zoom for each target area. The results are listed below in Figure 5.3:

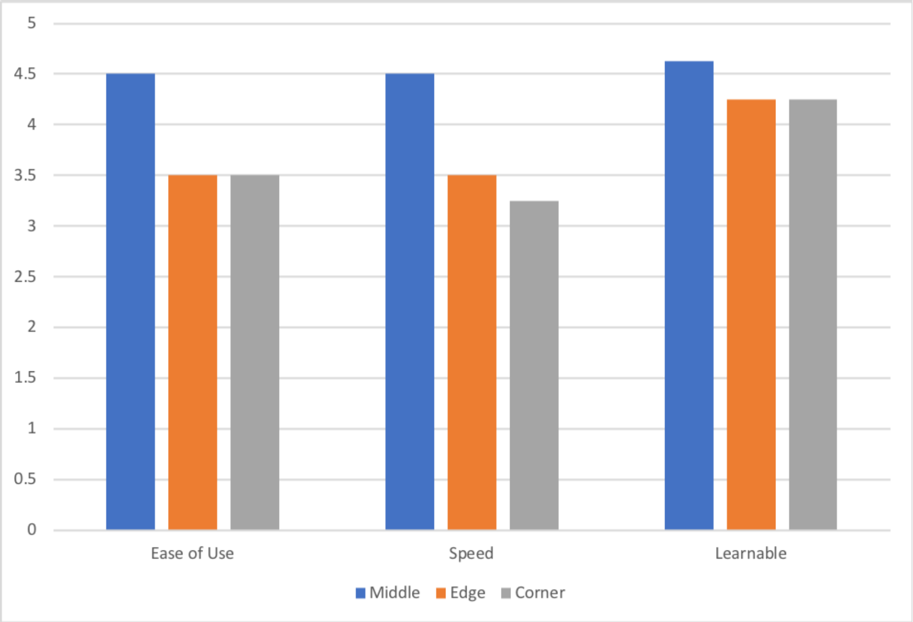


Figure 5.3: Feedback in pan-and-zoom tasks

Generally, pinch-to-zoom was well-regarded: "[Pinch] is so quick. The speed is great." (P01); "[Pinch] is "intuitive and natural." (P04). However, a few participants pointed out some specific limitations. "[Pinch] is not efficient for quickly zooming in and out." (P03); "Too much work to zoom out." (P05).

We made a number of observations in this study:

- Participants did not pan or zoom unless the task required it. They typically attempted to perform tasks using the simplest methods (from observed behaviour).
- Participants appeared to employ different strategies when manipulating content: some preferred to alternate between zoom and pan, while others would zoom to the desired resolution and then perform successive pans. Some preferred bimanual gestures, while others preferred unimanual operation. In other words, there were observable differences in how participants completed these tasks (from Figure 5.2).
- Panning and zooming appeared to be easier when the target was in the centre of

the screen. Users struggled when interacting along the edge or corners of the display (from Figure 5.3).

- Pinch-to-zoom is really optimized for occasional zooming, not for moving repeatedly between resolutions (from comments).

Our formative studies, which follow, are designed around these observations.

5.3 Formative Study 1

The goal of this first formative study was to examine clutching in pinch-to-zoom gestures using a simple docking task with the target and dock centred on the display (see Figure 5.4). In this task, panning was disabled, and only pinch-to-zoom was required; the target remained centred as the user changed resolution.

5.3.1 Participants

Twenty right-handed participants (3 female), with an average age of 27 (SD 5.44) participated in our first study. All participants had prior experience with multi-touch tablets or smartphones. $10 remuneration was provided.

5.3.2 Apparatus

This study was performed on a Toshiba AT200 1.2 GHz Dual Core tablet with 1 GB RAM and a 1280 × 800 pixel display (218 × 136 mm, 5.88 px-per-mm). This was the same device and experimental setup used in the earlier Observational study.

5.3.3 Design

The target was a blue opaque square and the docking area was a semi-transparent blue rectangle with a border thickness representing the fit tolerance. The background was textured to provide context and simulate a document task, like map navigation. At the start of the task, the dock was placed mid-screen with the target centred over it. Panning could be used to move the target, but the dock did not move from the centre. Thus, panning was allowed, but was not required to complete the task. The dock highlighted when the target fit correctly, and the task was completed when both fingers were released in that state. Note that all trials were ultimately successful, but intermediate errors caused by tighter fit tolerance increased the difficulty of the task.

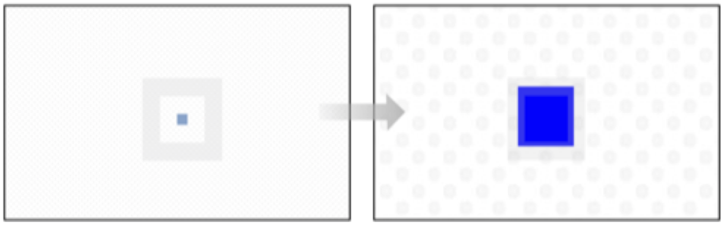


Figure 5.4: Study 1 docking task. Participants were asked to zoom the target (i) to make it fit within the docking area (ii). Targets turn bright blue when properly scaled to the docking area, indicating that the operation was successful.

The design is within-subjects with independent variables for hand condition (unimanual/bimanual), tolerance of docking region (4 mm, 7 mm, 10 mm tolerances), zoom direction (zoom-in, zoom-out), and zoom level (4, 7, 10, 13, 16, 19× for zoom-in, and multiplicative inverse levels for zoom-out). The zoom-out direction required participants to scale a large target down, and zoom-in required the opposite. The six zoom levels were chosen to cover typical pinch-to-zoom scenarios; for example, the maximum zoom level of 19× is roughly equivalent to zooming from a map of a large city (10 km per cm) down to a city block (525 m per cm). Hand condition (unimanual/bimanual) was counterbalanced between participants, and all tolerances, zoom levels, and zoom directions were fully crossed and presented in a random ordered block of trials, to balance potential learning effects. Five blocks were presented creating five repetitions.

An initial practice block of 12 trials presented a subset of tolerances 7 mm, 10 mm and zoom levels 4, 7, 15, 0.25, 0.14, 0.06. Not counting these practice trials, the experiment had 360 trials per participant. The experiment time was approximately 45 minutes.

5.3.4 Pre-Processing

We define an *action* as the sequence of two finger dragging movements occurring between touch down and touch up events. Note that a clutch occurs between actions. We classify each action as either a zoom or pan using thresholds determined in our pilot study:

- *Pan actions* are defined as actions where fingers spread less than 5 px (0.85 mm). Pans of less than 2 px (0.34 mm) were discarded as unintentional movements.
- *Zoom actions* are defined as actions where fingers spread more than 5 px (0.85 mm). We differentiate between zoom-in and out based on spread direction.

For each trial, we calculated the completion time and number of zoom-in, zoom-out, and pan actions. One participant was an outlier with all completion times more than two standard deviations from the overall mean, and was excluded from analysis. Learning effects of block on elapsed time were observed ($F_{2,18} = 7.79$, $p < .01$). Posthoc analysis indicated an effect on block 1, so it was excluded. Excluding practice blocks and block 1 tasks, the remaining 19 participants performed a total of 4104 tasks.

5.3.5 Results

Mean task completion time was 5.65 s (SD 2.71). Handedness had no significant effect on elapsed time ($F_{1,18} = 0.418$, NS). As expected, tolerance had an effect on the elapsed time, with smaller targets taking significantly longer to complete ($F_{2,18} = 210.85$, $p < .01$). Zoom level also had a significant impact on time ($F_{5,18} = 32.83$, $p < .01$), with the completion time increasing as greater zoom levels were required to achieve the target.

Mean counts of actions per task were 1.94 zoom-in actions (SD 2.33), 1.75 zoom-out actions (SD 1.86) and 7.76 pan actions (SD 4.66). Zoom-in count was not significantly affected by handedness ($F_{1,18} = 2.096$, NS), but was affected by tolerance ($F_{2,18} = 41.94$, $p < 0.01$) and scale ($F_{5,18} = 125.66$, $p < 0.01$). Zoom-out count was not significantly affected by handedness ($F_{1,18} = 0.374$, NS) but was affected by tolerance ($F_{2,18} = 199.792$, NS) and scale ($F_{5,18} = 125.148$, NS). Pan count was not significantly affected by handedness

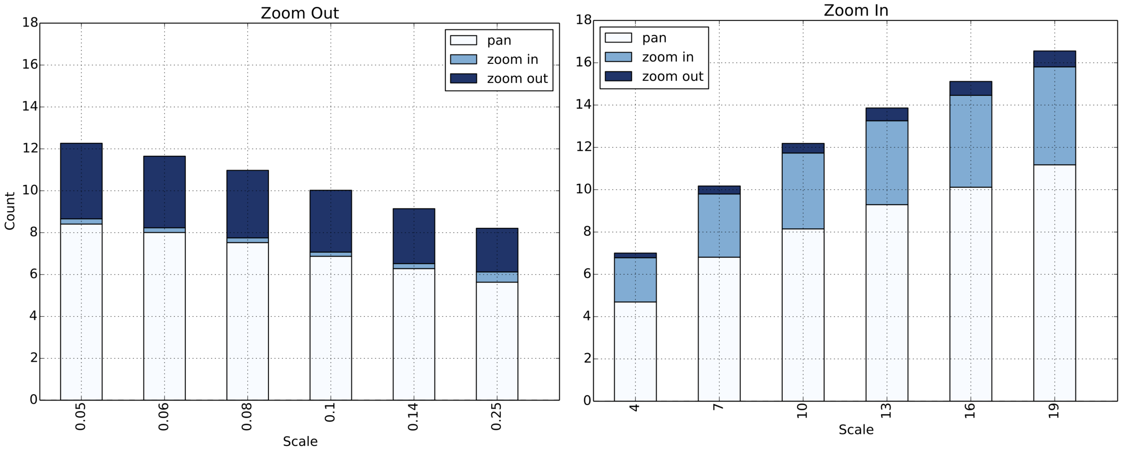


Figure 5.5: Count of zoom and pan actions by zoom level. More than 1 zoom or pan action requires a clutch (i.e. 1 action has 0 clutches, 2 actions require 1 clutch, etc.)

$(F_{1,18} = 0, \text{NS})$ but was affected by tolerance $(F_{2,18} = 89.49, p < 0.01)$ and scale $(F_{5,18} = 41.65, p < 0.01)$. Not surprisingly, smaller or larger targets, and low-tolerance targets, require more zoom and pan actions to acquire.

The pattern indicates that even a small change in zoom-level requires clutching: 1.94 clutches for zoom-in (4x had a mean of 2.1 zoom-in actions) and 1.75 clutches for zoom-out (0.25x had a mean of 2.1 clutches). Exceeding 10x or 0.01x zoom level required more than three clutches on average (5.5). The mean clutch time between two successive zoom actions was 180.5 ms (SD 250.7).

We were surprised by the number of pan operations. The target and dock were both positioned in the centre of the display, and overlapping, so panning was not required to complete the task. Even if one assumes that zooming in (i.e. enlarging content) would require pan operations to ensure that content did not migrate off the screen, the similarity in frequency of pan actions when zooming out is not explained by this. Even a small change in resolution, 4x for zoom-in and 0.25 for zoom out, was associated with a significant number of pan actions: averages of 4.69 and 5.62 pan actions, respectively. The prevalence of pan actions when a target was optimally positioned in the centre of the display raises the question of how and why participants chose to reposition content.

5.4 Formative Study 2

This goal of the second formative study is to expand the scope of the first study by placing the dock and target around the perimeter of the screen. In addition to the docking task from study 1, we add a drawing task that required a simple drawing to be performed after docking. In both tasks, participants were allowed to use pan and pinch-to-zoom. Panning was not required, but could be used to keep the target centred over the dock.

5.4.1 Participants

We recruited 17 right-handed participants (4 female), and 3 left-handed participants (all male) with an average age of 27.7 (SD 4.98). All participants had prior experience with multi-touch tablets or smartphones. A $10 voucher was offered as remuneration.

5.4.2 Apparatus

This study was performed on a Toshiba AT200 1.2 GHz Dual Core tablet with 1 GB RAM and a 1280 × 800 pixel display (218 × 136 mm, 5.88 px-per-mm). This was the same device and experimental setup used in the earlier Observational study.

5.4.3 Design

As in our first study, the docking task is to zoom a target in or out until it fits in a docking area, within a certain tolerance (Figure 5.6). Unlike our first study, targets are placed at one of 8 locations around the perimeter of the display. Panning caused both the target and dock to move together, so panning was not required to complete the task — successful docking only required zooming. However, based on the voluntary panning of study 1, we believed participants would use pan as part of the zoom operation and/or to optimize the placement of the target.

The second task simulates editing tasks like annotating maps or editing photographs. The presentation is the same as the docking task, except that the target and dock are

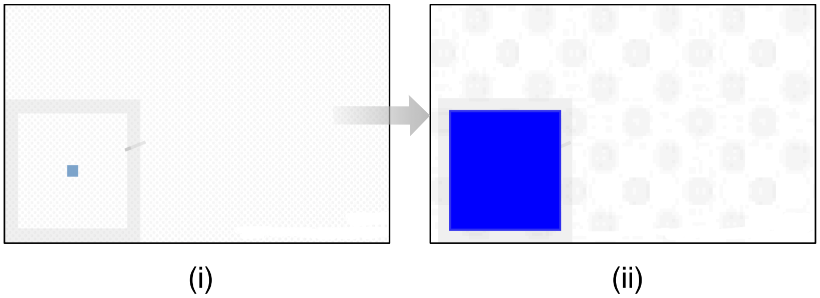


Figure 5.6: Study 2 docking task with targets positioned around the edge of the screen. Participants are asked to zoom the square (i) until it falls within the docking area (ii). Targets turn bright blue when properly scaled within the docking area, indicating that the operation was successful.)

circular. After reaching the required zoom level with the target fit inside the dock, participants also had to draw a circle within a confined region of the target (Figure 5.7). This was done to encourage participants to interact with the target, making it difficult to complete the task unless the target remained visible on the screen. For brevity, we label this second task a drawing task. Drawing tasks did not include a zoom-out condition, because the nature of the task required a large target in which to draw.

The screen was divided into a 3 × 3 grid of 9 regions, which were used to determine initial target position (the middle was excluded, so 8 regions were available). At the start of each task, a region was chosen randomly, and the dock and target were centred within that region. All tasks were performed with a 7 mm tolerance, the median tolerance from the first study. We used 2 zoom levels for the docking task, 13× for zoom-in, and the multiplicative inverse for zoom-out, and one zoom level for the drawing task, 13×. The decision to use a single tolerance and zoom level was motivated by the observation that any scale factor required clutch and pan actions, so fully varying across all tolerances and scales was not necessary and allowed us to limit our study to a reasonable length of time for our participants.

Participants performed docking tasks first, then after a 3 min break, performed all drawing tasks. Each task was repeated for bimanual and unimanual hand conditions, with the order counterbalanced across participants. In the docking task, a practice block of 6

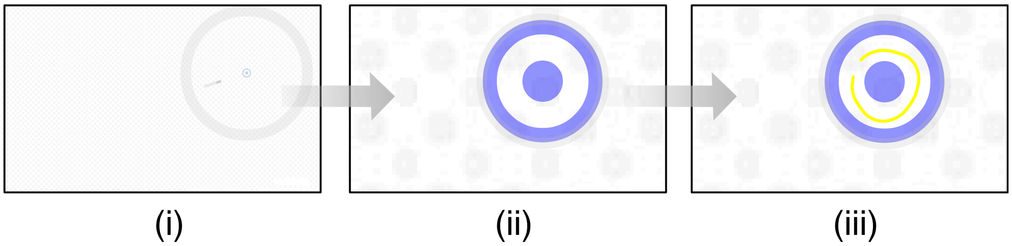


Figure 5.7: Study 2 drawing task. Participants are asked to zoom until the circular target (i) falls within the docking area (ii), and then draw a line along the inside path (iii).

trials was presented at the beginning of each condition (24 trials in total). Drawing tasks included 8 trials per block, representing the 8 positions. Not counting practice trials, the experiment had 384 trials per participant (256 docking tasks plus 128 drawing tasks). The total time for the experiment was 60 minutes. Two-factor ANOVA and Bonferroni-Dunn correction were used for pairwise comparisons including the effects of target position, and hand condition.

5.4.4 Pre-Processing

A learning effect existed across four blocks, for both docking tasks ($F_{1,20} = 18.52$, $p < 01$) and drawing tasks ($F_{1,20} = 13.75$, $p < .01$). Posthoc analysis indicated that this effect was limited to the first block, so it was excluded, resulting in a total of 5780 experimental tasks performed (3860 docking and 1920 drawing tasks). Actions were classified using the same tolerance as study 1 into pan actions (movement without significant zoom), zoom-in actions (scale increasing) and zoom-out actions (scale decreasing).

5.4.5 Results

Docking tasks had a mean task completion time of 4.84s (SD 2.16). Handedness had no significant effect on time ($F_{1,20} = 1.005$, NS). Starting target position had an effect on task time in docking tasks, both for zoom-in ($F_{7,20} = 4.934$, $p < .01$) and zoom-out ($F_{7,20} = 2.752$,

$p < .01$), although post hoc analysis did not find which position was significant (likely due to the number of tests required).

Drawing task completion time was defined as the time required to zoom to the desired zoom-level, and excluded drawing time. Mean task completion time was 7.64s (SD 2.10). Since users had to interact with the content after zooming in the drawing task, users appeared to spend more time positioning the target before proceeding. As with the docking task, handedness had no effect on task time ($F_{1,20} = 4.11$, NS). Starting target position had an effect on task time in drawing tasks as well ($F_{7,20} = 4.88$, $p < .01$). As with docking tasks, post hoc analysis was unable to indicate which position was significant.

Docking tasks had a mean count of 3.5 zoom-in actions (SD 4.17), 5.67 zoom-out actions (SD 5.87), and 10.94 pan actions (SD 8.98) per task. Drawing tasks had a mean count of 7.03 zoom-in actions (SD 4.72), a mean count of 1.37 zoom-out actions, and a mean count of 14.43 pan actions (SD 9.68). These are higher than study 1. For both docking and drawing tasks, hand condition (bimanual/unimanual) had no significant effect on zoom-in count ($F_{1,20} = 1.33$, NS; $F_{1,20} = 4.16$, NS) or zoom-out count ($F_{1,20} = 0.24$, NS; $F_{1,20} = 4.22$, NS). Handedness did have a significant effect on pan count for the docking task ($F_{1,20} = 67.39$, $p < 0.01$), but not on the drawing task ($F_{1,20} = 0.19$, NS). Position had no effect on zoom-in count ($F_{7,20} = 0.98$, NS) or zoom-out count ($F_{7,20} = 0.97$, NS) but did have an effect on pan count ($F_{7,20} = 22.39$, $p < .01$) with targets on the left-hand side requiring more pans than targets positioned to the centre or right-hand side. We observed participants panning targets towards their dominant hand when the target is further away, which is consistent with this data. For drawing, position affects zoom-in count ($F_{7,20} = 5.25$, $p < .01$), zoom-out counts ($F_{7,20} = 6.00$, $p < .01$), and pan counts ($F_{7,20} = 3.61$, $p < .01$). Targets at the horizontal midpoints of the display (top-centre and bottom-centre) appear to require fewer zoom actions.

5.5 Further Analysis

Multiple clutch and pan actions were observed in both of our studies, regardless of hand condition, location, or zoom direction. In this section, we further analyze zoom and pan gestures to explore ways to reduce the number of clutch and pan actions.

Given our observation of multiple zoom-in/zoom-out and pan operations, we considered whether some technique akin to cursor acceleration could be applied to pinch-to-zoom operations to reduce clutching. We analyzed zoom and pan velocities, shown in Figure 5.8, aggregated across all single-handed trials (blocks 3-4 only, to reduce learning effects). Zoom and pan gestures both exhibit a ballistic pattern, with an initial high-speed movement to resize or reposition the target. For zoom actions, an initial ballistic movement peaking at 20% of the total duration, and is followed by a longer, low-speed tail beginning at about 40% of the duration. Pan movements exhibit this same ballistic profile, but without the longer, low-speed tail.

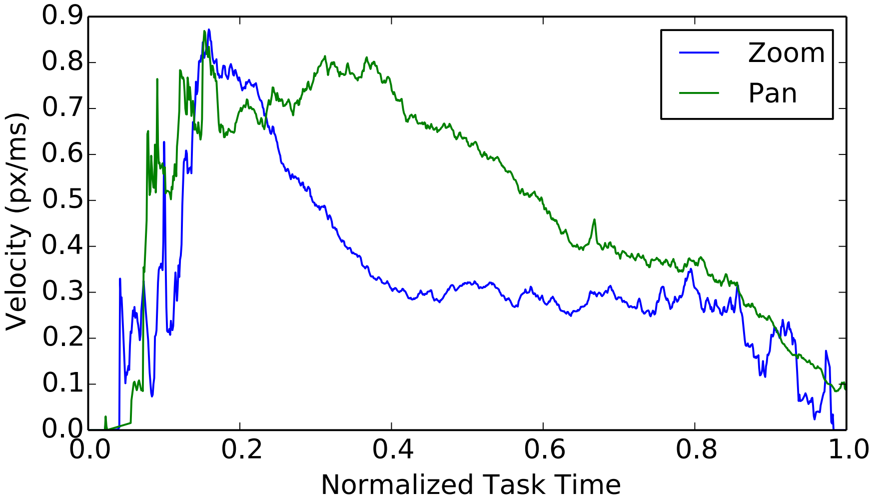


Figure 5.8: Mean pan and zoom velocity profiles for docking tasks with duration between 2681 ms to 7005 ms (mean time +/- 1 SD, normalized time for zoom-in, blocks 3 and 4 only).

Considering screen location data for both of our studies, panning was not required, but was a high-frequency operation. This suggests that participants were deliberately re-positioning content on the display. To understand why pan operations were occurring, we look at the final location of target and dock to determine why participants re-position content. At task completion, the mean distance between the target/dock and the centre of the screen is 248.9 px (SD 193.7), on a screen measuring 1280 × 800 px.

In a scatter plot of the final location of dock and target shown in Figure 5.9, we can see that target locations are grouped around the centre of the display and that locations near the edges are avoided. The observed bias to the right-of-centre in Figure 5.9 may be due to handedness of participants, but more data is needed to determine if this is significant.

