COMBINING AUXILIARY FINGER INPUT WITH THUMB TOUCH FOR SINGLE-HANDED MOBILE DEVICE INTERFACES

by

ANDREAS HOLLATZ

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
(October 2015)

Copyright ©Andreas Hollatz, 2015
Abstract

This thesis reports on the use of auxiliary finger input to complement touch-only interactions on mobile devices. While a majority of touchscreen based mobile devices support multi-touch input, mobile device interactions in one-handed usage scenarios are usually limited to a single point of contact with the screen. In most cases, the thumb is the preferred source of touch input. Selecting user interface elements, such as buttons and sliders, requires frequent movement of the thumb, occludes a display, and, to reach targets, demands frequent adjustments of grip.

To tackle these usability problems of single-handed usage scenarios, we explored the use of the auxiliary fingers — that is, the fingers that grip, support, and make contact with a mobile device — as additional input channels. Sensing input from the auxiliary fingers might lead to significantly less thumb movement, with target selection and other interactions distributed across all five digits. We built a series of mobile device prototypes that sense isometric pressure at different areas on their surfaces. To evaluate the performance of this interaction paradigm, we measured task completion times and error rates for common mobile tasks, including document formatting, application switching, and map navigation, and validated that the use of additional fingers for input led to performance gains. We follow-up with a study to measure each finger’s ability to apply pressure on the side of the device and measured the effect of this pressure on the thumb’s range of motion around the screen. Finally, we provide software and hardware design recommendations based on these studies.
Co-Authorship

The prototype design and experimental evaluation on pages 24-43 were conducted collaboratively with David Holman and Amartya Banerjee.
# Table of contents

## CHAPTER 1  INTRODUCTION

1.1 BACKGROUND AND MOTIVATION ................................. 1

1.2 CONTRIBUTIONS AND THESIS OUTLINE .......................... 2

## CHAPTER 2  RELATED WORK

2.1 SOFT MACHINES .................................................. 5

2.2 ONE-HANDED MOBILE INTERACTION .......................... 9

2.3 GRASP-BASED INTERACTION ................................... 12

2.4 AUGMENTED MOBILE INTERFACES (SENSOR FUSION) ....... 13

2.5 DEFORMABLE MOBILE INTERACTIONS .......................... 15

2.6 ISOMETRIC FORCE BASED INTERACTION ....................... 16

2.7 HAND ANATOMY AND FUNCTION ................................. 17

2.7.1 Anatomy ...................................................... 17

2.7.1.1 Forearm and Extrinsic Muscles ............................. 17

2.7.1.2 The Wrist ............................................... 18

2.7.1.3 The Hand ............................................... 19

2.7.1.4 The Digits ............................................... 19

2.7.2 Hand Function .................................................. 20

2.7.2.1 Internal and Manipulation Forces ......................... 20

2.7.2.2 Finger independence of movement ....................... 21

2.7.2.3 Movement Speed ......................................... 22

## CHAPTER 3  THE UNIFONE PROTOTYPE

3.1 UNIFONE DESIGN PROCESS ...................................... 24

3.1.1 Unifone Auxiliary Touch Gestures ............................ 30

3.2 DISTINGUISHING HOLDING GRASP AND INPUT SQUEEZE .... 31

3.3 FORCE SENSORS .................................................. 31

3.4 PHYSICAL CONNECTION TO DEVICE ........................... 33

3.5 SENSOR PROCESSING AND DEVICE COMMUNICATION ........ 34
CHAPTER 4 THE ISOPHONE PROTOTYPE ................................................................. 44
  4.1 ISOPHONE CONCEPT .................................................................................. 44
  4.2 HARDWARE .................................................................................................. 45
  4.3 SOFTWARE .................................................................................................... 47
    4.3.1 Device Communication ......................................................................... 47
  4.4 SIGNAL PROCESSING AND ANALYSIS ...................................................... 48
    4.4.1.1 Thresholding ..................................................................................... 48
    4.4.1.2 Blob Finding and Finger Positions .................................................... 48
    4.4.1.3 Gesture Recognition and Event handling ............................................ 49
  4.5 INTERACTIONS ............................................................................................ 50
    4.5.1.1 Home Gesture ................................................................................... 50
    4.5.1.2 Zoom Control .................................................................................... 51
  4.6 ISOPHONE EXPERIMENT .......................................................................... 51
    4.6.1 Experiment Description ......................................................................... 51
    4.6.2 Participants ............................................................................................ 53
    4.6.3 Results ..................................................................................................... 54
      4.6.3.1 Range of motion results .................................................................. 54
      4.6.3.2 TLX Results .................................................................................... 54
      4.6.3.4 Inter-response interval results ......................................................... 55
    4.6.4 Discussion ............................................................................................... 56
      4.6.4.1 Screen Area ..................................................................................... 56
      4.6.4.2 Inter-response Intervals ................................................................. 59
List of Figures

FIGURE 1. The Xerox 5700 printing system with a touchscreen interface [50] .................................................. 7
FIGURE 2. The IBM Simon was the world’s first smartphone [7] ........................................................... 8
FIGURE 3. The Neonode N1 was the first mobile to use swipe gestures [46] .............................................. 8
FIGURE 4. The original Apple iPhone marked the beginning of the current mobile paradigm [28] .......... 8
FIGURE 5. The Xerox PARCTab [49] ........................................................................................................ 11
FIGURE 6. Single Handed Zoom from Sony ................................................................................................ 12
FIGURE 7. Comparison of inter-response interval for finger motion for a range of activities [26] .......... 23
FIGURE 9. Sensor placement for the first iteration of the Unifone prototype ......................................... 26
FIGURE 10. An early iteration of the Unifone prototype: a silicon skin wraps around pressures sensor attached to an iPod touch. A Phidgets Interface Kit, attached to a Desktop computer, acquires the pressure data. ................................................................................................................. 27
FIGURE 11 (a) Top Squeeze Gesture: The user positions the index and middle fingers near the top corner of the mobile. Typically, the other fingers are not raised (the little finger is shown extended for clarity) (b) Middle Squeeze Gesture. The middle and ring finger are positioned around the middle of the device and push inwards, and (c) The ring and little fingers squeeze the bottom corner inwards ........................................... 31
FIGURE 13. The Unifone pressure sensor module: two aluminum planks sandwich pressure sensors on opposite ends. ........................................................................................................................................ 36
FIGURE 14 Results for Unifone Experiment. Standard error above and below the mean is shown. .......... 42
FIGURE 15. The ISOPHONE Prototype with live sensor feedback shown on screen .................................. 45
FIGURE 16. The ISOPHONE sensor design schematic .................................................................................. 46
FIGURE 17. The ISOPHONE sensor to device connection schematic .......................................................... 47
FIGURE 18. State diagram for ISOPHONE input and gesture classification ............................................... 50
FIGURE 19. Force vs. Time diagram used to illustrate ISOPHONE’s Inter-response Interval .................. 53
FIGURE 20. Thumb contact frequency difference from AVG. For the (a) index, (b) middle (c) ring and, (d) little fingers. Red areas represent screen locations that were contacted more frequently than AVG. .......... 58
FIGURE 21. Normalized pressure from the middle and ring (a), and index and ring (b) ......................... 59
FIGURE 22. Graphical representations of data collected during the ISOPHONE study. Averages across participants are shown for a) screen area reached by the thumb b) TLX score (Reminder: a lower TLX score indicates a more favorable result) c) inter-response interval d) force per finger ........................................... 66
List of Tables

**TABLE 1.** SCREEN AREA REACHED BY THUMB IN CM\(^2\) ................................................................. 54

**TABLE 2.** TLX SCORES ....................................................................................................................... 55

**TABLE 3** INTER-RESPONSE INTERVALS IN MILLISECONDS .......................................................... 56
Chapter 1

Introduction

1.1 Background and Motivation

Advancements in mobile technology have enabled highly portable devices that, compared to earlier feature phones, contain a significant amount of computational power. Before the iPhone’s introduction in 2007, many mobile devices had their interactive surface area divided between a small screen and a number of physical controls. Since then, there has been a trend of increased display size and a corresponding decrease in the physical space allotted for hardware controls. This effect is explained by the touch-sensitive display (touchscreen), and its dual purpose for input and output; it affords the replacement of physical controls with virtual buttons and gestures. Although devices with touchscreen make many tasks easier, such as web browsing or composing an email, they are still limited for more complex tasks, such as document editing and three-dimensional spatial navigation.

In particular, one limitation is that the viewable area of a touchscreen is partially occluded by a user's finger when in use. Although a minor obstruction creates a negligible impact on tasks that do not require precise targeting, such as scrolling through an email, it is prohibitive when more precise targeting or manipulation is required (e.g., selecting a sentence while editing a document). Additionally, complex tasks such as text entry, drawing, and 3D spatial interactions often require numerous on-screen buttons and additional Graphical User Interface (GUI) controls that occupy valuable display area.

Often, touch gestures are used to increase the range and number of interaction available without occupying additional display space. These gestures can be highly usable when they embody a
strong physical metaphor such as *swipe-to-scroll* and *pinch-to-zoom* navigation gestures found on most smartphones today. As the number of available gestures increases, so does complexity and, typically, user error. Even relatively simple gesture combinations, such as slide-to-scroll and tap-to-select, often result in false positives. This is further complicated when a user has only one hand available, leaving the thumb of their grasping hand to act as a single touch point. When designing interactions for one-handed use, the contact area between device and user becomes an even more valuable and premium resource. We must maximize its use as an interactive surface if we are to create better mobile user experiences.

### 1.2 Contributions and thesis outline

Although gestures that use tilt, acceleration, spatial location, and even deformation have been actively researched in one-handed scenarios [6,13,16,18] the user’s additional fingers that grasp, or rest along the device are often overlooked as input channels. They can be used to support or extend the thumb’s primary pointing behavior. In this thesis, we explore how input from these auxiliary fingers can impact the usability of mobile interactions.

