FLEXVIEW: AN EVALUATION OF DEPTH NAVIGATION ON DEFORMABLE MOBILE DEVICES

by

JESSE PAUL BURSTYN

A thesis submitted to the School of Computing

In conformity with the requirements for

the degree of Master of Science

Queen's University

Kingston, Ontario, Canada

August, 2012

Abstract

Mobile devices are frequently used to view rich content while on the go. However, they have a tradeoff between increased screen size and portability; mobile devices, by definition, are constrained to a fraction of a desktop computer's display area. This constraint means a user has to frequently navigate to content that lies outside the display.

We present FlexView, a prototype system and set of interaction techniques, which allows users to navigate through depth-arranged large information spaces using display curvature as an additional input channel. FlexView augments the planar (X-Y) navigation currently performed by touch input with two forms of bend input to navigate through depth (Z). With *leafing*, the user holds one side of display and bends the opposite side. *Squeezing* involves gripping the display in one hand and applying pressure on both sides to create concave or convex curvatures, and supports concurrent interaction with touch input.

We performed two evaluations to investigate the performance of FlexView's interaction techniques. In Experiment 1, we measured the efficiency of participants when searching through pages of a document, and compared touch input to *squeezing* and *leafing* used in isolation. Experiment 2 introduced X-Y navigation in a pan-and-zoom pointing task where multi-touch pinch gestures were compared against *squeezing* and *leafing* for zoom operations. Panning, across all conditions, was performed with touch input using the index finger.

Our experiments demonstrated that touch and bend interactions are comparable for navigation through depth-arranged content, and *squeezing* to zoom recorded the fastest times in the pan-and-zoom pointing task. Overall, FlexView allows users to easily browse depth-arranged information spaces without sacrificing traditional touch interactions.

Acknowledgements

This work would not have been possible without the advice and mentorship of my supervisor Dr. Roel Vertegaal. I thank him for creating an environment where outside-the-box ideas are explored, fostered, and realized. The members of the Human Media Lab have also played a large role in my success. Thank you Aneesh Tarun, David Holman, Doug Wightman, John Bolton, and Nicholas Fellion; I am grateful to have been able to work with such excellent friends and colleagues. I would like to give specific recognition to Amartya Banerjee, for the many hours of collaboration in both the successes and the only ever temporary setbacks.

Finally, I would like to thank my family and friends for their continued support. Thank you Brynn, whose work ethic and optimistic outlook is a model for all to follow. Thank you to my mother and father who are always confident of my success and never let me forget it.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Overview and Motivation

Mobile devices are frequently used to view rich content while on the go. Smartphones, such as the iPhone [1], use multi-touch displays that afford fine-grained interactions with richer content, such as detailed vector maps, high resolution photos, and web-based applications. In this scenario of richer content, there is a tradeoff between increased screen size and portability; mobile devices, by definition, are constrained to a fraction of a desktop computer's display area. This constraint means a user has to frequently navigate to content that lies outside the display. At present, this problem is addressed by using an adjustable viewport [12] or, by another strategy, segmenting the content and stacking it, like the pages in an e-book reader. In general, these solutions illustrate how arranging digital content by depth better manages dense information spaces that are difficult to fit into a viewport all at once.

Depth is typically represented by occlusion, by projecting three-dimensional effects onto the two dimensions of the display, or by varying the content zoom-level [24]. Some examples are digital paged documents, window management visualizations [49], and digital maps. In other words, input along a two-dimensional plane (X-Y) must be transformed into navigation along a perpendicular third dimension (Z). When a user navigates through three dimensions, both input and output occur on planar surfaces. Consequently, interacting with depth involves the use of an intermediary two-dimensional interaction technique, such as swiping, scrolling, or other touch gestures, that are not coupled to the affordances that the graphics convey.

In addition, interacting with rich content in a touch-enabled application often requires necessary interface controls that take up valuable screen real estate [15], such as the forward and back buttons on an iPhone's web-browser. Offloading controls from the display, to reduce clutter and increase usable screen size, would be beneficial and could be achieved using additional input channels. To this purpose, several systems have augmented touch input with additional sensors [17,19,32,43]. Despite increasing the bandwidth of input and reducing occlusion, few solutions follow the direct manipulation paradigm [40].

With the introduction of flexible displays in mobile form factors [28], there exists an opportunity to augment touch input with bend gestures [28]. Bend gestures could allow a designer to transform interface elements used for depth navigation into physical, shape-based control in the third dimension (Figure 1).

Previous efforts in flexible interaction techniques investigated bend input in isolation from other input channels [25,28,38,46]. Since direct-input with displays [21], is becoming a primary means by which people interact with computers, this led us to explore augmenting touch input with device curvature.

1.2 Contribution

In this thesis, we present the design and evaluation of *FlexView*, a prototype system and set of interaction techniques, which allows users to navigate through depth-arranged information using display shape as an additional input channel. Two forms of display deformation are explored. In its first form, *leafing*, a user holds one side of the display and bends the opposite side. The second form, *squeezing*, involves gripping the display in one hand and applying pressure on both sides to create concave or convex curvatures.

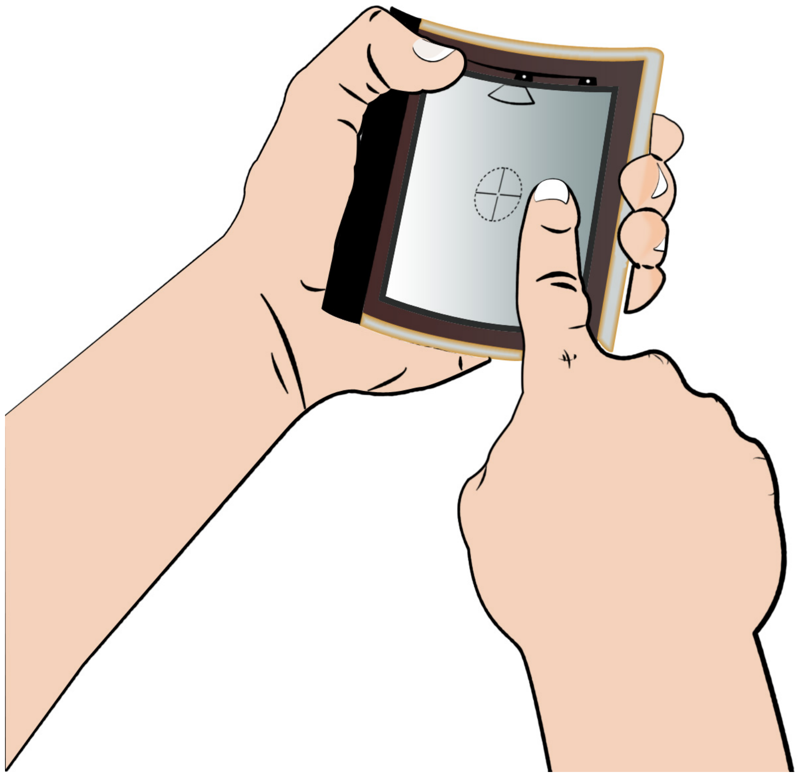

Figure 1. Augmenting touch input with bend gestures. The user simultaneously bends the display while providing touch input.

Squeezing supports concurrent input through two channels, bend input with the non-dominant hand and touch input with the dominant hand.

We performed two evaluations to investigate the performance of FlexView. In Experiment 1, we measured the efficiency of participants when searching through pages of a document, and compare touch input to *squeezing* and *leafing* used in isolation. Experiment 2 reports on a pan-and-zoom pointing task where multi-touch pinch gestures were compared against *squeezing* and *leafing* for zoom operations. Panning, across all conditions, was performed with touch input using the index finger.

The contribution is three-fold: we present FlexView which provides embodied, multi-modal interactions for depth-based information spaces, while augmenting touch input; our first experiment demonstrates that touch and bend interactions are comparable for navigation through depth; the second experiment demonstrates that, when used in parallel, combining bend and touch improves task performance.

1.3 Outline of Thesis

This thesis is presented in seven chapters. The first chapter introduces the topic of augmenting touch input with bend gestures and reveals the motivation and objective of the work.

The second chapter presents a review of related literature in the areas of navigating large information spaces on mobile devices, including panning, zooming, overview + detail, and focus + context views; augmenting direct-touch with additional input channels; bimanual interactions; and flexible display interactions.

The third chapter provides our design goals and describes FlexView's interaction techniques in detail. The fourth chapter describes the hardware and software components that compose the FlexView apparatus, as well as the process for creating flexible printed circuits.

The fifth chapter provides detail about our first experiment, where participants searched within an e-book, a one-dimensional depth navigation task. This chapter includes descriptions of the task, experimental design, hypotheses, measurements, and the participants. A discussion of the results is also presented in this chapter.

The sixth chapter provides detail about our second experiment, where participants traversed both depth and the X-Y plane in a pan-zoom task. This chapter, like the previous, includes description of the task, experimental design, hypotheses, measurements, and a discussion of the results.

The seventh chapter summarizes the main findings of the thesis, provides a set of design recommendations, and describes the limitations and future work in this area.

Chapter 2

Related Work

FlexView builds upon several areas of previous research: navigating large information spaces on mobile devices, augmenting direct-touch with other input channels, bimanual input, and flexible display interaction techniques.

2.1 Navigating Large Information Spaces on Mobile Devices

We identified four approaches to navigate large information spaces: panning, zooming, overview + detail, and focus + context.

2.1.1 Panning

Scrolling, panning, and paging are techniques that use the relative movement between a fixed viewport and the underlying document to navigate through information spaces that extend beyond the span of the viewport. Johnson [22] compared several mechanisms for panning on a touch-enabled device; he found that panning by dragging the background was superior to either pushing the viewport or touching the side of the screen (Figure 2). Dragging the background is now ubiquitous in direct-touch phones, such as the iPhone [1].

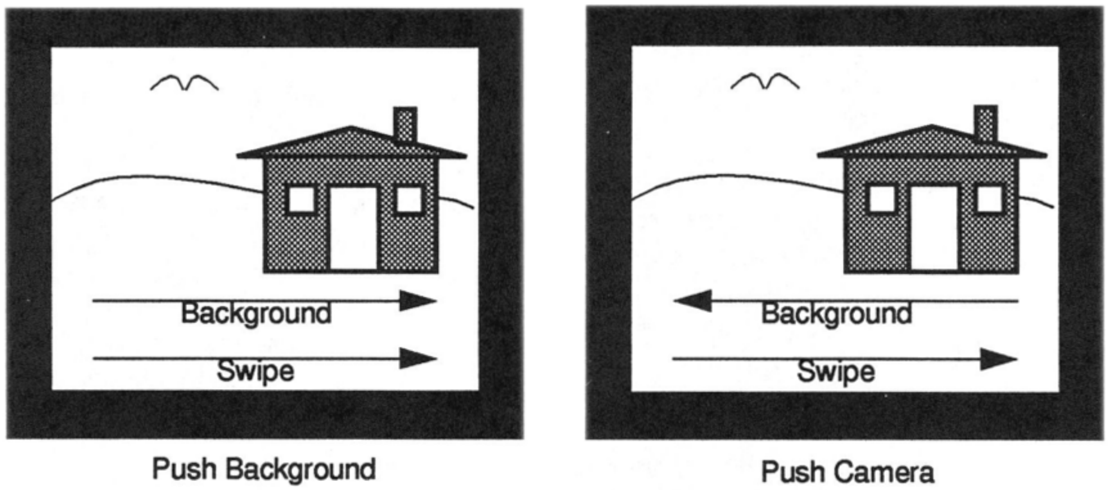

Figure 2. Two forms of panning evaluated by Johnson [22].

2.1.2 Zooming

When the information space is particularly large, panning alone is tedious, requiring repeated clutching actions. A zooming interface allows a user to either continuously regulate a magnification level or move between preset zoom levels specific to the task (Figure 3). Zooming both improves navigation and provides users with multiple levels of detail. Kaptelinin demonstrated that a zoom-enabled system was faster than basic panning interfaces in a information space that extended beyond the bounds of the display [24].

Guiard et al. [13] evaluated a series of targeting tasks in large multi-scale information spaces. They demonstrated that Fitts's Law [8] can be extended to multi-scale spaces, and that task difficulty is proportional to the size of viewport: the smaller the display area, the more critical zooming becomes.

Although multi-touch interactions, such as pinch-to-zoom, have gained popularity recently [1], techniques such as two-finger scaling and translation of graphical objects were demonstrated

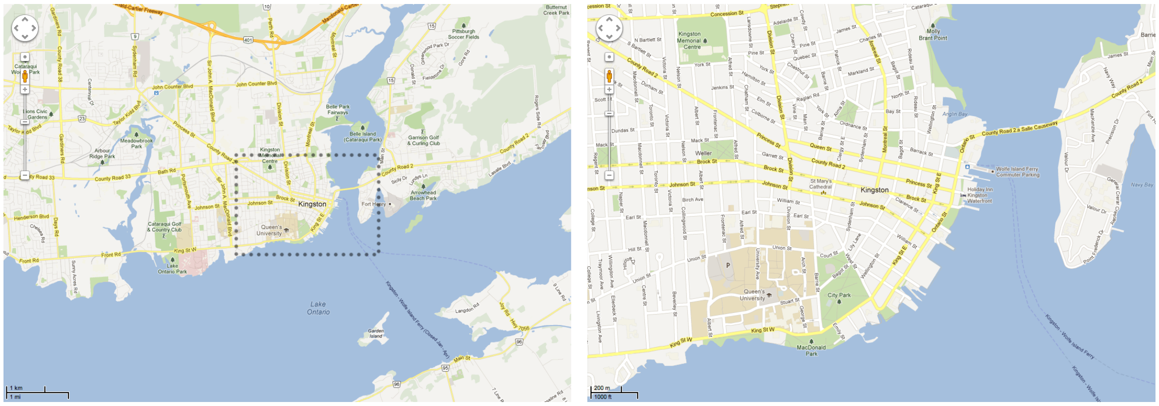

Figure 3. Two views in a zooming interface. Detailed and contextual views are segregated temporally.

in 1991 with the DigitalDesk Calculator [47]. Other techniques use additional sensors, such as accelerometers and gyroscopes, to perform zoom operations [7,27,32].

2.1.3 Overview + Detail Views

A zoom-enabled interface allows a user to visualize data at different levels of detail, but is restricted to one view at a time. Overview + detail views allow users to have more than one perspective of an information space concurrently (Figure 4). The most prominent example is a minimal overlay that shows a scaled down version of the information space, combined with a larger and more detailed view at a higher levels [12]. It has been shown that this overview + detail approach outperforms both panning and fisheye [37] representations for some navigation tasks [16]. Nevertheless, this approach increases cognitive load by requiring users to frequently reorient themselves when switching between views [3]. Additionally, overview windows occlude a portion the detail view, occupying part of the display area that is already at a premium on mobile devices.

Figure 4. An example of an overview + detail view. The overview inset in the top right corner allows users to see the context of the detailed region.

2.1.4 Focus + Context Views

A variation on the multiple-view approach is providing both overview and detail in the same viewport. Focus + context techniques include bifocal displays [36], fisheye views [37], and table lenses [35]. These techniques improve interaction with large information spaces by distorting the spatial layout according to the level of interest of the component areas.

Bifocal displays use a linear transformation function that provides a discontinuous visual distortion in the information space; areas of interest are unaffected, while the surrounding context is de-magnified. Fisheye views (Figure 5) use an exponential transformation function that provides a more continuous visual distortion; the distortion resembles the output of a fisheye camera lens. Table lenses are used to browse though large information spaces consisting of tables. The transformation function of the table lens technique is based on a cell as the smallest unit of distortion, instead of pixels.

Figure 5. A fisheye view is an example of a focus + context view. The focal region is magnified and displayed within its context.