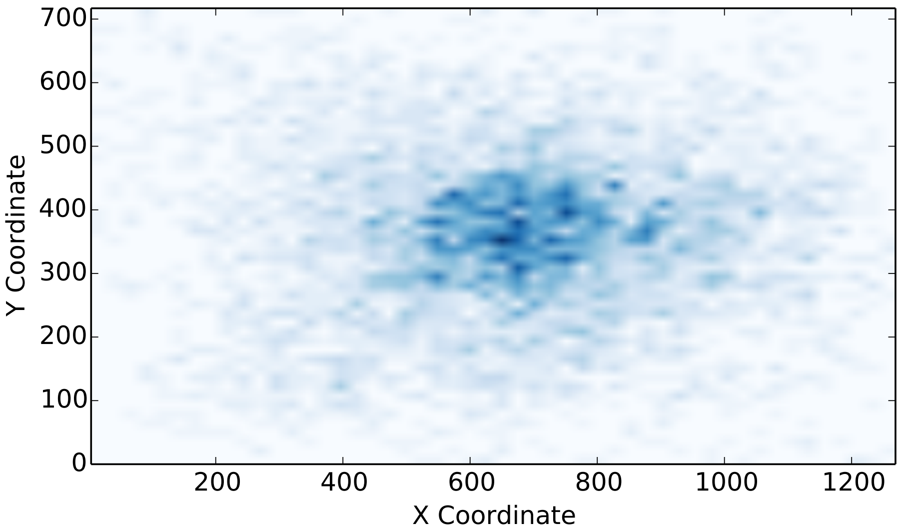


Figure 5.9: Heat map showing screen coordinates of completed experiment 2 docking and drawing tasks, demonstrating that users tend to pan the target to the centre of the display.

In the next chapter, we leverage these study insights – ballistic velocity profiles when clutching, and the tendency to pan content to the centre when zooming – to design an enhanced pinch-to-zoom technique.

Chapter 6

Pinch-to-Zoom Plus

Despite its popularity, the classic pinch-to-zoom gesture used in modern multi-touch interfaces has drawbacks: supporting movement through a range of resolutions requires clutching, and the need to keep content within the view window requires frequent panning. We apply these observations to design an enhanced zooming technique called Pinch-to-Zoom-Plus (PZP) that reduces clutching and panning operations compared to standard pinch-to-zoom behaviour. An evaluation study demonstrates the performance gains of PZP over standard pinch-to-zoom in a single trial. We also demonstrate that performance and mastery of this technique continues over a three-day period, stabilizing on day three, and consistently outperforming standard pinch-to-zoom across all three days.

6.1 Motivation

Many multi-touch gestures have analogous actions on a desktop interface: touch is similar to single-click to activate a widget, dragging is similar to scrolling to reposition content. However, the 'pinch-to-zoom' technique, based on a change in spread between two contact points, is unique, and has become the standard gesture for changing document zoom level on multi-touch devices [Hinckley et al., 1998, Wellner, 1991, Spindler et al., 2014]. This 'pinch-to-zoom' interaction is direct, simple, and intuitive, but not perfect: to reach distant zoom levels, frequent clutching is needed i.e. both fingers need to be periodically lifted to reset the distance between touch points to continue zooming. Also, specifying the intended zoom target can be difficult and often requires panning to keep the location of interest from moving off the screen while zooming. These problems are exacerbated during extreme focus+context tasks which require frequent zooming in-and-out between a close-up view to perform an action, and an overview of the entire document [Käser et al., 2011, Lank and Phan, 2004, Negulescu et al., 2011]. These problems commonly occur in tasks like documentation annotation, image manipulation or any operations that require frequent changes in resolution. Pinch-to-zoom can also pose challenges when enlarging content located near the edge of the screen because it is difficult for a user to position their fingers such that the point of magnification is centred over the content to be magnified [Wagner et al., 2012]. Zooming from the edge of the screen can cause the target to be inadvertently moved off-screen, which can require corrective pans.

Prior work has acknowledged deficiencies in pinch-to-zoom, and has suggested replacing it with new techniques to eliminate clutching [Forlines and Shen, 2005, Hinckley and Song, 2011, Malacria et al., 2010] or facilitate easier switching between an overview and close-up views [Cockburn et al., 2009]. However, these techniques are designed to replace pinch-to-zoom completely, which will be challenging given how 'natural' and 'intuitive' users tend to find it. Pinch-to-zoom is seen as *the* standard interaction technique for multi-scale navigation [Spindler et al., 2014], and it will be difficult for users to accept any replacement.

So, instead of replacing pinch-to-zoom, we designed Pinch-to-Zoom-Plus (PZP) as an enhancement of the standard pinch-to-zoom technique, designed to improve its efficiency by reducing the number of clutches and pans required to acquire a target. It is fully compatible with current pinch-to-zoom and it works regardless of device size, handedness, or whether under bimanual or unimanual control. It is designed as a subtle augmentation to the existing technique, to reduce training time and facilitate adoption of this gesture [Scarr et al., 2011].

The design parameters in PZP are based on the results of earlier formative studies, where we discovered that the velocity of pinch and spread behaviours follows a ballistic pattern, and used this to justify and parameterize pan and zoom acceleration to reduce clutching. We also observed a tendency to pan targets towards the centre of the screen, and used this to justify automatic pan-to-centre while zooming. Our results complement recent work on pinch-to-zoom performance and ergonomics [Hoggan et al., 2013a, Tran et al., 2013] and provides new insights and practical applications for results.

Note that even though the goal of this research is to directly address issues with pinch-to-zoom, pan and pinch-to-zoom are so closely related, that designing for one often means considering both gestures together. The design that we consider is strongly influenced by the observation that pan and pinch-to-zoom often co-mingle, so that we need to consider both gestures together in our design.

Finally, we also wanted to examine how learning of a novel technique occurs over time. We demonstrate that peak performance with PZP takes more than a single session to achieve, and that performance drastically improves over time to reach a stable threshold on day 3. This suggests a single session is insufficient to measure a user's capabilities when learning a novel interaction technique.

Our contributions are threefold:

1. The design of an enhanced pinch-to-zoom technique shaped by our earlier analysis.
2. The evaluation of this enhanced technique compared to baseline pinch-to-zoom.
3. The examination of learning effects over an extended period of time.

The resulting technique, PZP, reduces clutching and panning operations compared to standard pinch-to-zoom, can be implemented on all current multi-touch devices, and can be effectively mastered in a relatively short period of time, even by novice users.

6.2 Pinch-to-Zoom Plus (PZP) Design

The design of PZP was driven by two primary observations from our earlier study in Chapter 5: when zooming, users prefer small pinch and spread motions, which require clutching. Secondly, users tend to pan frequently to reposition content, sometimes to correct inadvertent target movement while zooming. PZP is designed to reduce these effects, and reduce the effort required for common pinch-to-zoom tasks. To accomplish this, we incorporate two distinct augmentations to pinch-to-zoom: zoom acceleration is a zoom transfer function that increases the relative zoom factor during rapid spreading or pinching movements, and reduces the need to clutch while zooming. Pan-to-centre is an automatic translation that moves the zoom target toward the centre of the display, and reduces the need for corrective pans. These two features are described more fully below.

6.2.1 Zoom Acceleration

With standard pinch-to-zoom, two contact points are positioned on the screen near the area of interest, and spread apart to zoom-in, or pinched together to zoom-out. This results in a linear zoom scale change proportional to the change in distance between the two contact points. However, velocity data from Study 2 in Chapter 5 shows the gesture is not performed at constant speed: pinch-to-zoom behaviour exhibits an early ballistic pattern, where the user uses an initial high-speed movement to perform larger scale manipulations, followed by a corrective phase for final positioning (Figure 5.8). This speed profile is similar to other aimed mouse movement [Casiez et al., 2008].

To reinforce this behaviour, we designed a pinch-to-zoom acceleration function to dynamically adjust a zoom scale multiplier based on the velocity of pinch and spread movement. With high velocity, the zoom scale multiplier increases the rate of zoom beyond the standard pinch-to-zoom rate. With slow velocity, the standard pinch-to-zoom zoom is used. Note that Jellinek and Card [Jellinek and Card, 1990] found that mouse acceleration functions do not provide performance benefits, although recent work has been more encouraging [Casiez et al., 2008, Hinckley et al., 2002]. More relevant to us, Casiez et al. [Casiez et al., 2008] show that mouse acceleration functions can reduce clutching. An acceleration function will let users adjust the zoom scale more quickly, and we expect that this will decrease clutching and reduce effort required to achieve a target.

Our acceleration function is not only motivated by the results of the formative studies, but uses study results to tune the function parameters. Figure 6.2 provides pseudo code for the acceleration function.

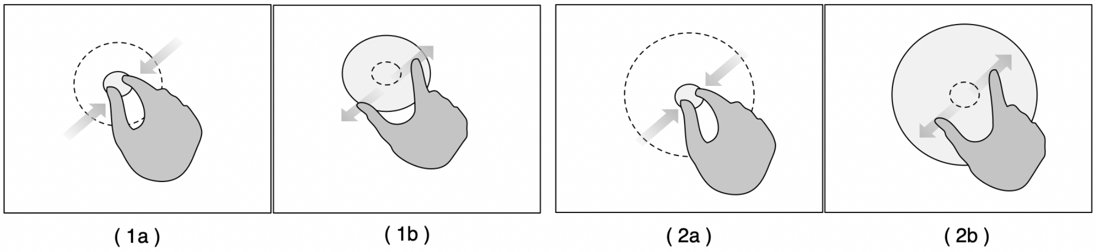
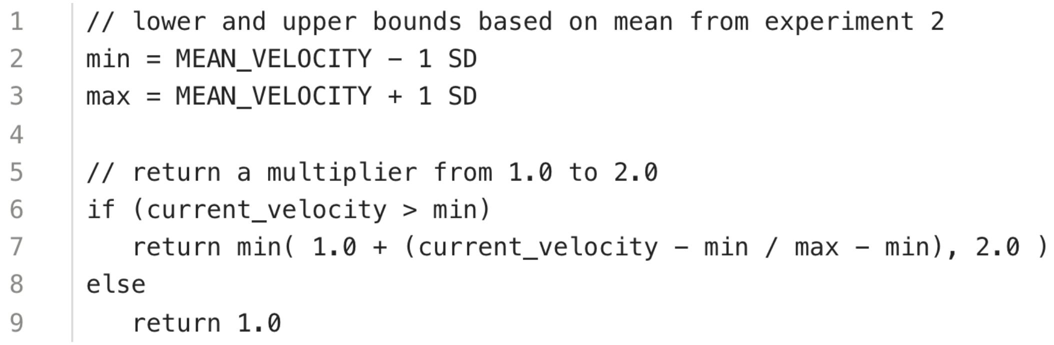


Figure 6.1: Zoom acceleration. Standard pinch-to-zoom showing pinch (1a) and spread (1b). PZP pinch (2a) and spread (2b) magnify the effects of these movements.



```
// lower and upper bounds based on mean from experiment 2
min = MEAN_VELOCITY - 1 SD
max = MEAN_VELOCITY + 1 SD

// return a multiplier from 1.0 to 2.0
if (current_velocity > min)
   return min( 1.0 + (current_velocity - min / max - min), 2.0 )
else
   return 1.0
```

Figure 6.2: Function to return the scale multiplier, which returns a value from 1.0 to 2.0 based on the current velocity.

Two thresholds, min and max, are calculated using the average pinch and spread velocity and standard deviation (SD) from study 2. Min is equal to mean velocity minus 1 SD, and max is set equal to mean velocity plus 1 SD. Study 2 yields a mean velocity of 1260 px/ms (SD 1190), creating a min threshold of 70 px/ms and a max threshold of 2450 px/ms. Using these thresholds and the current velocity of pinch or spread movement, we calculate a scale multiplier to increase the standard linear zoom scale change (or decrease it when zooming out):

- If current velocity is below min, scale multiplier = 1.0.
- If velocity is between min and max, the scale multiplier is in [1.0, 2.0], linearly interpolated by current velocity.
- If velocity is above max, scale multiplier = 2.0.

This change to the control-gain function results in acceleration at the start of the pinch-to-zoom gesture, which smoothly changes to standard linear movement during the low-speed tail of movement. This behaviour is similar to existing mouse pointer acceleration techniques [Casiez et al., 2008].

6.2.2 Pan-to-Centre

Zooming requires that the user place their fingers directly over the area of interest, which can cause target occlusion [Sears and Shneiderman, 1991]. We observed in our earlier studies, particularly in study 2 of Chapter 5, that users typically panned this area of interest to the centre of the screen before or during the zoom operation. Our goal is to partially automate this pan operation during a pinch-to-zoom gesture by moving the content of interest to the central area of the display (Figure 6.3).

Looking at the pan velocity in Figure 5.8, we note that pan follows a ballistic profile, but without the longer, low-speed tail found in zoom operations. This ballistic profile is more similar to unaimed movements of the kind characterized by Flash and Hogan [Flash and Hogan, 1985]. This suggests that participants panned toward the centre of the display

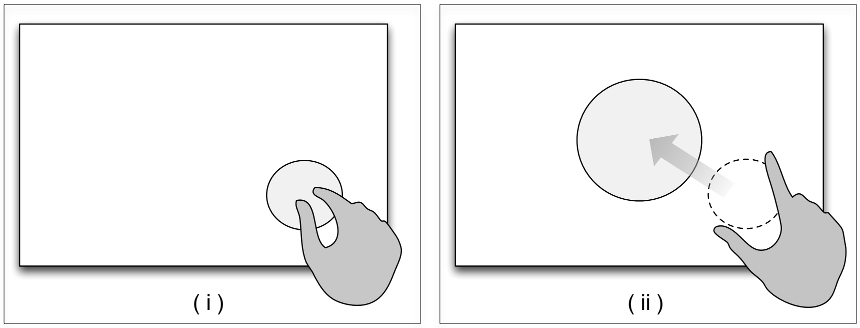


Figure 6.3: Pan-to-centre. Zooming on a target near the edge (i) causes it to be scaled and translated to the centre (ii).

without a precise goal for the final content position. They simply wished to avoid the edge of the display. The scatterplot in Figure 5.9 supports this characterization of behaviour.

Pan-to-centre is expressed as a translation applied directly to the pinch-to-zoom gesture. In study 2, we found that the mean distance from the target to the centre of the screen at task end was 247.8 px, SD 193.7. We use a similar threshold for pan-to-centre, specifically mean plus 1 SD, or 441.6 px, which represents 65% of the distance to the centre of the screen. In particular, if we examine Figure 5.9, we see that only a band around the edge of the display is avoided. This band that participants avoid represents approximately 35% of the distance from the edge of the display to the centre.

Our pan-to-centre approximates movement into this centre-region. The pan-to-centre algorithm is shown in Figure 6.4.

The algorithm performs as follows:

- A user places their fingers on the display, yielding a point of interest calculated as the mid-point between the user's fingers [Malacria et al., 2010]. This point's coordinates are $center_x, center_y$.
- The algorithm calculates a circular region in the centre of the display, as described in the previous paragraph. This is the target region for the pan.

```
// mean target distance derived from experiment 2
// defines a target region around center of screen
// auto-pan to this region, then stop panning
center_radius = MEAN_TARGET_DISTANCE - 1 SD

// remaining distance from current position to center
remaining_x = (center_x - current_x)
remaining_y = (center_y - curent_y)

// move distance of a single clutch, stopping close to center
dx = min(remaining_x / 6, remaining_x * center_radius)
dy = min(remaining_y / 6, remaining_x * center_radius)

// adjust signs so that (dx, dy) will translate to center
if current_x > center_x then dx = -dx
if current_y > center_y then dy = -dy

return dx, dy
```

Figure 6.4: Pan-to-centre distance calculation

- Next, the algorithm calculates a distance remaining between the midpoint and target (lines 7 and 8).
- The algorithm gradually pans to the centre of the display. The translation is attenuated such that the pan to centre is gradual (less than 1/6th of the remaining distance) and stops once the target point has moved into the centre region of the display (lines 11 – 16 in Figure 6.4). The 1/6th distance attenuation in our algorithm was an empirically determined value found to smooth the movement. During a pilot of this technique, we found that users preferred a gradual movement over a quick "snap" to the final target.

The goal of our design was to allow a 13× scale using a single clutch, which reduces but does not eliminate the need to clutch. To support multiple clutches, the starting midpoint $center_x, center_y$ is saved and re-used for each subsequent clutch operation. In this way, the initial target location, defined as the initial midpoint between finger contact points [Malacria et al., 2010], is panned toward the centre of the display (Figure 6.3). If no contact is made for a period of time (682 ms, the mean clutch time from study 1, + 2 SD), the midpoint resets.

6.3 Evaluation Study

We performed a controlled experiment to evaluate the performance of PZP compared to standard pinch-to-zoom. Our hypothesis is that PZP will reduce task effort, measured as the number of clutches and pans required to complete the experimental task, resulting in a reduced task time.

6.3.1 Task and Apparatus

We use the drawing task from study 2 (Figure 5.7) with the initial circular target placed at one of eight locations around the perimeter of the screen. The task is complete after the user draws a circle within the indicated region, and the zoom-level and pan are automatically reset for the next task. The same tablet from the formative studies was also used for this experiment. As in the formative studies, an overhead video camera recorded the experiment to capture additional user behaviours

6.3.2 Participants

We recruited 24 participants, 16 male (15 right-handed, 1 left-handed) and 8 female (6 right-handed and 2 left-handed) with an average age of 26.0 (SD 4.7). Participants were asked about their smartphone or tablet usage, and self-reported a mean usage of 15.3 hours per week (SD 13.8). When recruiting participants, we asked for people who "regularly used" a smartphone or tablet. Based on this filter, and the self-reported usage, we consider these users to be proficient with pinch-to-zoom. A $10 voucher was offered as remuneration.

6.3.3 Design

The experiment was a mixed between-subjects and within-subjects design. Within-subject factors were 2 technique conditions (Pinch-to-Zoom-Plus PZP and standard pinch-to-zoom PZ), 3 zoom levels (7×, 13×, 19×), one zoom direction (zoom-in), one tolerance (7 mm), and 8 positions along the perimeter of the screen (top-left, top-centre, top-right, middle-left, middle-right, bottom-left, bottom-centre, bottom-right). Hand condition (unimanual versus bimanual) was treated as a between-subject factor, with participants being randomly assigned to either the unimanual or bimanual condition. Hand condition ensures that PZP works for both unimanual and bimanual pinch-to-zoom gestures.

Technique conditions were fully crossed and presented in a randomly ordered block of trials, to balance potential learning effects. Four repetitions of each task were presented, to provide adequate time to learn the technique. Practice blocks of 12 tasks were presented at the start of each block. Not counting practice tasks, the experiment had 192 tasks per participant. At the end of the experiment, we conducted a short interview to record participant's subjective impressions of the two techniques. The experiment took 45 minutes to complete.

6.3.3.1 Pre-Processing

Excluding practice blocks, a total of 4632 tasks were performed across 24 participants. 140 tasks with an elapsed time of more than two standard deviations from the mean were excluded, which reduced the total tasks to 4492. Excessive time per trial was chosen for filtering data, as it presented the best indication of abnormal performance when completing a task (e.g. a participant stopping to ask a question mid-task, answering their mobile phone, etc.).

6.3.4 Results

Two-factor ANOVA and Bonferroni-Dunn correction were used for pairwise comparisons including the effects of handedness, acceleration and zoom level. Dependent measures were zoom-count, pan-count and elapsed time for each task.

6.3.4.1 Zoom and Pan Actions

The mean number of zoom actions was 3.01 (i.e. 2.01 clutches). PZP reduced the number of zoom actions by 21% (a 43% reduction in clutches). Tasks completed without acceleration required a mean of 3.36 zoom actions (SD 2.26), but tasks completed with the accelerated technique only required a mean of 2.65 zoom actions (SD 1.45). This was a significant effect ($F_{1,23} = 11.63$, $p < 0.01$), indicating a reduction in zoom clutch actions based on technique. There was no observed learning effect for zoom actions by block ($F_{1,23} = 0.89$, NS).

PZP also reduced the amount of panning required by 14%: tasks completed without acceleration required a mean of 7.1 pans (SD 2.64), while tasks completed with acceleration required a mean 6.1 pans (SD 2.42). This was a significant effect ($F_{1,23} = 13.80$, $p < 0.01$). There was no learning effect of panning on block ($F_{1,23} = 0.28$, NS).

Other variables (hand condition, zoom level, position) had no significant effect on zoom or pan actions.

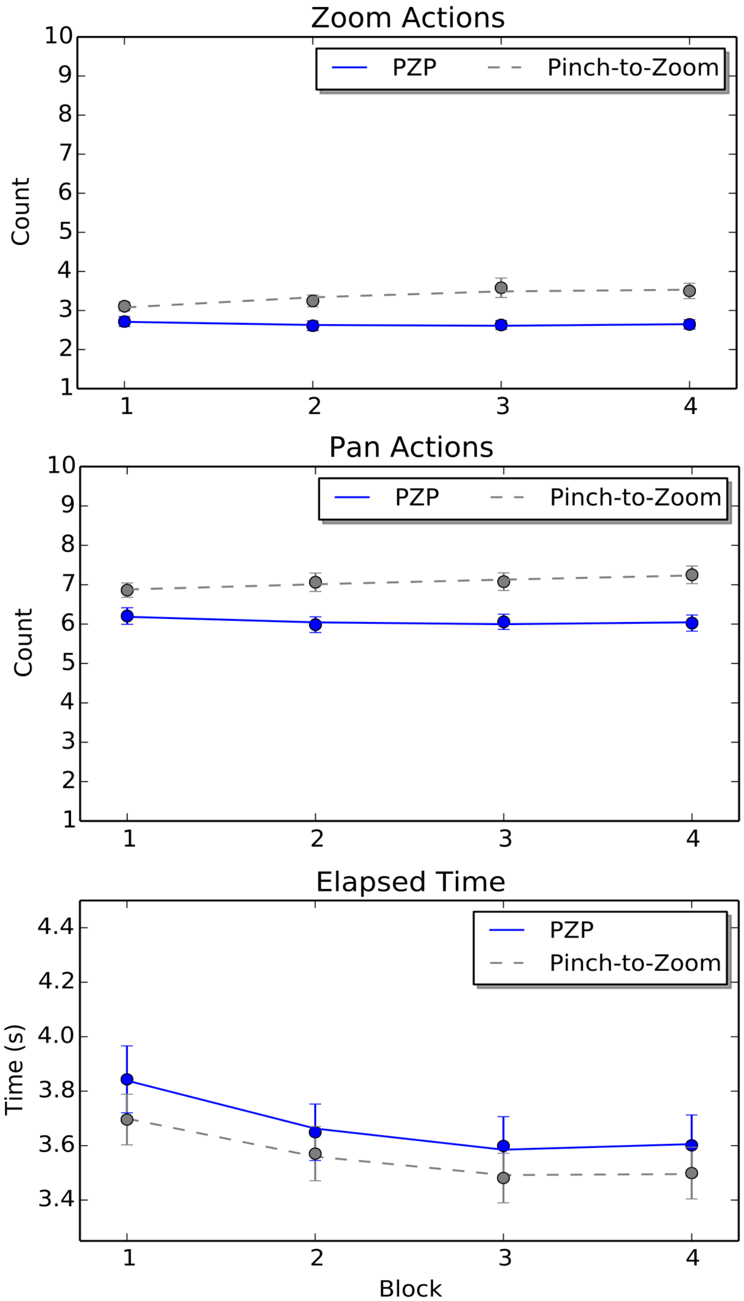


Figure 6.5: Zoom and pan count by block. PZP required far fewer zoom actions and pans than standard pinch-to-zoom, with comparable times required to complete the task.