As an initial step in this space, we focus on one-handed interactions that rely on a user’s nondominant grasp. Users often adopt one-handed strategies when interacting with mobile devices; this leaves a hand free to manage the demands of the real world. The thumb, stabilized by the hand’s supporting fingers, acts as the primary pointing digit in these scenarios. The supporting fingers are sometimes delegated to controlling hardware buttons for settings like volume or power. Most often, they are unused. Though poorly suited for precision, their movement is not completely inhibited. Even when the thumb is actively targeting, the individual fingers of a user’s nondominant grasp are capable of coarse isometric manipulations; the fingertips protrude and curl around the edge of the device. When grasping a mobile phone this way, gestures like squeezing the edges of a device are feasible as an input modality.
Leveraging this auxiliary finger input also influences the design tradeoff between screen size and interface complexity. Modern mobile devices, such as the Apple iPhone and Samsung Galaxy, have adopted high-resolution multi-touch displays that make it easier to interact with complex content. With greater demands on display area, some applications, like Google Earth, limit the functionality of their mobile version. Others leave their functionality intact and distribute workflows across numerous screens and repetitive interface button presses. Instead, using auxiliary input supports transferring interface control elements off the display; for example, auxiliary input gestures can be mapped to general navigational controls, leaving more area free to render content. Although dedicated hardware buttons, such as the trumpet-like buttons in Weiser’s PARCTab [51], can theoretically achieve similar results, an industrial design that is ergonomic for a range of hand and finger sizes is challenging (especially if buttons are placed along the top edge of a device larger than the ParcTab).

To address these problems and to explore the use of auxiliary finger input, we present two prototypes that improve one-handed mobile interaction: Unifone and Isophone. The first prototype, Unifone, senses isometric manipulations using a pressure-distributing accessory placed along its outer edge. Using this hardware sensor, Unifone affords coarse targeting [17]: instead of a precisely located hardware button, users squeeze the phone near its middle or corners, an ergonomic improvement that can adjust sensor location for each user and enhance the pointing behavior of the thumb (see section 4.3). We report on the evaluation of Unifone’s three squeeze-based gestures—a middle squeeze, top corner squeeze, and bottom corner squeeze—and compare them against thumb-only interaction in a set of common mobile tasks. When tightly coupled with the movement of the thumb, Unifone’s squeeze gesture results in superior performance for a set of common mobile tasks.

Unifone’s force-distributed sensor layout reduced the need for precise targeting, but it still placed significant constraints on how the device needed to be held, with some fingers on its upper half
and others on the lower half. Isophone, the second prototype, uses a more sophisticated hardware design with a high spatial resolution sensor array, capable of differentiating individual fingers placed anywhere along its length. This further eliminates the need for precise finger placement during interaction and allowed us to perform a study informing decisions about the placement of hardware controls on the side of the device as well as the placement of onscreen software based controls.

This thesis is presented in six chapters. This first chapter has introduced the limitations of a conventional mobile touchscreen device and revealed the motivation behind including input from auxiliary fingers into their design. The second chapter provides an overview of past work in mobile device interaction design, sensor augmentation and the related mechanics of the human hand. Our first prototype, Unifone, is discussed in chapter three along with a detailed discussion of finger pressure sensing requirements and their enabling technologies. Our discussion moves from discovery — in a series of iterative prototypes we tried a number of different sensor placements and interaction types — to refinement in the Unifone prototype where we improve on the interaction schemes that showed the most promise. We then discuss a comprehensive user study. The second major prototype, Isophone, is presented in chapter four. Using this prototype we evaluated the impact of using different fingers on the thumb’s ability to interact with the screen. We provide a summary of our hardware and software recommendations gathered from our user studies in chapter five. We conclude with a discussion of how we foresee the research continuing and how new technologies will enable this work to be applied to non-planar deformable devices in chapter six.
Chapter 2

Related Work

Our research builds upon and draws cues from the following areas of previous research: (1) the genesis of ‘soft machine’ mobile interfaces, one-handed mobile usage, and interaction techniques related to grasp; (2) extending one-handed interactions through spatial sensing, surface deformation, and examples that leverage pressure-based input for one-handed and bimanual interaction; (3) an overview of the anatomy and function of the human hand. This final section draws many references relating to human hand performance from the fields of anatomy, ergonomics and neuroscience, and is particularly relevant to the study reported in chapter four.

2.1 Soft Machines

Soft machines were proposed by Nakatini and Rohrlieh in [31] as a digital alternative to “hard machines” which they describe as "machines such as stoves, radios and copiers operated with knobs, switches, keys, pushbuttons and other familiar controls. Hard machines have many characteristics that make for ease of learning, efficiency of operation and ease of transfer, but they are ultimately limited by their ‘hardness.’” The authors point to several aspects of hard machines that offer usability advantages over the general-purpose computers of their time. The modularity of a special purpose machine keeps the complexity of interaction within reasonable limits. Scrutiny of the machine’s form leads a user to form conjectures about its function and operation. The one-to-one mapping between controls and their associated operations limits these conjectures to a reasonable number and the immediate feedback of physical controls allows a user to test their conjectures and stimulates the formation of new conjectures in a short amount of time. In contrast to the symbolic operations that require the learning of a human invented
language, the manual controls of a machine, “conform to a universal language based on the physical laws that govern the interactions between physical objects” and can often be learned without instruction or training. This ability to casually learn a machine, the ability to transfer knowledge between machines, and the efficiency of specialized controls are identified by the authors as the primary advantages of a hard machine over a general-purpose computer. While physical controls on hard machines clearly offer usability advantages, their material inflexibility is limiting. As stated by Nakatini and Rohrlich, “we are now in an awkward situation where the functionality of machines is easily changed by software, but the inflexibility of hard controls severely limits the changes that can be accommodated without changing the hardware or compromising the operability of the machine.” They suggest that a way out of this “awkward situation” is through the use of what they refer to as a “soft machine,” or a virtual representation of a physical machine composed of computer images displayed on a touch-screen display. Such a machine offers the “universal language” of physical controls as well as the flexibility and support of sequential disclosure of controls associated a general-purpose computer.

The earliest commercial implementation of a soft machine is found on the control console of the 1980 XEROX 5700 photocopier. Xerox introduced the system out of necessity, a growing number of photocopier features such as copying, duplexing, reducing, collating, stapling, typesetting and printing, would have required more than 130 buttons to control [8]. Instead of creating an extremely large, physically cumbersome, and overwhelming console, Xerox engineers opted for a black and white screen with an infrared touch sensor. A home screen presented soft button representations of the features available machine, and once touched the screen was replaced by the specific controls available to that feature.
In 1992, IBM and Bell South brought the soft machine philosophy to a mobile device with the release of the Simon smartphone. With limited network coverage, and with miniaturization of technology still in nascent stages, the Simon was could not gain widespread adoption. However, the interface style became quite popular on other mobile devices such as the Apple Newton and the Palm Pilot. The large number of features and constrained dimensions of these handheld devices made soft controls even more attractive than they had been on a desktop counterpart. Largely relying on discrete soft buttons combined with pen input, these early mobile soft machines lacked the continuous direct input gestures and more sophisticated physical metaphors that researchers had been developing on larger mobile, and stationary systems [23].
The Neonode N1 was the first commercially available mobile device to make extensive use of swipe gestures appropriate for one-handed use, including a browser that scrolled content vertically with swipes. The gestures were more limited than the continuous direct input based gestures we are all familiar with today. Swiping up scrolled content up and swiping down scrolled content down similar to the PC touchpads of the time. However, its menu lists did have a highlighted element that moved down with a downward swipe and up with an upward swipe in what was very close to a direct physical mapping. It was not until the release of the first iPhone in 2007 that a commercially available mobile device made extensive use of direct mapping based interface navigation. Pinch to zoom and direct mapped scrolling interfaces have become common ever since.

Figure 2. The IBM Simon was the world’s first smartphone [7]
Figure 3. The Neonode N1 was the first mobile to use swipe gestures [46]
Figure 4. The original Apple iPhone marked the beginning of the current mobile paradigm [28]
2.2 One-Handed Mobile Interaction

Modern mobile devices can be described as capacitive sensing direct input multitouch surfaces. In the one-handed use case, these multitouch surfaces are generally reduced to single touch because the fingers of the hand are used to grasp the device and only with great difficulty can they reach the device’s screen. A notable exception to this is the Galaxy Note Edge [58], which has a non-planar screen wrapping around the edge of the device, allowing touch input similar to our force input schemes. Mosovich and Hughes [29] contrast the use of one and two hands for multi-touch interactions with both direct and indirect mapping between finger positions and onscreen content. Their findings suggest that from an interaction design perspective, one-handed and bimanual operations can be used interchangeably with little impact on usability. Mosovich and Hughes also found that in the single-handed case, gain functions could be applied to exaggerate the motion of the finger, and while the direct mapping of the finger positions to on-screen objects is lost, it does not appear to interfere with interaction and can indeed improve task performance.

Karlson et al. surveyed 228 mobile users to capture detailed information about mobile usage patterns [20]. Of the 18 common mobile activities defined, 9 were performed more often with one hand, 6 more often with two hands, and 3 with either. Overall, 45% of users recorded they used one hand for most interactions as compared to only 19% for two-handed use. Task type often dictated handedness: 14% of users used one hand for simple tasks and 63% reported they used two hands when it was the only way to complete a task.

User preference results, however, suggested that users overwhelmingly wanted to use one hand exclusively. Overall, 66% stated they would prefer primarily one-handed interaction, 9% preferred two-handed, and 23% did not have a preference. Also, many of the users who preferred one-handed interaction only adopted two-handed usage patterns if the application absolutely required it. These findings indicate that one-handed mobile operation is common, is preferred, and should be appropriately supported.
In the mid 1990s, Want et al. introduced the experimental Xerox PARC Tab. At the time the project was undertaken as an exploration of ubiquitous computing, and although limited by the technology of the time, the designs address many central mobile design considerations that are still of interest today. The device was designed so that it could be used in one hand with the thumb interacting with a touch screen similar to today’s touchscreen smartphones, and the index, middle and ring fingers resting on discrete switch style buttons fixed to the opposing side of the device. The vertical symmetry of the device allowed it to be rotated by 180 degrees for use with either hand. The device was designed so that more than one of the buttons may be pressed together in a corded sequence [51].

In their discussion of the Tab’s user interface, the authors mention a particular interaction scheme that they converged upon over the course of developing many software applications for the device. For single-handed use, an interaction is initiated by pressing the middle button with the middle finger of the grasping hand. A menu list with one highlighted item is displayed on screen. The index fingers and ring fingers buttons scroll the selected item and a second press of the middle finger or a screen touch confirms selection of the menu item. This implementation gives the user a choice of intuitive options for navigating the system where she can switch between button and touch input at any point during the interaction. The authors claim that intelligent menu choices based on context can often reduce the execution of common commands to two presses of the middle finger.
In recent years, Sony Mobile has considered single-handed usage scenarios in both the hardware and software design of its smartphones. A single-handed zoom interaction developed by the company allows users to press and hold for a moment, switching the device into zoom mode. Moving the thumb up and down along the screen, similar to a vertical swipe gesture, zooms in and out. While this does allow for zoom with a single hand, it is a relatively complex gesture and a time delay must be added to switch between panning and zooming modes, slowing down the interaction.