The consequence of using focus + context views is the implementation of either a symbolic or distorted representation of distance. Therefore, users require a legend or must hold a complex mental model to interpret how distance and size cues translate into true measurements.

While designing FlexView, we drew on this body of prior research to explore the affordances associated with navigating large information spaces. In Experiment 2, we chose to implement panning by dragging the background, and afforded faster navigation though zooming.

2.2 Augmenting Direct-Touch with Other Input Channels

Rekimoto's use of tilt in handheld interaction is an early example of augmenting devices with spatial sensors [32]. With one hand, a user expresses orientation in three degrees of freedom, allowing for the use of tilt input to navigate menus, browse large maps, and select targets in pie menus.

Figure 6. Augmenting touch input with tilt sensing. No zooming takes place until the thumb is placed on the screen (middle). Zooming is then mapped to the change in pitch (right).

Similarly, Harrison et al. [17] explored more general embodied mobile interactions. They designed three handheld prototypes that mimicked real world metaphors by sensing tilt, corner touches (to simulate page turns), and handedness. Prior research has also explored using accelerometers to trigger implicit grip sensing while a device was held still [26]. Graspables [43] use accelerometers to sense the orientation of objects and to trigger grip sensing. With *touch-enhanced motion* and *motion-enhanced touch*, Hinckley et al. [19] explored techniques for mobile devices that leveraged the strengths of multi-modal interaction (Figure 6). Bergman et al. [4] explored the use of orientation of a mobile phone to augment touch input on a tablet computer.

Inspired by this body of research, we focused on exploring device curvature as an input channel to augment direct-touch input on mobile devices. A common thread among these multi-modal interfaces is the minimal use of additional on-screen elements, which informed the design of FlexView's interactions.

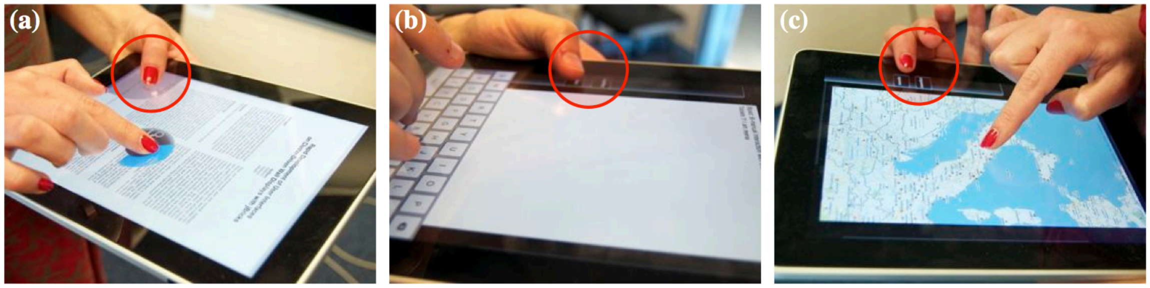

Figure 7. Bimanual interaction with BiPad: a) navigating a PDF, b) shifting to uppercase, c) zooming on a map. Red circles indicate the non-dominant hand acting as a modifier for the dominant hand.

2.3 Bimanual Input

Guiard's analysis of the human Kinematic Chain [14] emphasizes the asymmetric relationship commonly observed between the two hands. Guiard argues that the relationship between the non-dominant and dominant hands is that the former provides the spatial and *contextual* frame of reference for the detailed task action of the latter: the movements of the non-dominant hand are generally less frequent, and less precise, than the higher frequency, more detailed actions, of the dominant hand.

Myers and Buxton found that, given appropriate context, users were capable of providing continuous data from two hands simultaneously, without significant overhead, and the speed of performing a task was directly proportional to the degree of parallelism [33]. In another example, Latulipe et al. [29] compared the performance of single mouse input to symmetric and asymmetric dual mouse input in an image alignment task and found that bimanual input significantly improved performance.

The non-dominant hand often acts as a modifier to contextualize the actions of the dominant hand. Nancel et al. [34] used bimanual interaction techniques on a mobile device to pan-and-zoom content on a large display. Because pan-zoom operations inherently benefit from a high degree of parallelism, they are well afforded by the use of bimanual input techniques [14]. Wagner et al. [45] added novel bimanual interaction to tablet computers by using the positions of the thumb or fingers of the non-dominant hand as a modifier for the dominant hand (Figure 7). The supporting hand can tap, make gestures, or perform chords, thus modifying the interaction of the dominant hand. Bergman et al. [4] used the orientation of a mobile phone in the non-dominant hand as a frame of reference for the more frequent and precise actions of touch input with the dominant hand on a tablet computer.

The use of bimanual interactions, including those where the non-dominant hand can be used to increase the level of parallelism, was also a design goal of FlexView.

2.4 Flexible Display Interaction Techniques

FlexView strongly builds upon the emerging space of interacting with flexible input sensors and displays.

With PaperPhone, Lahey et al. [28] presented an evaluation of several pairs of bend gestures on a flexible display (Figure 8, left). They performed a study in which users designed bend gestures for common computing actions and concluded that bend gestures that take directional cues into account are more natural to users. Similarly, Kildal et al. [25] presented initial guidelines for deformable user interfaces using the Nokia Kinetic Device (Figure 8, right). They found continuous deformation gestures were well suited to controlling the magnitude of a parameter, and bending to zoom an image or scrolling through a list were intuitive use cases.

Schwesig et al. presented Gummi [38], a bendable handheld computer. They demonstrated the feasibility and potential benefits of compact, flexible mobile form factors. Gummi was designed with flexibility as an affordance; navigation was achieved through bending the display with two hands. Using global device curvature allowed for both discrete events, considered at a

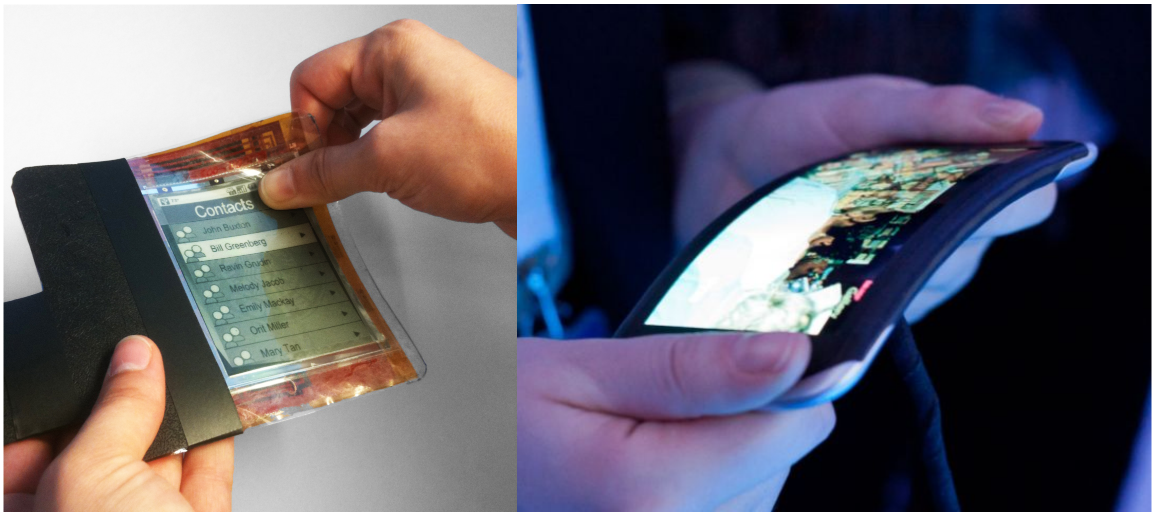

Figure 8. PaperPhone (left) is a flexible cell phone that maps bend gestures to common interactions [28]. Nokia Kinetic Device (right) uses bending and twisting to navigate large information spaces. Both prototypes use deformation as the primary method of interaction [25].

maximum bending threshold, and analog events, by measuring continuous transition states between thresholds.

Herkenrath et al. developed TWEND [18], a hardware prototype that allowed for navigation by bending and twisting. TWEND could recognize nine different directional deformations using eight fiber optic bend sensors. Similarly, Watanabe et al. presented Bookisheet [46], a pair of flexible input devices, made from thin acrylic sheets augmented with bend sensors. Bookisheet translated the degree of deformation into either discrete page flips or a continuous scroll.

With PaperWindows, Holman et al. [20] explored interaction techniques for thin and flexible displays by augmenting paper with projected digital images. By tracking the paper's shape and location using motion capture, users could use PaperWindows as if they were interacting with a flexible display. Document navigation required flipping over the sheet to advance. Gallant et al. designed Foldable User Interfaces [11], a prototyping tool for flexible displays by augmenting passive sheets of paper with patterns of infrared retro-reflective markers.

Physical bend gestures provide a user with kinesthetic feedback for efficient and enjoyable navigation. Using flexibility as an affordance, similar to those explored by this body of prior research, informed FlexView's design and development.

Chapter 3

Design Rationale

3.1 Design Goals

Direct manipulation [40] is easy for users to learn because gestures resemble physical actions. More recently, direct manipulation, in the form of display manipulation, is becoming a primary means by which people interact with computers. This rise in popularity is reflected – and spurred on – by the amount of novel multi-touch interaction techniques, both in the market and in active research. Now, with the introduction of flexible display technology [9], there has been a new focus on alternative forms of direct input by exploring bend gestures [28,30].

A majority of these efforts, however, continue to research bend input without the incorporating the benefits of direct-touch [18,28,38,46]. In certain scenarios, such as navigating large information spaces, a user benefits from performing a sequence of *parallel* gestures, rather than just one in isolation [33]. This led us to reflect upon situations where a user could choose between unimanual and bimanual input, and choose to touch and bend simultaneously, or in isolation.

Although augmenting touch input with additional sensors has been explored [4,19], interactions with flexible displays suggests a new vocabulary for the interplay between touch input and bend.

To this purpose, FlexView incorporates a series of interaction techniques for flexible mobile devices (Figure 9, Figure 10). Navigation can be performed with touch, bend, or both (Figure 1), where appropriate.

We created these techniques with four design goals: *augmenting touch*, providing *embodiment*, *maximizing display area*, and allowing for *parallelism.*

3.1.1 Augment Touch

Our first goal was that FlexView's interaction techniques should serve as an addition to touch interaction, not replace or impede it.

Touch screens have become the primary means of interaction with mobile devices. At the same time, there has been an increase in the number of interactions for deformable materials [25,28,38]. Many interaction designers overlook the benefits of touch by focusing on bend as the primary method of interaction. The benefits of direct-touch are well explored and need not be ignored. The principal goal of FlexView was ensuring that bend interaction complements touch, whether separately or together.

3.1.2 Embodiment

Our second goal was to provide interaction techniques that are embodied [19]. When a user provides input to a mobile device, the gesture used to navigate should physically represent the manipulations carried out. Dragging on a direct-touch enabled display translates the object displayed under the finger. Similarly, FlexView uses the metaphor of navigating though a stack, i.e. depth-arranged content, when deforming the display along its normal. By following this model, FlexView can be categorized as an *organic user interface* [21]. The use of a thin-film flexible display tightly couples FlexView's physical shape to the functionality its interface graphics afford, representing an item in a stack, such as a page of an e-book.

Additionally, the normal force applied when bending a display serves as haptic feedback that further expresses direct interaction [39]. This is achieved without resorting to physical buttons, visible widgets, or active haptics that compete for screen space and a user's attention.

3.1.3 Maximize Display Area

The display size of a mobile device determines the amount of information it can present. For small displays, task difficultly depends on the view size. The smaller the view, the more difficult the task, i.e. bandwidth is proportional to view size [13]. One solution is to increase the size of the display, but there is a tradeoff between size and portability: at a certain point, a "mobile" device is no longer lightweight and portable. Introducing a higher pixel density may ameliorate the issue, but this is a solution that does not scale.

Many interfaces feature buttons to allow a user to navigate between views, such as the forward and back buttons on a web-browser. These graphical elements reduce the space dedicated to representing useful information and are often distracting. Instead, a designer can reduce clutter on screen by offloading interactive elements away from the display and onto to additional input sensors [19]. By reducing the number of fingers touching the screen, or the frequency of a single finger touching, users can view more information on screen without occlusion.

3.1.4 Parallelism

In the appropriate contexts, bend and touch input must be effective interaction techniques when operated alone. Importantly though, FlexView should allow for a high degree of parallelism between touch and bend where possible. Guiard's Kinematic Chain theory [14] emphasizes the asymmetric relationship commonly observed between the two hands. Using bend input with the non-dominant hand provides a frame of reference for the detailed touch interaction of the dominant hand, and should improve performance in tasks that benefit from a high degree of parallelism.

Although the interactions presented in this paper are designed for navigating depth, the use of bend and touch in parallel can be extended to other scenarios.

3.2 Interaction Techniques

To navigate through depth-arranged information with FlexView, users bend the display around the vertical axis, i.e., longitudinally. Lateral bends were not considered as Lee et al. found bending around the horizontal axis had low user preference [30]. We explored two different methods of longitudinal bending: *leafing* and *squeezing*.

3.2.1 Leafing

With *leafing*, a user grips the left or right side of the device and, with his free hand, bends the other side (Figure 9). Although this bend interaction is only performed with the dominant hand, it requires the non-dominant hand to keep the opposite side of the display stable. Therefore, *leafing* follows Guiard's Kinematic Chain when bend interaction is performed in isolation.

Leafing is a style of bending that traces its roots to several previous flexible interfaces [11,28,38,46]. Each of these systems differs in the position of the dominant hand and where curvature is created. With *leafing*, the dominant hand always bends the display longitudinally. The range of valid axes of deformation spans from the center of the display to the edge where the dominant hand is grasping. We recommend that users bend the display closer to the edge; it requires less effort, as surface tension is minimized here, and the full range of interaction can be performed using the fingers, without engaging wrist motion.

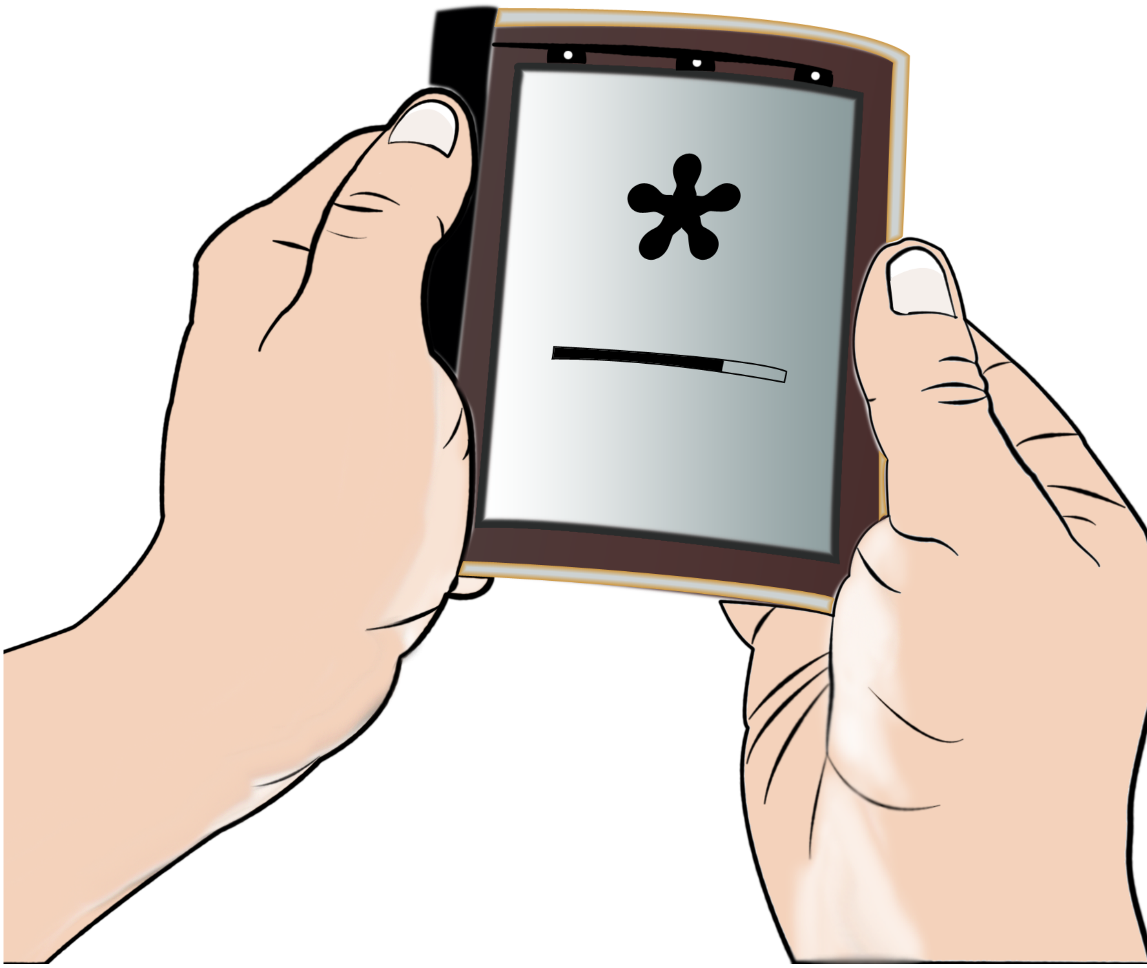

Figure 9. Leafing interaction technique. The user grips the left or right side of the device and, with his free hand, bends the other side.