6.3.4.2 Task Time

Mean task completion time was 3.62s (SD 1.25). Mean times for all tasks (including PZ and PZP) were 3.77s, 3.61s, 3.54s and 3.55s for blocks 1 through 4. There was a significant learning effect for elapsed time, where time improved across blocks ($F_{3,23} = 8.76$, $p < 0.01$). Posthoc analysis failed to distinguish affected blocks.

Zoom-level, as expected, directly correlated to completion time: 7x scale tasks took 3.06s, 13x took 3.65s and 19x took 4.16s to complete, which was a significant effect ($F_{2,23} = 118.45$, $p < 0.01$). Position of the target had no significant impact on task completion time ($F_{7,23} = 4.95$, NS). Technique (PZP vs. PZ) had no statistically significant effect on time ($F_{1,23} = 1.56$, NS). Presentation order of PZP vs. PZ conditions was not significant ($F_{1,23} = 1.25$, NS) suggesting no technique carry-over effect.

6.3.4.3 User Feedback

Participants seemed quick to grasp the PZP technique. Several participants indicated that it was a fairly seamless change from current pinch-to-zoom behaviour. P2 “couldn’t see a difference” between PZP and PZ behaviour. P6 also indicated that she “couldn’t really tell the difference between [PZP] and [PZ]”. Those that noticed the difference seemed to like PZP, but did not always find it easy to adjust to using it. P1 said “For single-handed operation, the [PZP] technique was much better because the automatic movement was taken care of.” P4 said “I think the [PZP] case is better when I need to move the target to the centre, like Google Maps.” However, P5 commented, “I feel like I’m fighting [PZP], but I’m getting used to it”.

6.3.5 Discussion

PZP had a significant impact on the number of clutch and pan actions. The number of clutches was reduced by 21% and the number of pans reduced by 14%. This demonstrates that PZP achieved its primary goals. We did not see a significant time saving associated with the reduced clutch and pan behaviours. However, we did see a persistent learning effect with decreasing time, and when observing our participants, we noted that their ingrained behaviour with standard pinch-to-zoom may have reduced the benefit from PZP. Many of our participants used multi-touch smartphones and tablets extensively prior to this study, and had high familiarity with the expected behaviour of standard pinch-to-zoom. The "fighting" comment from P5 above is a fair description of participants who struggled to adapt to PZP. For these reasons, we conducted a follow-up longitudinal study.

6.4 Longitudinal Study

A short evaluation is useful to assess the immediate impact of a new technique, but it's unlikely that user performance after a short trial reflects a user's final performance. With practice, we would expect additional gains – but how much of a gain, and over what period of time? Does learning progress at a smooth pace? At what point does it level out?

To address these questions, we conducted a series of longitudinal studies. Our hypothesis is that performance will continue to improve over subsequent days, and eventually reach a plateau of stable performance.

6.4.1 Pilot

6.4.1.1 Design

We ran an initial pilot to determine the appropriate characteristics for the larger study.

Using the same hardware and task as the controlled experiment, participants performed zooming tasks over a series of sessions (one session per day over five days). The tasks included 3 zoom levels (7×, 13×, 19×), one zoom direction (zoom-in), one tolerance (7 mm), and 8 positions around the perimeter of the display (top-left, top-centre, top-right, middle-left, middle-right, bottom-left, bottom-centre, bottom-right). These reflect a range of common parameters from our evaluation study. This design yielded 24 tasks per block, representing all combinations of given zoom levels and positions. All conditions were presented randomly in each block. Six task blocks (i.e. 6 repetitions), were presented each day for across five consecutive days.

Our intention was to provide concentrated bursts of training each day, with adequate time between sessions to maximize learning. By comparing user performance across days, we hoped to determine how user performance changed, i.e. how much they learned, across the five day period.

6.4.1.2 Participants

For this pilot, we recruited 5 participants, 2 male (both right-handed) and 3 female (1 right and 2 left-handed) with a mean age of 23.0 (SD 5.1). Excluding practice blocks, 6 blocks of 24 tasks were presented to each participant during each daily session (144 tasks per day, or 720 in total per participant).

All tasks were performed unimanually using the participant's dominant hand. Practice blocks of 12 tasks were presented at the start of each block; data was recorded but not included in the analysis. Log files captured task time and the number of zoom/pan actions performed.

6.4.1.3 Results

Repeated measure ANOVAs are used to determine significance, and Bonferroni-Dunn for posthoc analysis.

Zoom action count across days showed little variation: mean values ranged from 2.48 on day 1 to 2.50 on day 5. Pan action count across days also had little variation: mean values ranged from 5.97 on day 1 to 5.89 on day 5 . This suggests PZP's efficacy for consistent reduction of clutch and pan actions. However, task time decreased significantly ($F_{4,4} = 6.4$, $p < 0.01$). Over the course of 5 days, time improved by 27.1%. Mean task time fell from 3.51s on day 1 to 2.56s on day 5.

Posthoc analysis indicated that performance improved significantly from day 1 to day 2, then progression slowed and reached a quick plateau on day three. This suggests that a three-day window is adequate time for a user to become proficient in our technique. This is the window that we use in our larger, follow-up study.

6.4.2 Evaluation

6.4.2.1 Design

One challenge in any type of evaluation is determining whether participants are gaining competency in the task, or competency with a specific technique. We initially considered alternating normal and accelerated conditions in a within-subject design, so that we had a baseline for comparison, but we felt that alternating techniques would compromise learning of our specific PZP technique. However, ignoring standard pan-and-zoom (PZ) performance would give us no way to gauge how performance was objectively improving, and might lead us to incorrectly attribute task learning gains to the technique.

To address this, we 'book-end' blocks of PZP tasks with a starting and ending block of PZ tasks. i.e. our design presents 1 block of tasks using normal PZ, 6 blocks of PZP tasks, and 1 final block of normal PZ tasks each day. This gives us a baseline of task performance each day both before and immediately after training with PZP, that serves as a general measure of relative competency for that user with that task. Also, this design helps to isolate gains in task performance from gains in technique performance; if PZP performance improves over subsequent blocks, but normal performance does not change significantly, then it is reasonable to assume that the gains are related to improvement in using PZP.

Using the same hardware configuration and task parameters as the pilot study, participants performed zooming over a series of sessions (one session per day). The tasks included 3 zoom levels (7×, 13×, 19×), one zoom direction (zoom-in),one tolerance (7 mm), and 8 positions around the perimeter of the display (top-left, top-centre, top-right, middle-left, middle-right, bottom-left, bottom-centre, bottom-right). This design yielded 24 tasks per block, representing all combinations of zoom level and position. All conditions were presented randomly in each block. As discussed above, one block was presented using 'normal' PZ tasks, followed by six blocks of PZP tasks, and one final block of normal PZ tasks, for a total of 8 blocks per day. This pattern was repeated across three consecutive days.

Based on our pilot, we anticipated PZP performance to continue to improve across blocks, and flatten out on day 3. We expected some performance improvement with normal tasks, reflecting general task improvement, but did not anticipate normal task performance to improve as radically as PZP performance.

6.4.2.2 Participants

We recruited 18 participants, 10 male (9 right and 1 left handed) and 8 female (all right handed), with a mean age of 26.5 years (SD 9.0). All participants were experienced multi-touch users, but none had previously been exposed to PZP. Each day for three consecutive days, participants were presented with 2 normal blocks and 6 PZP blocks in the order described above. Excluding practice blocks, a total of 432 PZP tasks (24 per block × 6 blocks × 3 days) and 144 normal tasks (24 per block × 2 blocks × 3 days) were completed by each participant.

All tasks were performed unimanually using the participant's dominant hand. Practice blocks of 12 tasks were presented at the start of each block; data was recorded but not included in the analysis. Log files captured task time and the number of zoom/pan actions performed.

6.4.2.3 Results

Repeated measure ANOVAs are used to determine significance, and Bonferroni-Dunn for posthoc analysis.

The summary results across three days are shown in Figure 6.6. The purpose of this study was not to directly compare PZ and PZP performance, but it is useful to note that performance profiles for both techniques are consistent with previous studies (see Figure 6.5).

PZP continues to beat standard PZ performance, exhibiting 24.4% fewer zoom actions (mean 2.3 PZP vs. 3.0 PZ) and 20.2% fewer pan actions (mean 5.8 PZP vs. 6.8 PZ). Both of these results represent a significant reduction in required zoom ($F_{1,17} = 902.4$, $p < 0.01$) and pan actions ($F_{1,17} = 1120$, $p < 0.01$) based on technique.

Over a three day period, we can compare task times for both techniques. PZ improved from 3.13 s on day 1 to 2.49 s on day 3, for an improvement of 20.4%. PZP improved from 2.95 s on day 1 to 2.31 s on day 3, for an 21.6% improvement. Although performance improved with both techniques, PZP made greater gains, and became significantly faster than PZ after 3 days: mean 2.58 s for PZP compared to 2.76 for standard PZ ($F_{1,17} = 103.9$, $p < 0.01$). In other words, with sufficient training, PZP became 8.0% faster in performing our target task compared to PZ. This suggests that optimal performance with PZP cannot be achieved in a single practice session, and requires at least two sessions to achieve.

We observe that there is a significant improvement in PZP performance over the three-day period.

PZP zoom count significantly decreases from 2.42 to 2.22 actions over three days, for an 8.5% improvement over day 1 values ($F_{2,17} = 44.2$, $p < 0.01$). Pan count also decreases across days, decreasing from 5.62 to 5.37, for an 4.5% improvement over day 1 values ($F_{2,17} = 19.22$, $p < 0.01$). Posthoc analysis suggests a significant improvement on days 1 and 2.

PZP task time improved significantly from 2.95 s on day 1 to 2.31 s on day 3 ($F_{2,17} = 896.1$, $p < 0.01$). Posthoc analysis indicated that performance improved significantly across day 1 ($F_{1,17} = 25.84$, $p < 0.01$), day 2 ($F_{1,17} = 6.2$, $p < 0.05$) but failed to significantly improve on day 3. This suggests that our performance has stabilized by the third day.

However, standard PZ also demonstrates similiar improvement over this same period of time.

PZ zoom count significantly decreases from 3.26 to 2.91 actions over three days, for an 10.9% improvement over day 1 values ($F_{1,17} = 21.26$, $p < 0.01$). Pan count also decreases across days, decreasing from 7.17 to 6.65, for an 7.3% improvement over day 1 values ($F_{1,17} = 15.3$, $p < 0.01$). Posthoc analysis suggests a significant improvement on days 1 and 2.

PZ task time improved significantly from 3.13 s on day 1 to 2.49 s on day 3, for a 20.4% improvement over time ($F_{1,17} = 403.6$, $p < 0.01$). Posthoc analysis indicated that

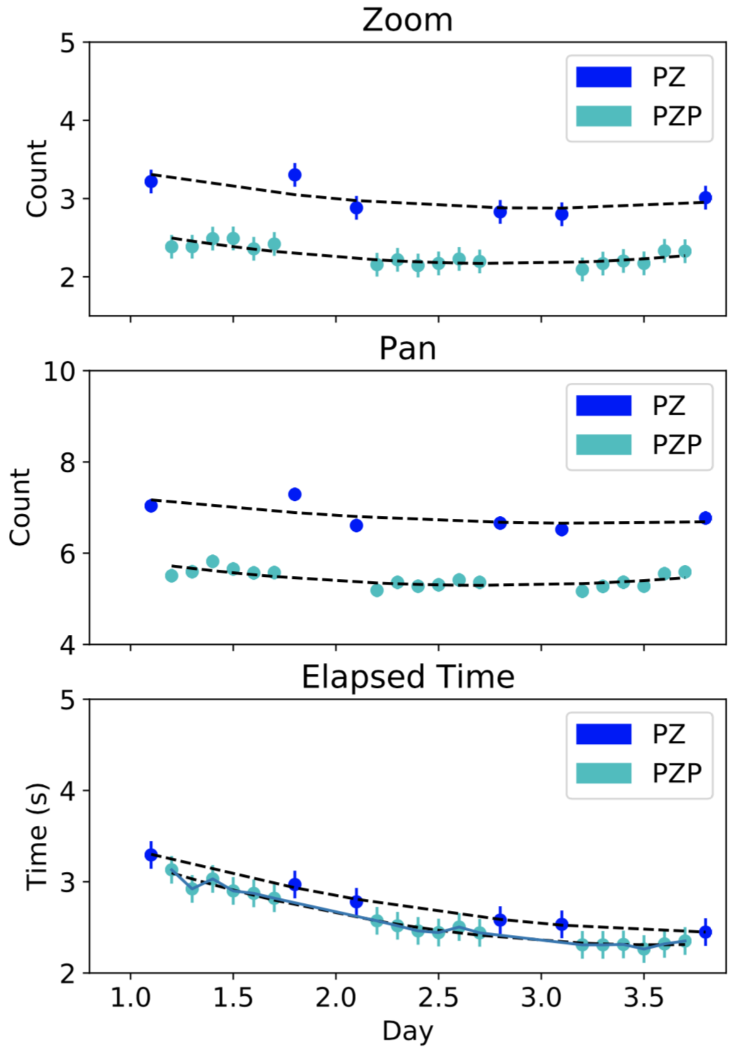


Figure 6.6: Mean action counts and task times over 3 days.

performance improved significantly across day 1 ($F_{1,17} = 31.8$, $p < 0.01$), day 2 ($F_{1,17} = 20.66$, $p < 0.01$) but failed to significantly improve on day 3. Gains are significant, but less pronounced for PZ on the final day.

One interesting observation is that gains occur in both techniques: both techniques show consistent and relatively even performance improvements over a three-day period. This was somewhat unexpected, but shows that consistent practice of this task results in task improvements regardless of the technique being used. However, as demonstrated in Figure 6.6, PZP continues to significantly dominate PZ performance across all three days, something that was not obvious after a single training session.

The relative improvements in both techniques strongly suggests that a single session is not sufficient to capture performance, but instead we need 2-3 sessions to adequately measure absolute performance, regardless of the technique being evaluated.

6.5 Discussion

We are optimistic about the performance results that we have achieved with PZP, but there are caveats: zoom acceleration was designed to reduce effort - specifically the number of pans and clutches - and not necessarily task time. As Casiez et al. and others point out [Casiez et al., 2008, Jellinek and Card, 1990, Nancel et al., 2015], accelerated CD gain functions may not always reduce task time. With that in mind, our evaluation results, showing immediate improvement in effort, are encouraging.

However, our longitudinal study demonstrates the full benefits of PZP are not immediately apparent; task time significantly improved over a two day period, stabilizing on the third day. Some reduction in effort is immediately apparent when learning the technique, but task time reduction takes more practice to achieve. Effectively learning PZP technique takes more than a single session, and optimal performance takes a full two days of training to achieve.

Overall, we were impressed that expert users were able to overcome their ingrained knowledge of standard pinch-to-zoom to harness the benefits of PZP enhancements.

In general, the PZP automatic pan assumes user always wish to keep a single point of interest visible by moving it near the centre of the screen. However, this may not be a universally applicable strategy, since not all tasks have a well-defined single point of interest. Some map navigation tasks, such as route finding, require zooming two or more points. Users can initiate PZP at the centroid of multiple points of interest and still gain a benefit, but this may not be intuitive. A technique to selectively disable automatic panning may be a useful future refinement for PZP.

6.6 Conclusions and Future Work

Since its inception pinch-to-zoom has become an effective, usable technique for enlarging content on multi-touch displays. However, as content increases in size and displays increase in resolution, the need to support ever-larger scale factors has resulted in inefficiencies.

In this chapter, we analyzed two of those inefficiencies: the performance of multiple sequential clutch operations while zooming to attain a desired scaling of content;and the performance of multiple panning translations to ensure content remains within the field of view on the display. First, we show that zoom and pan actions are quite common, even with relatively modest scale factors. Next, leveraging the kinematics and positional data associated with zoom and pan actions, we design PZP, a modified form of pinch-to-zoom that seeks to reduce these operations. Through a controlled study, we show that PZP can significantly reduce the number of zoom and pan operations.

Additionally, we present the results of training experienced users over a three-day period, and demonstrate that mastery of this technique requires more than two full training sessions, with performance stabilizing on the third day. We do not believe this is unique to PZP, but suggests that mastery of any novel technique requires more than a single training session.

Chapter 7

Transient Gestures

In this chapter, we introduce transient pan and zoom, i.e. pan and zoom manipulation gestures that temporarily alter the view and can be rapidly undone. Often working across different document view levels requires a user to rapidly navigate between these view levels, which is inefficient and time-consuming. We designed our transient gestures such that they co-exist with traditional pan and zoom interaction, and facilitate rapid movement between document states. We use an evaluation study to demonstrate that this technique reduces navigation time when working with multi-view documents, and aids rapid navigation through document view levels. We conclude with a discussion of user feedback, and directions for future research.

7.1 Motivation

Pan-and-zoom is considered the standard gesture for multi-scale navigation, used by both novice and expert users across use cases such as map navigation, drawing tasks, and document annotation. One common optimization to pinch-to-zoom is double-tap or contextual zoom, to snap between two fixed zoom levels. While this is simple to understand and easy to adopt, it requires target-aware systems [Olwal et al., 2008] that are designed around assumptions of target zoom level. This may work for very specific scenarios (e.g. map navigation), but may not be flexible enough for expert usage, where the user requires precise control over the target zoom level. Other research has been done to optimize pan-and-zoom [Avery et al., 2014, Hoggan et al., 2013a, Tran et al., 2013], but few of these optimizations have become mainstream.

Our work expands the design space around canonical gestures such as pan and pinch-to-zoom, to add functionality for expert use. Quickly changing zoom level, or shifting between areas of interest is a common strategy in information-rich displays used by expert users (e.g. real-time strategy games, image editing applications). However, these actions often suffer from inefficiencies. Previous work has demonstrated that users need to clutch excessively when performing data manipulations [Avery et al., 2014, Nancel et al., 2015]. For example, suppose a pinch is done and content is lost on the screen, or the user zooms in to some content and then wishes to re-perform the zoom to look at additional low-level details. In both cases, excessive zoom and pan operations would be needed to reposition the view and allow the user to find the correct data.

We explore the use of expert gestures that support fluid transitions between document states and eliminate the need for excessive clutching (Figure 7.1). We focus on tablet interaction, where this type of complex interaction is commonly used.

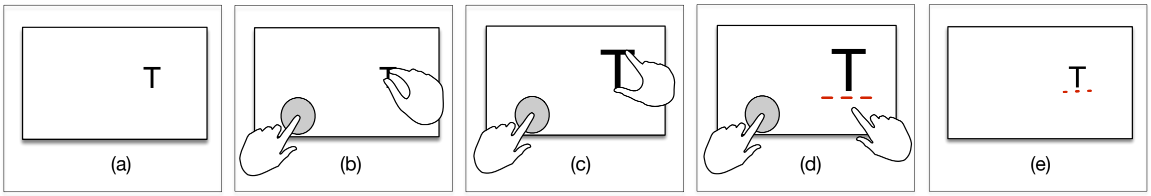


Figure 7.1: Illustration of transient zooming. The user wants to precisely annotate the letter T (a); she bookmarks the starting viewport using her non-dominant hand (b), zooms in, (c) annotates the T (d), and then lift her fingers to restore the bookmarked viewport (e). Normal operations can be performed during or after the bookmark operation.

Our transient gestures allow users to quickly undo the effects of pan and zoom operations, and support manipulations that can easily be undone during interaction without recourse to clutching. This provides support for more complex scenarios, like real-time strategy gaming (where the user needs to quickly reposition the viewport around content at different locations and resolutions), image editing, and active reading (where the user needs to navigate around a document efficiently) [Price et al., 1998]. Our hypothesis is that a transient zoom will decrease effort (e.g. number of actions) and time required to perform these types of tasks, compared to standard pan-and-zoom techniques.

These transient techniques can be considered as augmented versions of pan and pinch-to-zoom, that attempt to layer transient functionality over existing gestures. As with PZP, this technique augments existing gestures, although the introduction of a transient quasi-mode increases complexity compared to PZP.

The chapter is organized as follows. We present the design of our transient gestures, including an initial pilot and a revised design based on user feedback. We then report on an experiment where we compare standard pan-and-zoom, double-tap to zoom, and our transient pan-and-zoom, and demonstrate the performance of transient techniques over traditional navigation techniques. We present the results of a post-experiment survey, including user feedback on these techniques. Finally, we conclude with a discussion of future directions for this research.

Our contributions in this chapter consist of:

- An exploration and iteration of transient gesture designs.
- The design of bimanual gestures that support fluid movement between transient and atomic operations.
- The description and realization of a system supporting expert gestures.
- An evaluation of performance and user satisfaction with these gestures.

By way of acknowledgement, our initial design for transient pan-and-zoom was motivated by the bezel pins and swipes introduced in BiPad [Wagner et al., 2012], and our prototype used similar gestures to control the transition between states. We also leveraged previous work when deciding to define a bimanual interaction style where the off-hand anchors and defines the task for the dominant hand [Goguey et al., 2016, Guiard, 1987, Ruiz et al., 2008, Käser et al., 2011].

7.2 Transient Design

Transient pan and zoom is designed around a bookmarking metaphor, where we allow users to define a pre-specified UI state as a bookmark to which they can return. In this, it is similar to UIMarks Chapuis and Roussel [2010], a mouse-and-keyboard interaction technique where interface elements (e.g. toolbar buttons) are specified as a target where the pointer can be teleported. This required two distinct functions: the ability to save the current resolution, and a means to return to that resolution at a later time.

Inserting this functionality meant defining at least one new gesture, to activate the bookmarks. To be consistent with enhanced gesture design principles (Chapter 4), we needed to ensure that our techniques co-existed with standard pan and pinch-to-zoom. See Figure 7.2.

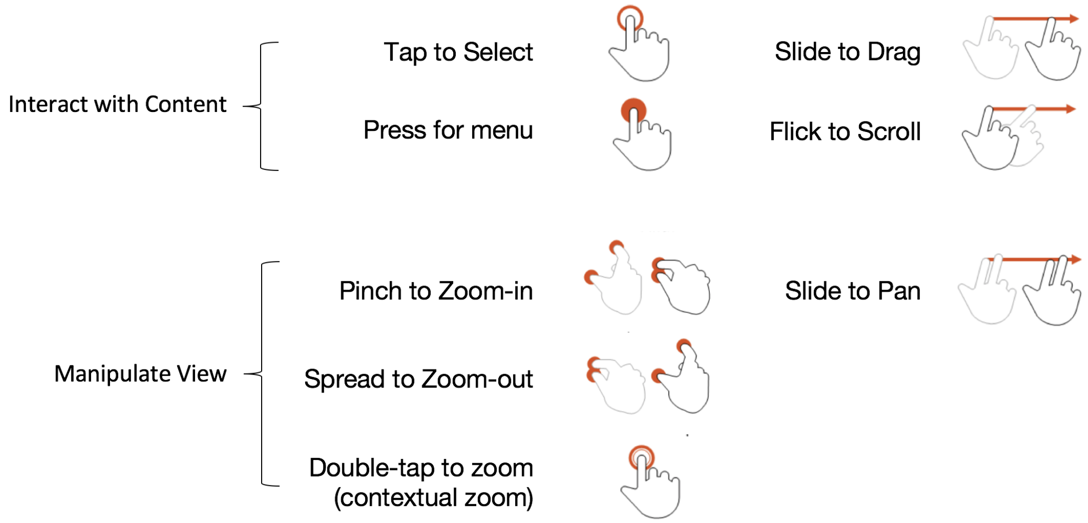


Figure 7.2: Standard gestures for multi-level navigation and interaction. Transient techniques needed to complement these existing techniques, including standard pan and pinch-to-zoom.