The devices in Sony’s Xperia line of smartphones all have hardware buttons along the right edge of the device. Typically, the home/power buttons, a two sided volume button and a camera button have been made accessible to the grasping fingers when the phone is held in the user’s left hand. Sony has taken the concept a step further in it’s mapping application where the volume button has been appropriated to control zoom level, giving the single handed user an extra degree of control [1].
2.3 Grasp-based Interaction

Taylor et al. created Graspables [45], a hardware and software platform that detects how a user is holding a device and attempts to infer a user’s intention based on this grasp. Orientation and hand position are implicitly sensed and the most relevant interface representation is displayed. For example, when the user holds the mobile like a camera, the photo capture software is displayed. Our studies contrast this by focusing on the use of individual finger input instead of grasp-based sensing.

Similarly, Song et al.’s MTPen prototype wraps a capacitance multi-touch sensor around a pen to detect gripping gestures during drawing tasks [42]. Unlike our prototypes, the grip is used implicitly to initiate different drawing modalities and an interaction vocabulary that varies by grasp.
2.4 Augmented Mobile Interfaces (sensor fusion)

Rekimoto’s use of tilt [37] in handheld interaction is an early example of augmenting devices with spatial sensors. With one hand, a user can express inclination in three degrees of freedom. This allowed users to use tilt input to navigate menus, browse large maps, and select targets in pie menus. Informal evaluation showed that users could use tilt input very accurately if appropriate visual feedback was available.

In similar work, Harrison et al. explored more general embodied mobile interactions [15]. They designed three handheld prototypes that mimicked real world metaphors by sensing tilt, corner touches (to simulate page turns), and handedness. With these prototypes, users were able to scroll lists using tilt, turn virtual pages like a book, and automatically pad annotation margins depending on which hand they used. Although no evaluation of these interactions was discussed, feedback from user explorations suggested users were able to easily transfer existing knowledge of real world manipulations to an augmented handheld device.

To improve text entry performance on a numeric keypad, Wigdor et al. [54] combined button presses with input from a tilt sensor. Each numeric button on their device was associated with up to four alphanumeric characters. A user selects an individual character by tilting the device to the left, right, away from themselves, or toward themselves as they press a numeric button. In their user study, the authors showed that this scheme significantly outperforms the conventional Multitap technique.

Hinckley et al. [16] showed how multiple sensors (e.g. proximity, touch, tilt) could infer a user’s context. In many ways, this work outlines common interactions available in modern mobile devices: using orientation to rotate between portrait and landscape, using proximity to modulate screen power, and recognizing when the device is shaken. Hinckley et al. also show how sensor fusion enables rich embodied gestures. In the Voice Memo application, multiple sensors determine when the user is holding a mobile like a stage microphone. Tilt sensors, in combination
with knowledge of the user’s grasp, trigger a recording when the mobile is tilted toward the user’s face. Informal user studies suggested completing this embodied interaction required less cognitive and visual effort than on-screen controls.

Although Hinckley et al.’s sensors enable rich interaction, they do not register input on the mobile’s back surface. Baudisch et al. [3] explored how pointing can be supported on tiny screens using the back of the device. Small screens, like watch-sized displays, are often difficult to operate as the pointing finger occludes screen content. Evaluations showed that front target selection failed for screens with a diagonal length smaller than 1 inch. However, rear-surface interaction performed well, independently of display size. Baudisch argues how this generalizes to other types of target selection techniques, such as back-of-device interactions, that rely on two-handed interactions [3]. The non-dominant hand stabilizes the devices using a minimal grasp and the dominant hand points on the back-side. This is difficult to support in one-handed usage scenarios.

In most of the previous work, interaction rarely takes place beyond the mobile’s surface. Although Hinckley et al. [42] measure proximity to an object, they do so to make implicit assumptions about the mobile’s context of use, a result alternatively achieved by Strachan et al.’s use of muscle tremor to determine grasp [44]. Seipp et al. have used a combination of microphone and accelerometer input to identify ‘pats’ on the back of the device and differentiate between finger input [40]. Using this scheme, they have created a system that augments thumb interaction in various tasks such text selection, reaching targets distant from the thumb, multiple file selection and zooming.

Spelmezan et al. have attached a combination pressure and proximity sensor to the edge of a mobile device, allowing it to detect a series of gestures made by the thumb [43]. Their method extends the capabilities of the thumb for single-handed interaction and enables eyes free use of the device. The thumb can only be used to operate their gesture recognizer or interact with the
touchscreen at a single time so the work presents a useful alternative to touch screen input rather than an augmentation.

Butler et al. [6], however, embed infrared proximity sensors along both edges of a mobile. Detecting presence and position of fingers on both sides, they supported common mobile gestures like pointing, panning, and zooming. They also explored multi-modal gestures that leverage the nondominant hand: in an annotation task, the user pans right and left with the nondominant hand while marking with the dominant. Although this approach is conceptually similar to our emphasis on nondominant interaction, it differs in two ways: (1) the mobile phone must be placed on a flat surface to work, is less generalizable and (2) the input device does not distinguish between unique fingers.

More recently, Saponas et al.’s [38] PocketTouch enabled interactions through fabric layers. Using a flexible capacitive multi-touch sensor, finger strokes are detected through a textile, affording more graceful management of notifications when a mobile is pocketed.

2.5 Deformable Mobile Interactions

Although augmented and embodied interaction has been actively explored, mobile deformation is increasingly receiving attention [20,37,45]. Gummi [39] is a design concept that envisions deformable interaction techniques contained in a thin, bendable, credit card sized display. Although rigid, Gummi explores these bendable interactions using a physical prototype that mimics the feel of a flexible display.

PaperPhone [24] uses a flexible electrophoretic ink display to afford a gesture vocabulary centered on the deformation of a mobile’s shape. A user scrolls through a list of contacts, for example, by bending a corner of the display. Unlike our study's focus on one-handed grasp, both Gummi and PaperPhone require either a symmetric or asymmetric bimanual grasp to operate their respective prototypes. Although our style of auxiliary input might be useful for bending the
perimeter of a deformable display, understanding the design implications for rigid versus flexible interaction requires further research.

Nguyen et al. [32] introduced BendID, a soft, deformable device made from layers of conductive foam, plastic and fabric. Electrodes placed throughout the central plane of the device measure conductivity of the foam in each sensor's local area. The system is roughly the size of a typical smartphone, but does not have a display for output. A support vector machine was used to recognize location, amplitude and direction of deformations applied to the device.

2.6 Isometric Force Based Interaction

Ramos et al.’s [35] investigated the use of continuously sensing pressure input to operate multi-state interface widgets on a tablet device. Although they present a taxonomy of the design of pressure sensing widgets, they focused only on the use of a stylus for fine-grained input manipulations.

In an exploration of smaller mobile form factors, Wobbrock et al. [55] show the benefit of isometric pressure input; using two isometric joysticks, one on the front and one on the back of a device, they designed a gestural text entry method that achieves input comparable in terms of words-per-minute to Multitap and T9. In a similar design space, Shi et al. [41] examined discretization functions that map continuous pressure levels to discrete input events. Using a pressure sensor placed along the side of a mouse, they found fisheye-based mappings could discretize continuous pressure input while increasing accuracy without compromising speed. This approach, however, assumes a fine-grained sensor manipulation by the thumb. The coarser and more limited movement of the auxiliary input of the fingers may not be able to operate a fisheye-based mapping with the same performance.

Blaskó et al. [4] leveraged fine-grained manipulations of four pressure-sensitive strips on a small pad to simulate a number of common interaction metaphors. Although effective for direct
manipulation, it is uncertain whether the nondominant hand can achieve the same effectiveness while grasping a mobile device. As well, Watanbe et al.’s interface [52], although effective at using pressure input to simulate a book-like interaction metaphor, does not clearly translate to a one-handed scenario.

2.7 Hand Anatomy and Function

The human hand is an incredibly complex machine and has been the subject of rigorous scientific study since the late 19th century. Even a basic summary of what is known about the hand is well beyond the scope of this paper. What follows is a minimal summary of the structure and function of the hand as it relates to holding a mobile device. For a technically inclined reader wishing to know more about the hand from an engineering perspective, [19] compiled by Jones and Lederman is an excellent resource, and was heavily leveraged in the following overview. A good reference for the anatomy of the hand is the University of Washington radiology department’s muscle atlas [59]. Also, a great deal of understanding can be gained from observing a professional dissection. The BBC produced [60], which assumes no previous knowledge of the subject and the dissection is performed on a well preserved specimen. For a more in-depth view, several medical schools have made videos of their dissections available for free online.

2.7.1 Anatomy

2.7.1.1 Forearm and Extrinsic Muscles

In order to understand the mechanics of the hand, we must start with the forearm in which the major muscle groups that drive the hand and fingers are contained. The forearm contains two major bones running in parallels from the elbow to the wrist, the ulna on the little finger side, and
the thicker radius on the thumb side. The extrinsic muscles of the hand originate from these bones and the humorous of the upper arm, with their main muscle bellies contained between them.

The inside of the forearm contains the important flexor digitorum profundus and flexor digitorum superficialis muscles. These muscles connect, respectively, to the medial and distal phalanges of all four fingers via separate tendons. They provide the force to bend the fingers around and apply pressure to an object during a power grip, e.g., when holding a hammer. The thumb has its own extrinsic flexor, the flexor pollicis longus.

The extensor digitorum muscle can be found on the other (dorsal) side of the forearm. It extends all four fingers via tendons running through the wrist and back of the hand. The index and little fingers can be extended independently by the extensor indicis and extensor digiti minimi muscles which run alongside the extensor digitorum. The thumb is extended using two muscles, the extensor pollicis longus and extensor pollicis brevis. Typically, there are no independent extensor muscles for the middle and ring fingers, however about one in ten individuals do have an independent middle finger extensor. Independent ring extensors do occur in some individuals but are even less common.