3.2.2 Squeezing

With *squeezing*, one hand holds the device and bends the display surface into a convex or concave arc (Figure 10). *Squeezing* resembles exerting pressure on the deformable skin of a mobile device, but the display itself bends. Gallant et al. [11] implemented a version of *squeezing* a passive paper sheet with Foldable User Interfaces.

Squeezing can be performed with either the dominant or non-dominant hand, as it is a unimanual interaction technique. By *squeezing* the display in the non-dominant hand though, a user is free to use her dominant hand for touch input on the surface.

Users typically perform the *squeezing* technique by cradling the display between the base of their thumb and the inside of the third knuckles, or using the pads of their thumb and fingers. Display curvature occurs down the device’s center line, longitudinally.

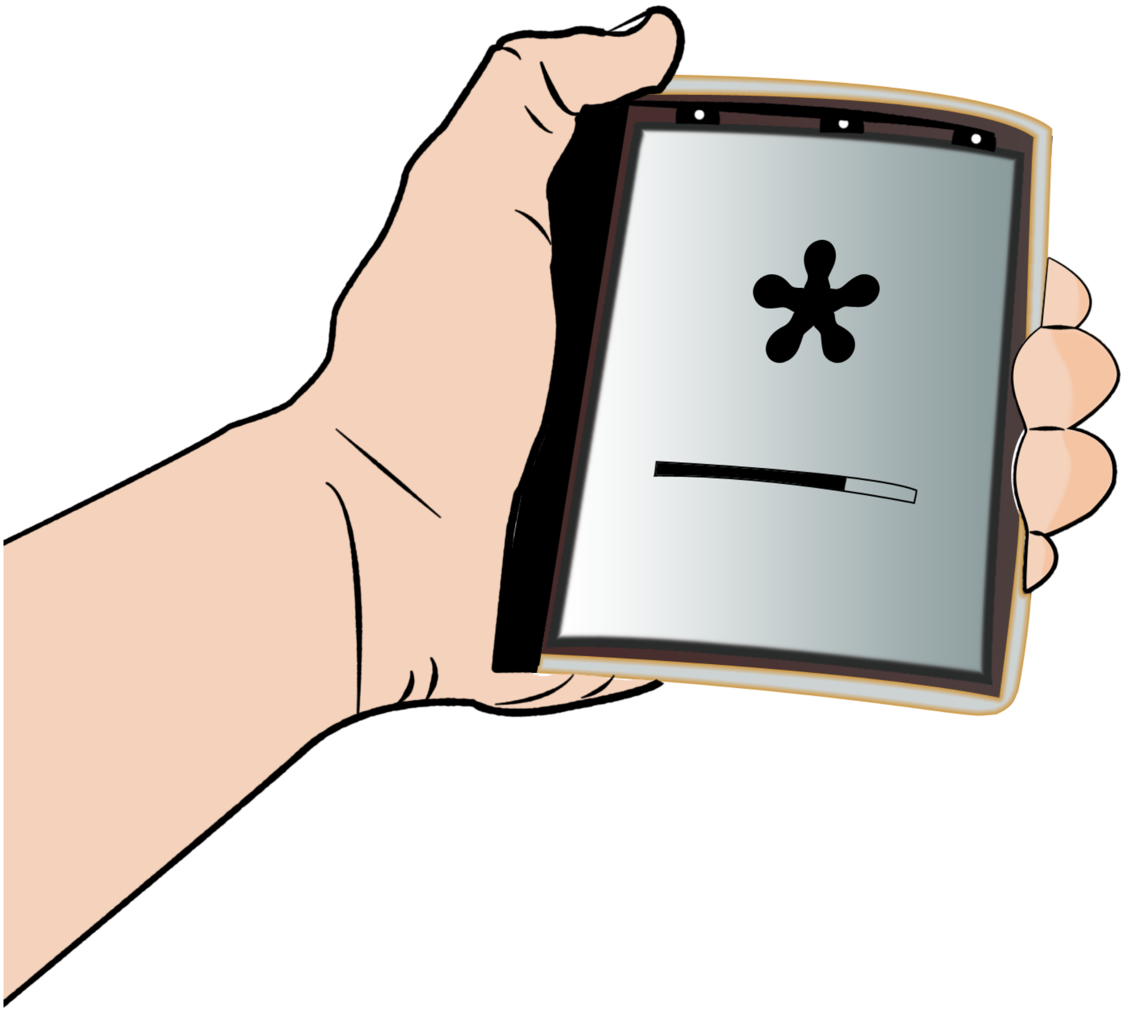

Figure 10. Squeezing interaction technique. One hand holds the device and bends the display surface into a convex or concave arc.

We recognized two different input mappings to translate the bend information into navigation through depth: absolute and relative.

3.2.3 Absolute Mapping

An absolute mapping indicates that the sensor values should translate into the state of the output directly. With FlexView, the shape of the display directly corresponds to the depth within the information stack. The resting, flat state of the display presents the top of the depth-arranged content. The further a user bends the sides of the display outwards, the deeper the interface navigates. The bottom of the stack corresponds to the maximum allowable bend. Without user intent, FlexView's display cannot retain its deformation. Therefore, with the absolute mapping, releasing the bend returns the user to the beginning of the stack. For scenarios where the resting state is not the top, e.g. zooming, bending the display towards the user navigates towards the top of the stack. For direct-touch input, an absolute mapping is analogous to the seek bar in a media player. Sliding the finger along the surface of the display (horizontally or vertically) navigates though the stack.

3.2.4 Relative Mapping

With the relative mapping, the extent of deformation is also proportional to the amount of depth traversal. Instead though, the resting state of the display presents the current position in the stack, and any deformation results in depth traversal away from the current position in the same direction. If a user deforms the sides of the display outwards, content deeper in the stack is displayed. Bending the display inwards displays content closer to the top of the stack. The relative mapping is similar to the iPhone's flick scrolling [1], though the direction of navigation is in the direction of actuation [28]. The amount of input provided to a sensor, either the degree of a bend or the length of a touch swipe, linearly maps to the initial amount of navigation.

With FlexView's relative mapping, we included inertia similar to the iPhone's. The amount of inertia is determined by the speed of deformation and a multiplication factor determined through pilot testing.

Chapter 4

Apparatus

4.1 Hardware

Figure 12 shows the assembled FlexView apparatus. The apparatus envisions the look and feel of a commercially available, lightweight, flexible mobile device. As the custom-designed prototype circuitry would be miniaturized in production, most of the supporting electronics are housed away from the display.

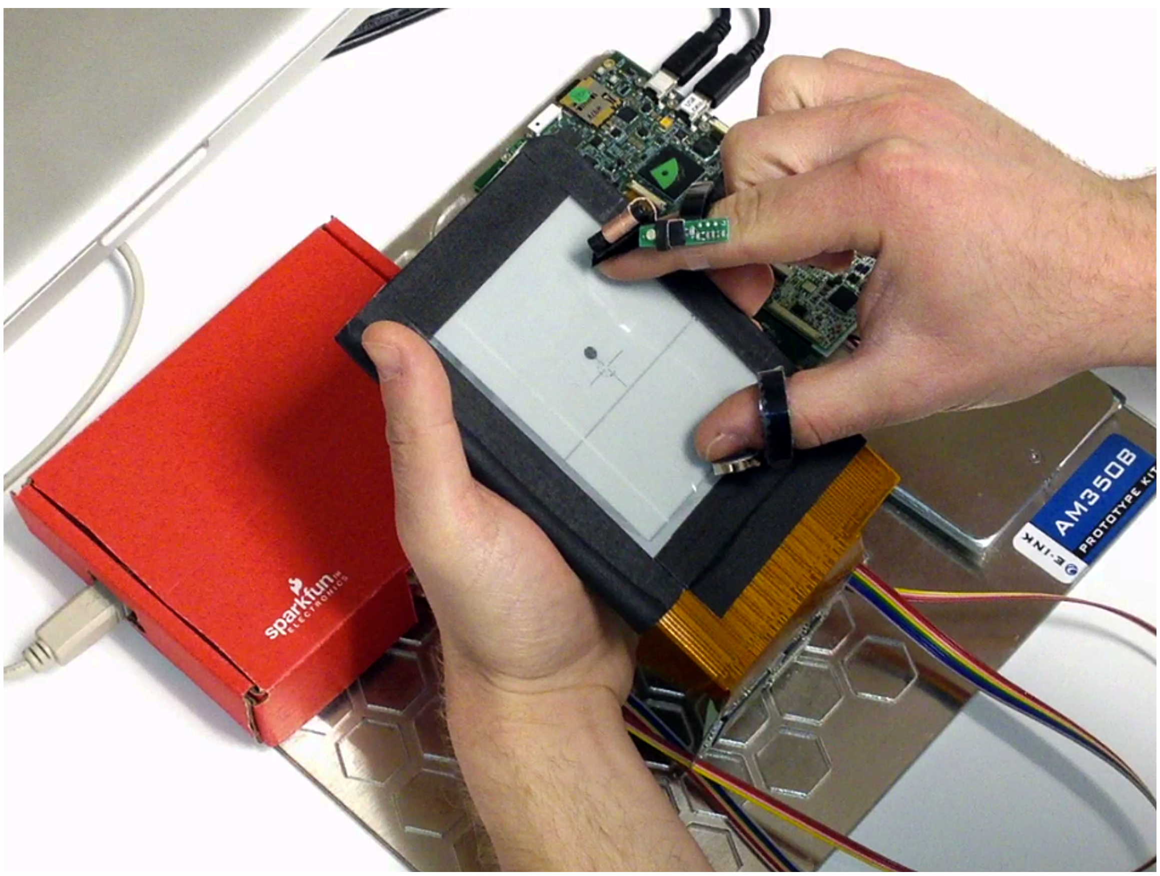

Figure 11. FlexView in use under experimental conditions. The flexible display is augmented by a Wacom touch sensor, a bend sensor, and hall-effect sensors, and is powered by an AM350.

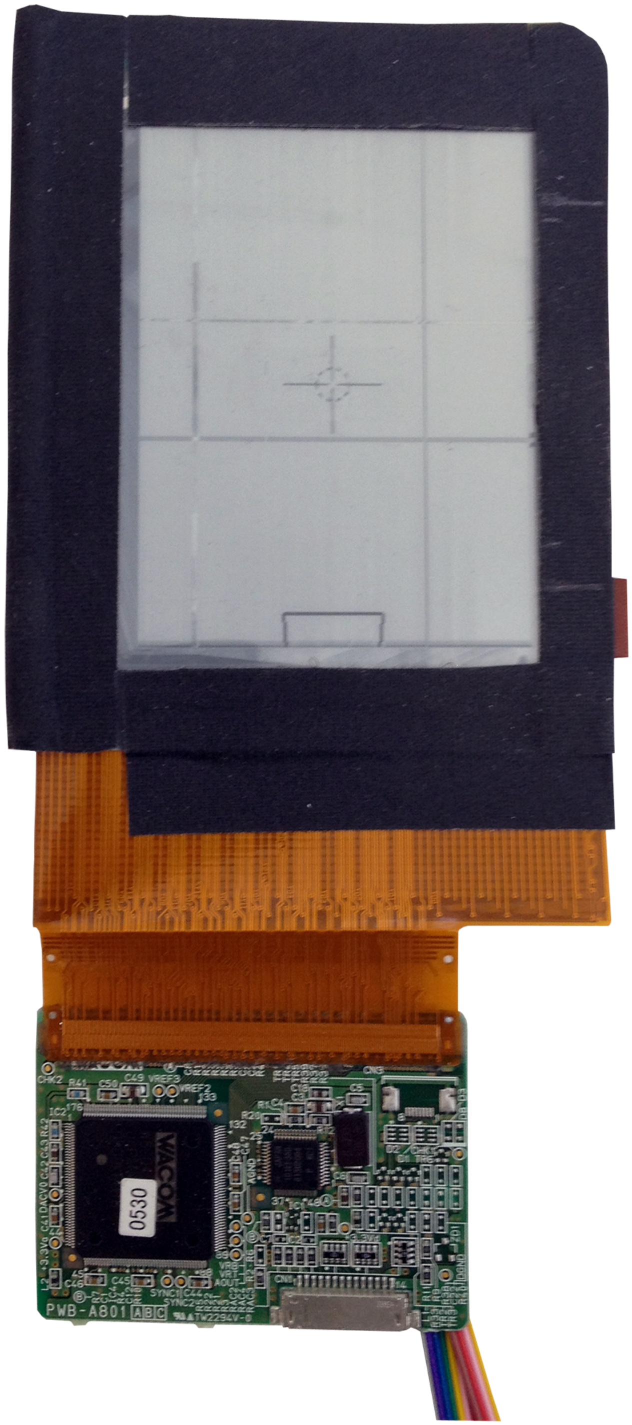

Figure 12. Assembled FlexView display and sensor layers.