To determine further design constraints, we conducted a series of pilot studies, where we collected user feedback on different activation mechanisms. From this feedback, we identified the following design principles:

- Avoid automatic mode switches and allow users to explicitly control when they enter and leave transient mode.
- Use the off-hand for triggering transient mode and leave the dominant-hand free to perform normal operations.
- Respect and mirror the semantic meaning of gestures in our transient gesture (e.g. transient pan should resemble standard pan).
- Implement transient as a quasi-mode that can be layered over existing gestures.

We used this feedback to arrive at our final design, illustrated in Figure 7.1. We modelled our transient mode as a quasi-mode over the standard gestures introduced in Figure 7.2. This allowed us to to avoid any collisions, and allowed us to retain similar semantic meanings for our gestures compared to standard gestures (i.e. standard pan and transient pan were activated and behaved similarly to one another).

One-finger is reserved for interacting with the view (e.g. dragging, activating controls), while two-fingers are used for panning, zooming and other standard operations.[1]

To activate transient mode, the user presses and holds down a third finger, which saves (”bookmarks”) the current location. A shaded circle appears around the pinning finger to make the mode change clear. As long as this pin is held, the system stays in transient mode. Releasing the pin causes the view to snap back to the previous position and resolution. The pinning finger can be moved around the screen so that it does not occlude content; the user can continue to use standard gestures and interactions while maintaining the pin. In this implementation, transient can be considered a quasi-mode Li et al. [2005] that can be applied to any subsequent set of operations.

The bookmark is saved at the moment when the third contact point is placed; the user can choose to start in transient mode by ensuring that the third contact is placed

---

[1] In some mobile operating systems (e.g. iOS), many applications leverage one-finger pan. We discuss modifications to address one-finger pan in the Discussion section (see 7.4).

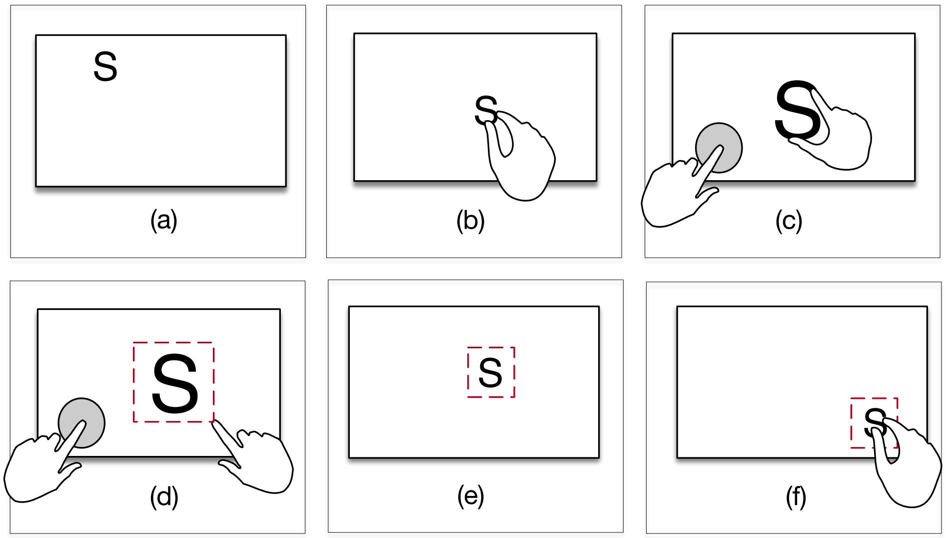


Figure 7.3: Transient pan/zoom design supports interleaving atomic and transient actions. From a starting screen (a), a user performs a two-finger pan to centre content (b); she then holds down third finger to enable transient mode, indicated by a shaded region under the pinning finger, and performs a transient zoom (c) before drawing (d). Lifting both fingers cancels transient mode and returns to the bookmarked resolution (e). She can then continue with other standard operations like panning (f).

before significant movement, or alternately, they can set in the middle of pan/zoom actions to bookmark an intermediate point. Figure 7.3 shows a more complex example, where standard and transient gestures are intermixed. The use of a quasi-mode in our design makes movement between atomic and transient states fluid.

A simplified state diagram (see Figure 7.4) illustrates the activation mechanism for this transient quasi-mode, and the transition between transient and atomic operations. A more detailed state diagram to support easy replication of our interaction technique is included in the appendix.

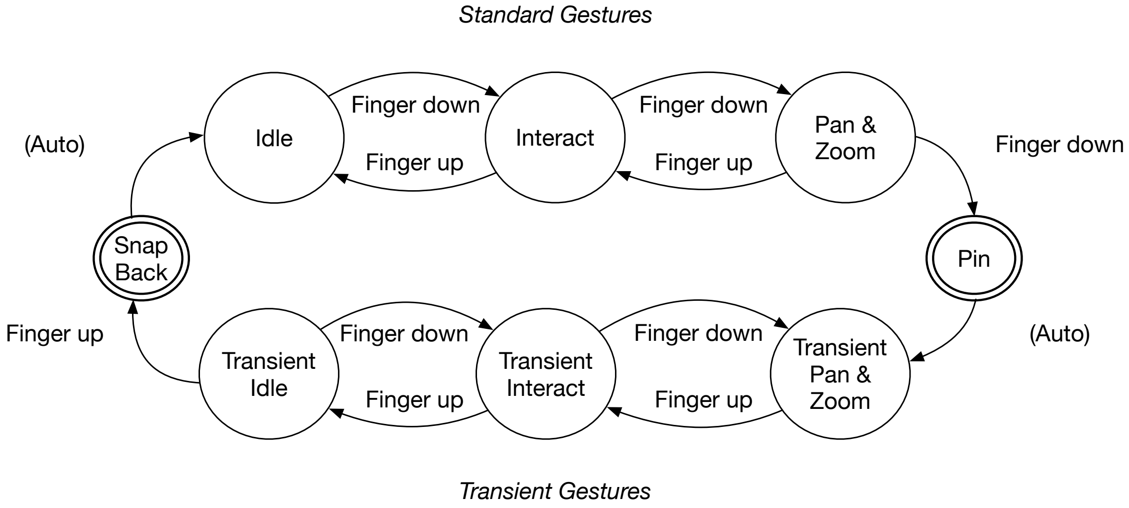


Figure 7.4: State diagram showing depicting standard (states 0-2) and transient (states 3-5) versions of common operations. Users start in state 0, and transition to different states by adding or removing fingers. Adding a third finger shifts to a transient state, where gestures are interpreted in that context; lifting all fingers returns to the normal interaction state. View interaction is supported in both states.

As is common with document and image-editing applications, this implementation makes the assumption that our system utilizes a single-finger to interact, and two fingers to pan and zoom [2].

---

[2] We are aware that some applications support one-finger pan and two-finger zoom/rotate; we discuss

In determining how to implement transient-mode, we examined the design space for our application to determine which gestures were under-utilized, and chose the off-hand pin for activation. In this case, adding a third-finger to the screen, away from the two fingers being used to pan/zoom, causes transient-mode to be activated. The factors used in this case are the number of fingers on-screen, and the distance between the active fingers and the third finger being added. In the absence of a more robust finger-tracking mechanism (using touch-aware screens or computer-vision systems) this mechanism reasonably infers which hand is being used to touch the screen based on proximity. More robust finger-identification mechanisms Masson et al. [2017] in commodity hardware might require changes in our design (which we will address further in the Discussion section).

7.3 Evaluation

To assess the effectiveness of our revised transient technique, we devised a task which simulated a real-time strategy game interface, where the user has to move between multiple resolutions to solve a task. The goal of this design was to focus on just zoom and pan actions, and minimize the other actions required for task completion.

7.3.1 Task

Our task consists of a 4×3 grid of cells on-screen, with each cell titled A through L. Each of these cells contains a list of number-letter pairs (numbers 1-9, with random letter for each number), all displayed using very small text (6 point font). One of the cells is randomly chosen as the starting cell and highlighted red, and the text is replaced with a target (e.g. ”B 7”). See Figure 7.6 for a high-level view of the task. Note that at the overview level, the text within an individual cell is so small as to be unreadable, necessitating a zoom-in to complete the task.

Figure 7.6 illustrates the steps. When the task starts, the view is already zoomed-in to the starting cell (a). The goal is to navigate to the cell identified in the starting cell

---

variants to support this interaction later in this chapter.

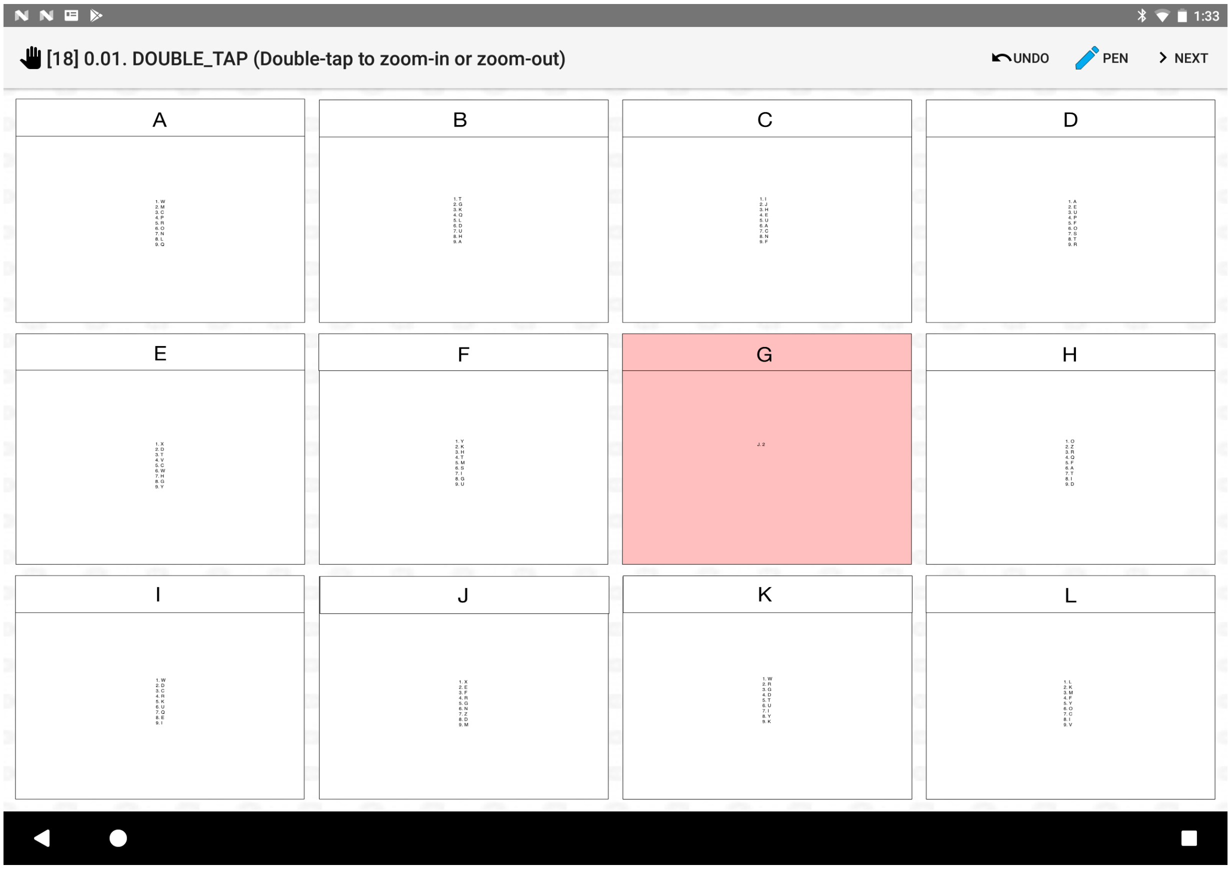
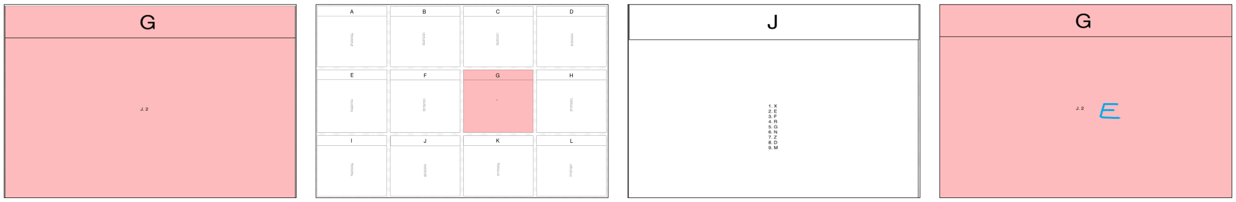


Figure 7.5: Task overview, showing the 4x3 grid of cells. The red cell is the starting cell for this particular task.

(b), find the value located in the target cell and row (c), return to the starting cell and write it down (d). For instance, in Figure 7.6, ”J 2” in the starting cell indicates that the participant needed to find the value in cell ”J”, row ”2”, which is the letter ”E”). When complete, the participants needed to press the ”next” button on-screen to move to the next task.



(a) (b) (c) (d)

Figure 7.6: Navigation task. The user is presented with a detailed view of the starting cell, which shows the target cell and row (a). The user needs to zoom-out to locate the target cell (b), zoom-in to read the value (c) and finally return to the starting cell to write down the result (d). An on-screen ”next” button (not shown) moves to the next task.

This design forced participants to move from a detail resolution in one cell (Figure 7.6a), to an overview (Figure 7.6b), and then down to a detail (Figure 7.6c) in a different cell to read the data. Next, they had to reverse the set of steps to return to the starting position (Figure 7.6d). This is similar to the sequence of steps that would be performed when navigating levels in a real-world scenarios. e.g. moving around a PDF document to write notes in the margin, consulting the full author list and title of a reference in a scientific paper, or examining the details of multiple locations on a map in, for example, command-and-control or civilization-style gaming.

7.3.2 Design

We compared our transient zooming technique to both the standard pan-and-zoom, and the omniscient double-tap (or contextual zoom), where the system automatically switches to the correct resolution during the task. These techniques were implemented following these guidelines:

- **Normal**: implemented as standard two finger pan and pinch-to-zoom, where the midpoint for the zoom action is the midpoint of the two contact points. No acceleration was implemented.
- **Double-tap**: standard single-point double-tap zooms in and out to the optimal zoom levels (i.e. snapping between overview and detail level for the cell, with cell contents centred). Pan and zoom were disabled for tasks that specified double-tap.
- **Transient**: a third finger is used to trigger transient mode, but otherwise it performs identically to normal pan and zoom (as described above).

We chose these baselines since standard two-finger pinch-to-zoom and pan is common in multi-touch (especially when a single-stroke is used for interaction such as drawing or writing), so it serves as a reasonable baseline for this comparison. Double-tap can be considered optimal, since it automatically changes to the correct resolution, and centres on the target, removing any guesswork that the user might have to perform. Both approaches are used in commercial systems in support of similar navigation tasks (e.g. tablet applications for reading two-column PDFs, or moving around parts of an image in drawing applications), though double-tap on commercial systems can occasionally select the wrong target resolution. We considered comparing our technique to accelerated pan-and-zoom techniques [Negulescu et al., 2011, Käser et al., 2011, Tran et al., 2013], but wanted to limit our investigation to explicit, user-controlled gestures.

We designed a within-subject experiment, using technique (normal, double-tap and transient), and starting cell position (A-L, or 12 positions) as conditions, and measuring task duration (ms) and number and type of discrete actions identified by our recognizer (e.g. number of pan vs. zoom actions). Our hypothesis was that duration of the task and number of zoom actions required would be reduced for transient and double-tap compared to standard pan-and-zoom. We expected double-tap to be the fastest technique (since it was designed as an optimal implementation that always guess the correct resolution and eliminates the need for continuous zooming), transient to perform less effectively than double-tap, and standard pan-and-zoom to be the least efficient technique. We did not expect starting cell position to significantly affect performance on any technique.

Each block in our design consisted of 12 tasks (representing the 12 starting positions, randomly presented), repeated across each of the three techniques. When a technique was first introduced, the experimenter demonstrated the optimal use of the technique, answered questions, and then the participant did 5 practice tasks in order to ensure he reaches his best performance for each technique. The order of techniques was counter-balanced across participants to ensure that there was no training effect between techniques (although it is a given that normal and double-tap will already be familiar to most participants). Finally, this was repeated across four blocks to check for learning effects. Each participant completed 12 tasks × 3 techniques × 4 blocks = 144 tasks. At the end of the experiment, we asked participants to rate each technique on a 1-5 Likert scale for ease-of-use and effectiveness, and also asked them to rank the techniques by preference.

We recruited 18 participants (mean age 28.9 years (SD=11.0), 7 female, 4 left handed) who were recruited from graduate students, working professionals and retirees.

7.3.3 Apparatus

Our software was written in-house in Java 1.8.0, Android 7.1.1 (Nougat) and deployed on a Pixel C tablet. The orientation was locked in landscape, and system controls were restricted so that participants could not accidentally exit a task. All data was collected into a log file on the device.

7.3.4 Results

Practice data was recorded, but excluded from our analysis, therefore we collected in total data from 144 × 18 = 2592 tasks. Repeated measure ANOVAs are used to determine significance, with Tukey correction applied for post-hoc analysis. We used Shapiro-Wilk to check normality, and for non-normal data that required a non-parametric test, we used Friedman to test significance and Mann-Whitney for post-hoc analysis.

7.3.4.1 Task Duration

We measured task duration, defined as the time in seconds required to complete each task, starting when the "next" button was pressed to start the task, and finishing when the last stroke of the character was made on-screen. The amount of time spent drawing was minimal (mean 0.44 seconds) and did not significantly vary between techniques, so variations in task duration are attributable to the time spent on navigation.

Tasks were presented in four blocks to each participant. Block has a significant effect on duration ($F_{1,18} = 142.98$, $p < 0.0001$, $\eta_p^2 = 0.54$), with mean task times progressing from 6.12s in block 1 to 4.82s in block 4. This suggests that there is an overall learning effect as participants progressed through the experiment. Task duration decreased across all of the blocks in succession, with significant differences between each block ($p < 0.001$). For this reason, we will use block 4 for further analysis, unless otherwise specified.

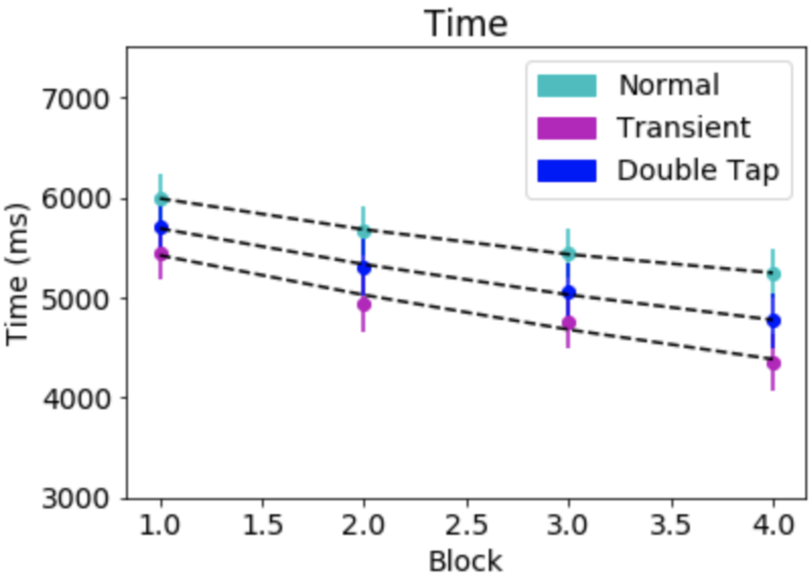


Figure 7.7: Time required for each technique across four blocks.

Technique had a significant effect on task duration ($F_{1,18} = 17.14$, $p < 0.01$, $\eta_p^2 = 0.40$), with transient and double-tap both being faster than standard techniques ($p < 0.001$).

There is no significant difference between transient and double-tap. Mean task duration was 5.32s (SD 1.72) for normal, 4.73s (SD 1.66) for double-tap and 4.40s (SD 1.58) for transient techniques. For double-tap, this represents a 11.1% improvement over normal. Transient is 17.3% faster than standard pan-and-zoom, but it is not significantly different than double-tap. This is surprising; we designed transient as an improvement over standard pan-and-zoom, but did not expect it to be comparable to our optimized double-tap (which always selects the exactly correct resolution), given that we are comparing a novel technique to a more familiar and practiced technique.

7.3.4.2 Actions

We measured the number of pan and zoom actions used to complete each task. Actions are considered to start when the users fingers are placed down, and end when they lift their fingers, or transition to a different gesture by lifting or adding a finger. This means that a user can zoom-out, pan, and zoom-in again as a single continuous gesture, if they do not remove their fingers. This mimics standard pan/zoom behaviour.

We hypothesized that zoom actions would be reduced for double-tap and transient compared to standard pan-and-zoom, but did not anticipate a reduction in pan actions when comparing techniques (since transient is not intended to eliminate panning). Pan and zoom counts were not normally distributed, so we used Friedman to test significance, and used Mann-Whitney for post-hoc analysis.

There was no significant difference in pan count when comparing normal and transient techniques. Double-tap did not allow participants to pan, so it had a constant zero pan count. Transient pan required a mean 3.31 pans, compared to 3.11 pans for standard zoom-and-pan, which represents a 6.4% difference (not significant). This is not unsurprising, since the task encouraged participants to zoom directly on the target.

As expected, there was a significant difference in zoom count by technique ($\chi^2 = 354.77$, $p < 0.001$). Post-hoc analysis reveals that this is significant for all three techniques ($p < 0.001$). Double-tap was a fixed 4.0 (SD 0.0) zoom actions, representing the fixed number of zoom actions that were enforced for that technique. Transient zoom required a 1.43 (SD 0.80) zoom actions, compared to 3.36 (SD 0.92) for standard pinch-to-zoom. Transient shows a 57.6% reduction in the number of zoom actions compared to standard techniques, and a 64.4% reduction compared to double-tap. This is better than expected performance for the transient technique, since participants still need to zoom-out to overview, and zoom-in to the target. However, both standard and transient gestures allow pan/zoom to be performed as a single continuous action, so that participants could (for instance), zoom-out and zoom-in as a single gesture. This method is available for both standard and transient techniques, but may have been leveraged more effectively with transient pan/zoom.

7.3.4.3 User Feedback

After completing all of the tasks, participants were asked to rate each of the techniques on ease-of-use and usability on a 5-point Likert scale (1-very negative to 5-very positive). The results are summarized in Figure 7.8.