2.7.1.2 The Wrist

The forearm joins the hand through the first row of four carpal bones. A second row of four carpal bones articulates along with the first row, the radius, and the ulna during flexion and extension of the wrist. Toward the other (distal) end, the carpal bones form saddle joints with the five metacarpal bones of the hand. The carpometacarpal joints of the 4 fingers have relatively limited independence and range of motion. However, the joint between the trapezium carpus and the first metacarpal provides approximately 90% of the thumb’s range of motion. As well as enabling abduction/adduction and flexion/extension the joint also allows 45 degrees of rotation around its longitudinal axis [60].
2.7.1.3 The Hand

The five metacarpal bones form the inner framework in the palm of the hand. The first metacarpal at the base of the thumb is covered with the three largest muscles of the thumb’s intrinsic muscle group forming a bulge known as the thenar eminence. The abductor pollicis brevis moves the thumb away from the palm. The flexor pollicis brevis flexes the thumb, curling it inwards towards the palm. The opponens pollicis brevis brings the thumb toward the fingers, giving us our opposable thumbs. Not considered part of the thenar eminence, but running deep behind it, the adductor pollicis connects the proximal phalanx of the thumb to the second and third metacarpals. When flexed, the adductor pollicis brings the thumb toward the plane of the palm and rotates it inward.

On the ulnar side of the anterior surface of the hand, another mound of muscles known as the hypothenar eminence provide motion to the little finger. The flexor digiti minimi, abductor digiti minimi, and opponents digiti minimi are all analogues to the thumb’s muscles found in the thenar eminence. This relatively large group of intrinsic muscles gives the little finger the greatest range of motion and highest independence of any of the digit other than the thumb.

2.7.1.4 The Digits

Each of the four fingers has a proximal, medial and distal phalanx moving outward from the metacarpal bone. The thumb has only proximal and distal phalanges. The metacarpal-interphalangeal (MP) joint at the base of each finger is capable of extension and flexion as well as a degree of adduction and abduction. The index finger has the greatest range of adduction/abduction other than the thumb, normally around 30 degrees, followed by the little finger, the middle and the ring. Each finger is capable of approximately 260 degrees of flexion/extension. The proximal interphalangeal (PIP) joint has the greatest range of flexion/extension at
110 degrees, followed by the MP joint at 85 degrees and the distal interphalangial (DIP) joint at 65 degrees [2].

As noted above, most of the force behind finger flexion and extension comes from the extrinsic muscles in the forearm, however, these forces are modified by smaller intrinsic muscles to produce precise and independent finger control. Four lumbrical muscles, originating from the tendon of the extrinsic flexor digitorum, run along side the metacarpals of the fingers. Each lumbrical attaches at the radial side of proximal phalanx of its digit and provides independent flexion at the MP joint and extension at the interphalangial (IP) joints. Also running along the metacarpal bones are the three palmer and four dorsal interossus muscles. The palmer interossei originate from the metacarpal bones and insert into the (extensor hoods of the) index middle and ring fingers. When contracted, the interossei serve to adduct the fingers as well as assisting the lumbricals in flexing the MP joint and extending the IP joints. The four dorsal interossei originate from the metacarpals as well as the corresponding extensor digitorum tendon. One interossus inserts into each of the index, middle and ring fingers to abduct the fingers when flexed. Taut ligaments severely inhibit the adduction/abduction of the fingers when the MP joint is flexed.

2.7.2 Hand Function

2.7.2.1 Internal and Manipulation Forces
Several researchers have found it useful to distinguish between manipulation force, which results in the movement of an object, and internal force, a set of contact forces, which can be applied to an object without disturbing its equilibrium [27,30]. Augmenting touch interactions with pressure from the grasping fingers on a mobile device requires both types of force, a manipulation force, that moves the device relative the pointing thumb, and internal force that is used for communication, ideally without disturbing the position of the thumb relative to the device.
Gao et al. [11] described the construction of a device consisting of five six-component force/torque transducers arranged on an aluminum handle such that they become the points of contact between the digits of one hand and the handle when it is grasped. In addition to the transducers, a triaxial accelerometer was mounted to the handle. This system allowed the authors to measure both the internal and manipulation forces as experiment participants moved the device with their hands.

While participants where able to maintain a constant internal force on the device as they moved it around, they would vary their grip force to reduce exertion when they were allowed to move naturally. The authors suggest that this is evidence that the central nervous system will take on more “computationally expensive” tasks (dynamically modifying the grip force) rather than produce excessive force. They also observed that coordination between the thumb and fingers changes between manipulations that are tangent to grip force, where the thumb and finger force change symmetrically, and manipulations that are normal to the grip force. The thumb and fingers respond asymmetrically, when pressure increases on the thumb it decreases on the fingers and vice versa.

2.7.2.2 Finger independence of movement

It is commonly known that there are limits to the degree we can move one finger while maintaining the position of our other fingers. If you hold your hand up with fingers extended and attempt to bend only the little finger, likely your ring finger will follow. This interdependence is known as enslavement. There are both biomechanical and neurological theories about its cause, but thorough anatomical studies are limited to monkeys who have much more limited finger independence (a higher degree of enslavement) than humans. As a result, the mechanism behind enslavement remains somewhat mysterious [56].

In [14], the researchers placed a motion tracking glove on study participants and asked them to move only one finger at a time, first with self paced-motion and then timed to a metronome at
3 Hz. When self-paced, participants flexed and extended their fingers at an average rate of 2Hz.
The study found that subjects consistently moved other fingers when asked to move only one.
The thumb, index and little fingers were able to move more independently than the middle and ring fingers and similarly, were more stationary while the other fingers moved. Additionally, the self-paced movement frequencies of the middle and ring fingers were lower on average when compared to the other digits. Fingers of the dominant hand were not more independent than those of the nondominant hand.

In [10], skilled typists had their finger motions captured on high speed video. It was observed that adjacent fingers moved together more than others, for example, the middle and ring fingers moved together more than the index and little. A decrease in synergistic motion between fingers was observed when one of the fingers was used to press a key. The authors suggest common extensor muscles and descending control of musculature as possible explanations for their observations.

2.7.2.3 Movement Speed
Many studies have investigated the speed at which skilled practitioners can move their fingers during specialized tasks. In [26], MacKenzie et al. provide a survey of studies ranging from typing speed to telegraph operation, piano playing and just tapping hands.
The authors sourced the minimum and median inter-response intervals from a number of studies to make the chart reproduced in (Figure 7).

Figure 7. Comparison of inter-response interval for finger motion for a range of activities [26]
3.1 Unifone Design Process

We took an iterative design approach to determine whether the typical finger placement could be leveraged when a person operates a touchscreen mobile device with one hand. Our hardware prototypes were built around Apple iPhone and iPod Touch devices as these devices had the most responsive touchscreens available at the time.

User feedback from early design iterations was in line with Karlson et. al’s [20] observation that most users adopt (and prefer) one-handed strategies when possible. This behavior fits with Guiard’s Kinematic Chain (KC) theory [13]. Guiard saw the hands as links in a serially assembled kinematic chain, with the nondominant hand as an anchor and the dominant hand as the terminal link. The nondominant hand executes coarser manipulation and provides the context for the action of the dominant hand. The sequence of motion during bimanual manipulation is nondominant followed by dominant. For mobile interaction, this suggests the following observations: (1) users will reach for the phone with their nondominant hand and (2) a task will terminate at the nondominant hand if it can be achieved with a coarse level of control (in this case, the thumb). Karlson’s finding, that users often employ two-handed strategies when a corresponding one-handed solution does not exist, supports these observations.

In early design iterations, we explored a variety of gestures that could be controlled with coarse manipulations. In one test, we created a keyboard application that used the velocity of a squeeze gesture to trigger a space, period, or uppercase letter. The top two corners were used to move the cursor forward or backward. Although users could, after training, type with little error, most
complained that this interaction felt unnatural. We suspect that users lacked an appropriate metaphor to describe deformable interaction and had to recall the gestures.

Motivated by this, we moved away from modal gestures. Instead of overloading a gesture, like the squeeze in the typing task mentioned, we treated force input to activate quasimodes [25]. A quasimode is a mode change that is maintained by some persistent action of the user, like capitalizing a letter using the Shift Key of a keyboard. This form of input is sometimes referred to as chording, however, we reserve chording to describe simultaneous input from multiple fingers and use a quasimode to describe input from the fingers while the thumb performs gestures and targeting tasks.

In a one handed grasp, the thumb is naturally positioned to be the most effective pointing digit. It has the highest degrees of freedom and, compared to the fingers, it can fluidly select screen elements. The other fingers have predictable positions: the index finger often supports the back of the device near the top corner opposite the thumb, and the remaining fingers along its side. Although poorly suited as primary pointing digits, these fingers can express a coarse level of input that augment the thumb’s action. By restricting the other finger’s movement, the nondominant thumb provides the context of action during one-handed mobile interaction.

The first iteration of the prototype that we subjected to significant user testing consisted of three components: an iPod Touch, a deformable silicon skin, and three pressure sensors (see Figure 9). The three sensors were mounted (a) along the side, (b) on the top left corner, and (c) behind the top right corner. These pressure sensors were thin and flexible, making them easy to bend around curved surfaces.
Figure 9. Sensor placement for the first iteration of the Unifone prototype

An off-the-shelf silicon skin enclosed the sensors as they measured the force generated by isometric gestures. Initially, we explored many different silicon skins. Using liquid silicon-rubber, we poured skins of varying widths. Ultimately, we did not use these skins, as it was difficult to create an appropriate consistency in the silicone. Custom skins would shift while in use. This was limiting ergonomically and it reduced the quality of sensor data. A commercially available skin produced the most accurate pressure values and could be operated comfortably.
Figure 10. An early iteration of the Unifone prototype: a silicon skin wraps around pressures sensor attached to an iPod touch. A Phidgets Interface Kit, attached to a Desktop computer, acquires the pressure data.

Using this early prototype, we implemented a set of interaction techniques motivated by those commonly found in mobile interaction. We designed four interaction techniques that augment the thumb’s action:

1. *Scroll:* The scrolling action is initiated by squeezing the device to scroll, leaving the thumb free for interaction. Applying greater amounts of pressure increases speed. Pressure on the top corner was used to scroll up and pressure lower on the side of the device was used to scroll down.