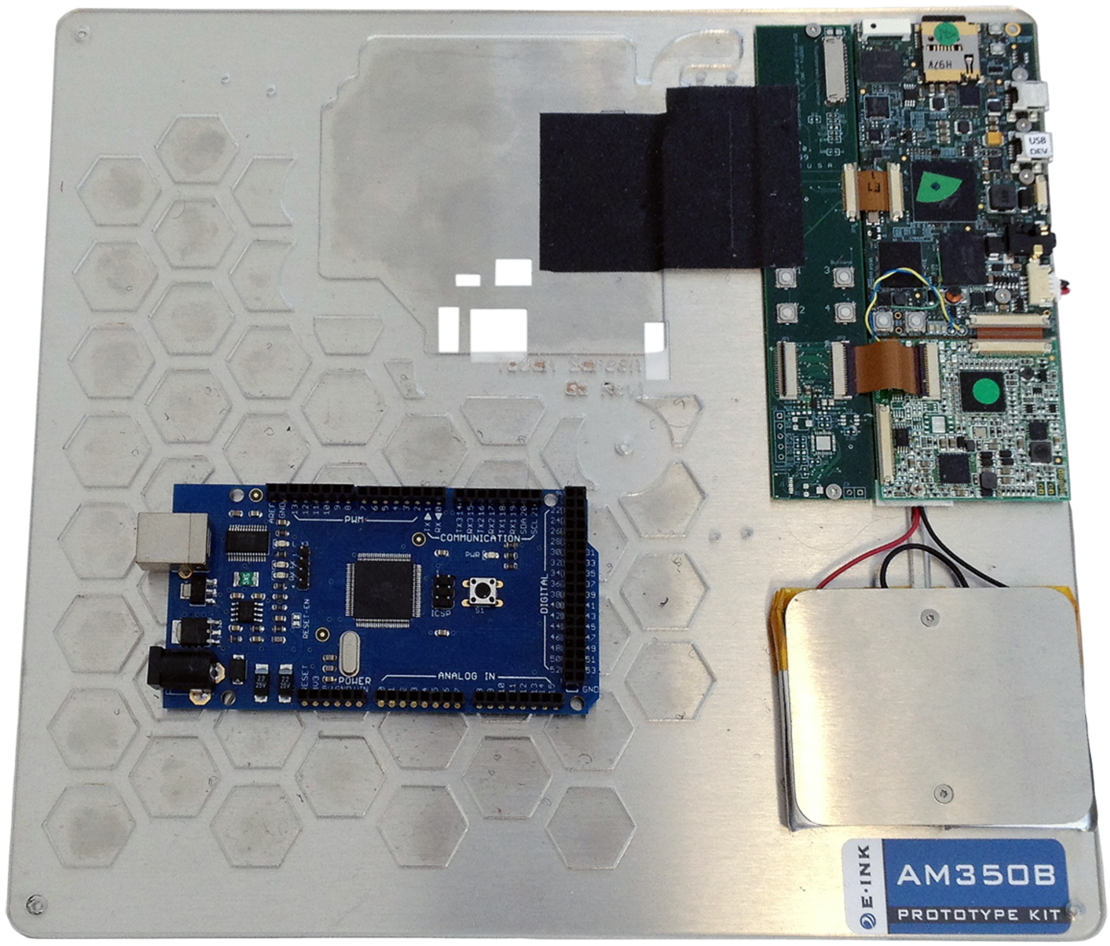

Figure 13. E Ink AM350 Development Kit and Arduino Mega.

4.1.1 Electrophoretic Displays

FlexView uses a kind of electronic paper known as an electrophoretic display. To create the appearance of ink on paper, electrophoretic displays use similar pigments as those in the printing industry and exploit the phenomenon of electrophoresis [31]. A thin film, consisting of millions of microcapsules, is suspended between two electrodes. The top electrode is part of the display surface and is made from a transparent plastic. Each microcapsule is filled with black and white pigment particles suspended in a mixture of a clear nonpolar liquid and surfactant additives. The black pigment particles have a negative charge, and the white pigment particles have a positive charge. By applying a negatively charged electrostatic field to the bottom electrode, the black pigment particles move to the top of the microcapsule where they can be seen through the transparent electrode; the white particles move to the bottom of the microcapsule where they are hidden. Applying a positively charged field to the bottom electrode clears the display. Creating positive and negative electrostatic fields underneath specific microcapsules provides the ability to display images and text. Some electrophoretic display technologies can apply fields at a sub-microcapsule level, enabling higher resolutions.

4.1.2 Flexible Electrophoretic Display

FlexView uses a 3.7” (77 mm x 55 mm active area) E Ink Bloodhound flexible electrophoretic display (Figure 14) from the Arizona State University Flexible Display Center [9]. An E Ink AM350 EPD Kit (Figure 13) running Android 1.5 on a Marvell Armada 166E processor drives the prototype EPD. The display runs at a resolution of 240x320, and has a refresh rate of 75 ms (13.3 Hz). A complete refresh of an E Ink display may take up to 260 ms, but we ensured that during trials, all refreshes consisted of A2 Waveform partial updates. In other words, we forced the display to only update with a 75 ms refresh by guaranteeing all pixel transitions were from white to black (or vice-versa), and updating less than half of the pixels on screen. The display is laminated to prevent rapid deterioration of the substrate.

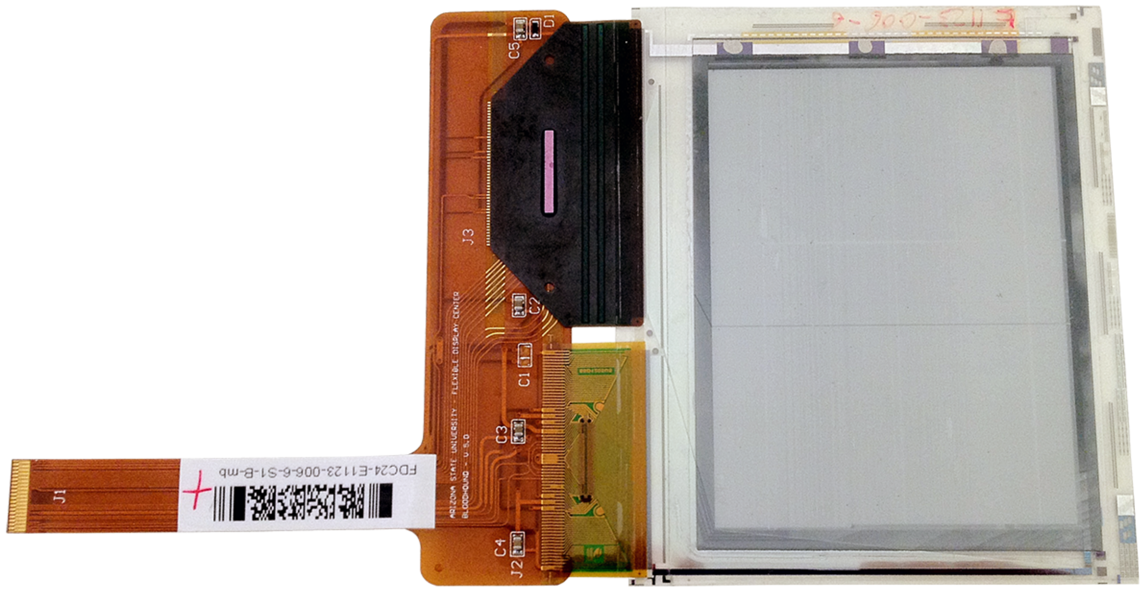

Figure 14. Prototype E Ink Bloodhound flexible electrophoretic display.

Although we present FlexView as a mobile device, FlexView is tethered; there are supporting rigid electronics that are required to drive the display prototype. To keep FlexView the approximate size and weight of a flexible mobile device, a thin cable bundle connects the display setup to the AM350 and a PC running Windows 7. This design maximizes the flexibility and mobility of the display, and keeps its weight to a minimum. The laptop processes sensor input and sends it to the AM350, which updates the display through a custom Android application.

4.1.3 Flexible Printed Circuit

The prototype display is augmented with a flexible printed circuit (FPC) that houses FlexView's sensors (Figure 15). To create the FPC, we followed the process performed by Tarun [42].

We printed a circuit diagram onto singled-sided DuPont Pyralux [6], a copper-clad flexible composite material. The printed areas created paths where copper, our conductive material, remained.

Etching the circuit consisted of placing the copper sheet into a two-to-one solution of hydrogen peroxide and hydrochloric acid. The resulting chemical reaction is as follows:

$$2HCl + H_2O_2 + Cu = CuCl_2 + 2H_2O$$

By using wax-based printer ink, the desired copper areas were masked and did not react to the etching solution. After 20 minutes, the exposed copper dissolved, leaving only the desired circuit paths. The FPC was washed and the wax ink was removed with isopropyl alcohol.

A 2” Flexpoint bi-directional bend sensor [10] and eight analog hall-effect sensors (that detect the presence of magnetic fields) were soldered directly onto the FPC. An Arduino Mega microcontroller [2] obtains data from the sensors.

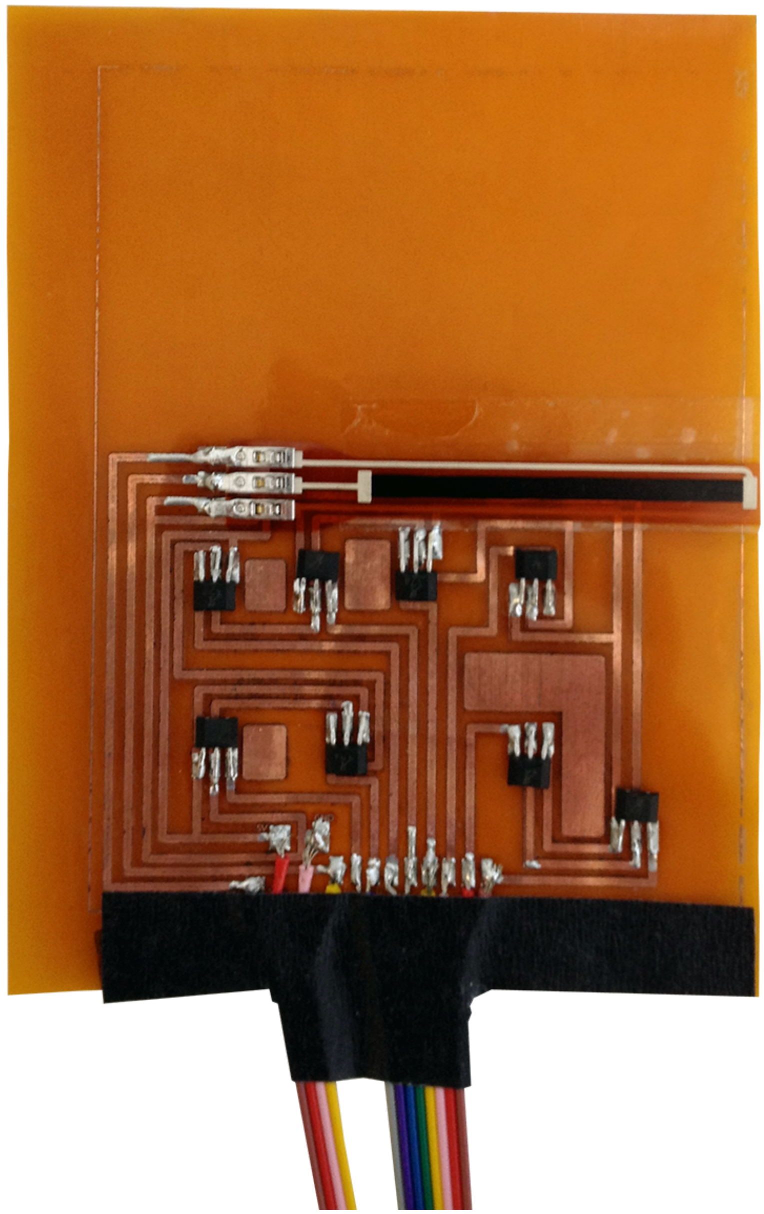

Figure 15. Flexible printed circuit. A bend sensor and hall-effect sensors are soldered to the copper paths.

4.1.4 Wacom Digitizer

To sense touch input, a flexible Wacom [44] digitizer (Figure 16) is housed between the FPC and the display. The Wacom has a resolution of 2540 dots per inch, sensitive to 0.01 mm, and has a response time of 8 ms (125 Hz). For our experimental evaluations, we needed to sense both single and dual touch input on the device. To detect a single touch, we extracted a Wacom pen's input coil and mounted it on the participants' thumb. Because the Wacom only supports a single touch point at a time, instrumenting a participant with two coils to detect dual touch – one on the index and one on the thumb – was not a viable solution. Since the only dual-touch input we needed to detect was a pinch gesture, we accommodated this requirement by mounting the coil on the index finger and a magnet on the thumb (Figure 11). These passive devices were made as light as possible, rested on the fingernail, and did not directly come into contact with the display surface. A matrix of hall-effect sensors on the FPC determines the thumb's contact with the display (as they detect the presence of the magnet on the thumb), and the gestural data is extrapolated through exploiting the symmetry of a pinch gesture. We extensively tested the efficacy of this solution during our pilot study. In addition, participants went through a calibration process, ensuring their perceived touch point mapped correctly to the coordinates detected by the sensor.

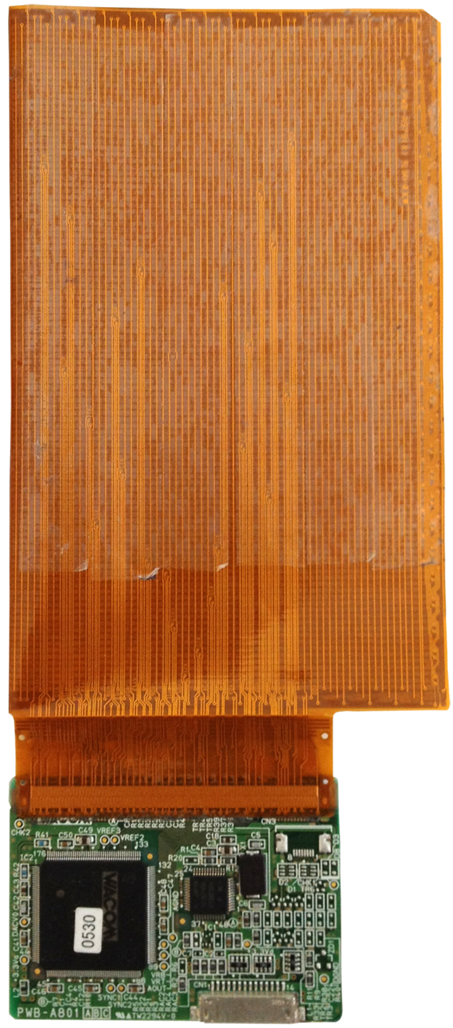

Figure 16. Folded flexible Wacom digitizer board.

4.1.5 Other Touch Solutions

Instrumenting a user, as in the flexible Wacom, is typically considered less preferable to natural 'un-augmented' touch. Initially, we explored other hardware touch sensing solutions that worked with a user's uninstrumented finger. Early iterations of FlexView's hardware used ZyFilm, a flexible projective capacitive multi-touch sensor, by Zytronic [51]. Such solutions were not as accurate as the Wacom, sensitive to 3 mm compared with 0.01 mm, especially while bending the device.

In addition, the E Ink display interferes with the finger detection of projective capacitive touch sensors. Projective capacitive touch sensors operate by detecting disturbances in a dielectric field, initiated by a finger, a capacitive stylus, or any other weak conductor. Their presence distorts the touch screen's electrostatic field, creating a measurable change in capacitance. The backplane of the E Ink display's substrate is a thin-film transistor, consisting of a layer of metallic contacts suspended in a polymer layer. This metal layer is a conductive surface that, in close proximity to a projected capacitive touch sensor, constantly disturbs the electrostatic field, making the presence of a finger undetectable. Although slightly different, the refresh of the E Ink's electrophoretic display also exacerbates this problem. The electrostatic field, used to align the charged pigment particles in the E Ink's microcapsules, pulses multiple times per second and, during this state change, fluctuates the touch sensor's electrostatic field; by doing so, it registers false touch events. Both kinds of unintentional and erratic touch sensor activations are prohibitive, especially for the demands of an empirical study.

We also considered resistive touch solutions. Resistive touch sensors operate using pressure; closing the gap between two spaced materials generates input. Bending these sensors may trigger a false input signal along the axis of deformation.

Although the Wacom digitizer required instrumenting a user during experiments, it consistently performed well when over the E Ink and when deformed. For this reason, it was chosen as the touch solution for our experiments. Finally, users found the lightweight components worn on their fingers minimally intrusive.