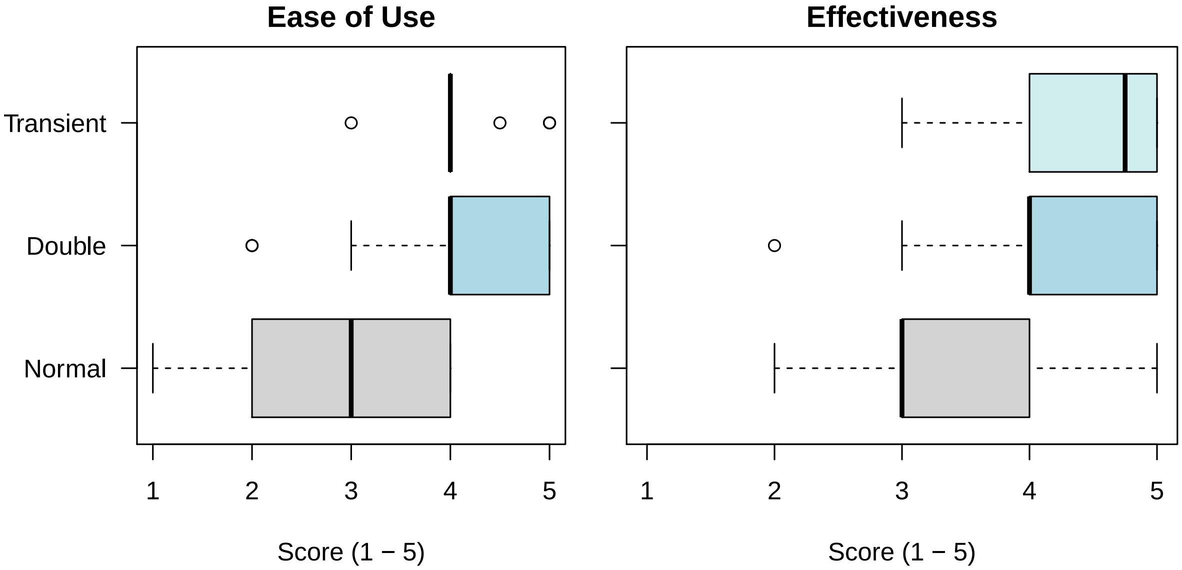


Figure 7.8: User ratings of ease-of-use and effectiveness of each techniques on a 5-point Likert scale (1-very negative to 5-very positive)

The ease of use that our participants reported varied significantly by technique ($\chi^2 = 13.46$, $p < 0.01$). Double tap ($p < 0.01$) and transient ($p < 0.01$) were both considered to be significantly easier to use than standard pan-and-zoom. There was no significant difference between them, and both were reported at 4.1 on the Likert scale (positive). Standard pan-and-zoom was reported at 2.9 (slightly less than average).

Technique also had a significant effect on the reported effectiveness ($\chi^2 = 16.26$, $p < 0.001$). Double-tap was not rated as significantly different from either of the other two techniques. Transient was considered to be more effective than standard pan-and-zoom ($p < 0.01$), but not significantly different than double-tap. Transient rated at a 4.4, compared to 4.1 for double-tap and 3.3 for standard pan-and-zoom.

Participants were also asked to rank the techniques from most preferred to least preferred. Transient pan-and-zoom were overwhelmingly the preferred techniques; standard pan-and-zoom were the least preferred. See Table 7.1:

Table 7.1: Technique Ranking

Rank	1	2	3
Transient	13	4	1
Double-Tap	5	9	4
Normal	0	5	13

When asked to explain their preferences, participants reported that they liked the transient gesture because it was easy to learn, familiar, and similar to techniques that they already knew. ”I thought the transient gesture made a lot of sense because it was [built-on standard gestures that] I already knew how to use” (P01) ”When I got used to it, I missed the bookmark when I didn’t have it. It felt like a normal extension of that technique” (P03).

Also, many participants found the transient technique to be an effective tool for this task. ”It was faster to snap back [compared to other techniques]” (P01). ”I like this, it’s much faster!” (P02). ”I liked the transient technique because I’m lazy” (P13).

Many participants also liked double-tap, but found that it felt like "more work"(P14). "Double-tap had too many taps compared to the other techniques. i.e. it takes more actions and costs more energy" (P14). However, for novice users, double-tap may be simpler. One participant, who described herself as a novice, preferred double-tap: "I like this [double-tap], it's the easiest to use, and I already know how to do it" (P09).

7.4 Discussion

Our study demonstrates that transient pan-and-zoom are more effective than standard pan-and-zoom, and are at least as effective as double-tap for navigation tasks that require switching between resolutions. Transient techniques reduce the number of zoom actions required in proportion to the amount of zooming required, so more complex tasks will see greater benefit. Double-tap is often effective, but has limitations, since it requires the system designer to anticipate what the user wants to do (i.e. is double-tap zoom-in, or out, and to what level?) Transient pan-and-zoom allows the user to specify their own target resolution and removes this limitation, making it more suitable for ad-hoc interactions.

Furthermore, users prefer transient pan-and-zoom to both standard gestures and double-tap to zoom. They find the technique easy to learn, complementing standard techniques that they already know how to use. This suggests that the transient techniques are approachable for novice to intermediate users, and may serve as an easy transition to expert use [Kurtenbach and Buxton, 1994].

One limitation to our work is the assumption of a specific gesture set. We based our design on common tasks where the user would likely use one finger to interact or draw, and two fingers to perform manipulations like pan and zoom. This is commonly used in tasks like image editing (e.g. Adobe Sketch), or document annotation (e.g. GoodReader) on tablets, where the user intends to manipulate and interact with content. Our implementation assumes no more than two contact points, where we can reliably use a third finger to active the transient quasi-mode.

Moving to a system with a different gesture set with different mappings (e.g. iOS single-finger to pan) may require changes to the activation mechanism, to ensure that there is no collision with existing gestures. For instance, on a system with single-finger pan, does a second finger switch into transient mode, or trigger a zoom? We can use proximity or some other factors to infer if this is a second finger on the same hand, intended to trigger zoom, or a finger on the other hand, intended to trigger transient mode. Alternately, we could use a more robust hand identification mechanism (e.g. computer vision) to determine which hand was used to trigger the second touch. We could also re-leverage a bezel-based activation area for explicit transient activations. BiPad, for example, suggests a number of underutilized gestures that we could employ [Wagner et al., 2012].

7.5 Conclusions and Future Work

Standard pan-and-zoom techniques suffer from inefficiencies, and often require excessive repeated transitions when performing tasks that require constant transitions between multiple resolutions and locations in a zoomable interface. We introduce a transient technique that expands the design of standard pan-and-zoom interaction to include the fluid movement between states to reduce the need for repetitive zooming to revisit previous states. In an experiment with 18 participants, we demonstrate that our technique requires 57.6% fewer zoom actions than standard pan-and-zoom, for an overall 17.3% improvement in task completion times. Transient pan-and-zoom is comparable to double-tap in ease-of-use and performance, but is found to be more effective than double-tap and preferred by our participants for navigation tasks.

Future directions for this work were suggested directly by participants after the study. When asked for ways to improve the transient techniques, some users found the offhand pin awkward, since it sometimes occluded the target (i.e. sliding the pinning finger out of the way required explicit effort). ”Sometimes having an extra finger down was awkward” (P06). ”I would use my left hand to bookmark, and sometimes it would cover the target, which made it awkward” (P18).

One design consideration might be to implement transient as an explicit mode switch, so that there was no need to hold down a finger to maintain state. However, one of the benefits of using a quasi-mode (i.e. ”pinning”) is that it makes the state explicit to users, and allows easy layering over existing gestures. Removing the pin might make the technique easier to perform at the cost of increasing the cognitive load, as users need to explicitly keep track of the state.

One other common suggestion was to combine double-tap and transient, to try and gain the benefits of both techniques (i.e. the quick-movement of double-tap in the initial zoom-out and transient to return to the starting state). We suspect that, as with the pilot, the added complexity may make the transient technique less usable, but it would be interesting to find out exactly how far we can extend this approach while still maintaining performance and usability benefits.

Finally, some participants disliked needing to use their offhand to pin. ”I don’t like using two hands” (P09). ”[It’s] not as easy coordinating two hands” (P08). Although there are benefits to bimanual interaction with complex tasks [Ruiz et al., 2008], unimanual interaction has benefits in specific situations, and could be considered for future designs.

Chapter 8

Conclusions

> The scientist is not a person who gives the right answers, he's one who asks the right questions.
>
> —– Claude Lévi-Strauss

In this chapter, we discuss the implications from previous chapters, and revisit the research questions that we proposed in Chapter 1. We also consider future directions for this research.

8.1 Introduction

Our research was grounded in the observation that, despite the popularity of multitouch devices and a desire by users to use them for complex tasks, expert multitouch gestures were under-represented in standard gesture sets, and under-utilized by users. Our fundamental goal was to explore the design space around expert gestures, and using pinch-to-zoom as an example, build more efficient techniques that users would be willing to adopt.

In the Introduction (Chapter 1), we proposed the idea of **enhanced-gestures**: *variants of standard gestures, designed to increase the efficiency or expressivity of gestural input* *languages by modifying the behaviour of familiar gestures in predictable, or learnable ways.* In other words, enhanced gestures are designed to resemble standard gestures to reduce cognitive dissonance and encourage adoption.

Chapter 3 demonstrated that users, when introduced to expert-level gestures, were eager and willing to adopt them, provided that they were easy to learn, and provided a clear benefit. We used the results from this survey in Chapter 4, where we defined the scope of our research, and established how we would apply the notion of an enhanced gesture when developing our pan and pinch-to-zoom gestures. Subsequent chapters focused on pinch-to-zoom as an example of a standard gesture that we could augment, investigated characteristics of that gesture, and designed and evaluated two enhanced pinch-to-zoom gestures.

8.2 Enhanced Gestures

In Chapters 6 and 7, we addressed research questions related to the design and implementation of enhanced gestures.

1. How can we promote the use of enhanced gestures? Are there factors inhibiting the adoption of enhanced or expert gestures?

This question was addressed by our user study in Chapter 3. We demonstrated that users were interested in expert-gestures, and willing to adopt them, but that discoverability was a key factor that inhibited their use. This study reaffirmed our belief that there was no necessary correlation between experience level and use of an expert-technique, and suggested that novice and expert techniques could be useful for users of all experience and skill levels.

2. Can we leverage these observations to design effective, enhanced versions of pinch-to-zoom? Are users willing to adopt these gestures?

This question was addressed across Chapters 5, 6 and 7.

We conducted observational studies, presented in Chapter 5, that exposed patterns in how users pan, pinch and spread during standard navigation tasks, specifically tendencies to pan-to-centre, and perform small, incremental zoom actions.

In Chapter 6, we leveraged the observations from Chapter 5, to designed and built PZP, which provided automatic pan-to-centre and zoom acceleration functions. PZP was intentionally designed as a subtle augmentation to pan and pinch-to-zoom, to make it familiar to users, and more easily learned. PZP demonstrated significant performance gains over standard pan and pinch-to-zoom in standard navigation tasks, and users were able to quickly achieve proficiency.

In Chapter 7, we designed and build Transient gestures, designed to aid multi-level navigation. Transient gestures were implemented as a quasi-mode, layered over standard pan and pinch-to-zoom; they were intended to be approachable and familiar, but took some more time to learn relative to PZP. TZ demonstrates significant performance gains in multi-level navigation tasks, and was well-received by users.

3. Can users learn and become proficient with these techniques in a reasonable timeframe?

Our last question, learnability of these gestures, was addressed during the evaluation studies in Chapters 6 and 7. In 6 in greater detail, we presented a longitudinal study over a three-day period that demonstrated that further gains could be achieved over successive training sessions. This suggests that for enhanced techniques, performance continues to improve across multiple training sessions, reaching a plateau on the third day.

8.2.0.1 Contributions

In this thesis, made the following contributions:

- *Design guidelines on facilitating transition to complex gestures by decreasing barriers to adoption.* Our notion of enhanced gestures, with the design guidelines proposed earlier, suggest a path towards designing low-friction, approachable expert gestures, and provide guidelines for future research. This approach was validated in the designs of our novel interaction techniques presented in Chapters 6 and 7.

- *An understanding of user perceptions of complex gestures, and identification of barriers to adoption of novel techniques.* Our user survey, presented in Chapter 3, validates our perception that slow adoption is mainly an issue of discoverability, and removing perceived barriers to adoption. Users are willing to adopt novel, more efficient gestures, if we can lower the barriers to adoption.
- *Foundational understanding of pan and pinch-to-zoom for typical multi-resolution tasks.* Our exploration of pan and pinch-to-zoom behaviour identified behavioural tendencies that we were able to exploit in our design of PZP.
- *Realization and evaluation of two interaction techniques that offer significant improvements to standard pinch-to-zoom.* The design and implementation of PZP (Chapter 6) and Transient Pan and Zoom (Chapter 7) provide examples of novel interaction techniques, designed as enhanced gestures, that are demonstrably efficient, usable and preferred by users.
- *An understanding of the costs of gaining expertise in a novel interaction technique.* We were surprised to discover, as part of our longitudinal study of PZP (Chapter 6) that user performance doesn't stabilize until multiple training sessions have been performed. When evaluating interaction techniques, we often make the assumption that performance stabilizes after a single session, and report those results. This study suggests that we may need to accommodate longer training periods when evaluating new techniques.

8.3 Future Work

As often happens, attempting to address our initial research questions led to more questions and areas for further exploration.

8.3.1 Designing for User Expertise

There are clear differences between skill-levels of different users, but it is unclear that characterizing users as novice or expert is appropriate, or an accurate representation of their usage patterns. Anderson's model of skill acquisition implies that users can have varying levels of expertise in related skills, so that a user may be proficient in one technique, but expert in another, for reasons that may have nothing to do with their experience level. We believe there are a number of related questions related to how skills are acquired, and how we design for issues of mixed-skills levels, and if and how we should encourage 'expert-use' among users.

- Is the classification of users into novice and expert categories sufficient? Are there more appropriate ways of characterizing users?
- How do you design for mixed skill levels? i.e. users with different skill levels with different , but related tasks.
- What denotes a novice user, and how do they differ from an expert user? Is it just the techniques that they choose to use, or can we identify patterns in their behaviour, or differences in their strategies?
- Is there such a thing as an expert-user? On what do we base this determination - expertise, strategies used, user attitudes, or some other factor?
- Do expert users apply expert strategies and behaviours in all situations? What determines the strategy that a particular user will choose at a particular time?

We believe there is a lot of work that can be done to specifically investigate how users develop expertise with related skills, that is, skills that are used together towards solving a complex task, over a prolonged period of time. As a discipline, we tend to evaluate interaction techniques in isolation, and not within the context of a larger or more complex task, where they are used in conjunction with other techniques. Observing users as they interact to solve real-world problems would provide a context to address many of the questions that we have raised. For instance, users may adopt different strategies to solve a problem based on past experience, or a mix of skills levels, where they bias their interaction towards familiar techniques.

8.3.2 Discoverability

Discoverability is well-understood to be major factor preventing adoption [Lane et al., 2005], which we confirmed in Chapter 3. As personal devices like smartphones continue to evolve, we expect user expectations for these devices to also grow. This likely means the continued introduction of new gestures, or new ways of applying standard, familiar gestures. If discoverability and adoption are issues now, they will become even more critical as we scale up to more complex multitouch interactions, in support of more complex tasks.

From a user perspective, to encourage adoption we have to address discoverability; to encourage learnability we have to address co-existence with existing gestures. We would like to address key questions about discoverability and learnability of complex gestures, at the system level:

1. Are there system-wide discoverability mechanism that we can develop to encourage exploration of multi-touch interfaces? This may need to be addressed at both the micro-level, by making gestures more explorable and "guessable", and at the macro-level, with global-level discovery and visualization mechanisms (e.g. Octopocus feed-forward mechanisms).
2. How can we extend our discoverability mechanisms to include mixed modalities? For example, can they be adapted for multi-touch with finger-tracking, or multi-touch and speech?
3. How can we scale discoverability mechanisms to a large set of gestures?

4. Do discoverability and learnability interact?

We intend to explore these questions by designing and evaluating novel gestures to encourage discoverability. For example, Wobbrock et al. [Wobbrock et al., 2009] suggest a series of gestures like copy and paste, which have not been developed commercially. By designing multitouch techniques for this type of unsupported functionality, we can assess how users discover and learn sets of techniques that are designed to work together.

8.4 Final Thoughts

It is difficult to assess user perceptions of expert gestures in-the-wild because users have not been exposed to very many; vendors have focused primarily on simple gestures that are easy to discover, and recall, and that require little training to use. The expert gestures that do exist in commercial products tend to be designed as alternatives to existing gestures, providing the same functionality but with different activation mechanisms e.g. tap-and-a-half on Google Maps to zoom-in vs. using pinch-to-zoom; the pinch -to-home gesture on iOS vs. using the home button. In a way, these gestures serve as a counterpoint to our position. Where enhanced gestures attempt to use familiar activation mechanisms and semantics, but provide additional functionality, vendor designed expert-gestures use different activation mechanisms for existing functionality. From the results in Chapter 3, we suspect that users prioritize new functionality over new activation mechanisms, and that our approach of adding advanced functions will resonate more with users, and motivate them to learn novel gestures.

Appendices

Appendix A

Expert-Gesture Survey

This appendix includes the full survey that was presented in Chapter 3.

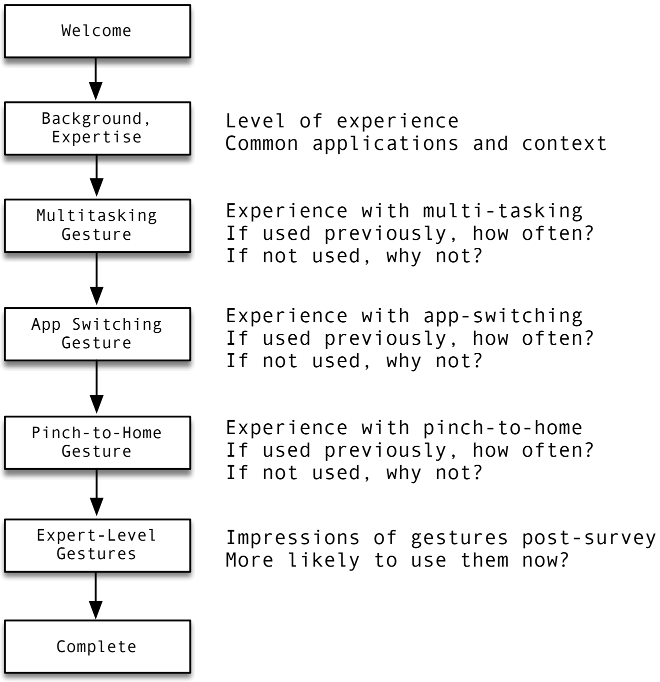


Figure A.1: Structure of the survey. Users were asked to try techniques with which they were unfamiliar.

Answers were constrained to be one of the following, depending on the type of question:

- Open-ended questions: text field
- Choose from a list: radio buttons constrained to a single choice
- Choose multiples from a list: checkboxes that allow multiple choices
- Ratings: 5-point Likert scale

A.1 Introduction

General questions about your experience using an iPad.

1. What model of iPad do you normally use?
    - iPad Air 2 (2014)
    - iPad Air (2013)
    - iPad 4 (Fall 2012)
    - iPad Mini 4 (2015)
    - iPad Mini 3 (2014)
    - iPad Mini 2 (2013)
    - Other (please specify)
2. Overall, how many years experience do you have using an iPad?
    - Less than 1 year
    - 1-2 years
    - 2-3 years
    - Other (please specify)
3. How frequently do you use your iPad?

- ○ Less than 30 mins per day
- ○ 30 - 60 mins per day
- ○ 60 - 90 mins per day
- ○ More than 90 mins per day

4. What are your main uses for your iPad? Check all that apply.
   - ♢ Playing games
   - ♢ Browsing the web/using websites
   - ♢ Watching movies/video (incl. iTunes, Netflix, Hulu)
   - ♢ Social/messaging/chat (e.g. Email, Facebook, Pinterest, Reddit, Imgur, Messenger, Snapchat)
   - ♢ Productivity apps (e.g. MS Office, Evernote)
   - ♢ Creative uses (e.g. drawing, sketching, taking pictures/movies)
   - ♢ Reading (e.g. Kindle, iBooks, Comics)
   - ♢ Listening to music (iTunes, Spotify, Pandora)
   - ♢ Other (please specify)
5. Where do you typically use your iPad? Check all that apply.
   - ♢ Work/school
   - ♢ Bathroom
   - ♢ Bedroom
   - ♢ Kitchen
   - ♢ Living room
   - ♢ Office
   - ♢ Car/bus/plane
   - ♢ Other (please specify)

6. How do you feel about your iPad? Please select one.
    - ○ Strongly dislike it
    - ○ Dislike it
    - ○ Neutral
    - ○ Like it
    - ○ Strongly like it
7. If Apple could do one thing to improve your experience with the iPad, what would it be?

A.2 Multitasking Gesture

The ”multitasking” gesture of your iPad allows you to use a swipe-up gesture with four or five fingers to show a list of running applications (you can also do this by double-pressing the home button). These questions address your experience with this feature.

8. Have you used this feature previously?
    - ○ Yes, I've used swipe-up.
    - ○ Yes, I've used double-press of the home button.
    - ○ Yes, I've used both swipe-up and double-press.
    - ○ No. I'm aware of this feature but haven't used it before.
    - ○ No. I've never heard of this feature until now.

Answering "Yes" to this question branches to Multitasking Experienced; *answering "No" branches to* Multitasking Inexperienced.

A.2.1 Multitasking Experienced

9. How often do you use swipe-up to activate multitasking?
    - I rarely use it.
    - I occasionally use it.
    - I frequently use it.
10. How do you prefer to activate the multitasking feature?
    - Swipe-up gesture.
    - Double-press the home button.
    - I use both methods and have no clear preference.

Survey continues with Multitasking Gesture (cont'd).

A.2.2 Multitasking Inexperienced

11. If you don't use the swipe-up gesture, explain why.
    - I didn't know it existed.
    - I don't need this functionality.
    - It's difficult to use.
    - Other (please specify).

Survey continues with Multitasking Gesture (cont'd).

A.2.3 Multitasking Gesture (cont'd)

If you've never used the multitasking gesture before, take a moment to try it on your iPad before continuing with the survey. Try swiping "up" with four or five fingers, to reveal a list of running applications. From this list, you can switch to an app in the list, or push an app "up" to close it.

12. Rate how you feel about this feature.
    - ○ Strongly dislike it
    - ○ Dislike it
    - ○ Neutral
    - ○ Like it
    - ○ Strongly like it
13. Rate how easy you find the swipe-up gesture to use.
    - ○ Strongly dislike it
    - ○ Dislike it
    - ○ Neutral
    - ○ Like it
    - ○ Strongly like it
14. How would you improve the swipe-up gesture?
15. Will this survey change how often you use this feature?
    - ○ Yes, I'm more likely to use multitasking now.
    - ○ Maybe, I might try it and see.
    - ○ No, it won't affect how often I use multitasking.

A.3 App Switching Gesture

The "app switching" gesture allows you to swipe left-right with four or five fingers to switch between recent applications. This section asks about your experience with this feature.