2. *Context Menu:* Squeezing is well suited for quasimodal gestures. Depending on what item the thumb has selected, squeezing the phone transitions the interface to a context menu. For example, if a piece of text is selected while composing an email, squeezing transitions to a screen that contains font, bold, italics, and other settings. Context menus are often activated on personal computers using a right-click of the mouse. Typical
touchscreens lack an analogue to the right click, making this form of interaction difficult to perform, either relying on a long time delay to activate the secondary action of a ‘tap’, or, requiring screen real estate consuming controls.

3. **Browse**: Navigating content, whether in a web browser or photo application, is a common activity on mobile phones. For example, the iPhone uses Cover Flow [61] to provide an overview of pages. Since this is a quasimode in our prototype, the user squeezes the phone to trigger this overview.

4. **Tabbing**: With limited display size, mobile interfaces often distribute screens across tabbed items. In a music player, for example, the user can access different views of their music library by pressing tabs for playlists, artists, albums, and so on. We treated each tab as a quasimode as well. To move through individual tabs with auxiliary fingers input, the user squeezes serially until they reach the desired tab.

In early user experiments it became clear that the thumb push and index push gestures hindered time performance. Rather than embarking on an exploration of why these gestures underperformed, we simply refocused our efforts on different applications of the squeeze gesture, which showed more promise. However, the prototype still did not provide a time advantage across our selected tasks. Three out of four tasks had their average completion time increase when using auxiliary finger input. We did learn that under certain conditions, adding input from the grasping fingers could improve performance over the thumb acting alone. We made several observations that explain when the squeeze gesture can improve performance and that informed our later prototype designs.

In the scrolling task, scrolling was controlled through a squeeze gesture. The thumb was left to handle selection of the list item. Initially, we suspected auxiliary input would improve task time, as squeezing requires less movement of the thumb. However, after the thumb has initiated the
scrolling motion it is very well placed to stop the scroll and tap the target. The indirection caused by auxiliary finger input, where the fingers are responsible for part of the task and the thumb another, appears to determine performance.

Tabbing, like scrolling, adds a level of indirection. The tabbing task using auxiliary finger input required a sequence of squeezes to select one of 5 different tabs, followed by a tap from the thumb to activate a soft button contained within the tab. On average, the time required for the sequence of squeezes took more time than for the thumb to select one of 5 controls and then reposition the thumb to a selection button.

We did, however, see strong evidence that at least under some circumstances force sensing from the grasping fingers could improve performance in terms of task completion time. Text formatting was, on average, 15% faster when auxiliary finger input was used. In this task most of the screen was covered with selectable text. Performing the task using only the thumb required a soft button press to reveal additional controls. This button had to be placed at an extremity of the phone (in this case the bottom) to avoid interfering with the layout of the text. As well as taking up screen real estate that could otherwise be used for text, it required a shift of attention away from the document. Leveraging the auxiliary fingers, by contrast, required only a simple squeeze. Participants could focus on the document until they squeezed and it was replaced by formatting controls. When they released their squeeze, both their gaze and thumb were left near the center of the document.

Based on our observations from this design iteration, we decided not to implement interactions that require pressure input from the thumb in future prototypes. We found that using the thumb for pressure input limited its use for precision pointing. See Chapter 5 for further discussion of this topic.

For our next prototype, which we called Unifone, we focused on interactions for augmenting the actions of the thumb and therefore only equipped the prototype with sensors on one side.
Although we imagined the sensors directly integrated into the industrial design of the phone, we prototyped it using a force-distributed pressure sensing metal attachment made of two separated aluminum planks. These planks housed two pressure sensors at each end that, when compressed, allow three areas—top, middle, and bottom—to sense input. This transitions a precisely located button into a target zone, a shift that enables coarse input controls that (1) allow ergonomic use with each of the grasping fingers and (2) afford imprecise targeting within each zone.

### 3.1.1 Unifone Auxiliary Touch Gestures

To support course auxiliary finger gestures, we focused on three key gestures.

1. *Top Squeeze:* This corner push is focused 4 cm around the top right corner of the device and performed by the index finger (see Figure 11a).

2. *Middle Squeeze:* The enclosure of the user’s grasp is used to press the mobile’s edge inward along its vertical axis. This gesture is easiest to complete when the mobile is placed squarely in the user’s hand. Squeezes 3 cm around the middle axis are supported (see Figure 11b).

3. *Bottom Squeeze:* A bottom corner squeeze is performed by the ring or little fingers, or both simultaneously at an area of approximately 4 cm around the bottom corner. (see Figure 11c).

Each of these gestures has the natural position of a user’s grasp in mind. They can also be chorded: it is theoretically possible that a user might squeeze the middle and top corner at the same time. However, for the sake of simplicity, chorded gestures were not used in our prototype interactions.
A critical challenge to be met when designing an interactive device that senses grasp is to differentiate between a gentle supporting grasp and an intentional squeeze. For the experimental prototypes, it is necessary for the user to be able to hold the device comfortably and provide input without requiring a lengthy time to reposition the fingers. This ruled out the possibility of using binary capacitive or resistive sensors that can only indicate the location of a touch. Instead, we had to measure the relative force between a gentle supporting grasp and an intentional squeeze and decide on a force threshold to differentiate between the two.

### 3.3 Force Sensors

There are numerous force sensor designs produced commercially, by researchers and by hobbyists. Arguably any material property that changes with pressure can be used for sensing. Changes in optical properties such opacity and polarization, in magnetic transduction, and in
piezoelectric properties have all been used to measure force in existing sensor designs [48]. Sensors made with materials that change in their electrical resistance as pressure is applied are commonly used in human interfaces.

These force sensors consist of conductive contacts that are physically separated with a conductive, deformable polymer composite material, leaving an air-gap and electrically insulating the contacts. As pressure is applied to the sensor, the polymer compresses and is forced against the contacts, lowering the electrical resistance between them [62].

In many designs, the metal contacts are interdigitated across a membrane with the deformable conductive polymer making contact across them. An alternative design [63], popular among hobbyists, sandwiches the conductive polymer between the two contacts. Pressure moves the contacts closer together as the polymer deforms between them.

Many different variants on the form and composition of the conductive polymer are employed for specific applications. Polymer films, foams, and adhesives are all used to as non-conductive mediums in different designs. Varying the ratio of conductive particles to the polymer medium controls the resistance range of a sensor. The size, shape, and material used for the particles can all change how the electrical properties of the material will vary as a function of force.

In the early design iterations, we experimented with carbon doped polymer foam to sense force changes along the side of the device. Pieces of foam approximately one cm in thickness where sandwiched between two strips of conductive copper coated fabric. Copper wires were attached to the fabric with an electrically conductive adhesive. The sandwiched materials were then coated with silicon rubber as part of a pressure sensitive skin. The resulting sensors had a high degree of deformability with a consistent drop in resistance when compressed. Unfortunately the sensors degraded too quickly to be useful in a user study. They became permanently compressed and lost much of their variability in resistance. The conductive polymer foam was scavenged from electronics packaging and was not designed for elasticity. The large degree of deformability
relative to force makes foamed conductive materials well suited for use when sensing over a large range of travel. We had the most success with prototypes that contained commercial IESF-R-5L pressure sensor made by CUI [64]. These sensors have an air gap separating a pellet of conductive foam from interdigitated metal contacts. When force is applied the conductive layer shorts the copper traces with a proportional resistance.

3.4 Physical Connection to Device

As we developed our prototypes we investigated several different combinations of sensor designs as well as methods for attaching them to a mobile device. In early iterations, we considered designs that had a great deal of deformability. One-centimeter thick polyethylene foam based sensors were mounted inside of a custom mold. Liquid silicone rubber was poured into the mold and allowed to set. The result of the process was a silicon skin that fit around an iPod touch similar to commercially available cases.

While a thick and very flexible skin with custom made carbon impregnated foam provided the most responsive and novel interactions, but it was not durable enough to give consistent results over the course of a user study. As the design progressed, we decided to abandon deformation in order to achieve reliable interactivity. Commercial force sensors were attached directly to the iPod touch and a thin silicone skin was wrapped around the entire unit. This created a nearly isomorphic interaction surface with a consistent response.

In later prototypes, we moved away from the silicone skin to focus on sensing capabilities. The sensor module of the Unifone prototype was attached directly to an iPhone with an adhesive. We believe that our work on interaction would be best employed when integrated directly into a device rather than an add-on case and will generalize to deformable surfaces when materials allow.
3.5 Sensor Processing and Device Communication

Unifone used force sensors connected to a Phidgets interface kit that passed the values on to a laptop over USB. A simple program running on the laptop registered any changes in sensor values and sent differences to the iPod touch over a wireless network connection using TCP/IP communication. This provided a very flexible prototyping infrastructure that allowed different sensor arrangements to be tested without spending a long time on hardware development.

3.6 iOS Experiment Software Architecture

For each prototype, the bulk of the software system was implemented as an iOS app. The app was divided into a communication and change notification, sensor calibration, experiment management, and individual user task components. All components were written in Objective-C.

When the app launched, an interface was presented requesting an IP address for sensor communication. Once connected, the communication code ran in its own thread. The communication component provided an interface for other Objective-C classes to register as listeners and receive messages when sensor values were updated.

After being connected to the networked sensors, the app could enter an optional calibration mode. In this mode, the maximum and minimum sensor values were recorded while a user applied pressure to and then released, each of the force sensors. These values were then used to linearly scale all future sensor values before they are processed by the user interface code. This allowed us to write experiment code that assumed a consistent sensor response as sensors degraded or were replaced by a different type of sensor.

The experiment manager component provided all the sequencing and timing of an experimental session. It read a text file that defined the sequencing of experiment tasks as well as target values for task completion. It then presented the experiment participant with tasks in the defined
sequence and recorded the participant’s task completion time in a text file referenced by a unique subject identifier.

The interactive user tasks were implemented using Apple's Cocoa Touch framework. Each individual task provided its own instruction text, which the experiment manager presented to the user before they begin. Each task received a target value from the experiment manager when it began and messaged the manager upon completion so that the task time could be recorded.

The Unifone prototype used two sensors mounted at each corner and suspended between two thinly spaced aluminum planks. These planks extend the pressure sensors beyond a focused point and, using only two sensors, enabled a middle squeeze gesture to be detected. Without this force-distributed design, it is difficult to achieve comfort across a range of hand sizes.

Figure 12. The Unifone prototype: two aluminum planks house two pressure sensors attached to an iPhone 4S.
3.7 Unifone Interactions

We designed four interaction techniques that augment the thumb’s action:

1. **Scroll**: In thumb-only scrolling, the thumb is used to scroll a viewport and select targets.