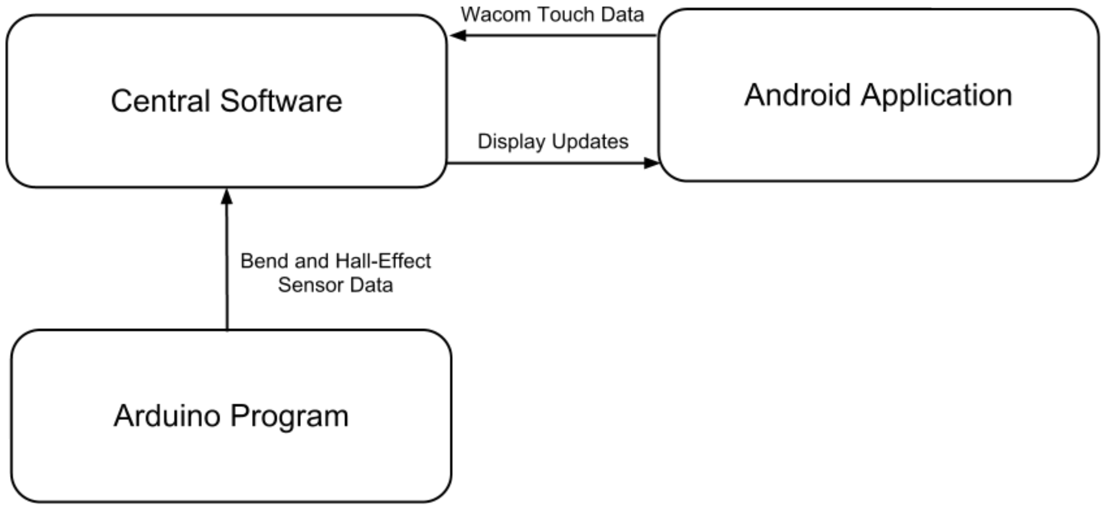

Figure 17. Software Architecture.

4.2 Software

4.2.1 Arduino

An Arduino program was written and uploaded to the Arduino Mega microcontroller [2] to receive and process data from the bend sensors and the hall-effect sensors. When the script first executes, it stores an initial 50 frames of data that are averaged to create a baseline value for each sensor. On each subsequent frame, sensor values are updated and reported relative to their baseline values.

When using the Arduino's 10-bit analog-to-digital converter on physically manipulated analog sensors, readings may be noisy, especially at high frequencies. To address this issue, the Arduino program processes and reports sensor values as a weighted moving average of the previous 6 frames. This calculation ensures that slight changes in voltage are not considered input deltas and are effectively filtered out. A 6 frame weighted moving average ignores small fluctuations in sensor readings, without impeding true input deltas. The final sensor values are sent to serial output (USB) of the Arduino at a rate of 20 Hz.

4.2.2 Central FlexView Software

FlexView's main software runs on a PC running Windows 7. The central software is the hub for all the hardware communication and contains the experimental logic. The program was written in the C# language, with the Microsoft .NET 4.0 Framework, and uses the Windows Presentation Foundation as its GUI library. It receives touch sensor data from the Wacom digitizer over a local Wi-Fi socket connection to the AM350 and bend sensor and hall-effect sensor data from a serial connection through USB.

For each experiment and condition within, the software converts sensor values into appropriate navigation deltas. For our first experiment, sensor values are mapped to both absolute and relative document page flips. For our second experiment, sensor values are mapped to panning and zooming navigations. This software also stores the pages of the document and determines the current page for Experiment 1.

The central FlexView software also logs movement times, errors, and all other measurements for both experiments.

During the evaluations, the central software was directly interfaced through the PC. The central software provides control over the flow of the experiments; it has commands for moving to the next trial and repeating trials if necessary.

4.2.3 Android Application

The AM350, which drives the flexible display prototype, runs a custom version of Android 1.5 with display waveforms for the EPD. For each experiment, an Android application was written to perform two tasks: to send Wacom sensor data to the central software and to draw images to the display. Both applications receive Wacom sensor values through the AM350's UART connection and report them as Android touch events. The values are sent to the central software through a local Wi-Fi socket connection running on a separate thread. For our first experiment, the Android application receives a page number and immediately draws the appropriate image. In the second experiment, the Android application receives the target distance and size, as well as pan and zoom deltas. The application creates a bitmap with the appropriate targets, in the desired locations, and then scales and translates the viewport to the correct levels. The viewport determines which pixels are drawn on the display.

Chapter 5

Experiment 1

In our first experiment, we measured the performance of the two FlexView techniques in comparison with touch in an e-book searching task. Here, touch and bend interactions are used independently in a one-dimensional task, i.e., to navigate depth exclusively. Our objective was to gain an insight into the performance of each interaction technique and input mapping in isolation.

5.1 Task

Participants were shown an image of a bullseye on the display and were asked to search for it within a 30 page e-book. They could then start interacting with the display to begin the trial. Each trial began at the first page. The 29 distractor images were randomized every trial, with the restriction that no image was duplicated on consecutive pages. Each participant was asked to navigate through pages of the e-book until they found the target image and remain at that position for 200 ms. The dwell time ensures that navigation to the target page was intentional and that the participant did not scroll past the target. We did not use a confirmation button, as such an action requires explicit participant action, which could potentially cause unintentional navigation or modify recorded movement times. Figure 18 shows a typical view of the software presented to the participant in Experiment 1.

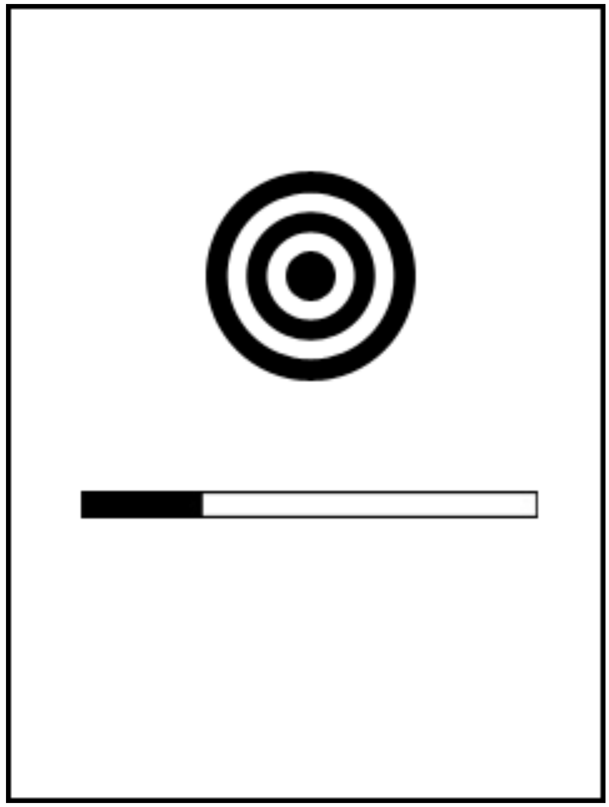

Figure 18. Experiment 1 configuration.

Two measures were recorded: movement time and navigation errors. Movement time represents the time it took for the participant to find the target image in milliseconds. Movement times are calculated from when participants first provide input to the device during a trial, until they find the target page. The dwell time at the end of the trial is not included in this measure. Navigation errors report how many times the participant overshot the target and had to reverse direction.

5.2 Design

We used a 4x2x3 factorial repeated-measures within-subject design. Our factors were: *interaction technique* (squeezing, leafing, and both horizontal and vertical touch – to not bias direction), *input mapping* (absolute and relative), and *target distance* (page 5, 15, and 25). Each participant performed 3 trials per combination of factors, for a total of 72 trials (4 interaction techniques x 2 input mappings x 3 target distances x 3 trials). Condition order was counter-balanced between participants. The experimental sessions lasted about 60 minutes, including training. Participants trained with each combination of *interaction technique* and *input mapping* until they achieved less than 10% improvement between trials.

5.2.1 User Feedback

Participants were asked to rank each combination of *interaction technique* and *input mapping* in their preferred order. In addition, participants were asked to rate each combination of *interaction technique* and *input mapping* on whether it was *easy to complete the task* and whether it was *comfortable to use*. The questions were structured using a 5-point Likert scale (1 = Strongly Disagree to 5 = Strongly Agree).

5.2.2 Participants

12 participants (10 male and 2 female), between the ages of 19 and 28, took part in the study, as well as the following evaluation. Each participant had some familiarity with multi-touch gestures, e.g., on a smartphone or a laptop. All participants were right-handed. They were paid $20 for their participation both studies.

5.3 Hypotheses

We hypothesized that vertical touch would have faster navigation times than horizontal touch scrolling (H1). This hypothesis was based on the physiology of the thumb, where the thumb, in a grip position, has a higher freedom of Y-axis movement [41], and vertical scrolling's proliferation in mobile touch interfaces. We predicted that navigation using *leafing* would be faster than using *squeezing* (H2). We expected that touch conditions would be faster than bend conditions (H3), though user feedback would indicate a higher preference for bend, especially *leafing* (H4), due to its page flipping metaphor. We also hypothesized that techniques with the absolute mapping would be faster than those with the relative mapping (H5) as it requires less motion to perform. Finally, we hypothesized that closer targets would result in faster navigation times (H6), and the relative mapping would result in fewer errors (H7).

5.4 Results

5.4.1 Performance Analysis

We analyzed the collected measures by performing a repeated measures factorial analysis of variance (ANOVA) using *interaction technique* (4) x *input mapping* (2) x *target distance* (3) on movement time and navigation errors.

For movement times (Figure 19), the analysis showed that *input mapping* was a significant factor ($F_{1,11} = 20.27$, $p<0.001$). Pairwise post-hoc tests, with Bonferroni corrected comparisons between *input mappings*, revealed that the absolute mapping was faster than the relative mapping. *Target distance* ($F_{2,22} = 561.72$, $p<0.001$) was also a significant factor, with shorter distances resulting in shorter navigation times. In addition, we found a significant interaction effect between *input mapping* and *target distance* ($F_{2,22} = 9.62$, $p<0.001$).

With respect to navigation errors (Figure 20), the analysis revealed *interaction technique* as a significant factor ($F_{3,33}$ = 29.97, p<0.001). Pairwise post-hoc tests, with Bonferroni corrected comparisons, showed that both horizontal and vertical touch resulted in fewer errors than either *squeezing* or *leafing*. *Input mapping* was also revealed to be a significant factor ($F_{1,11}$ = 14.19 p<0.05); the relative mapping recorded fewer errors than the absolute mapping. We also found a significant interaction effect between *interaction technique* and *input mapping* ($F_{3,33}$ = 12.30, p<0.05).

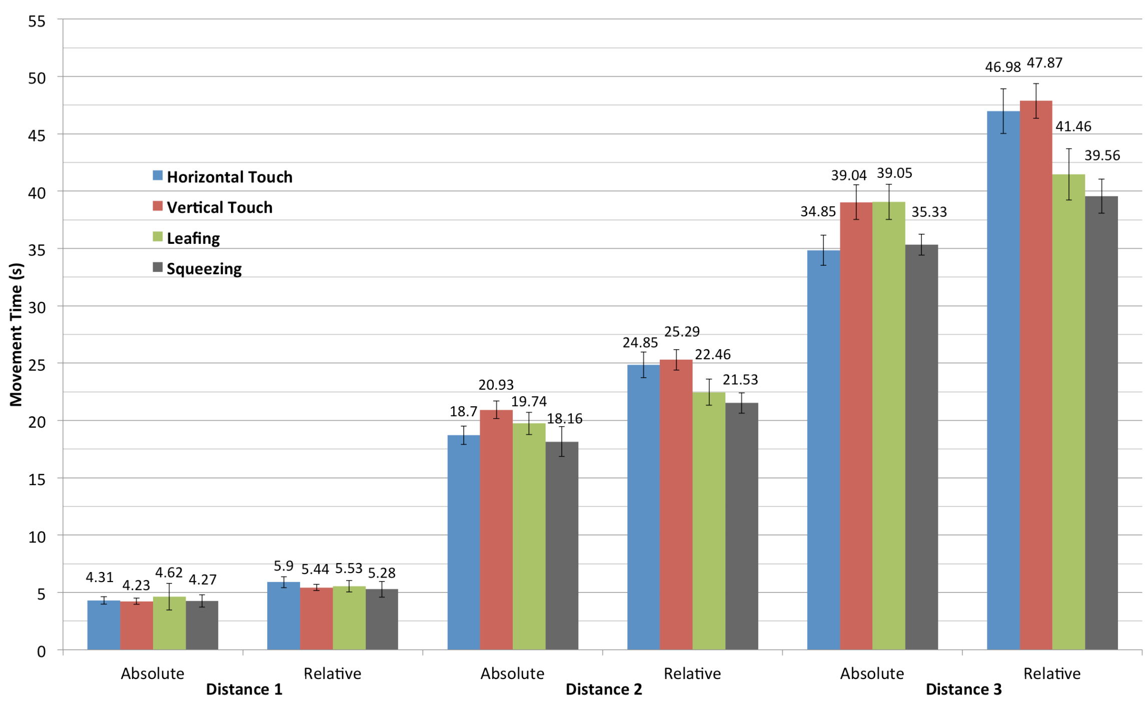

Figure 19. Mean movement times across interaction technique, input modality, and target distance.

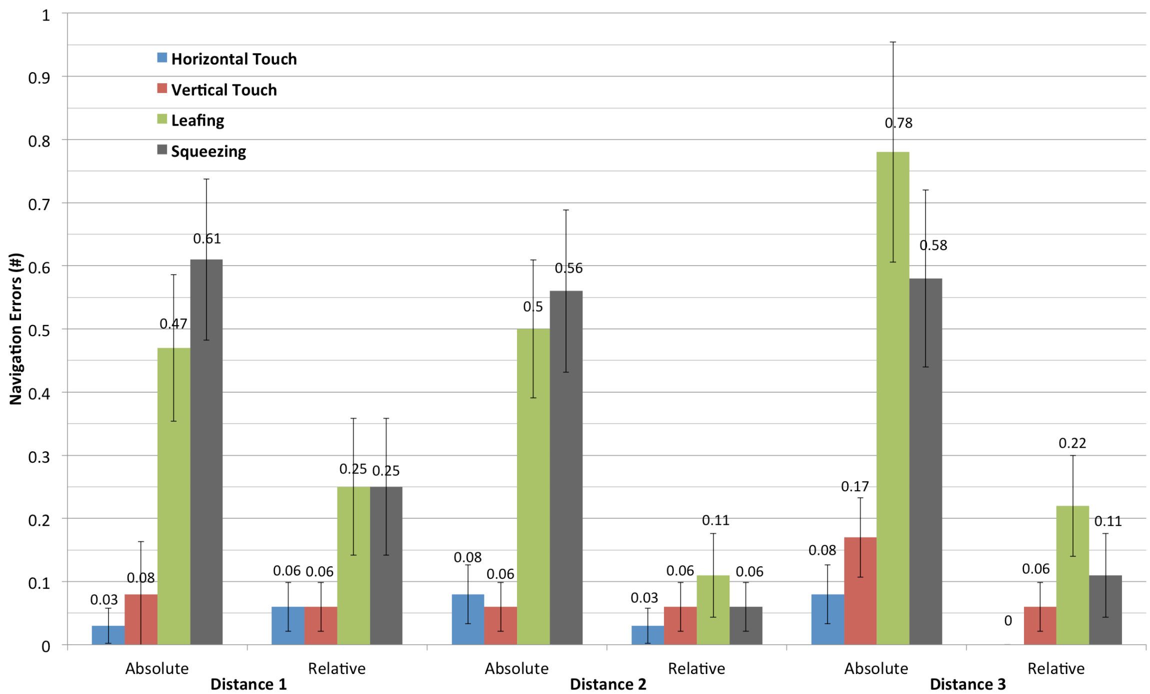

Figure 20. Mean navigation errors per trial across interaction technique, input modality, and target distance.