16. Have you used this feature previously?

- ○ Yes, I've used this feature.
- ○ No, I haven't used this feature before.
- ○ No, I've never heard of this feature until now.

Answering "Yes" to this question branches to App Switching Experienced; *answering "No" branches to* App Switching Inexperienced.

A.3.1 App Switching Experienced

17. How often do you use swipe-left or swipe-right to switch apps?
    - I rarely use it.
    - I occasionally use it.
    - I frequently use it.

Survey continues with App Switching Gesture (cont'd).

A.3.2 App Switching Inexperienced

18. If you don't use the app-switching gestures, explain why.
    - I didn't know they existed.
    - I don't need this functionality.
    - They're difficult to use.
    - Other (please specify).

Survey continues with App Switching Gesture (cont'd).

A.3.3 App Switching Gesture (cont'd)

If you've never used the app-switching gesture before, take a moment to try it on your iPad before continuing with the survey. Try ”swiping left” or ”swiping right” with four or five fingers, to switch between running applications.

19. Rate how you feel about this feature.
    - ○ Strongly dislike it
    - ○ Dislike it
    - ○ Neutral
    - ○ Like it
    - ○ Strongly like it
20. Rate how easy you find the app-switching gestures to use.
    - ○ Strongly dislike it
    - ○ Dislike it
    - ○ Neutral
    - ○ Like it
    - ○ Strongly like it
21. How would you improve the app-switching gestures?
22. Will this survey change how often you use this feature?
    - ○ Yes, I'm more likely to use app-switching now.
    - ○ Maybe, I might try it and see.
    - ○ No, it won't affect how often I use app-switching.

A.4 Pinch-to-Home Gesture

The ”pinch-to-home” gesture allows you to ”pinch” four or five fingers together to exit an application and return to the home screen (this can also be performed by pressing the home button). This section asks about your experience with this feature.

23. Have you used this feature previously?
    - ○ Yes, I’ve used this feature.
    - ○ No, I haven’t used this feature before.
    - ○ No, I’ve never heard of this feature until now.

Answering “Yes” to this question branches to Pinch-to-Home Experienced; *answering “No” branches to* Pinch-to-Home Inexperienced.

A.4.1 Pinch-to-Home Experienced

24. How often do you use pinch-to-home?
    - I rarely use it.
    - I occasionally use it.
    - I frequently use it.

25. How do you prefer to switch to the home screen?
    - Pinch-to-home gesture.
    - Press the home button.
    - I use both methods and have no clear preference.
    - I don’t use either method.

Survey continues with Pinch-to-Home Gesture (cont’d).

A.4.2 Pinch-to-Home Inexperienced

26. If you don't use the pinch-to-home gesture, explain why.
    - I didn't know it existed.
    - I don't need this functionality.
    - It's difficult to use.
    - Other (please specify).

Survey continues with Pinch-to-Home Gesture (cont'd).

A.4.3 Pinch-to-Home Gesture (cont'd)

If you've never used the pinch-to-home gesture before, take a moment to try it on your iPad before continuing with the survey. With an application running in the foreground, try placing four fingers on the screen and "pinching" them together to return to your home screen.

27. Rate how you feel about this feature.
    - ○ Strongly dislike it
    - ○ Dislike it
    - ○ Neutral
    - ○ Like it
    - ○ Strongly like it
28. Rate how easy you find this gesture to use.
    - ○ Strongly dislike it
    - ○ Dislike it
    - ○ Neutral

- ○ Like it
- ○ Strongly like it

29. How would you improve the pinch-to-home gesture?

30. Will this survey change how often you use this feature?

    - ○ Yes, I'm more likely to use pinch-to-home now.
    - ○ Maybe, I might try it and see.
    - ○ No, it won't affect how often I use pinch-to-home.

A.5 Impressions

31. Overall, what are your impressions of multitasking, app switching, and pinch-to-home gestures on the iPad?

    - ○ Strongly dislike it
    - ○ Dislike it
    - ○ Neutral
    - ○ Like it
    - ○ Strongly like it

32. What, if anything, do you like and dislike about these gestures?

Appendix B

System Architecture and Design

This appendix describes aspects of the system design that was used for PZP and Transient Zoom. We hope this is useful for anyone that wishes to replicate or build on this work.

B.1 Architecture

The system architecture is loosely based on Echtler et. al's multi-touch framework [Echtler and Klinker, 2008], which separates raw user input from recognizable actions, and supports customizable gestures for each action. In our case, we extended this model to include standard Model-View-Controller constructs, and used the model to track states in our application (Figure B.1).

This design allows us to distinguish actions that the user performs from gestures i.e. the command or action that we wish to activate. This provides flexibility in determining the meaning of a particular gesture based on context, and supports the idea of a discrete application state for regular, transient and drawing modes.

Based on this design, the system was designed around the following pipeline:

- System actions are delivered by the system to a Java listener class, which packages and delivers them to an ActionRecognizer. These system actions include raw pointer events like ACTION_DOWN and ACTION_MOVE, and contain raw coordinate data for all active pointers.

- The ActionRecognizer uses current state (and a list of previous actions) to generate a lightweight application action, and place that on a pending_actions queue. The results are passed to a GestureRecognizer.
- The GestureRecognizer looks at all of the pending actions and uses the current context to interpret one or more actions (matching against one or more actions allows the GestureRecognizer to handle action sequences or chords, although they are unused in this implementation). If they are classified as a gesture, the actions are consumed (removed from the pending_actions queue) and a gesture is created, and placed on the gestures queue. The new gesture is passed to the model.
- The model processes the gesture (i.e. calling on it to do any work, like panning the screen), and notifies the views to update their state.

B.2 Implementation

One challenge in designing this system is determining how to implement ”standard” features so that they behave in a familiar way. We used the build-in Android double-tap recognizer, to ensure that we used standard and familiar thresholds. Translation and pinch-to-zoom was implemented from scratch, but followed conventions and scaling factors for the platform.

The second major challenge was building a configurable recognizer that could differentiate spurious from intentional actions. At the lowest level, we filtered out small movements (less then 2 pixels, based on pilot data). Movement above this threshold was processed as an action (e.g. ACTION_DOWN, ACTION_MOVE, ACTION_UP in Android). Our gesture recognizer looked for specific pattern of actions, and application context to determine whether a gesture should be triggered. For example, panning gestures were ignored when the application was constrained to work with double-tap input.

B.3 Modelling Transient States

In designing Transient Pan and Zoom, we wanted our transient gestures to resemble their standard counterparts (e.g. transient pan compared to standard pan), but recognized that the underlying implementation of these two gestures was quite different. This drove the requirement for explicit transient and standard states. To determine if a gesture was valid, we defined a complete state model for standard and transient modes, with defined transitions between these states (see Figure B.2). Adding and removing fingers moves between states, and application state is explicitly saved and restored when we enter or leave the transient state.

We also introduced the notion of wait-states into our design, where the system is waiting for significant movement to transition to a drawing or interactive state. From the user's perspective, this helped retain the "stickiness" of the transient state, and avoid inadvertently switching modes: once in traditional draw or pan/zoom, you need to perform an explicit action to switch to transient mode. Similarly in transient mode, switching between zoom/pan and draw mode requires a user to utilize a default pinned transient state with one contact point.

From a designer's perspective, this approach made it simple to configure and remap our system while prototyping. It also makes it relatively simple to extend our system, and overlay the transient quasi-mode on other gestures like double-tap.

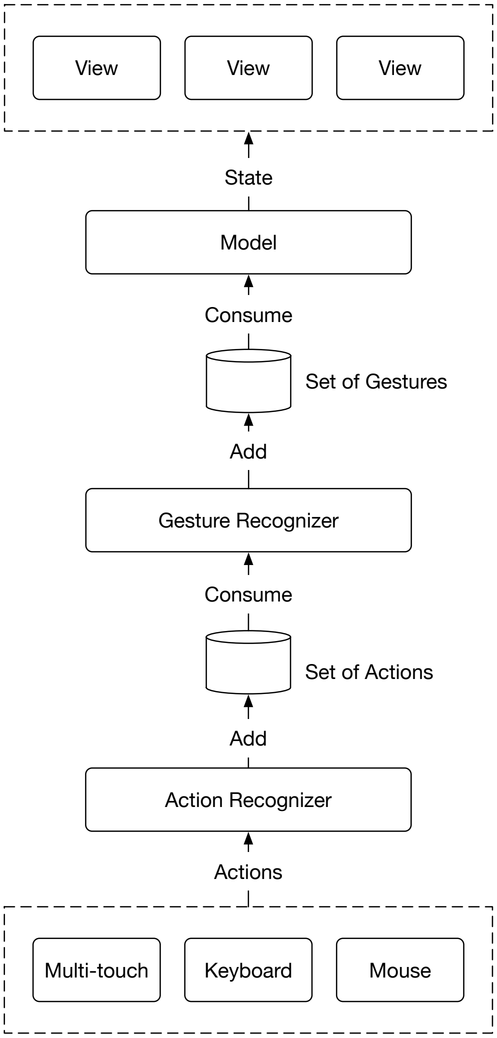


Figure B.1: An adaptation of Echtler's multi-touch architecture, showing the relationship between actions, gestures and the underlying application model.

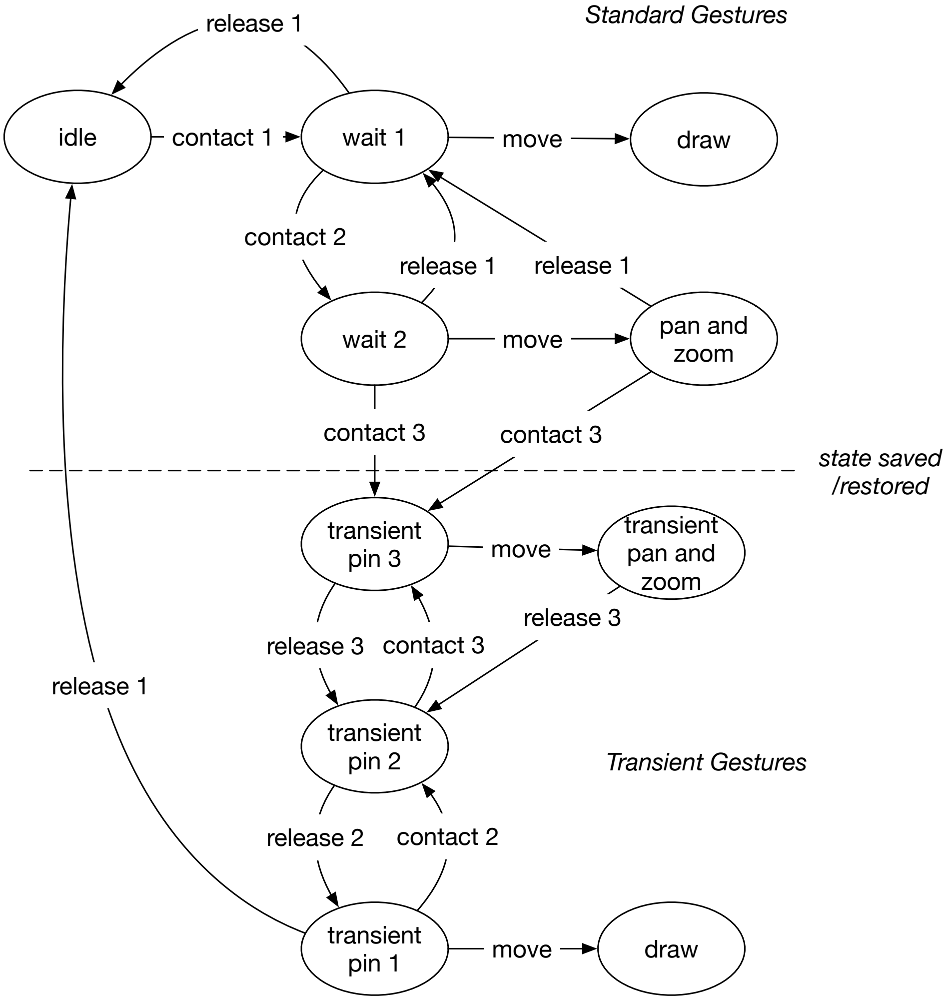


Figure B.2: Full state diagram showing idle states and movement triggered transitions. This expands the earlier diagram by explicitly showing non-movement states as pins; pan/zoom are inferred from movement. State is saved when the user moves into a transient state, and restore when moving back to the standard idle state.

References

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

Fraser Anderson and Walter F. Bischof. Learning and performance with gesture guides. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1109–1118, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2466143. URL `http://doi.acm.org/10.1145/2470654.2466143`.

J. R. Anderson. *Learning and memory: an integrated approach*. John Wiley & Sons Inc., New York, NY, USA, 2nd edition, 2000. ISBN 0471249254.

J. R. Anderson. An integrated theory of the mind. *Psychological Review*, 111(4):1036–60, 2004.

Michelle Annett, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. Medusa: A proximity-aware multi-touch tabletop. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 337–346, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047240. URL `http://doi.acm.org/10.1145/2047196.2047240`.

Michelle Annett, Anoop Gupta, and Walter F. Bischof. Exploring and understanding unintended touch during direct pen interaction. *ACM Trans. Comput.-Hum. Interact.*, 21(5):28:1–28:39, November 2014. ISSN 1073-0516. doi: 10.1145/2674915. URL `http://doi.acm.org/10.1145/2674915`.

Caroline Appert, Olivier Chapuis, and Emmanuel Pietriga. Dwell-and-spring: Undo for direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1957–1966, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208339. URL http://doi.acm.org/10.1145/2207676.2208339.

Caroline Appert, Olivier Chapuis, Emmanuel Pietriga, and María-Jesús Lobo. Reciprocal drag-and-drop. *ACM Transactions on Computer-Human Interaction*, 22(6):29:1–29:36, September 2015. ISSN 1073-0516. doi: 10.1145/2785670. URL http://doi.acm.org/10.1145/2785670.

Apple Inc. ios developer library, Dec 2015a. URL https://developer.apple.com. Accessed December 12, 2015.

Apple Inc. Adopting multitasking enhancements on ipad, Dec 2015b. URL https://developer.apple.com. Accessed December 12, 2015.

Apple Inc. Take advantage of 3d touch, 03 2018. URL https://developer.apple.com/ios/3d-touch/. Accessed 20 Mar 2018.

Patrick Lucas Austin. I love apple's 3d touch, why don't you?, April 2018. URL https://gizmodo.com/i-love-apples-3d-touch-why-don-t-you-1825603964. Accessed 4 June, 2018.

Jeff Avery and Edward Lank. Surveying Expert-Level Gesture Use and Adoption on Multi-Touch Tablets. In *DIS '16: Proceedings from the ACM Conference on Designing Interactive Systems*, New York, NY, USA, 2016. ACM. doi: 10.1145/2901790.2901895. URL http://doi.acm.org/10.1145/2901790.2901895.

Jeff Avery, Mark Choi, Daniel Vogel, and Edward Lank. Pinch-to-zoom-plus: an enhanced pinch-to-zoom that reduces clutching and panning. In *UIST '14: Proceedings from the ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2014. ACM. doi: 10.1145/2642918.2647352. URL http://doi.acm.org/10.1145/2642918.2647352.

Jeff Avery, Sylvain Malacria, Mathieu Nancel, Géry Casiez, and Edward Lank. Introducing transient gestures to improve pan and zoom on touch surfaces. In *CHI '18: Proceedings from the ACM CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2018. ACM. doi: 10.1145/3173574.3173599. URL http://doi.acm.org/10.1145/3173574.3173599.

Julie L. Baher and Bill Westerman. The usability of creativity: Experts v. novices. In *Proceedings of the Seventh ACM Conference on Creativity and Cognition*, C&C '09, pages 351–352, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-865-0. doi: 10.1145/1640233.1640296. URL http://doi.acm.org/10.1145/1640233.1640296.

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, New York, NY, USA, 2000. ACM. ISBN 1-58113-216-6. doi: 10.1145/332040.332404. URL http://doi.acm.org/10.1145/332040.332404.

Olivier Bau and Wendy E. Mackay. Octopocus: A dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, pages 37–46, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-975-3. doi: 10.1145/1449715.1449724. URL http://doi.acm.org/10.1145/1449715.1449724.

Patrick Baudisch, Edward Cutrell, Ken Hinckley, and Adam Eversole. Snap-and-go: Helping users align objects without the modality of traditional snapping. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 301–310, New York, NY, USA, 2005. ACM. ISBN 1-58113-998-5. doi: 10.1145/1054972.1055014. URL http://doi.acm.org/10.1145/1054972.1055014.

Alessio Bellino. Two new gestures to zoom: Enhancing online maps services. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 167–170, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3473-0. doi: 10.1145/2740908.2742823. URL http://doi.acm.org/10.1145/2740908.2742823.

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. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124963. URL http://doi.acm.org/10.1145/1124772.1124963.

Hrvoje Benko, Daniel Wigdor, and Christian Müller-Tomfelde. *Imprecision, Inaccuracy, and Frustration: The Tale of Touch Input*, pages 249–275. Springer New York, New York, NY, 03 2010.

Niels Ole Bernsen. *Multimodality Theory*, pages 5–29. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-78345-9. doi: 10.1007/978-3-540-78345-9_2. URL https://doi.org/10.1007/978-3-540-78345-9_2.

Mathieu Berthellemy, Elodie Cayez, Marwan Ajem, Gilles Bailly, Sylvain Malacria, and Eric Lecolinet. Spotpad, locipad, chordpad and inoutpad: Investigating gesture-based input on touchpad. In *Proceedings of the 27th Conference on L'Interaction Homme-Machine*, IHM '15, pages 4:1–4:8, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3844-8. doi: 10.1145/2820619.2820623. URL http://doi.acm.org/10.1145/2820619.2820623.

Suresh K. Bhavnani and Bonnie E. John. From sufficient to efficient usage: An analysis of strategic knowledge. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 91–98, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: 10.1145/258549.258615. URL http://doi.acm.org/10.1145/258549.258615.

Suresh K. Bhavnani, Frederick Reif, and Bonnie E. John. Beyond command knowledge: Identifying and teaching strategic knowledge for using complex computer applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pages 229–236, New York, NY, USA, 2001. ACM. ISBN 1-58113-327-8. doi: 10.1145/365024.365107. URL http://doi.acm.org/10.1145/365024.365107.

Google Mobile Ads Blog. Tablets are changing the way consumers engage with content. http://googlemobileads.blogspot.ca/2011/04/tablets-are-changing-way-consumers.html, April 2011. Accessed May 26, 2018.

Sebastian Boring, Marko Jurmu, and Andreas Butz. Scroll, tilt or move it: Using mobile phones to continuously control pointers on large public displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*, OZCHI '09, pages 161–168, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-854-4. doi: 10.1145/1738826.1738853. URL `http://doi.acm.org/10.1145/1738826.1738853`.

Frédréric Bourgeois and Yves Guiard. Multiscale pointing: Facilitating pan-zoom coordination. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, pages 758–759, New York, NY, USA, 2002. ACM. ISBN 1-58113-454-1. doi: 10.1145/506443.506583. URL `http://doi.acm.org/10.1145/506443.506583`.

Andrew Bragdon, Robert Zeleznik, Brian Williamson, Timothy Miller, and Joseph J. LaViola, Jr. Gesturebar: Improving the approachability of gesture-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2269–2278, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1519050. URL `http://doi.acm.org/10.1145/1518701.1519050`.

Bill Buxton. Multi-touch systems that i have known and loved. http://www.billbuxton.com/multi-touchOverview.html, 2007. URL `http://www.billbuxton.com/multi-touchOverview.html`. Accessed 8 February 2018.

Bill Buxton. The long nose of innovation. https://www.billbuxton.com, Jan 2009a. URL `https://www.billbuxton.com`. BusinessWeek.com. Accessed 8 February 2018.

Bill Buxton. The mad dash toward touch technology. http://www.billbuxton.com, Oct 2009b. URL `http://www.billbuxton.com`. BusinessWeek.com. Accessed 8 February 2018.

Bill Buxton. The case against lemmings and sheep. http://www.billbuxton.com, 2018. URL `http://www.billbuxton.com`. Accessed 8 February 2018.

William Buxton. Lexical and pragmatic considerations of input structures. *SIGGRAPH Comput. Graph.*, 17(1):31–37, January 1983. ISSN 0097-8930. doi: 10.1145/988584.988586. URL `http://doi.acm.org/10.1145/988584.988586`.

William Buxton. A three-state model of graphical input. In *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction*, INTERACT '90, pages 449–456, Amsterdam, The Netherlands, 1990. North-Holland Publishing Co. ISBN 0-444-88817-9. URL `http://www.billbuxton.com/3state.html`.

William Buxton, Ralph Hill, and Peter Rowley. Issues and techniques in touch-sensitive tablet input. *SIGGRAPH Comput. Graph.*, 19(3):215–224, July 1985. ISSN 0097-8930. doi: 10.1145/325165.325239. URL `http://doi.acm.org/10.1145/325165.325239`.

François Bérard and Amélie Rochet-Capellan. Measuring the linear and rotational user precision in touch pointing. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 183–192, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1209-7. doi: 10.1145/2396636.2396664. URL `http://doi.acm.org/10.1145/2396636.2396664`.

Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1983. ISBN 0898592437.

Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. Multimedia interface design. chapter The Design Space of Input Devices, pages 217–232. ACM, New York, NY, USA, 1992. ISBN 0-201-54981-6. doi: 10.1145/146022.146058. URL `http://doi.acm.org/10.1145/146022.146058`.

John M. Carroll and Mary Beth Rosson. Interfacing thought: Cognitive aspects of human-computer interaction. chapter Paradox of the Active User, pages 80–111. MIT Press, Cambridge, MA, USA, 1987. ISBN 0-262-03125-6. URL `http://dl.acm.org/citation.cfm?id=28446.28451`.

Didier Casalta, Yves Guiard, and Michel Beaudouin-Lafon. Evaluating two-handed input techniques: Rectangle editing and navigation. In *CHI '99 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '99, pages 236–237, New York, NY, USA, 1999. ACM. ISBN 1-58113-158-5. doi: 10.1145/632716.632862. URL `http://doi.acm.org/10.1145/632716.632862`.

Géry Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. The impact of control-display gain on user performance in pointing tasks. *Human–Computer Interaction*, 23(3):215–250, 2008. URL http://doi.acm.org/10.1080/07370020802278163.

Huiwen Chang and Michael F. Cohen. Panning and zooming high-resolution panoramas in virtual reality devices. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pages 279–288, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4981-9. doi: 10.1145/3126594.3126617. URL http://doi.acm.org/10.1145/3126594.3126617.

Olivier Chapuis and Nicolas Roussel. Uimarks: Quick graphical interaction with specific targets. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 173–182, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0271-5. doi: 10.1145/1866029.1866057. URL http://doi.acm.org/10.1145/1866029.1866057.

Kelvin Cheng and Dingyun Zhu. Tablet interaction techniques for viewport navigation on large displays. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 2029–2034, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2474-8. doi: 10.1145/2559206.2581148. URL http://doi.acm.org/10.1145/2559206.2581148.