   In Unifone, the middle squeeze gesture is used to augment the thumb by dampening and, if squeezed firmly enough, halting the scrolling. Physics-based scrolling interaction, like that of the iPhone, affords a larger range of motion in the thumb and, ultimately, a wider scroll extent; using a middle squeeze to halt scrolling means that the thumb does not have to touch the device’s display as frequently.

2. **Navigate**: When using a map application in a one-handed use scenario, navigating zoom levels is typically achieved by overloading taps (i.e., double tap zooms in). Using Unifone, a user zooms in using a bottom squeeze and zooms out using a top squeeze. This allows the thumb to focus exclusively on panning and targeting.
3. **Context Menu**: A middle squeeze is well suited for contextual quasimodal gestures. Depending on what item the thumb has selected, squeezing the device transitions the interface to a contextual menu. For example, if a piece of text is selected while composing an email, a middle squeeze could curl up the bottom right corner of the interface screen to reveal font settings (e.g., bold, italics, and underline). This is achieved with thumb-only input by using a dedicated interface button.

4. **Switching**: With limited display size, mobile interfaces often distribute applications across multiple screens, running in the background until switched to by the user. To move through running applications with Unifone the user performs a middle squeeze to reveal the running applications and then swipes to the necessary application, before releasing the squeeze to indicate a selection.

### 3.8 Unifone Evaluation

The goal of our study was to measure Unifone’s impact on task completion in frequently used one-handed thumb interactions. Users completed tasks with Unifone gestures and the thumb (thumb+Unifone) or only the thumb. We measured the task completion time of five tasks that correspond to Unifone’s use as a direct, supportive, continuous, and discrete input device:

1. **Direct Scroll (Direct & Continuous)**: The user scrolls through a list of 100 ordered numbers in a single column. At the top of the screen, a text label specifies the target number. To shorten search times, the target number is highlighted red in both conditions. Upon finding the number, the user touches it with the thumb. In the thumb condition, the thumb scrolls the list up or down. In the thumb+Unifone condition, the top squeeze gesture scrolls up and the bottom squeeze gesture scrolls down. In both conditions,
acceleration is matched: the maximum acceleration using the thumb and the thumb+Unifone’s pressure base input are the same.

2. *Halted Scroll (Supportive & Continuous)*: The previous scrolling task actually counters Unifone’s design goal of supporting the thumb’s primary behavior via coarse auxiliary finger input. For the purpose of comparing a direct and supportive interaction, Unifone’s middle squeeze is used to slow down and halt scrolling. The thumb condition is the same as the previous task. The thumb+Unifone condition is identical, yet the middle squeeze slows and, once fully pressed, halts altogether. In both tasks, the target number is also highlighted in red.

3. *Map Navigation (Supportive & Discrete)*: In this task, users target on a two-dimensional map application by locating, translating and zooming until selecting a target city on a map. In the thumb+Unifone condition, users zoom in one level using a bottom squeeze and zoom out one level using a top squeeze. The thumb translates the map until the city fully occupies the viewport and, by doing so, indicates a selection. In the thumb condition, the thumb also translates position. However, in this condition, single finger input is overloaded; users zoom in one level by double tapping and zoom out one level by triple tapping. We introduced the triple tap because there was no existing single-handed gesture to zoom out. The use of a soft button would require the user to first target it and move their thumb to a different screen area away from where it was positioned to scroll.

4. *Context Menu (Supportive & Discrete)*: In this task, the user selects and formats a single word in a chunk of text. A text label asks users to “Set word to bold.” To shorten search times, the target word and text on the formatting button is highlighted red in both conditions. In the thumb condition, the user touches the target word with the thumb, and then the ‘formatting’ button to reveal a context screen containing the formatting settings. The screen peels and reveals the bold, italics, and underline buttons in the bottom right
corner. The user then touches the appropriate formatting button. In the thumb+Unifone condition, the screen peel is triggered by the middle squeeze gesture. This gesture is released to confirm the formatting selection.

5. **Switching (Supportive & Discrete):** In this task, users switch between a set of running applications. In the thumb condition, the user taps the screen to see a zoomed view of running applications. The user scrolls through the applications horizontally with their thumb. When the target application is found, the user touches the screen. In the thumb+Unifone condition, the squeeze gesture triggers the zooms view. The user scrolls through the application using their thumb and releases the squeeze to select the target application.

### 3.9 Unifone Experiment Design

We used a within-subjects design, with two independent variables for each task: input type (thumb vs. thumb +unifone) and target distance. Target distance is a term that encompasses a target's configuration in each of the five tasks, e.g., for scrolling, target distance implies variable length of vertical scrolling, while for context switching (formatting), target distance implies different positions of the target word in a paragraph. There were five tasks that were repeated in five trials for each condition. The tasks were structured using a latin square and order of input type was balanced across subjects. Our dependent measure was task completion time. At the start of the study, users were familiarized with Unifone. While seated in a chair, they were presented with a screen that displayed each sensor’s feedback and were asked to perform each gesture until they were comfortable with these interactions. All users were able to learn and perform Unifone’s gestures easily. No high level details about Unifone’s application to mobile interaction were provided. Before each task, users completed a training phase. On the mobile device, users were presented with a text instruction screen that instructed them how to complete the task. Verbal
instructions were only provided if the users needed additional help understanding the text instructions. The training was repeated until subjects could complete five trials successfully. In Unifone interactions in one-handed mobile applications where the thumb performed the same target selection in all conditions, we defined an error to be a trial that exceeds a practical time limit or where the user verbally stated they did not want to finish the task. This was more inclusive to the thumb+unifone condition and measured how long it took to complete each trial; overshooting or mis-selections were not considered errors.

3.9.1 Participants

10 participants (6 male, 4 female) were recruited from a pool of friends and students.

3.10 Unifone Results

Overall, users completed all tasks within each practical time limit and did not withdraw from any task.

3.10.1 Scrolling task

We analyzed the measures collected by performing a two-way repeated measures factorial analysis of variance (ANOVA) using input type and target distance on task completion time. The three input types were: halted scroll and direct scroll for thumb+unifone design, and thumb input (i.e. touch) exclusively. For task completion time, the analysis showed that both input type (F(2, 22)=6.11, p<0.05) and target distance (F(4, 44)=15.36, p<0.001) were significant.

Pairwise post-hoc tests with Bonferroni corrected comparisons between input types reveal that touch was significantly faster than both direct scroll and halted scroll. Mean task completion times for direct scroll, halted scroll and touch input were 5.35, 6.76 and 4.57 seconds respectively (Figure 14, scrolling).
3.10.2 Formatting task

Similar to the scrolling task, we performed an ANOVA using input type and target distance on task completion time. Unlike the scrolling task, the formatting task had one thumb+unifone condition that was compared to thumb (i.e. touch) input. The analysis showed a significant main effect for input type (F(1, 9)=8.58, p<0.05).
Thumb+unifone condition was significantly faster than touch input, mean task completion times for thumb+unifone and touch input were 4.20 and 5.62 seconds respectively (Figure 14, formatting).

3.10.3 Application switching

An ANOVA using input type and target distance on task completion time. This task had one thumb+unifone condition that was compared to thumb (i.e. touch) input. We found that there was a significant effect of input type (F(1, 9)=5.89, p<0.05) and target distance (F(4, 36)=156.26, p<0.001).
Thumb+unifone condition was significantly faster than touch input; mean task completion times for thumb+unifone and touch input were 12.25 and 13.6 seconds respectively (Figure 14, app. switching).

3.10.4 Map navigation

Similar to the previous tasks, a two-way repeated measures factorial analysis of variance (ANOVA) was performed using input type and target distance on task completion time. The two input types were: thumb + unifone, and thumb input (i.e. touch) exclusively. For task completion time, the analysis showed that both input type (F(1, 9)=5.47, p<0.05) and target distance (F(4, 44)=135.00, p<0.001) were significant.
Thumb+unifone condition was significantly faster than touch input; mean task completion times for thumb+unifone and touch input were 17.32 and 19.81 seconds respectively (Figure 14, map navigation).
3.11 Discussion

The results of our Unifone study suggested that auxiliary finger input, when used to control isometric gestures, could improve the performance of thumb-only interaction. Initially, the direct scrolling that was strictly controlled through top and bottom squeeze gestures performed 28% slower than thumb-only input. However, when the fingers were adjusted to dampen or halt the thumb’s scrolling behavior, it performed only 17% slower than thumb-only scrolling. For the tasks that required displaced movement of the thumb (formatting, application switching, and map navigation), Unifone performed better than thumb-only tasks.

The formatting task was the same as we used in the earlier prototype but was performed on average 25% faster with squeeze than the thumb-only condition. Similarly, the application switching task that had performed slightly slower in the original prototype condition was 9.8%
faster when augmented by squeeze. This suggests that our modifications to the sensing hardware had improved the system’s usability. Although both tasks essentially replaced a soft button with a squeeze gesture, the formatting task required precise targeting with the thumb to select words and then buttons, whereas application switching used the thumb for swipe gestures. It therefore seems likely that the required shift in gaze to the soft button when formatting may account for the relative difference in task performance.

Map navigation, a new interaction in the Unifone study, performed 12.5% faster when augmented by squeeze. We found this result particularly encouraging since it added a degree of freedom to spatial navigation that was unavailable in single-handed scenarios on smartphones of the time. Since the publication of the Unifone study Sony Mobile has introduced a single handed zooming with a combined tap and swipe gesture. However, it still requires a time delay and limits the use of tap for activating controls on the zoomable view.
Chapter 4

The Isophone Prototype

4.1 Isophone Concept

In the previous study, we hypothesized that the benefit of allowing interaction between a mobile device and the fingers grasping it was that it brings the point of interaction to the fingers rather than making the fingers move towards the point of interaction. The use of a GUI button on a touchscreen requires a user to move their gaze to the button and then move the thumb (in the case of single-handed use) moving the user’s attention away from on-screen information. The position of a physical button or frequently used soft button on the device could be learned, but still requires travel time for a finger or thumb and still occludes the screen.

In the Unifone study, a user completed five successful training interactions before their task time was measured. During these training tasks, the users were able to find an optimal finger position for the interaction. However, this repositioning is undesirable for many real-world scenarios. A user may want to activate a device, quickly check a notification and then stow it. For such brief interactions, the time taken to align one's fingers with a pressure sensitive target area would eclipse any benefit gained by using the fingers for auxiliary input. In order to completely eliminate homing time, a system needs to be able to sense finger position and pressure anywhere the user happens to grasp.