5.4.2 User Feedback

Responses to the 5-point Likert scale question of whether an input condition was *easy to complete the task* with can be found in Appendix B. We tested for significance using a repeated measures ANOVA on the Aligned Rank Transform (ART) [50] of the Likert-scale scores. Typically, one might perform a Friedman's Two-Way ANOVA by Ranks over the eight input conditions. Each input condition, however, consists of the independent factors, *interaction technique* and *input modality*. The Friedman test can only handle one factor and cannot be used to examine interactions effects. The ART procedure preprocesses the data to be aligned and ranked for each main effect and interaction effect. A full-factorial (repeated measures in our case)

ANOVA can then be run on each set of aligned responses. A basic overview of the Aligned Rank Transform procedure follows:

1. For each raw response *Y*, compute its residual.
2. Compute estimate effects for all main and interaction effects.
3. Compute aligned response *Y'*. This process strips all effects from *Y* but the one of interest (a specific main effect or interaction effect).
4. Assign ranks *Y''*. In the case of a tie among *k* values, each rank is the sum of ranks divided by *k*.
5. Perform a full-factorial ANOVA on *Y''*. All main effects and interaction effects are included in the model, but only the result corresponding to the aligned effect should be considered.
6. If necessary, perform post-hoc comparisons only for the effect for which the data was aligned.
7. Repeat steps 1-6 for each main effect and interaction effect.

The repeated measures ANOVA found a significant main effect of *interaction technique* ($F_{3,33} = 4.34$, $p<0.05$). Pairwise post-hoc tests, with Bonferroni corrected comparisons between *interaction techniques*, revealed that vertical touch was easier to complete the task with than *squeezing*. *Input modality* was not a significant factor, nor was there an interaction effect between *interaction technique* and *input modality*.

For an example comparison, we tested significance using a Friedman's Two-Way ANOVA by Ranks on the Likert-scale scores for *easy to complete the task*. No significant differences between input conditions were found.

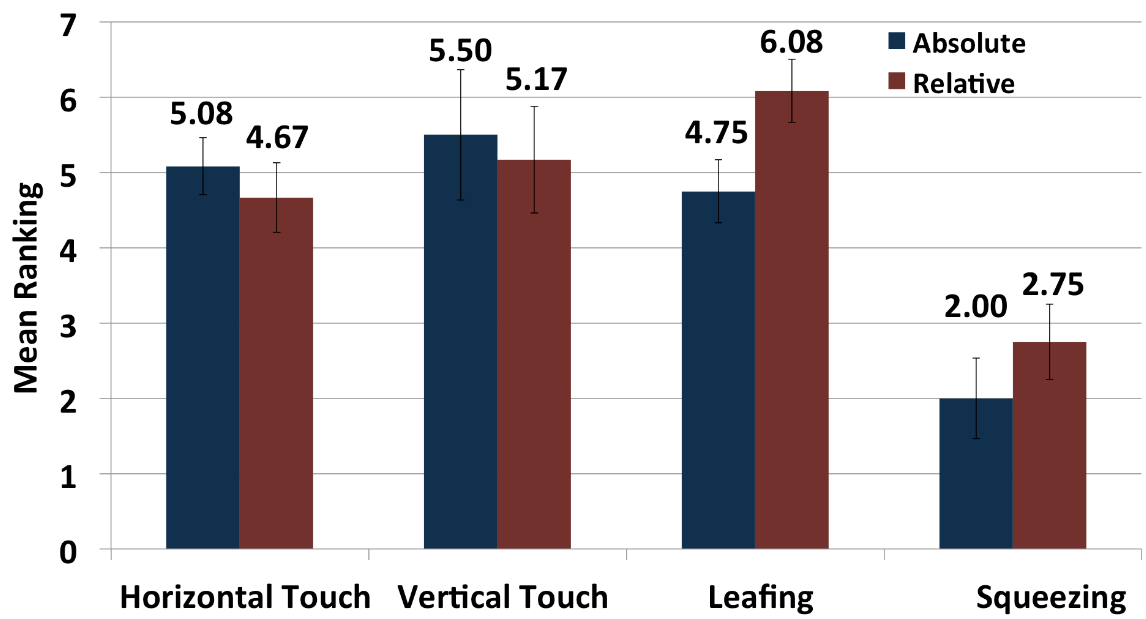

Figure 21. Mean preference rankings across interaction technique and input modality. Results are inverted: larger numbers indicate higher preference.

Responses to the 5-point Likert scale question of whether a condition was *comfortable to use* can be found in Appendix B. We tested significance using a repeated measures ANOVA on the ART of the Likert-scale scores. There was a significant main effect of *interaction technique* ($F_{3,33} = 7.45$, $p<0.05$). Pairwise post-hoc tests, with Bonferroni corrected comparisons, revealed that *squeezing* was not as comfortable to use as vertical touch and *leafing*.

Figure 21 shows the mean ranks from participants' rankings of the eight input conditions. We tested for significance using a repeated measures ANOVA on the ART of the rankings. The analysis showed that *interaction technique* was a significant factor ($F_{3,33} = 10.40$, $p<0.001$). Pairwise post-hoc tests, with Bonferroni corrected comparisons, revealed that *squeezing* was ranked significantly lower than all other *interaction techniques*. Kendall's W, or Kendall's coefficient of concordance, was 0.33.

Detailed ranking information, and *easy to complete the task* and *comfortable to use* ratings, can be found in Appendix B.

5.5 Discussion

Across all interaction techniques, conditions with the absolute mapping were faster than those using the relative mapping, as hypothesized (H5). This result is not surprising and can be traced to the pattern of motion associated with the two mappings. The relative mapping requires the user to perform discrete, repeated actions to move within the document. After each gesture, participants are forced to release contact with the sensor to perform subsequent navigation (releasing a bend or clutching with the thumb). These short breaks, in both time and contact with the sensor, are avoided with the absolute mapping. In addition, absolute maps a larger amount of navigation to a smaller range of sensor values, thereby reducing the total amount of movement required to traverse the entire document. The absolute mapping, however, produces a larger number of navigation errors, as minute movements may result in exaggerated or unintentional navigation (H7).

We observed that some of our hypotheses were not confirmed with this experiment. We had predicted that within touch input, vertical scrolling would record faster movement times than horizontal scrolling (H1). Both techniques were mapped to the same physical distance on the touch sensor, but during vertical touch, the thumb has a more comfortable range of motion afforded by gripping the side of the display. Nevertheless, no significant differences were found. We found that user feedback, in the form of preference rankings (Figure 21) sided with this prediction.

We had also hypothesized that *leafing* would be faster than *squeezing* (H2). We made this prediction on the basis that more subtle, and therefore shorter, more accurate bends could be performed with a supporting hand. Although the mean movement times might suggest this was the case (Figure 19), we did not find any significant difference between the two interaction techniques.

An interesting result is that our prediction that touch interaction would be faster than bend interaction (H3) was not confirmed. Again, no significant differences were found between the interaction techniques. For searching tasks on a flexible mobile device, we do not have evidence to suggest that either type of interaction is faster.

At the same time, the analysis showed that bend interactions resulted in a statistically significantly larger number of errors (Figure 20). We believe four possibilities may account for the large discrepancy in the number of errors: the differences between touch's positional motion and bend's rotational motion, residual noise from the bend sensor, the material properties of our flexible display, or a greater familiarity with touch gestures. Any noise is likely specific to the particular implementation of our bend sensing and unlikely representative of the interaction techniques in general. In addition, after releasing a bend, the lamination on the E Ink display forced the substrate to return to its resting state faster than an unaltered flexible EPD. If a participant overshot the target and had to release his grip, the stiffness of the lamination might have caused him to navigate backwards further than anticipated, even after training. This results in a compounding of errors, especially in the absolute conditions. We found that all participants recorded multiple errors per trial in 30-50% of their trials with errors.

With respect to the user feedback, the most apparent result is that participants disliked the *squeezing* technique to navigate depth exclusively. Although the coefficient of concordance over all input conditions was low, the dislike for *squeezing* can be observed. Our analysis showed that *squeezing* was not easy to complete the task with, or comfortable to use, and was ranked significantly lower than the other interaction techniques. Much of the dislike can be traced back to the material properties of the FlexView prototype. Participants commented that it was "*uncomfortable to keep grasping for a long time at precisely different pressures*" and pointed out in these conditions that the device "*was very stiff*". The stiffness of the laminated display fought back against the participants' deformation and made it difficult to be precise. Although squeezing was not significantly slower than any other technique, the feedback suggests that participants would not choose to use it when navigating to precise points within depth.

Although we did not confirm our hypothesis that vertical touch would be faster than horizontal touch (H1), user feedback indicates that vertical touch was easier to complete the task with, more comfortable to use, and preferred more overall. The basis for our prediction was that was it was an existing technique and because of the ergonomics of the thumb. Participants arrived at similar conclusions. One participant said of horizontal touch, "*I wouldn't want to move my thumb in this direction. The motion felt odd*," and another felt that "*it was easy for the thumb to feel cramped in that smaller space*," indicating that the orientation of the device may also be a factor. Citing previous experience with mobile touch interfaces, one participant suggested that relative vertical touch was "*like scrolling on an iPhone*." A different participant mentioned that relative vertical touch was "*easier than [relative] horizontal touch*." Again, the physiology of the thumb in vertical touch granted greater comfort, according to participants. Participants commented that, "*the thumb position felt more natural in the vertical motion than it did in the horizontal motion*" and that vertical touch was "*much more comfortable than horizontal. My thumb moved more naturally*."

There was also a strong preference for *leafing* in this experiment. One participant said of absolute *leafing*, "*This was an intuitive way to flip through the pages of the document, just like* *flipping through a book!*" and "*I didn't have to think about how to turn the page*." *Leafing*, however, suffered similar stiffness problems as *squeezing*: "*Maintaining the bend was hard. Especially for the maximum bend conditions*." There was an even stronger preference for relative *leafing*. One participant found that it "*[worked] better than the touch based flicks*." Another participant also picked up on the same metaphor as absolute leafing, "*This was easy. Very similar to flipping through a book*."

Chapter 6

Experiment 2

In this experiment we investigated the performance of touch and bend interaction in a three-dimensional navigation scenario: traversing a multi-scale map in a pan-zoom task. That is, navigation at multiple levels of detail (i.e. depth) and panning along a plane. Panning was performed consistently with touch input and the interaction technique varied for zooming. Our goal was to investigate whether the addition of bend interaction could improve performance in tasks that benefit from a high degree of parallelism.

6.1 Task

Participants performed a variation of Guiard's multi-scale pointing task [13] which demonstrated the validity of Fitts's Law when targeting in multi-scale information spaces.

Participants were required to navigate back and forth between two circles placed off-screen, past the top and bottom boundaries of the display. The targets were initially positioned along a vertical axis that passed through a crosshair drawn on the center of the display. Figure 22 illustrates the experimental configuration.

Each trial began at the default zoom: 100%. The target was painted black and the second circle was painted a faint gray. Participants could use a combination of pan and zoom operations until the target hit the crosshair. Although the task may involve intermediate zoom levels, the task is only registered as complete at 100% zoom, with a small (2%) margin of error for the correct zoom level. A dashed circle around the crosshair indicates the size of the target required to complete a trial. When this is achieved, the targets swap positions and the next trial begins immediately.

Unlike Guiard's multi-scale pointing, our task did not use a series of concentric circles to indicate the zoom level and the position of off-screen targets. Instead, we used Gustafson et al.'s wedge visualizations [15] to communicate the location and zoom-level of the target. The wedge visualizations prevent the phenomenon of desert fog [23], a common problem of zooming interfaces, where one gets lost if one zooms into an empty region of the navigation space. To circumvent this issue, the software draws a triangle straddling the appropriate edge of the viewport. Two vertices of the triangle are displayed and the third is located at the target. As the participant navigates, the two clipped edges converge and diverge indicating how close the target is from the viewport. The third edge expands and contracts to indicate the current zoom level. When the target was out of sight, participants always knew in which direction and how far they needed to navigate. Participants in our pilot studies found that on the low-resolution E Ink display, the wedge visualizations were less distracting and disorienting compared to concentric circles.

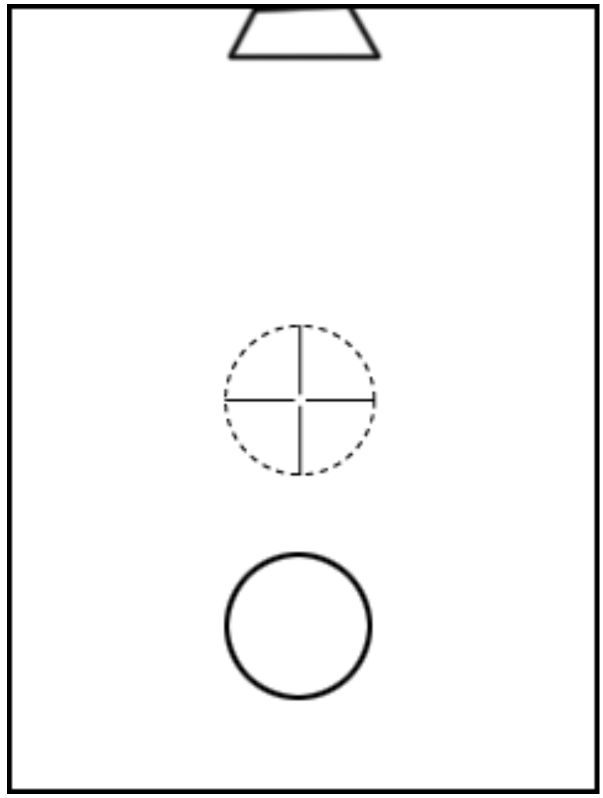

Figure 22. Experiment 2 configuration.

6.2 Design

We used a 3x3x3 factorial repeated-measures within-subject design. Our factors were: *interaction technique* (absolute squeezing, relative leafing, and multi-touch), *target distance* (640, 920, and 1280 pixels, corresponding to 2, 3, and 4 screen heights), and *target size* (20, 60, and 180 pixels). Each participant performed a set of 4 reciprocal trials per combination of factors, for a total of 108 trials (3 interaction techniques x 3 target distances x 3 target sizes x 4 trials). Condition order was counter-balanced between participants. The experimental sessions lasted about 40 minutes, including training. Participants trained with each interaction technique until they achieved less than 10% improvement between trials.

6.2.1 User Feedback

Participants were asked to rate each *interaction technique* on whether it was *easy to complete the task* with. The questions were structured using a 5-point Likert scale (1 = Strongly Disagree to 5 = Strongly Agree). In addition, participants were asked to rank the *interaction techniques* in their preferred order.

6.3 Hypotheses

We hypothesized that *squeezing* to zoom would have the fastest movement times, followed by multi-touch and *leafing* to zoom (H1). This hypothesis was based on the *squeezing* technique's ability to pan and zoom with a higher degree of parallelism, resulting in faster task completion. We hypothesized that larger targets would result in faster movement times (H2), and likewise, smaller distances would result in faster movement times (H3).

6.4 Results

6.4.1 Performance Analysis

We analyzed the collected measures by performing a repeated measures factorial analysis of variance (ANOVA) using *interaction technique* (4) x *target distance* (3) x *target size* (3) on movement time.