Kelvin Cheng, Jane Li, and Christian Müller-Tomfelde. Supporting interaction and collaboration on large displays using tablet devices. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 774–775, New York, NY, USA, 2012a. ACM. ISBN 978-1-4503-1287-5. doi: 10.1145/2254556.2254708. URL http://doi.acm.org/10.1145/2254556.2254708.

Lung-Pan Cheng, Fang-I Hsiao, Yen-Ting Liu, and Mike Y. Chen. irotate: Automatic screen rotation based on face orientation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2203–2210, New York, NY, USA, 2012b. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208374. URL http://doi.acm.org/10.1145/2207676.2208374.

Lung-Pan Cheng, Fang-I Hsiao, Yen-Ting Liu, and Mike Y. Chen. irotate grasp: Automatic screen rotation based on grasp of mobile devices. In *Adjunct Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST Adjunct Proceedings '12, pages 15–16, New York, NY, USA, 2012c. ACM. ISBN 978-1-4503-1582-1. doi: 10.1145/2380296.2380305. URL `http://doi.acm.org/10.1145/2380296.2380305`.

Mauricio Cirelli and Ricardo Nakamura. A survey on multi-touch gesture recognition and multi-touch frameworks. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ITS '14, pages 35–44, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2587-5. doi: 10.1145/2669485.2669509. URL `http://doi.acm.org/10.1145/2669485.2669509`.

Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1):2:1–2:31, January 2009. ISSN 0360-0300. doi: 10.1145/1456650.1456652. URL `http://doi.acm.org/10.1145/1456650.1456652`.

Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. Supporting novice to expert transitions in user interfaces. *ACM Comput. Surv.*, 47(2):31:1–31:36, November 2014. ISSN 0360-0300. doi: 10.1145/2659796. URL `http://doi.acm.org/10.1145/2659796`.

Elaine B. Coleman, Randy Hellman, Desiree Davis, Kari Cafouros, and Aric Ackerman. The ipad effect, Oct 2010. URL `https://www.catalystmr.com`. Alert! Accessed on 8 February 2018.

Joëlle Coutaz, Laurence Nigay, and Daniel Salber. Multimodality from the user and system perspectives. In *Workshop on Multimedia Multimodal User Interfaces*, ERCIM'95, 1995.

Chi Tai Dang and Elisabeth André. Usage and recognition of finger orientation for multi-touch tabletop interaction. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part III*, INTERACT'11, pages 409–426, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-23764-5. URL `http://dl.acm.org/citation.cfm?id=2042182.2042219`.

Chi Tai Dang, Martin Straub, and Elisabeth André. Hand distinction for multi-touch tabletop interaction. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 101–108, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2. doi: 10.1145/1731903.1731925. URL `http://doi.acm.org/10.1145/1731903.1731925`.

Horace Dediu. Winning against non-consumption. http://www.asymco, 2015. URL `http://www.asymco.com/2015/05/11/winning-against-non-consumption/`. Accessed February 8, 2018.

Bruno Dumas, Denis Lalanne, and Sharon Oviatt. *Multimodal Interfaces: A Survey of Principles, Models and Frameworks.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-00437-7. doi: 10.1007/978-3-642-00437-7_1. URL `https://doi.org/10.1007/978-3-642-00437-7_1`.

John C. Dvorak. Understanding the impact of the iphone, 2008. URL `https://www.marketwatch.com/story/understanding-the-impact-of-the-iphone`. MarketWatch.com. Accessed 8 February 2018.

Florian Echtler and Gudrun Klinker. A multi-touch software architecture. In *Proceedings of the 5th Nordic Conference on Human-computer Interaction: Building Bridges*, NordiCHI '08, pages 463–466, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-704-9. doi: 10.1145/1463160.1463220. URL `http://doi.acm.org/10.1145/1463160.1463220`.

W. Keith Edwards, Takeo Igarashi, Anthony LaMarca, and Elizabeth D. Mynatt. A temporal model for multi-level undo and redo. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, pages 31–40, New York, NY, USA, 2000. ACM. ISBN 1-58113-212-3. doi: 10.1145/354401.354409. URL `http://doi.acm.org/10.1145/354401.354409`.

Leah Findlater, Jon E. Froehlich, Kays Fattal, Jacob O. Wobbrock, and Tanya Dastyar. Age-related differences in performance with touchscreens compared to traditional mouse input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing*

Systems, CHI '13, pages 343–346, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2470703. URL http://doi.acm.org/10.1145/2470654.2470703.

Tamar Flash and Neville Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience*, 5:1688–1703, 1985.

Clifton Forlines and Chia Shen. Dtlens: Multi-user tabletop spatial data exploration. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 119–122, New York, NY, USA, 2005. ACM. ISBN 1-59593-271-2. doi: 10.1145/1095034.1095055. URL http://doi.acm.org/10.1145/1095034.1095055.

Dustin Freeman, Hrvoje Benko, Meredith Ringel Morris, and Daniel Wigdor. Shadowguides: Visualizations for in-situ learning of multi-touch and whole-hand gestures. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 165–172, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2. doi: 10.1145/1731903.1731935. URL http://doi.acm.org/10.1145/1731903.1731935.

Emilien Ghomi, Stéphane Huot, Olivier Bau, Michel Beaudouin-Lafon, and Wendy E. Mackay. Arpège: Learning multi-touch chord gestures vocabularies. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 209–218, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512795. URL http://doi.acm.org/10.1145/2512349.2512795.

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. ISBN 978-1-4503-1580-7. doi: 10.1145/2380116.2380184. URL http://doi.acm.org/10.1145/2380116.2380184.

Mayank Goel, Alex Jansen, Travis Mandel, Shwetak N. Patel, and Jacob O. Wobbrock. Contexttype: Using hand posture information to improve mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI

'13, pages 2795–2798, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2481386. URL http://doi.acm.org/10.1145/2470654.2481386.

Alix Goguey, Géry Casiez, Daniel Vogel, Fanny Chevalier, Thomas Pietrzak, and Nicolas Roussel. A three-step interaction pattern for improving discoverability in finger identification techniques. In *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST'14 Adjunct, pages 33–34, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3068-8. doi: 10.1145/2658779.2659100. URL http://doi.acm.org/10.1145/2658779.2659100.

Alix Goguey, Mathieu Nancel, Géry Casiez, and Daniel Vogel. The performance and preference of different fingers and chords for pointing, dragging, and object transformation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4250–4261, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858194. URL http://doi.acm.org/10.1145/2858036.2858194.

Alix Goguey, Daniel Vogel, Fanny Chevalier, Thomas Pietrzak, Nicolas Roussel, and Géry Casiez. Leveraging finger identification to integrate multi-touch command selection and parameter manipulation. *International Journal of Human-Computer Studies*, 99: 21–36, March 2017. doi: 10.1016/j.ijhcs.2016.11.002. URL https://hal.inria.fr/hal-01558712.

Alix Goguey, Géry Casiez, Daniel Vogel, and Carl Gutwin. Characterizing finger pitch and roll orientation during atomic touch actions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 589:1–589:12, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5620-6. doi: 10.1145/3173574.3174163. URL http://doi.acm.org/10.1145/3173574.3174163.

Google Inc. Android developer library, Dec 2015. URL http://developer.android.com. Accessed December 12, 2015.

Wayne Gray and John Lindstedt. Plateaus, dips, and leaps: Where to look for inventions and discoveries during skilled performance. 41, 10 2016.

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. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124898. URL `http://doi.acm.org/10.1145/1124772.1124898`.

Tovi Grossman, Pierre Dragicevic, and Ravin Balakrishnan. Strategies for accelerating online learning of hotkeys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1591–1600, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9. doi: 10.1145/1240624.1240865. URL `http://doi.acm.org/10.1145/1240624.1240865`.

Tovi Grossman, George Fitzmaurice, and Ramtin Attar. A survey of software learnability: Metrics, methodologies and guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 649–658, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518803. URL `http://doi.acm.org/10.1145/1518701.1518803`.

Yves Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19:486–517, 1987.

Yves Guiard, Michel Beaudouin-Lafon, Julien Bastin, Dennis Pasveer, and Shumin Zhai. View size and pointing difficulty in multi-scale navigation. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, pages 117–124, New York, NY, USA, 2004. ACM. ISBN 1-58113-867-9. doi: 10.1145/989863.989881. URL `http://doi.acm.org/10.1145/989863.989881`.

Aakar Gupta, Thomas Pietrzak, Nicolas Roussel, and Ravin Balakrishnan. Direct manipulation in tactile displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3683–3693, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858161. URL `http://doi.acm.org/10.1145/2858036.2858161`.

Carl Gutwin and Chris Fedak. Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques. In *Proceedings of Graphics Interface*, Graphics Interface 2014, pages 1–8. Canadian Human-Computer Communications Society, May 2004.

Carl Gutwin, Andy Cockburn, and Benjamin J. Lafreniere. Testing the rehearsal hypothesis with two fasttap interfaces. In *Graphics Interface*, 2015.

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. ISBN 1-59593-271-2. doi: 10.1145/1095034.1095054. URL http://doi.acm.org/10.1145/1095034.1095054.

Jefferson Y. Han. Multi-touch interaction research. In *ACM SIGGRAPH 2006 Computer Animation Festival*, SIGGRAPH '06, New York, NY, USA, 2006. ACM. ISBN 1-59593-364-6. doi: 10.1145/1179196.1179256. URL http://doi.acm.org/10.1145/1179196.1179256.

Mark S. Hancock and Kellogg S. Booth. Improving menu placement strategies for pen input. In *Proceedings of Graphics Interface 2004*, GI '04, pages 221–230, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society. ISBN 1-56881-227-2. URL http://dl.acm.org/citation.cfm?id=1006058.1006085.

Chris Harrison, Julia Schwarz, and Scott E. Hudson. Tapsense: Enhancing finger interaction on touch surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 627–636, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047279. URL http://doi.acm.org/10.1145/2047196.2047279.

Chris Harrison, Robert Xiao, Julia Schwarz, and Scott E. Hudson. Touchtools: Leveraging familiarity and skill with physical tools to augment touch interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages

2913–2916, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557012. URL http://doi.acm.org/10.1145/2556288.2557012.

Khalad Hasan, David Ahlström, and Pourang P. Irani. Comparing direct off-screen pointing, peephole, and flick & pinch interaction for map navigation. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, SUI '15, pages 99–102, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3703-8. doi: 10.1145/2788940.2788957. URL http://doi.acm.org/10.1145/2788940.2788957.

Ken Hinckley. A background perspective on touch as a multimodal (and multisensor) construct. In Sharon Oviatt, Björn Schuller, Philip R. Cohen, Daniel Sonntag, Gerasimos Potamianos, and Antonio Krüger, editors, *The Handbook of Multimodal-Multisensor Interfaces*, pages 143–199. Association for Computing Machinery and Morgan and Claypool, New York, NY, USA, 2017. ISBN 978-1-97000-167-9. doi: 10.1145/3015783.3015789. URL https://doi.org/10.1145/3015783.3015789.

Ken Hinckley and Mike Sinclair. Touch-sensing input devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 223–230, New York, NY, USA, 1999. ACM. ISBN 0-201-48559-1. doi: 10.1145/302979.303045. URL http://doi.acm.org/10.1145/302979.303045.

Ken Hinckley and Hyunyoung Song. Sensor synaesthesia: Touch in motion, and motion in touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 801–810, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979059. URL http://doi.acm.org/10.1145/1978942.1979059.

Ken Hinckley and Daniel Wigdor. Input technologies and techniques. In Andrew Sears and Julie A. Jacko, editors, *Human-Computer Interaction Handbook*. L. Erlbaum Associates Inc., 2007.

Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. A survey of design issues in spatial input. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST '94, pages 213–222, New York, NY, USA,

1994. ACM. ISBN 0-89791-657-3. doi: 10.1145/192426.192501. URL http://doi.acm.org/10.1145/192426.192501.

Ken Hinckley, Mary Czerwinski, and Mike Sinclair. Interaction and modeling techniques for desktop two-handed input. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pages 49–58, New York, NY, USA, 1998. ACM. ISBN 1-58113-034-1. doi: 10.1145/288392.288572. URL http://doi.acm.org/10.1145/288392.288572.

Ken Hinckley, Edward Cutrell, Steve Bathiche, and Tim Muss. Quantitative analysis of scrolling techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 65–72, New York, NY, USA, 2002. ACM. ISBN 1-58113-453-3. doi: 10.1145/503376.503389. URL http://doi.acm.org/10.1145/503376.503389.

Ken Hinckley, Francois Guimbretiere, Patrick Baudisch, Raman Sarin, Maneesh Agrawala, and Ed Cutrell. The springboard: Multiple modes in one spring-loaded control. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 181–190, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124801. URL http://doi.acm.org/10.1145/1124772.1124801.

Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton O'Hara, Gavin Smyth, and William Buxton. Pre-touch sensing for mobile interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 2869–2881, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858095. URL http://doi.acm.org/10.1145/2858036.2858095.

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, 2013a. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512817. URL http://doi.acm.org/10.1145/2512349.2512817.

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, 2013b. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2481423. URL `http://doi.acm.org/10.1145/2470654.2481423`.

Christian Holz and Patrick Baudisch. Understanding touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2501–2510, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979308. URL `http://doi.acm.org/10.1145/1978942.1979308`.

Steven Hoober. How do users really hold mobile devices? https://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php, 2013. URL `https://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php`. Accessed 3 June 2018.

Kasper Hornbæk, Benjamin B. Bederson, and Catherine Plaisant. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Transactions on Computer-Human Interaction*, 9(4):362–389, December 2002. ISSN 1073-0516. doi: 10.1145/586081.586086. URL `http://doi.acm.org/10.1145/586081.586086`.

Zheng Huang, Jed Limke, and Jun Kong. A comparative study on inter-device interaction: One-handed interaction vs two-handed interaction. In *Proceedings of the 9th International Symposium on Visual Information Communication and Interaction*, VINCI '16, pages 67–74, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4149-3. doi: 10.1145/2968220.2968236. URL `http://doi.acm.org/10.1145/2968220.2968236`.

J.J. Huber. *Applying Educational Psychology in Coaching Athletes*. Human Kinetics, 2012. ISBN 9780736079815. URL `https://books.google.ca/books?id=qmWmAQAACAAJ`.

George S. Hurst. Electrographic sensor for determining planar coordinates, March 1974. US Patent 3798370.

George S. Hurst and James E. Parks. Electrical sensor of plane coordinates, May 1972. US Patent 3662105.

T. K. Philip Hwang and Horng-Yi Yu. Accommodating both expert users and novice users in one interface by utilizing multi-layer interface in complex function products. In *Proceedings of the 4th International Conference on Internationalization, Design and Global Development*, IDGD'11, pages 159–165, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-21659-6. URL http://dl.acm.org/citation.cfm?id=2028768.2028788.

Ray Hyman. Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, 45(3):188, 1953.

Apple Inc. How to use the do not disturb while driving feature, June 2018. URL https://support.apple.com/en-ca/HT208090. Accessed 3 June, 2018.

Florence Ion. From touch displays to the surface: A brief history of touchscreen technology, April 2013. URL https://arstechnica.com/gadgets/2013/04/from-touch-displays-to-the-surface-a-brief-history-of-touchscreen-technology/. Accessed March 3, 2018.

Herbert D. Jellinek and Stuart K. Card. Powermice and user performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 213–220, New York, NY, USA, 1990. ACM. ISBN 0-201-50932-6. doi: 10.1145/97243.97276. URL http://doi.acm.org/10.1145/97243.97276.

Heejin Jeong and Yili Liu. Modeling touchscreen pinch and spread gestures in a computational cognitive architecture, 05 2017.

Hans-Christian Jetter, Svenja Leifert, Jens Gerken, Sören Schubert, and Harald Reiterer. Does (multi-)touch aid users' spatial memory and navigation in 'panning' and in 'zooming & panning' uis? In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 83–90, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1287-5. doi: 10.1145/2254556.2254575. URL http://doi.acm.org/10.1145/2254556.2254575.

E.A. Johnson. Touch display—a novel input/output device for computers. *Electronics Letters*, 1:219–220(1), October 1965. ISSN 0013-5194. URL http://digital-library.theiet.org/content/journals/10.1049/el_19650200.

E.A. Johnson. Touch displays: A programmed man-machine interface. *Ergonomics*, 10 (2):271–277, 1967. doi: 10.1080/00140136708930868. URL https://doi.org/10.1080/00140136708930868.

Dietrich Kammer, Jan Wojdziak, Mandy Keck, Rainer Groh, and Severin Taranko. Towards a formalization of multi-touch gestures. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 49–58, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0399-6. doi: 10.1145/1936652.1936662. URL http://doi.acm.org/10.1145/1936652.1936662.

L.R. Kasday. Touch position sensitive surface, Nov 1984. URL http://www.google.com/patents/US4484179. US Patent 4484179.

Jesper Kjeldskov and Jeni Paay. A longitudinal review of mobile hci research methods. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, pages 69–78, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1105-2. doi: 10.1145/2371574.2371586. URL http://doi.acm.org/10.1145/2371574.2371586.

Brian Krisler and Richard Alterman. Training towards mastery: Overcoming the active user paradox. In *Proceedings of the 5th Nordic Conference on Human-computer Interaction: Building Bridges*, NordiCHI '08, pages 239–248, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-704-9. doi: 10.1145/1463160.1463186. URL http://doi.acm.org/10.1145/1463160.1463186.

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, New York, NY, USA, April 1985. ACM.

Gordon Kurtenbach and William Buxton. Issues in combining marking and direct manipulation techniques. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology*, UIST '91, pages 137–144, New York, NY, USA, 1991. ACM. ISBN 0-89791-451-1. doi: 10.1145/120782.120797. URL http://doi.acm.org/10.1145/120782.120797.

Gordon Kurtenbach and William Buxton. The limits of expert performance using hierarchic marking menus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 482–487, New York, NY, USA, 1993. ACM. ISBN 0-89791-575-5. doi: 10.1145/169059.169426. URL `http://doi.acm.org/10.1145/169059.169426`.

Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 258–264, New York, NY, USA, 1994. ACM. ISBN 0-89791-650-6. doi: 10.1145/191666.191759. URL `http://doi.acm.org/10.1145/191666.191759`.

Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. The design of a gui paradigm based on tablets, two-hands, and transparency. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 35–42, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: 10.1145/258549.258574. URL `http://doi.acm.org/10.1145/258549.258574`.

Gordon P. Kurtenbach, Abigail J.Sellen, and William A.S. Buxton. An empirical evaluation of some articulatory and cognitive aspects of marking menus. *Human Computer Interaction*, 8(1):1–23, 1993. doi: 10.1207/s15327051hci0801_1. URL `https://www.tandfonline.com/doi/abs/10.1207/s15327051hci0801_1`.

Dominik P. Käser, Maneesh Agrawala, and Mark Pauly. Fingerglass: Efficient multiscale interaction on multi-touch screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1601–1610, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979175. URL `http://doi.acm.org/10.1145/1978942.1979175`.

Benjamin Lafreniere, Carl Gutwin, and Andy Cockburn. Investigating the post-training persistence of expert interaction techniques. *ACM Trans. Comput.-Hum. Interact.*, 24 (4):29:1–29:46, August 2017. ISSN 1073-0516. doi: 10.1145/3119928. URL `http://doi.acm.org/10.1145/3119928`.

D. M. Lane, H. A. Napier, S. C. Peres, and A. Sándor. Hidden costs of graphical user interfaces: Failure to make the transition from menus and icon toolbars to keyboard shortcuts. In *International Journal of Human-Computer Interaction*, volume 18, pages 133–144, 2005. doi: http://dx.doi.org/10.1207/s15327590ijhc1802_1.

Edward Lank and Son Phan. Focus+context sketching on a pocket pc. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 1275–1278, New York, NY, USA, 2004. ACM. ISBN 1-58113-703-6. doi: 10.1145/985921.986042. URL http://doi.acm.org/10.1145/985921.986042.

Songyang Lao, Xiangan Heng, Guohua Zhang, Yunxiang Ling, and Peng Wang. A gestural interaction design model for multi-touch displays. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, BCS-HCI '09, pages 440–446, Swinton, UK, UK, 2009. British Computer Society. URL http://dl.acm.org/citation.cfm?id=1671011.1671068.

Huy Viet Le. Fully hand-and-finger-aware smartphone interaction. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI EA '18, pages DC13:1–DC13:4, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5621-3. doi: 10.1145/3170427.3173023. URL http://doi.acm.org/10.1145/3170427.3173023.

Huy Viet Le, Sven Mayer, Katrin Wolf, and Niels Henze. Finger placement and hand grasp during smartphone interaction. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 2576–2584, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4082-3. doi: 10.1145/2851581.2892462. URL http://doi.acm.org/10.1145/2851581.2892462.

S.K. Lee, William Buxton, and K. C. Smith. A multi-touch three dimensional touch-sensitive tablet. *ACM SIGCHI Bulletins*, 16(4):215–250, apr 1985.

Alexandre Lemort and Stephane Vales. Continuous recognition of multi-touch gestures, 2011. EP2148268A2.

G. Julian Lepinski, Tovi Grossman, and George Fitzmaurice. The design and evaluation of multi-touch marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2233–2242, New York, NY, USA, 2010.

ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753663. URL http://doi.acm.org/10.1145/1753326.1753663.

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. ISBN 1-58113-998-5. doi: 10.1145/1054972.1055036. URL http://doi.acm.org/10.1145/1054972.1055036.

Joern Loviscach. Two-finger input with a standard touch screen. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 169–172, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-679-0. doi: 10.1145/1294211.1294239. URL http://doi.acm.org/10.1145/1294211.1294239.

Zhihan Lv, Alaa Halawani, Muhammad Sikandar Lal Khan, Shafiq Ur Réhman, and Haibo Li. Finger in air: Touch-less interaction on smartphone. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, MUM '13, pages 16:1–16:4, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2648-3. doi: 10.1145/2541831.2541833. URL http://doi.acm.org/10.1145/2541831.2541833.

Christine L MacKenzie and Thea Iberall. *The grasping hand.* Advances in psychology, Vol. 104. North-Holland/Elsevier Science Publishers, Amsterdam, Netherlands, 1994.

I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Hum.-Comput. Interact.*, 7(1):91–139, March 1992. ISSN 0737-0024. doi: 10.1207/s15327051hci0701_3. URL http://dx.doi.org/10.1207/s15327051hci0701_3.

I. Scott Mackenzie. Motor behavior models for human-computer interaction. In J. M. Carroll, editor, *HCI Models, Theories, and Frameworks: Toward A Multidisciplinary Science*, pages 27–54. Morgan Kaufmann, San Francisco, US, 2003.

I. Scott MacKenzie, Abigail Sellen, and William A. S. Buxton. A comparison of input devices in element pointing and dragging tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 161–166, New York, NY, USA,

1991. ACM. ISBN 0-89791-383-3. doi: 10.1145/108844.108868. URL `http://doi.acm.org/10.1145/108844.108868`.

Sylvain Malacria, Eric Lecolinet, and Yves Guiard. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: The cyclostar approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2615–2624, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753724. URL `http://doi.acm.org/10.1145/1753326.1753724`.

Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. Promoting hotkey use through rehearsal with exposehk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 573–582, New York, NY, USA, 2013a. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2470735. URL `http://doi.acm.org/10.1145/2470654.2470735`.