The principle behind Isophone is that an initial squeeze can be used as a gesture to both switch into a new mode, and inform the system of where each of the fingers are positioned along the side of the device. The system can then assign actions to each finger detected. This allows for interactions that are both eyes-free and that do not require any repositioning of the fingers.
An ideal device would have a continuous array of high-resolution pressure sensors around its entire perimeter, or at a minimum, down the length of both sides. This would allow for use with either hand. However, for the purpose exploring interaction techniques, it was decided that a single-sided sensor array would be sufficient.

### 4.2 Hardware

The Isophone hardware consists of an array of 32 sensors running a total of 11.17 cm down the length of the case of an iPhone 5. Each sensor consists of four parallel copper traces on a flexible polyimide substrate. Two of the traces are electrically connected to a reference voltage while the two interdigitated traces are connected to ground through a 1k-ohm resistor.
Figure 16. The Isophone sensor design schematic.

A piece of Linqstat [66] carbon doped polyethylene film is fixed above the copper traces with a strip of laser cut tape, leaving a small gap between the conductive film and the copper contacts. When pressure is applied to the polyethylene film it stretches in toward the copper traces, eventually making contact and allowing current to flow between them. As more force is applied, a larger area of the conductive film's surface makes contact with the traces, further reducing the resistance across the sensor.

The 32 pressure sensors are connected to two Texas Instruments CD74HCT4067-Q1 [67] 16 channel analog multiplexors. The select lines of the multiplexer are connected to the pd2-7 pins of an atmega328-au 8-bit microprocessor. The tx, rx, avcc and ground pins are connected to the rx,tx, dtr and ground pins of a Redpark C2-TTL [65] serial cable, providing serial communication and power connections to an iPhone 5.

An ABS plastic case attaches the sensor unit to the iPhone and protects the electronic parts. A shaped piece of vinyl tape fixes the sensor to the case and provides a layer of protection for the conductive plastic film.
4.3 Software

4.3.1 Device Communication

The Atmega processor has been fused to use its internal oscillator running at 8 MHz. An Arduino bootloader has been burned onto the chip. A simple Arduino sketch polls the multiplexers and writes the byte values to the serial connection.

On the iPhone, an iOS app uses the Redpark serial SDK to read bytes from the sensor module. The serial manager component of the app fills a buffer until it receives a stop byte from the sensor module. If the buffer contains exactly 32 values, the serial component pushes the new data to
another module for processing, otherwise a transmission error is assumed, the buffer is cleared and the system begins reading new bytes.

4.4 Signal Processing and Analysis

A sensor manager module handles all processing related specifically to the 32-byte stream of data coming from the sensor unit. It interprets the data coming in and sends high-level events to the app’s main user interface controller. The manager maintains a buffer of raw data coming from the sensors as well as buffers containing the data after various stages of processing. Each buffer can be written to a file for databasing, visualization and analysis.

4.4.1.1 Thresholding

Thresholding is necessary to differentiate grasp from input. If a sensor falls above a certain threshold, it is considered pressed and a high value will be written to the threshold buffer. If the value falls below the pressed threshold, but is still higher than a second lower threshold a high value will still be written to the buffer. If the two sensors directly adjacent to a sensor are high and the sensor's previous buffer value is also high, the new threshold buffer will also be high independent of its current value. In all other cases the threshold value will be low.

4.4.1.2 Blob Finding and Finger Positions

In our system, blob finding is performed on the thresholded sensor values in four passes. In the first pass, adjacent regions of high sensor values in the threshold buffer are collected as blobs represented as the index of the first sensor and last sensor in a continuous region of high values from the threshold buffer. The next step applies the rule that each finger must take up a certain minimum amount of space. Any two adjacent blobs, where the distance between the start index of the first blob and the end index of the second blob is smaller than a constant minimum finger
width, are merged into a single blob that has the start index of the first blob and end index of the second.

The third step is an application of the rule that each finger has a certain maximum size. Each blob that has a distance from its start index to its end index greater than a constant maximum finger width is divided into two blobs, one blob with the original start index and end index of midway between the original start and end indices, and another with a start index midway between the start and end indices and the end index of the original blob.

The final step is an application of the rule that there may be no more than four fingers grasping the side of the device. If more than four blobs exist after the previous three steps, the two blobs that minimize the total width of the two blobs minus the distance between them, are merged until only four blobs remain. Final finger positions are buffered as float values representing the midpoint of each blob in screen coordinates.

4.4.1.3 Gesture Recognition and Event handling

A state machine handles gesture recognition. By default, the system is in a waiting state with all thresholded sensor values low. When a sensor value goes high, the system transitions to the pressed state.

Upon entering the pressed state, the system stores the number of fingers that are pressed as determined by the finger detection system. With each sensor update, the finger count value is increased if it is higher than the previous count. There are two ways that the system can transition from the pressed state. First, if all thresholded values go low, it will move to the released state. Alternatively, if the system remains in the pressed state for longer than a specified time threshold, it will move to the long press state. Either way, the maximum number of fingers pressed at one time while in the pressed state is stored as the finger count for the gesture.
When in the released state, the system will transition back to the waiting state after a certain time threshold unless at least one sensor value goes high (another finger press is detected) first, in which case it will transition to the double press state. Both long press and double press states can only transition back to the waiting state. Until that happens, in either state, the system will continue to update the maximum finger count for the gesture.

User interface controllers can implement a delegate protocol and register with the sensor manager to receive notifications when any new sensor data is available, when detected finger positions change, and when the system changes state.

![State diagram for Isophone input and gesture classification.](image)

**Figure 18. State diagram for Isophone input and gesture classification.**

### 4.5 Interactions

#### 4.5.1.1 Home Gesture

The home gesture is performed with a three-finger squeeze. It is a good candidate to replace the home button found on many mobile devices because it is unlikely that it would be triggered
inadvertently while device is inside a pocket. It also has the advantage of detecting where each of the three fingers is located.

4.5.1.2  Zoom Control
The second interaction is zooming. First, a two-finger squeeze is used to initiate, and then "+" and "-" buttons appear on screen located at the two finger positions. Subsequent presses at those positions cause the view to zoom in or out. A three-finger-squeeze removes the buttons and a double-three-finger-squeeze exits the zooming app and returns to the home screen.

4.6  Isophone Experiment

The high resolution of the Isophone sensor array provides an opportunity to explore new questions about the use of auxiliary finger input and its relationship to the physical support of a device. The relative ability of each individual finger to produce force, as well as possible interference related to physical support, can be measured. For our final study, we used a within-subjects design, this time divided into two independent experiments. The objective of the first experiment was to measure the effect of different finger postures on the thumb’s range of motion and, simultaneously, measure the effect of the thumb’s movement on the fingers' ability to maintain pressure on the side of the device. The second experiment of the study measures the effect of using different combinations of fingers on time to perform a basic gesture.

4.6.1  Experiment Description
At the start of each session, participants were familiarized with the Isophone prototype. They were shown a screen that graphs the pressure values from the sensor array. Holding the device in the left hand, participants were asked to apply force with each of their left hand fingers. Viewing the visual feedback helped the participants understand how to properly transmit force to the side of the device.
Each trial in the first part of the experiment lasted 15 seconds. To measure the effect using the grasping fingers on the thumb’s ability to interact with different regions of the mobile screen, participants were instructed to maintain a consistent force with a specified combination of fingers of their left hand while they attempted to 'paint' as much of the screen as possible with the thumb of the same hand. The experiment software played a chime sound if the total force on the side of the device dropped below approximately 1.5 N. The stream of pressure data was stored to be analyzed in combination with the touch screen results. Immediately before each trial, the experiment software displayed a graphical representation of a hand with brightly coloured circles highlighting which fingers should be used to engage the sensor along with written instructions. Users were asked to hold the phone with their fingers close to its vertical center.

A 10 by 18 grid was presented on screen at the start of each trial. Individual squares on the grid measured 5 mm per side. Initially drawn in white, the squares turn dark gray when touched to provide feedback. During each trial, a spatially organized set of time stamps were recorded, each representing the moment the thumb first touched a particular grid square. After each trial, the participants filled out a digital version of the NASA TLX questionnaire. One trial was completed for each possible combination of index, middle, ring, and little fingers making a total of 15 trials. The ordering of finger combinations was structured using a pseudo-random Latin square. Our dependent measures were total number of grid squares reached by the thumb and total TLX survey score.

The second experiment of the study used the same 15 finger combinations as the first, but this time the participants were asked to apply force to the side of the device in two short bursts, similar to double-clicking a mouse. Each trial in the study represents a double-click gesture with one combination of fingers. The position of the thumb was not measured in this part of the study. When the values from the sensor array are summed and plotted against time, the
double-press-gesture appears, predictably, as two similar curves. To compare the time performance of different finger combinations, we measured the time between the peak of force on the first curve and the peak of force on the second curve. This measurement can be considered a measure of the period of a signal generated by the fingers, also known as the inter-response interval.

![Force vs. Time diagram used to illustrate Isophone's Inter-response Interval](image)

**Figure 19. Force vs. Time diagram used to illustrate Isophone's Inter-response Interval**

### 4.6.2 Participants

20 participants (10 male, 10 female) between the ages of 19 and 34 were recruited from a pool of undergraduate and graduate students at Queen's University. All participants had used smart phones in the past and 17 were smart phone owners.
4.6.3 Results

4.6.3.1 Range of motion results
The means and standard deviations for the screen area reached by the thumb are shown in Table 1. A 4-way repeated measures analysis of variance showed that applying pressure with different fingers produced significant changes in the screen area. There was a statistically significant effect of force applied with the index finger on the screen area reached by the thumb $F(1,19) = 8.174$, $p = 0.010$. There was also a statistically significant effect of force applied with the ring finger $F(1,19) = 20.559$, $p = 0.000$. There was no significant interaction between different combinations of fingers.