For movement time (Figure 23), the analysis showed that interaction technique was a significant factor ($F_{2,22} = 22.28$, $p<0.001$). Pairwise post-hoc tests, with Bonferroni corrected comparisons between *interaction techniques*, reveal that *squeezing* was the fastest technique, followed by multi-touch, then *leafing*. *Target distance* ($F_{2,22} = 35.79$, $p<0.001$) and *target size* ($F_{2,22} = 30.72$, $p<0.001$) were also significant factors, with larger targets and shorter distances

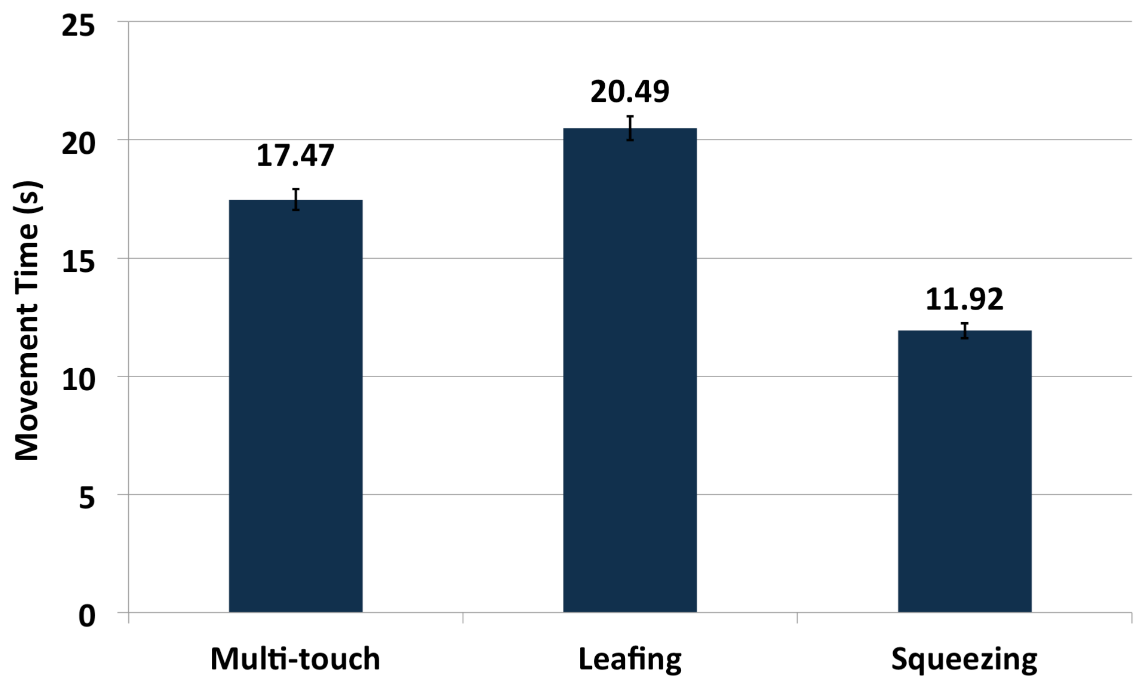

Figure 23. Mean movement times for panning with touch, while zooming with leafing, squeezing or touch.

resulting in lower movement times. In addition, we found a significant interaction effect between *interaction technique* and *target distance* ($F_{4,44} = 5.09$, $p<0.05$) as well as between *interaction technique* and *target size* ($F_{4,44} = 9.70$, $p<0.001$).

6.4.2 Behavioural Analysis

Figure 24 presents a graph of zoom levels recorded across all trials for each *interaction technique*. The data was normalized to the movement time for each trial. Data points represent the mean zoom level for all participants at each percentage of task completion.

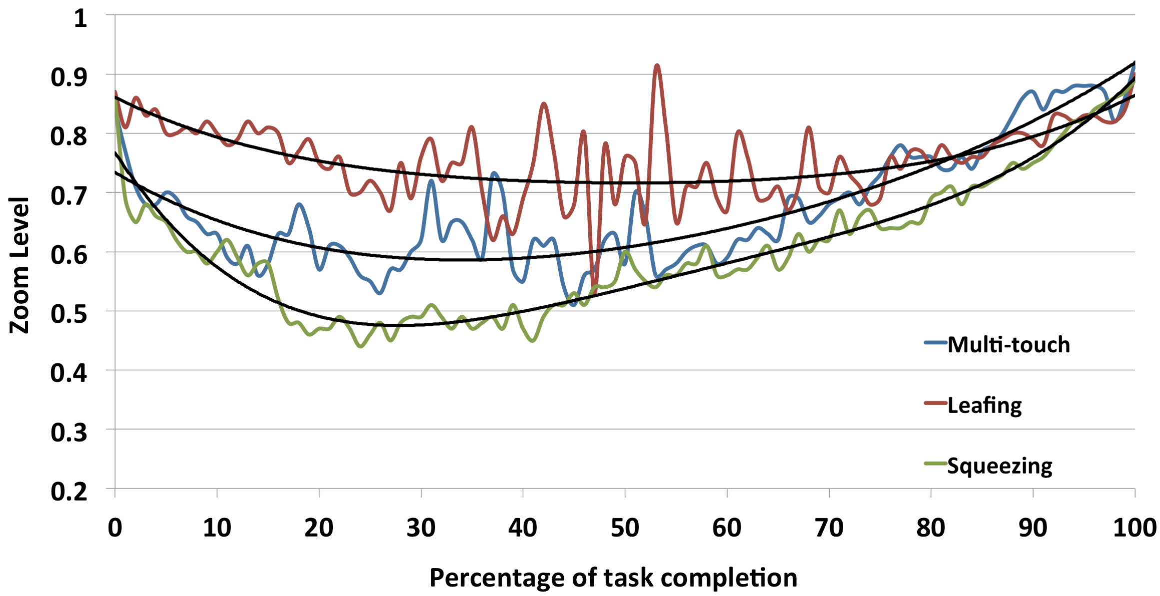

Figure 24. Variation in zoom levels recorded for Experiment 2. Trend lines are superimposed.

6.4.3 User Feedback

Responses to the 5-point Likert scale question of whether an interaction technique was *easy to complete the task* with can be found in Appendix B. We tested significance using a Friedman's Two-Way Analysis of Variance by Ranks on the Likert-scale scores. No significant differences between *interaction techniques* were found.

Figure 25 shows the results of the participants' preference rankings. We tested significance using a Friedman's Two-Way Analysis of Variance by Ranks on the rankings. No significant differences between mean rankings were found. The analysis showed that Kendall's coefficient of concordance was 0.21.

Detailed ranking information, and *easy to complete the task* ratings, can be found in Appendix B.

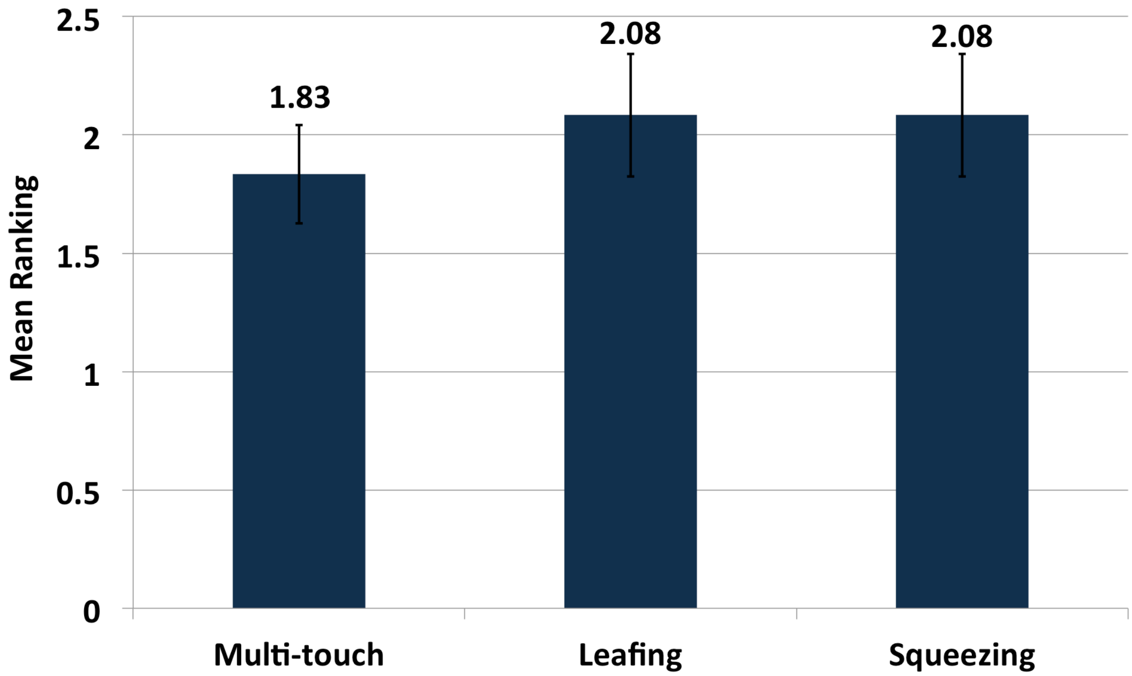

Figure 25. Mean preference rankings for each interaction technique. Results are inverted: larger numbers indicate higher preference.

6.5 Discussion

The results show that augmenting touch input with bend interaction can improve performance in tasks that benefit from a high degree of parallelism.

The observed movement times (Figure 23) confirm our hypothesis that *squeezing* to zoom would achieve the fastest movement times (H1). Also as predicted, multi-touch outperformed zooming using *leafing*. These results are indicative of how easily the sub-tasks of panning and zooming can be done in parallel, resulting in a 32% improvement in movement times between *squeezing* and multi-touch, and a 15% improvement of multi-touch over *leafing*. With all interaction techniques, participants held the device with their non-dominant hand and panned with the index finger of their dominant hand.

With *leafing*, participants had to stop panning, remove their hand from the surface, and then grasp the side of the display each time they wished to zoom. These interruptions may have both slowed the participant down and influenced them to zoom less frequently. When *leafing*, participants also zoomed out less than other techniques (Figure 24), increasing the amount of panning required. This resulted in slower movement times, and is in line with the results of Kaptelinin's evaluation of zooming interfaces [24].

When participants used multi-touch, they had to stop panning and switch to a pinch-to-zoom gesture. Although scaling may include a translation component, no *significant* amount of panning could be performed while zooming. The switch between panning and zoom was less disruptive to the task than *leafing,* and as a result, participants zoomed out more (Figure 24). Although both *leafing* and pinch-to-zoom are direct interaction techniques for zooming, when a user must switch to new action, the one most similar to the preceding action will be faster. Despite this, more participants found *leafing* easier to use than touch. Comments suggested that *leafing* allowed participants to zoom in place more easily, due to multi-touch scaling's translation component.

We believe *squeezing* the display to zoom achieved the fastest movement times because participants were not forced to choose between panning and zooming at any moment. *Squeezing* to zoom could also be assisted with the dominant hand by applying additional force when panning. When pan and zoom operations are performed in parallel, participants had no disruptions between sub-tasks, and zoomed out more than any other interaction technique (Figure 24).

With respect to user feedback, no statistically significant results were derived from the Likert question of whether an interaction technique was *easy to complete the task* with or the rankings. Although the performance analysis showed a significant difference between each of the techniques, it is interesting that the overall user feedback did not reveal a similar pattern (Figure 25). Unlike our first experiment, we only asked one Likert-style question, whether an interaction technique was *easy to complete the task with*, but did not ask whether it was *comfortable to use*. It is possible that with the omission of the latter question, participants treated the single question as their opportunity to rate the technique on both criteria, given their chance in the previous experiment.

The individual participant comments, however, lend insight into the preferences of interaction techniques for zooming. We predicted that *squeezing* to zoom was the fastest technique because of the high degree of parallelism between panning and zooming. Comments such as, "*It was the easiest to use and was completely independent of the controls used to move*" and "*Being able to move and zoom together was great. It seemed like one continuous motion could be used to complete the task*" indicate that participants picked up on this ability.

Squeezing was the least preferred interaction technique in Experiment 1, largely because of the material properties of the E Ink display; the laminated substrate tries to pull back to a flat state. These opinions carried over to the pan and zoom task. Participants commented that it "*may be hard to maintain the bending in longer term use*."

Multi-touch did not perform as well as *squeezing*, but was faster compared to *leafing* to zoom. Participants also picked up on why this may have been the case. "*This was easy, but I needed to move between move and zoom. I could keep my hand in one place, unlike the bend on the side*," one participant commented. When *leafing*, participants found the zooming action easy to control, much like the depth navigation in Experiment 1, but had trouble maintaining a continuous motion.

Chapter 7

Conclusion

In this thesis, we presented FlexView, a prototype system and set of interaction techniques, that allows users to navigate through depth-arranged information using display curvature as an additional input channel. With *leafing*, the user holds one side of display and bends the opposite side. *Squeezing* involves gripping the display in one hand and applying pressure on both sides to create concave or convex curvatures, and supports concurrent interaction with touch input. FlexView was designed to realize the following goals: to *augment touch*, provide *embodiment*, *maximize display area*, and to support *parallelism*.

We evaluated FlexView in two experiments designed to test these goals. The first experiment demonstrated that bend interaction is comparable to touch input for navigation through depth-arranged content as it *embodies* navigation through a stack. Our second experiment established that by *augmenting touch* with bend interaction in *parallel*, targeting performance in a pan-zoom task was significantly improved. FlexView techniques do not require additional on screen elements to operate and minimize screen occlusion, fulfilling the design goal of *maximize display area*.

7.1 Design Recommendations

The creation and evaluation of FlexView can be summarized into the following design recommendations for future flexible mobile devices used to navigate depth-arranged content:

1. When navigating depth-arranged content in a one-dimensional task, i.e. navigating depth exclusively, we recommend using either *leafing* or vertical touch. Although there were no significant differences between the movement times of interaction techniques, *leafing* and

vertical touch were found to be the easiest, most comfortable and most preferred methods of navigating depth.

2. When navigating depth-arranged content in a three-dimensional task, i.e. navigating a two-dimensional plane while also navigating through depth, we recommend using *squeezing* to navigate depth with the non-dominant hand and using touch input with the dominant hand to navigate the two-dimensional plane, in parallel. We also recommend that navigation through depth should be a more course interaction (e.g. zooming) compared to the finer manipulations along the plane (e.g. panning). This recommendation follows from the results of our second evaluation and is consistent with Guiard's Kinematic Chain theory [14], where the non-dominant hand provides the frame of reference for the detailed actions of the dominant hand.
3. The form factor and material qualities of the flexible device are important to how users perceive interaction and affect their ability to perform tasks. We distinguished three qualities that influence both affective and qualitative measures:
    a. *Size and Orientation*. To represent Guiard's Kinematic Chain [14] with *squeezing*, the mobile device must be operated in a portrait orientation, or one dimension of the device should comfortably sit in the palm. The sides of the flexible display should be able to rest at the base of the thumb and the third knuckle of the fingers for the greatest amount of control over curvature. If necessary, the tips of the thumb and fingers should be able to grasp the sides of the device.
    b. *Flexibility*. The degree to which the flexible mobile device can deform should strike a balance between complete malleability and having a supporting structure. Flexible displays that are extremely pliable are difficult to interact with using touch

input and requires the use of structural holds to create supporting zones [5]. FlexView's display was laminated because of technical constraints. Though still highly flexible, deforming the display and holding it in a specific position required more force than we believe necessary, especially during pronounced bends. This property of our E Ink display influenced the qualitative measures of our interaction techniques, most notably with *squeezing*. A flexible mobile device should be easily deformed and should not require excessive force to maintain the deformation. At the same time, it should be rigid enough to hold itself up under its own weight and while touching the surface. The right combination of these qualities is a difficult balance to maintain, but is critical for flexible mobile devices that respond well affectively and qualitatively.

c. *Texture*. With FlexView, not only did the display need to sit in the hand properly, it had to feel like a device a participant would want to hold. We covered the back and sides of the device with a bookbinding tape, giving it a paper-like quality. Combined with the ink-on-paper look of the electrophoretic display and the book navigation task, the bookbinding tape gave users the impression of navigating through a physical book. Easy operation of touch input requires low friction when interacting parallel to the display surface, but device deformation occurs perpendicular to the display. The bookbinding tape around the border of the display area added enough friction to reduce the force required to initiate bends. Our recommendation is line with Wightman et al.'s exploration of e-book navigation [48].