Sylvain Malacria, Joey Scarr, Andy Cockburn, Carl Gutwin, and Tovi Grossman. Skillometers: Reflective widgets that motivate and help users to improve performance. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 321–330, New York, NY, USA, 2013b. ACM. ISBN 978-1-4503-2268-3. doi: 10.1145/2501988.2501996. URL `http://doi.acm.org/10.1145/2501988.2501996`.

Damien Masson, Alix Goguey, Sylvain Malacria, and Géry Casiez. Whichfingers: Identifying fingers on touch surfaces and keyboards using vibration sensors. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pages 41–48, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4981-9. doi: 10.1145/3126594.3126619. URL `http://doi.acm.org/10.1145/3126594.3126619`.

Katerina Eva Matsa, Amy Mitchell, Lee Rainie, Aaron Smith, and Paul Hitlin. Mobile fact sheet, 02 2018. URL `http://www.pewinternet.org/fact-sheet/mobile`. Accessed 15 March 2018.

Microsoft Inc. Use touchscreen gestures on microsoft surface — zoom, scroll, tap, copy, paste, Dec 2015. URL `https://www.microsoft.com/surface/enca/support/`

touch-mouse-and-search/using-touchgestures-tap-swipe-and-beyond. Accessed December 12, 2015.

Tsukasa Mizumata and Ryuuki Sakamoto. A pinch up gesture on multi-touch table with hover detection. In *ACM SIGGRAPH ASIA 2010 Posters*, SA '10, pages 27:1–27:1, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0524-2. doi: 10.1145/1900354.1900384. URL http://doi.acm.org/10.1145/1900354.1900384.

Rafael Morales González, Caroline Appert, Gilles Bailly, and Emmanuel Pietriga. Touchtokens: Guiding touch patterns with passive tokens. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4189–4202, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858041. URL http://doi.acm.org/10.1145/2858036.2858041.

Rafael Morales González, Caroline Appert, Gilles Bailly, and Emmanuel Pietriga. Passive yet expressive touchtokens. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3741–3745, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4655-9. doi: 10.1145/3025453.3025894. URL http://doi.acm.org/10.1145/3025453.3025894.

Juan P. Moreno, Antonio Peñalver, and Federico Botella. Evaluating the usability and use efficiency of a desktop vs tablet version of one application. In *Proceedings of the XVI International Conference on Human Computer Interaction*, Interaccion '15, pages 48:1–48:2, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3463-1. doi: 10.1145/2829875.2829917. URL http://doi.acm.org/10.1145/2829875.2829917.

Meredith Ringel Morris, Jacob O. Wobbrock, and Andrew D. Wilson. Understanding users' preferences for surface gestures. In *Proceedings of Graphics Interface 2010*, GI '10, pages 261–268, Toronto, Ont., Canada, Canada, 2010. Canadian Information Processing Society. ISBN 978-1-56881-712-5. URL http://dl.acm.org/citation.cfm?id=1839214.1839260.

Tomer Moscovich. Multi-touch interaction. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1775–1778, New York, NY, USA,

2006. ACM. ISBN 1-59593-298-4. doi: 10.1145/1125451.1125786. URL http://doi.acm.org/10.1145/1125451.1125786.

Hendrik Müller and Aaron Sedley. Designing surveys for hci research. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pages 2485–2486, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3146-3. doi: 10.1145/2702613.2706683. URL http://doi.acm.org/10.1145/2702613.2706683.

Hendrik Müller, Jennifer L. Gove, and John S. Webb. Understanding tablet use: A multi-method exploration. In *Proceedings of the 14th Conference on Human-Computer Interaction with Mobile Devices and Services (Mobile HCI 2012)*, New York, NY, USA, 2012.

Hendrik Müller, Aaron Sedley, and Elizabeth Ferrall-Nunge. Survey research in hci. In Judith S. Olson and Wendy A. Kellogg, editors, *Ways of Knowing in HCI*, pages 229–266. Springer New York, New York, NY, 2014. ISBN 978-1-4939-0378-8. doi: 10.1007/978-1-4939-0378-8_10. URL https://doi.org/10.1007/978-1-4939-0378-8_10.

Hendrik Müller, Jennifer L. Gove, John S. Webb, and Aaron Cheang. Understanding and comparing smartphone and tablet use: Insights from a large-scale diary study. In *Proceedings of the 27th Australian Computer-Human Interaction Conference (OzCHI 2015)*, pages 427–436, New York, NY, USA, 2015.

Miguel A. Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. Memorability of pre-designed and user-defined gesture sets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1099–1108, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2466142. URL http://doi.acm.org/10.1145/2470654.2466142.

Vicente Nacher, Alfredo Ferreira, Javier Jaen, and Fernando Garcia-Sanjuan. Are kindergarten children ready for indirect drag interactions? In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*, ISS '16, pages 95–101, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4248-3. doi: 10.1145/2992154.2992186. URL http://doi.acm.org/10.1145/2992154.2992186.

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. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1978969. URL http://doi.acm.org/10.1145/1978942.1978969.

Mathieu Nancel, Daniel Vogel, and Edward Lank. Clutching is not (necessarily) the enemy. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 4199–4202, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702134. URL http://doi.acm.org/10.1145/2702123.2702134.

Matei Negulescu, Jaime Ruiz, and Edward Lank. Zoompointing revisited: Supporting mixed-resolution gesturing on interactive surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, pages 150–153, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0871-7. doi: 10.1145/2076354.2076382. URL http://doi.acm.org/10.1145/2076354.2076382.

A. Newell and P. S. Rosenbloom. The soar papers (vol. 1). chapter Mechanisms of Skill Acquisition and the Law of Practice, pages 81–135. MIT Press, Cambridge, MA, USA, 1993. ISBN 0-262-68071-8. URL http://dl.acm.org/citation.cfm?id=162580.162586.

Alexander Ng, John Williamson, and Stephen Brewster. The effects of encumbrance and mobility on touch-based gesture interactions for mobile phones. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, pages 536–546, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3652-9. doi: 10.1145/2785830.2785853. URL http://doi.acm.org/10.1145/2785830.2785853.

Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 0125184050.

Erik Nilsen, HeeSen Jong, Judith S. Olson, Kevin Biolsi, Henry Rueter, and Sharon Mutter. The growth of software skill: A longitudinal look at learning & performance. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human*

Factors in Computing Systems, CHI '93, pages 149–156, New York, NY, USA, 1993. ACM. ISBN 0-89791-575-5. doi: 10.1145/169059.169126. URL http://doi.acm.org/10.1145/169059.169126.

Don Norman and Bruce Tognazzini. How apple is giving design a bad name, November 2015. URL https://www.fastcodesign.com/3053406/how-apple-is-giving-design-a-bad-name. Accessed 6 June, 2018.

Donald A. Norman. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA, 2002. ISBN 9780465067107.

Chris North, Tim Dwyer, Bongshin Lee, Danyel Fisher, Petra Isenberg, George Robertson, and Kori Inkpen. Understanding multi-touch manipulation for surface computing. In Tom Gross, Jan Gulliksen, Paula Kotzé, Lars Oestreicher, Philippe Palanque, Raquel Oliveira Prates, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2009*, pages 236–249, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-03658-3.

Daniel L. Odell, Richard C. Davis, Andrew Smith, and Paul K. Wright. Toolglasses, marking menus, and hotkeys: A comparison of one and two-handed command selection techniques. In *Proceedings of Graphics Interface 2004*, GI '04, pages 17–24, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society. ISBN 1-56881-227-2. URL http://dl.acm.org/citation.cfm?id=1006058.1006061.

Halla Olafsdottir and Caroline Appert. Multi-touch gestures for discrete and continuous control. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, AVI '14, pages 177–184, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2775-6. doi: 10.1145/2598153.2598169. URL http://doi.acm.org/10.1145/2598153.2598169.

Alex Olwal, Steven Feiner, and Susanna Heyman. Rubbing and tapping for precise and rapid selection on touch-screen displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 295–304, New York, NY, USA,

2008. ACM. ISBN 978-1-60558-011-1. doi: 10.1145/1357054.1357105. URL http://doi.acm.org/10.1145/1357054.1357105.

Michel Pahud, Ken Hinckley, Shamsi Iqbal, Abigail Sellen, and Bill Buxton. Toward compound navigation tasks on mobiles via spatial manipulation. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 113–122, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2273-7. doi: 10.1145/2493190.2493210. URL http://doi.acm.org/10.1145/2493190.2493210.

Henri Palleis and Vanessa Niedermeier. Exploring two-handed indirect multi-touch input for a docking task. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, pages 36–39, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4131-8. doi: 10.1145/2909132.2909285. URL http://doi.acm.org/10.1145/2909132.2909285.

J.A. Pickering. Touch-sensitive screens: The technologies and their application. *Int. J. Man-Mach. Stud.*, 25(3):249–269, September 1986. ISSN 0020-7373. doi: 10.1016/S0020-7373(86)80060-8. URL http://dx.doi.org/10.1016/S0020-7373(86)80060-8.

Benjamin Poppinga, Alireza Sahami Shirazi, Niels Henze, Wilko Heuten, and Susanne Boll. Understanding shortcut gestures on mobile touch devices. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services*, MobileHCI '14, pages 173–182, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3004-6. doi: 10.1145/2628363.2628378. URL http://doi.acm.org/10.1145/2628363.2628378.

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. ISBN 0-201-14237-6. doi: 10.1145/57167.57171. URL http://doi.acm.org/10.1145/57167.57171.

Jacob Poushter. Smartphone ownership and internet usage continues to climb in emerging economies, February 2016. URL http://www.pewglobal.org/

2016/02/22/smartphone-ownership-and-internet-usage-continues-to-\
climb-in-emerging-economies/. Accessed February 15, 2018.

Morgan N. Price, Bill N. Schilit, and Gene Golovchinsky. Xlibris: The active reading machine. In *CHI 98 Conference Summary on Human Factors in Computing Systems*, CHI '98, pages 22–23, New York, NY, USA, 1998. ACM. ISBN 1-58113-028-7. doi: 10.1145/286498.286510. URL http://doi.acm.org/10.1145/286498.286510.

Jef Raskin. *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000. ISBN 0-201-37937-6.

Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han. Generalizing multi-touch direct manipulation. In *SIGGRAPH 2009: Talks*, SIGGRAPH '09, pages 12:1–12:1, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-834-6. doi: 10.1145/1597990.1598002. URL http://doi.acm.org/10.1145/1597990.1598002.

Yosra Rekik, Radu-Daniel Vatavu, and Laurent Grisoni. Understanding users' perceived difficulty of multi-touch gesture articulation. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, pages 232–239, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2885-2. doi: 10.1145/2663204.2663273. URL http://doi.acm.org/10.1145/2663204.2663273.

Mary Beth Rosson. Patterns of experience in text editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '83, pages 171–175, New York, NY, USA, 1983. ACM. ISBN 0-89791-121-0. doi: 10.1145/800045.801604. URL http://doi.acm.org/10.1145/800045.801604.

Anne Roudaut, Eric Lecolinet, and Yves Guiard. Microrolls: Expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 927–936, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518843. URL http://doi.acm.org/10.1145/1518701.1518843.

Jaime Ruiz, Andrea Bunt, and Edward Lank. A model of non-preferred hand mode switching. In *Proceedings of Graphics Interface 2008*, GI '08, pages 49–56, Toronto, ON, Canada, 2008. Canadian Information Processing Society. ISBN 978-1-56881-423-0. URL `http://dl.acm.org/citation.cfm?id=1375714.1375724`.

Jaime Ruiz, Yang Li, and Edward Lank. User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 197–206, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1978971. URL `http://doi.acm.org/10.1145/1978942.1978971`.

Krystian Samp. Designing graphical menus for novices and experts: Connecting design characteristics with design goals. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3159–3168, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2466432. URL `http://doi.acm.org/10.1145/2470654.2466432`.

Zhanna Sarsenbayeva, Niels van Berkel, Chu Luo, Vassilis Kostakos, and Jorge Goncalves. Challenges of situational impairments during interaction with mobile devices. In *Proceedings of the 29th Australian Conference on Computer-Human Interaction*, OZCHI '17, pages 477–481, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5379-3. doi: 10.1145/3152771.3156161. URL `http://doi.acm.org/10.1145/3152771.3156161`.

Joey Scarr, Andy Cockburn, Carl Gutwin, and Philip Quinn. Dips and ceilings: Understanding and supporting transitions to expertise in user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2741–2750, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979348. URL `http://doi.acm.org/10.1145/1978942.1979348`.

Andrew Sears and Julie A. Jacko. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications (Human Factors and Ergonomics Series).* L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2007. ISBN 0805858709.

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. ISSN 0020-7373. doi: 10.1016/0020-7373(91)90037-8. URL `http://dx.doi.org/10.1016/0020-7373(91)90037-8`.

Andrew Sears, Catherine Plaisant, and Ben Shneiderman. A new era for high precision touchscreens. In H. Rex Hartson and Deborah Hix, editors, *Advances in Human-computer Interaction (Vol. 3)*, pages 1–33. Ablex Publishing Corp., Norwood, NJ, USA, 1992. ISBN 0-89391-751-6. URL `http://dl.acm.org/citation.cfm?id=159433.159436`.

Thomas Seifried, Christian Rendl, Michael Haller, and Stacey Scott. Regional undo/redo techniques for large interactive surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2855–2864, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208690. URL `http://doi.acm.org/10.1145/2207676.2208690`.

Karsten Seipp and Kate Devlin. Landscape vs portrait mode: Which is faster to use on your smart phone? In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 534–539, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2273-7. doi: 10.1145/2493190.2494422. URL `http://doi.acm.org/10.1145/2493190.2494422`.

Abigail J Sellen, Gordon P Kurtenbach, and William A S Buxton. The prevention of mode errors through sensory feedback. *Human-Computer Interaction*, 7(2):141–164, June 1992.

Steven C. Seow. Information theoretic models of hci: A comparison of the hick-hyman law and fitts' law. *Hum.-Comput. Interact.*, 20(3):315–352, September 2005. ISSN 0737-0024. doi: 10.1207/s15327051hci2003_3. URL `http://dx.doi.org/10.1207/s15327051hci2003_3`.

Mindy Seto, Stacey Scott, and Mark Hancock. Investigating menu discoverability on a digital tabletop in a public setting. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 71–80, New York,

NY, USA, 2012. ACM. ISBN 978-1-4503-1209-7. doi: 10.1145/2396636.2396647. URL http://doi.acm.org/10.1145/2396636.2396647.

Ben Shneiderman. Touch screens now offer compelling uses. *IEEE Software*, 8(2):93–94, March 1991. ISSN 0740-7459. doi: 10.1109/52.73754.

Ben Shneiderman. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Proceedings of the 2Nd International Conference on Intelligent User Interfaces*, IUI '97, pages 33–39, New York, NY, USA, 1997a. ACM. ISBN 0-89791-839-8. doi: 10.1145/238218.238281. URL http://doi.acm.org/10.1145/238218.238281.

Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, 1997b. ISBN 0201694972.

Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. Fat finger worries: How older and younger users physically interact with pdas. In *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction*, INTERACT'05, pages 267–280, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-28943-7, 978-3-540-28943-2. doi: 10.1007/11555261_24. URL https://doi.org/10.1007/11555261_24.

Aaron Smith. Americans and their cell phones, 08 2011. URL http://www.pewinternet.org/2011/08/15/americans-and-their-cell-phones/. Accessed 03 June 2018.

Aaron Smith. Us smartphone use in 2015, 04 2015. URL http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/. Accessed 03 June 2018.

S.G. Smith and B.A. Sherwood. Educational uses of the PLATO computer system. *Science*, 192(4237):344–352, April 1976.

Caroline Snowdon and Christian Kray. Exploring the use of landmarks for mobile navigation support in natural environments. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 13:1–13:10, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-281-8. doi: 10.1145/1613858.1613875. URL http://doi.acm.org/10.1145/1613858.1613875.

Rajinder Sodhi, Hrvoje Benko, and Andrew Wilson. Lightguide: Projected visualizations for hand movement guidance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 179–188, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207702. URL http://doi.acm.org/10.1145/2207676.2207702.

Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. In-air gestures around unmodified mobile devices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 319–329, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3069-5. doi: 10.1145/2642918.2647373. URL http://doi.acm.org/10.1145/2642918.2647373.

Martin Spindler, Martin Schuessler, Marcel Martsch, and Raimund Dachselt. Pinch-drag-flick vs. spatial input: Rethinking zoom & pan on mobile displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1113–1122, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557028. URL http://doi.acm.org/10.1145/2556288.2557028.

S. Stellmach, M. Jüttner, C. Nywelt, and Dachselt R. Schneider, J. Investigating freehand pan and zoom. In Harald Reiterer and O. Deussen, editors, *Mensch & computer 2012: interaktiv informiert*, pages 303–312. Oldenbourg Verlag, München, DE, 2012.

Wolfgang Stuerzlinger, Olivier Chapuis, Dusty Phillips, and Nicolas Roussel. User interface façades: Towards fully adaptable user interfaces. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 309–318, New York, NY, USA, 2006. ACM. ISBN 1-59593-313-1. doi: 10.1145/1166253.1166301. URL http://doi.acm.org/10.1145/1166253.1166301.

SurveyMonkey Inc. Surveymonkey: Free online survey software and questionnaire tool, Dec 2015. URL https://www.surveymonkey.com/. Accessed December 12, 2015.

Lucia Tokárová and Melius Weideman. Understanding the process of learning touch-screen mobile applications. In *Proceedings of the 31st ACM International Conference on Design of Communication*, SIGDOC '13, pages 157–164, New York, NY, USA, 2013.

ACM. ISBN 978-1-4503-2131-0. doi: 10.1145/2507065.2507066. URL http://doi.acm.org/10.1145/2507065.2507066.

Jessica J. Tran, Shari Trewin, Calvin Swart, Bonnie E. John, and John C. Thomas. Exploring pinch and spread gestures on mobile devices. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 151–160, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2273-7. doi: 10.1145/2493190.2493221. URL http://doi.acm.org/10.1145/2493190.2493221.

Zeynep Tuncer, Mohamed Selim, Marcus Liwicki, and Georg Kormann. Evaluation of multi-touch gestures. In Arno Ruckelshausen, Hans-Peter Schwarz, and Brigitte Theuvsen, editors, *Informatik in der Land-, Forst- und Ernährungswirtschaft 2015*, pages 189–192, Bonn, 2015. Gesellschaft für Informatik e.V.

Daniel Vogel and Ravin Balakrishnan. Interactive public ambient displays: Transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 137–146, New York, NY, USA, 2004. ACM. ISBN 1-58113-957-8. doi: 10.1145/1029632.1029656. URL http://doi.acm.org/10.1145/1029632.1029656.

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. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753365. URL http://doi.acm.org/10.1145/1753326.1753365.

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. ISBN 978-1-59593-593-9. doi: 10.1145/1240624.1240727. URL http://doi.acm.org/10.1145/1240624.1240727.

Daniel Vogel and Géry Casiez. Hand occlusion on a multi-touch tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2307–2316, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208390. URL http://doi.acm.org/10.1145/2207676.2208390.

Daniel Vogel, Matthew Cudmore, Géry Casiez, Ravin Balakrishnan, and Liam Keliher. Hand occlusion with tablet-sized direct pen input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 557–566, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518787. URL http://doi.acm.org/10.1145/1518701.1518787.

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. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208391. URL http://doi.acm.org/10.1145/2207676.2208391.

Feng Wang and Xiangshi Ren. Empirical evaluation for finger input properties in multi-touch interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1063–1072, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518864. URL http://doi.acm.org/10.1145/1518701.1518864.

Ina Wechsung. *An Evaluation Framework for Multimodal Interaction: Determining Quality Aspects and Modality Choice*. Springer Publishing Company, Incorporated, 2014. ISBN 3319038095, 9783319038094.

Pierre Wellner. The digitaldesk calculator: Tangible manipulation on a desk top display. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology*, UIST '91, pages 27–33, New York, NY, USA, 1991. ACM. ISBN 0-89791-451-1. doi: 10.1145/120782.120785. URL http://doi.acm.org/10.1145/120782.120785.

Joel West and Michael Mace. Browsing as the killer app: Explaining the rapid success of apple's iphone. *Telecommunications Policy*, 34(5-6):270–286, 2010. URL https://EconPapers.repec.org/RePEc:eee:telpol:v:34:y::i:5-6:p:270-286.

W. Westerman. *Hand Tracking, Finger Identification, and Chordic Manipulation on a Multi-touch Surface*. University of Delaware, 1999. URL https://books.google.ca/books?id=MqN4HAAACAAJ.

W. C. Westerman. Multi-touch input discrimination, 2013. US8130203 B2.

Wayne Westerman, John G. Elias, and Alan Hedge. Multi-touch: A new tactile 2-d gesture interface for human-computer interaction. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 45(6):632–636, 2001. doi: 10.1177/154193120104500612. URL https://doi.org/10.1177/154193120104500612.

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. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979173. URL http://doi.acm.org/10.1145/1978942.1979173.

Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1083–1092, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518866. URL http://doi.acm.org/10.1145/1518701.1518866.

J. Wu. Accommodating both experts and novices in one interface. 2000. URL http://www.otal.umd.edu/UUGuide/jingwu/.

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. ISBN 1-58113-636-6. doi: 10.1145/964696.964718. URL http://doi.acm.org/10.1145/964696.964718.

Jay Yarow and Leah Goldman. How people actually use the ipad: Results from our exclusive survey, 2012. URL http://www.businessinsider.com/ipad-usage-surveyresults-2012-7. BusinessInsider. Accessed 12 Dec, 2015.

Ka-Ping Yee. Peephole displays: Pen interaction on spatially aware handheld computers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 1–8, New York, NY, USA, 2003. ACM. ISBN 1-58113-630-7. doi: 10.1145/642611.642613. URL http://doi.acm.org/10.1145/642611.642613.

Zenith Media. Smartphone penetration to reach 66% in 2018, 2018. URL https://www.zenithmedia.com/smartphone-penetration-reach-66-2018/. Accessed February 15, 2018.

Chen Zhao, Ke-Yu Chen, Md Tanvir Islam Aumi, Shwetak Patel, and Matthew S. Reynolds. Sideswipe: Detecting in-air gestures around mobile devices using actual gsm signal. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 527–534, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3069-5. doi: 10.1145/2642918.2647380. URL http://doi.acm.org/10.1145/2642918.2647380.

Jian Zhao, R. William Soukoreff, and Ravin Balakrishnan. A model of multi-touch manipulation. In *In Proc. Grand*, 2011.

Jingjie Zheng and Daniel Vogel. Finger-aware shortcuts. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4274–4285, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858355. URL http://doi.acm.org/10.1145/2858036.2858355.

Jingjie Zheng, Blaine Lewis, Jeff Avery, and Daniel Vogel. Fingerarc and fingerchord: Supporting novice to expert transitions with guided finger-aware shortcuts. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, New York, NY, USA, 2018. ACM. doi: 10.1145/2858036.2858355. URL DOI:http://dx.doi.org/10.1145/2858036.2858355.