7.2 Limitations and Future Work

FlexView was designed with specific design goals and under particular technical constraints. Although the empirical findings support our design recommendations, many of the details in the experimental results pertain only to our specific hardware implementation, which could be improved in several ways.

FlexView uses a flexible electrophoretic display that allows the device to bend. A consequence of using this E Ink display is its low refresh rate. The E Ink was fast enough to perform our experimental evaluations, but is not suitable for certain tasks (i.e. video navigation) or commercialization. Future flexible mobile devices could take advantage of newer technologies, such as flexible organic light-emitting diodes (FOLEDs). FOLED displays have higher refresh rates, approximately 60 Hz, and can operate in 24-bit colour.

Using the flexible E Ink display in FlexView also prevented us from using a touch solution that could detect a participant's uninstrumented finger. Again, with newer technologies such as FOLEDs, this constraint could be removed.

Much of our user feedback pertained to the stiffness of the display. Flexible display technologies are still in their infancy, and future devices need not be laminated. A full examination of the material qualities of flexible mobile devices would be of great benefit to the design space.

In addition, a non-trivial portion of the FlexView apparatus was housed away from the interactive part of the device. Flexible display technologies still require the use of rigid electronics to drive the displays. It would be interesting to explore flexible interactions in a truly mobile scenario, when it is possible to have an untethered solution.

We designed FlexView with the ability to navigate depth-arranged content in mind. Nevertheless, FlexView techniques could easily be extended to other domains. Other methods of navigating large information spaces could be controlled using the fusion of touch and bend input. Although with FlexView we explored single touch and bend input in parallel, future interfaces could take advantage of both bend and multi-touch. Our interaction techniques were designed to increase parallelism through the bimanual interplay of bend and touch in an asymmetric fashion. Here, the non-dominant bent the device while the dominant hand provided touch input. Future interactions could relax our design constraints by holding the flexible mobile device in both hands in a landscape orientation, where either or both hands could control device curvature and provide touch input using the thumbs.

References

1. Apple iPhone. http://www.apple.com/iphone. (as of April 2012).

2. Arduino. http://www.arduino.cc. (as of April 2012).

3. Baudisch, P., Good, N., and Bellotti, V. Keeping Things in Context: A Comparative Evaluation of Focus Plus Context Screens, Overviews, and Zooming. *Proc. CHI*, (2002), 259-266.

4. Bergman, J., Kauko, J., and Keränen, J. Hands on music: physical approach to interaction with digital music. *Proc. MobileHCI*, (2009), Article 33, 11 Pages.

5. Dijkstra, R., Perez, C., and Vertegaal, R. Evaluating effects of structural holds on pointing and dragging performance with flexible displays. *Proc. CHI*, (2011), 1293-1302.

6. Dupont Pyralux. http://www2.dupont.com/Pyralux/en_US/index.html. (as of April 2012).

7. Eslambolchilar, P. and Murray-Smith, R. Tilt-based automatic zooming and scaling in mobile devices–a state-space implementation. *Proc. MobileHCI*, (2004), 219–260.

8. Fitts, P. The information capacity of the human motor system in controlling the amplitude of movement. *Experimental Psychology*, 47(6), (1954), 381-391.

9. Flexible Display Center. http://flexdisplay.asu.edu. (as of April 2012).

10. Flexpoint. http://www.flexpoint.com. (as of April 2012).

11. Gallant, D. and Seniuk, A. Towards more paper-like input: flexible input devices for foldable interaction styles. *Proc. UIST*, (2008), 283-286.

12. Google Maps. http://maps.google.com. (as of April 2012).

13. Guiard, Y. and Beaudouin-Lafon, M. Target acquisition in multiscale electronic worlds. *Human-Computer Studies*, 61(6) (2004), 875–905.

14. Guiard, Y. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Motor Behavior*, (1987), 486-517.

15. Gustafson, S., Baudisch, P., and Gutwin, C. Wedge: clutter-free visualization of off-screen locations. *Proc. UIST*, (2008), 787-796.

16. Gutwin, C. and Fedak, C. Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques. *Proc. UIST*, (2004), 145–152.

17. Harrison, B.L., Fishkin, K.P., Gujar, A., Mochon, C., and Want, R. Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. *Proc. UIST*, (1998), 17-24.

18. Herkenrath, G., Karrer, T., and Borchers, J. Twend: twisting and bending as new interaction gesture in mobile devices. *Proc. CHI*, (2008), 3819–3824.

19. Hinckley, K. and Song, H. Sensor Synaesthesia: Touch in Motion, and Motion in Touch. *Proc. CHI*, (2011), 801-810.

20. Holman, D., Vertegaal, R., Altosaar, M., Kl, C., Troje, N., and Johns, D. PaperWindows : Interaction Techniques for Digital Paper. *Proc. CHI*, (2005), 591-599.

21. Holman, D. and Vertegaal, R. Organic user interfaces: designing computers in any way, shape, or form. *Communications of the ACM*, 51(6), (2008), 48-55.

22. Johnson, J. A comparison of user interfaces for panning on a touch-controlled display. *Proc. CHI*, (1995), 218-225.

23. Jul, S. and Furnas, G. Critical zones in desert fog: aids to multiscale navigation. *Proc. UIST*, (1998), 97-106.

24. Kaptelinin, V. A comparison of four navigation techniques in a 2D browsing task. Proc. CHI, (1995), 282-283.

25. Kildal, J., Paasovaara, S., and Aaltonen, V. Kinetic Device : Designing Interactions with a Deformable Mobile Interface. *Proc. CHI EA*, (2012), 1871-1876.

26. Kim, K., Chang, W., Cho, S., Shim, J., and Lee, H. Hand grip pattern recognition for mobile user interfaces. *Proc. IAAI*, (2006), 1789-1794.

27. Kratz, S., Brodien, I., and Rohs, M. Semi-automatic zooming for mobile map navigation. *Proc. MobileHCI*, (2010), 63–72.

28. Lahey, B., Girouard, A., Burleson, W., and Vertegaal, R. PaperPhone: understanding the use of bend gestures in mobile devices with flexible electronic paper displays. *Proc. CHI*, (2011), 1303–1312.

29. Latulipe, C., Kaplan, C.S., and Clarke, C.L.A. Bimanual and unimanual image alignment: an evaluation of mouse-based techniques. *Proc. UIST*, (2005), 123–131.

30. Lee, S., Kim, S., Jin, B., Choi, E., and Kim, B. How users manipulate deformable displays as input devices. *Proc. CHI*, (2010), 1647-1656.

31. Lyklema, J. Fundamentals of Interface and Colloid Science. 1995.

32. Miyaki, T. and Rekimoto, J. GraspZoom: zooming and scrolling control model for single-handed mobile interaction. *Proc. UIST*, (2009), 1–4.

33. Myers, B.A. and Buxton, W. A Study in Two-Handed Input. Proc. CHI, (1986), 321-326.

34. Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., and Mackay, W. Mid-air pan-and-zoom on wall-sized displays. Proc. CHI, (2011), 177-186.

35. Rao, R. and Card, S.K. The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information. *Proc. CHI*, (1994), 318–322.

36. Robertson, G.G. and Mackinlay, J.D. The document lens. *Proc. UIST*, (1993), 101–108.

37. Sarkar, M. and Brown, M.H. Graphical fisheye views of graphs. *Proc. UIST*, (1992), 83–91.

38. Schwesig, C., Poupyrev, I., and Mori, E. Gummi: a bendable computer. *Proc. CHI*, (2004), 263–270.

39. Sellen, A. and Kurtenbach, G. The prevention of mode errors through sensory feedback. *Human-Computer Interaction*, 7(2), (1992), 141-164.

40. Shneiderman, B. Direct Manipulation: A Step Beyond Programming Languages. *Computer*, 16(8), (1986), 57-69.

41. Tang, J., Zhang, X., and Li, Z.-M. Operational and maximal workspace of the thumb. *Ergonomics*, 51(7), (2008), 1109-18.

42. Tarun, A. Designing and building inexpensive flexible circuits. *Proc. TEI,* (2012), 375-377.

43. Taylor, B. Graspables: grasp-recognition as a user interface. *Proc. CHI*, (2009), 917-925.

44. Wacom. http://www.wacom.com. (as of April 2012).

45. Wagner, J., Huot, S., and Mackay, W. BiTouch and BiPad: Designing Bimanual Interaction for Hand-held Tablets. *Proc. CHI*, (2012), 2317-2326.

46. Watanabe, J.-ichiro, Mochizuki, A., and Horry, Y. Bookisheet: bendable device for browsing content using the metaphor of leafing through the pages. *Proc. UbiComp*, (2008), 360–369.

47. Wellner, P. The DigitalDesk calculator: tangible manipulation on a desk top display. *Proc. UIST*, (1991), 27-33.

48. Wightman, D., Ginn, T., and Vertegaal, R. Bendflip: examining input techniques for electronic book readers with flexible form factors. *Proc. INTERACT*, (2011), 117-133.

49. Windows 7. http://windows.microsoft.com. (as of April 2012).

50. Wobbrock, J.O., Findlater, L., Gergle, D., and Higgins, J.J. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only ANOVA Procedures. *Proc. CHI*, (2011), 143-146.

51. Zytronic. http://www.zytronic.co.uk. (as of April 2012).

Appendix A

Questionnaires

1. Experiment 1

1. I felt that it was easy to complete the task using horizontal touch with the absolute mapping.

   1 = Strongly Disagree … **5 = Strongly Agree**

1	2	3	4	5

Comment:

2. I felt that horizontal touch with the absolute mapping was comfortable to use.

   1 = Strongly Disagree … **5 = Strongly Agree**

1	2	3	4	5

Comment:

3. I felt that it was easy to complete the task using horizontal touch with the relative mapping.

   1 = Strongly Disagree … **5 = Strongly Agree**

1	2	3	4	5

Comment:

4. I felt that horizontal touch with the relative mapping was comfortable to use.

   1 = Strongly Disagree … **5 = Strongly Agree**

1	2	3	4	5

Comment:

5. I felt that it was easy to complete the task using leafing with the absolute mapping.

1 = Strongly Disagree				**5 = Strongly Agree**
1	2	3	4	5

Comment:

6. I felt that leafing with the absolute mapping was comfortable to use.

1 = Strongly Disagree				**5 = Strongly Agree**
1	2	3	4	5

Comment:

7. I felt that it was easy to complete the task using leafing with the relative mapping.

1 = Strongly Disagree				**5 = Strongly Agree**
1	2	3	4	5

Comment:

8. I felt that leafing with the relative mapping was comfortable to use.

1 = Strongly Disagree				**5 = Strongly Agree**
1	2	3	4	5

Comment:

9. I felt that it was easy to complete the task using squeezing with the absolute mapping.

1 = Strongly Disagree				**5 = Strongly Agree**
1	2	3	4	5

Comment:

10. I felt that squeezing with the absolute mapping was comfortable to use.

1 = Strongly Disagree **5 = Strongly Agree**

1	2	3	4	5

Comment:

11. I felt that it was easy to complete the task using squeezing with the relative mapping.

1 = Strongly Disagree **5 = Strongly Agree**

1	2	3	4	5

Comment:

12. I felt that squeezing with the relative mapping was comfortable to use.

1 = Strongly Disagree **5 = Strongly Agree**

1	2	3	4	5

Comment:

13. I felt that it was easy to complete the task using vertical touch with the absolute mapping.

1 = Strongly Disagree **5 = Strongly Agree**

1	2	3	4	5

Comment:

14. I felt that vertical touch with the absolute mapping was comfortable to use.

1 = Strongly Disagree **5 = Strongly Agree**

1	2	3	4	5

Comment:

15. I felt that it was easy to complete the task using vertical touch with the relative mapping.

1 = Strongly Disagree				**5 = Strongly Agree**
1	2	3	4	5

Comment:

16. I felt that vertical touch with the relative mapping was comfortable to use.

1 = Strongly Disagree				**5 = Strongly Agree**
1	2	3	4	5

Comment:

17. Please rank the interaction techniques in decreasing preference (one check per column).

	1 Most Preferred	**2**	**3**	**4**	**5**	**6**	**7**	**8 Least Preferred**
Horizontal Touch (Absolute)								
Horizontal Touch (Relative)								
Squeezing (Absolute)								
Squeezing (Relative)								
Leafing (Absolute)								
Leafing (Relative)								
Vertical Touch (Absolute)								
Vertical Touch (Relative)								

General Comments:

2. Experiment 2

1. To complete the task, I found leafing in order to zoom easy to use.

1 = Strongly Disagree				**5 = Strongly Agree**
1	2	3	4	5

Comment:

2. To complete the task, I found squeezing in order to zoom easy to use.

1 = Strongly Disagree				**5 = Strongly Agree**
1	2	3	4	5

Comment:

3. To complete the task, I found using touch in order to zoom easy to use.

1 = Strongly Disagree				**5 = Strongly Agree**
1	2	3	4	5

Comment:

4. Please rank the interaction techniques in decreasing preference (one check per column).

	1 Most Preferred	**2**	**3 Least Preferred**
Leafing to Zoom, Touch to Pan			
Squeezing to Zoom, Touch to Pan			
Touch to Zoom, Touch to Pan			

General Comments:

Appendix B

Questionnaire Response Tallies

1. Experiment 1

I felt that it was easy to complete the task using ________________.

	1 = Strongly Disagree				**5 = Strongly Agree**
	1	**2**	**3**	**4**	**5**
Horizontal Touch (Absolute)	0	1	1	8	2
Horizontal Touch (Relative)	1	1	3	5	2
Leafing (Absolute)	0	1	2	6	3
Leafing (Relative)	1	2	1	4	4
Squeezing (Absolute)	0	2	3	6	1
Squeezing (Relative)	0	2	3	6	1
Vertical Touch (Absolute)	0	1	1	5	5
Vertical Touch (Relative)	0	0	1	8	3

I felt that _________________ was easy to use.

	1 = Strongly Disagree				5 = Strongly Agree
	1	**2**	**3**	**4**	**5**
Horizontal Touch (Absolute)	0	3	3	1	5
Horizontal Touch (Relative)	1	2	4	2	3
Leafing (Absolute)	0	2	2	5	3
Leafing (Relative)	1	0	0	4	7
Squeezing (Absolute)	1	5	4	2	0
Squeezing (Relative)	1	2	3	5	1
Vertical Touch (Absolute)	0	0	3	5	4
Vertical Touch (Relative)	0	1	2	6	3

Please rank the interaction techniques in decreasing preference.

	1 Most Preferred	2	3	4	5	6	7	8 Least Preferred
Horizontal Touch (Absolute)	3	1	1	1	3	2	0	1
Horizontal Touch (Relative)	1	3	3	0	1	0	1	3
Squeezing (Absolute)	1	0	0	0	0	1	3	7
Squeezing (Relative)	0	0	0	2	1	2	6	1
Leafing (Absolute)	2	1	1	3	0	3	2	0
Leafing (Relative)	3	2	3	2	1	1	0	0

Vertical Touch (Absolute)	2	3	0	2	4	1	0	0
Vertical Touch (Relative)	0	2	4	2	2	2	0	0

2. Experiment 2

To complete the task, I found ________________ in order to zoom easy to use.

	1 = Strongly Disagree				**5 = Strongly Agree**
	1	**2**	**3**	**4**	**5**
Leafing	1	1	3	5	2
Squeezing	0	4	1	3	4
Touch	0	1	3	6	1

Please rank the interaction techniques in decreasing preference.

	1 Most Preferred	**2**	**3 Least Preferred**
Leafing to Zoom, Touch to Pan	5	3	4
Squeezing to Zoom, Touch to Pan	5	3	4
Touch to Zoom, Touch to Pan	2	6	4