<table>
<thead>
<tr>
<th>Finger</th>
<th>Present Mean</th>
<th>Absent Mean</th>
<th>Present SE</th>
<th>Absent SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>22.3</td>
<td>23.9</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>Middle</td>
<td>23.2</td>
<td>22.9</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>Ring</td>
<td>23.8</td>
<td>22.2</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>Little</td>
<td>22.7</td>
<td>23.3</td>
<td>0.5</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 1. Screen area reached by thumb in cm$^2$

4.6.3.3 TLX Results
The means and standard deviations for the raw TLX scores are shown in Table 2. A 4-way repeated measures analysis of variance showed that applying pressure with different fingers produced significant changes in score. There was a statistically significant effect of having force applied with the index finger on the total TLX score $F(1,19) = 5.990$, $p = 0.024$. There was also a statistically significant effect of having force applied with the ring finger on the total TLX score $F(1,19) = 5.680$, $p = 0.028$. There was a significant interaction between having force applied with the index and middle fingers on the total TLX score $F(1,19) = 4.619$, $p = 0.045$. 
### Finger Scores

<table>
<thead>
<tr>
<th>Finger</th>
<th>Present Mean</th>
<th>Absent Mean</th>
<th>Present SE</th>
<th>Absent SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>50.0</td>
<td>46.8</td>
<td>1.7</td>
<td>1.7</td>
</tr>
<tr>
<td>Middle</td>
<td>49.1</td>
<td>47.9</td>
<td>1.6</td>
<td>1.8</td>
</tr>
<tr>
<td>Ring</td>
<td>47.1</td>
<td>50.2</td>
<td>1.6</td>
<td>1.8</td>
</tr>
<tr>
<td>Little</td>
<td>49.0</td>
<td>48.0</td>
<td>1.7</td>
<td>1.8</td>
</tr>
</tbody>
</table>

*Table 2. TLX scores*

#### 4.6.3.4 Inter-response interval results

The timing portion of the study required participants to tap a soft button to begin recording sensor data, perform the gesture, and then tap the button again to stop recording. While the vast majority of trials were completed as expected, there were a small number of occasions where the participants began to record between gestures and stop the recording as the gestures were performed. In other cases, they simply did not perform the gesture correctly. Since the study was designed for a repeated measures analysis, it was determined that the safest way to handle the bad data was to assume it missing and perform list-wise deletions of the participants before analysis. Of the original 20 participants, 7 performed one or more trials incorrectly and were removed, leaving a sample of N= 13 participants for analysis.

The means and standard deviations for the inter-response intervals are shown in Table 3. A 4-way repeated measures analysis of variance showed that applying pressure with different fingers produced significant changes in interval. There was a statistically significant effect of having force applied with the ring finger on the inter-response interval $F(1,12) = 5.327, p = 0.040$. There was also a statistically significant effect of having force applied with the little finger on the inter-response interval $F(1,12) = 20.559, p = 0.000$. There was a significant interaction between having force applied with the index middle and ring fingers $F(1,12) = 5.174, p = 0.042$. 

55
### Table 3 Inter-response intervals in milliseconds

<table>
<thead>
<tr>
<th>Finger</th>
<th>Present Mean</th>
<th>Absent Mean</th>
<th>Present SE</th>
<th>Absent SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>615</td>
<td>615</td>
<td>19.7</td>
<td>22.8</td>
</tr>
<tr>
<td>Middle</td>
<td>615</td>
<td>616</td>
<td>18.9</td>
<td>23.7</td>
</tr>
<tr>
<td>Ring</td>
<td>591</td>
<td>643</td>
<td>18.5</td>
<td>23.8</td>
</tr>
<tr>
<td>Little</td>
<td>651</td>
<td>575</td>
<td>22.5</td>
<td>18.4</td>
</tr>
</tbody>
</table>

4.6.4 Discussion

4.6.4.1 Screen Area

In Isophone’s first experiment, the self-assessment of performance that users recorded in the TLX questionnaires generally agreed with the measured screen area from the experimental task. Both measures strongly indicate that performance is negatively impacted when participants apply force with their index finger, and performance is improved when they apply force with their ring finger. We can identify several possible reasons for the relatively poor performance of the index finger.

To begin, we look at the difference between the number of times a particular screen location was touched when a particular finger was applying force and the number of times the same location was touched when that finger was not applying force. This has been presented graphically for each finger in figure 20. Notice that in figure 20a, a relatively large number of touches can be seen (in red) on the top left of screen with relatively poor performance (blue) toward the bottom.

A likely explanation is that when an inward force, normal to the edge of the device, is applied above the thenar eminence, it creates torque on the device that tends to rotate the top right corner toward the thumb and the bottom of the device outward and away from the thumb. A similar effect occurs when a force is applied very low on the device. This can be seen for the little finger in figure 20d. The force will tend to rotate the bottom of the device inward until the opposite edge of the device is tangential to the thenar eminence at the point where it is intersected by the force...
vector. The latter case does not have the same negative impact on the thumb’s range of motion on the screen, first, because when the pressure is applied toward the top of the device, the area reachable by the thumb is largely above the screen. Second, when pressure is low on the device, it can be increased by pushing the opposite edge deeper into the soft muscle bellies of the thenar eminence, allowing the thumb to reach further toward the opposing side of the screen, increasing task performance.

Looking at the relative frequency per screen area for the ring finger (figure 20c) does not appear to explain why providing force with this finger would increase task performance so significantly. There is an area of significantly increased touch frequency near the midpoint of the screen on the right hand side, near where the ring finger touches the device, but it is relatively small and it appears that the larger numbers of touches are distributed more evenly across the screen. Various studies [10,14,56] have identified the ring finger as the least independent and typically cite a combination of cognitive and biomechanical factors as the reason. Anatomically, the middle and ring fingers lack the independent extensor muscles that are connected to the thumb, index and little fingers. You can observe the effect of this by extending your index and little fingers while you flex your middle and ring fingers to make a ‘devil horns’ gesture. Next try the opposite by extending the middle and ring fingers and flexing the index and little fingers. The latter gesture is much more difficult than the former because without extrinsic extensor muscles you depend on the intrinsic lumbrical and interosi muscles to fight against the flexion of the extrinsic flexor digitorum profundus and flexor superficialis muscles. The adductor pollicis, the muscle that brings the thumb toward the plane of the palm, has two heads that originate from the bases of the second and third metacarpals.
It seems plausible that the function of the interoseus and lumbrical muscles, located within the same region, would interfere with thumb adduction, limiting its range of motion and hence its ability to move around the screen. However, credible studies have shown that biomechanical factors alone cannot explain the variations of finger independence [21,56]. Zatsiorsky et al. theorize that peripheral tendon connections and multi-digit extrinsic muscles (the flexor digitorum profundus and superficialis) may be the primary sources of enslavement while plasticity of the central nervous system may lead to learned secondary sources [56].

To better understand the effect of the thumb’s movement on the fingers’ ability to maintain pressure on the side of the device, we normalized the pressure data and then calculated the mean pressure that was applied by the fingers for each location on the screen as it was touched by the thumb, across participants, for each trial. These data, for all trials, are included graphically in Appendix A. At a glance it is apparent that pressure generally increases toward the upper right-hand corner of the screen (figure 21a) and in some cases (figure 21b) toward the bottom of the screen. Pressure generally decreases toward the left vertical center.
To explain these regions of high and low force we must consider the role of the first metacarpal and the thenar eminence on which the device rests. In order to reach the opposite side of the screen, the thumb must be opposing and adducted with the first metacarpal and thenar eminence pointing inward against the device. The further the point on the screen is away from the base of the thumb, the harder the thenar eminence will be pressed into the device and the device pressed into the opposing fingers. For the thumb to make contact with a region of the screen near its base, the first metacarpal and proximal phalanx are extended with the distal phalanx flexed. Without the thenar eminence providing any opposition, the fingers are unable to deliver as much internal force without altering the position of the device.

4.6.4.2 Inter-response Intervals

Compared to studies examining the inter-response intervals (IRIs) in skilled manual tasks, the average times observed in our study were quite long. Expert typists and pianists commonly have inter-key intervals between 100 and 200 ms, and as low as 40 ms. Participants in our study tended to have IRIs in the 500-600 ms range, with the fastest participants getting to 200-300 ms. Our participants had no previous experience with the experimental task, were constrained to the use of one hand with the same fingers performing both pressure inputs, and were allowed to self-pace.
their movements, so it is not surprising that they were slower on average. Our interest lies in the relative performance of the fingers.

Once again, the presence of the ring finger increased task performance with close to a 10% reduction in IRI compared to tasks where it did not contact the device. It seems likely that difficulties in keeping the ring finger extended while the other digits flex are once again related. Interestingly, the index finger did not decrease performance. It appears that the index finger itself is quite dexterous. The same cannot be said for the little finger. While it does have the largest number of independent muscles, and the greatest range of motion of the fingers, it does not appear to be fast when moved independently.
Chapter 5

Design Guidelines

5.1 General Design Guidelines

The overarching design goal of Unifone and Isophone was to use auxiliary finger input to increase the task performance of common one-handed mobile interactions. The results of the Unifone study indicate that using these grasping fingers in conjunction with the thumb can be faster than the thumb alone. However, both the hardware and the software interactions need to be carefully designed to support this performance improvement. Based on lessons learned from our prototype iterations and from empirical results, we present the following design guidelines for one-handed mobile interactions that use auxiliary finger input:

1. Auxiliary input is supportive.

The thumb is naturally positioned as a mobile display’s pointing digit. Using the fingers to directly control interaction, on the other hand, adds a layer of indirection to this fine-grained input control. This also decouples the haptic and visual feedback that takes place when the thumb guides interaction. Therefore, the remaining fingers should act as secondary controls that extend the thumb’s behavior.

2. Auxiliary input is a discrete.

Early design iterations showed that chorded gestures, and pressure-based gestures with multiple discretization levels [25] were frustrating to users. Although pressure as an input channel is expressive, a complex gesture requires training, dexterity, and recall. The complexity of modal operations can be avoided by treating pressure as a simple quasimodal input state. If multiple
levels are used, as it is in the halted scrolling task, it should be mapped across a continuum of the same dimension.

3. Auxiliary input is coarse.

The fingers are extremely poor at providing fine-grained input. In the scrolling task, many users had a slower performance using Unifone as a continuous input device, mainly because of the level of precision it required. The simpler manipulations, like performing a middle squeeze to trigger application switching, are more practical by the very fact that they require less precise targeting. Given the limited range of motion of the auxiliary fingers, designing for coarse interaction is preferred and supporting input zones—instead of exact target locations—is critical. Although quasimodal squeezes simplify interaction, they also cause tension in the user’s grasp and limit the thumb’s movement. In the map navigation task, the top and bottom squeeze gestures guided the thumb’s position around the center of the display. In the halted scroll, users moved their thumb to the side of the display to stabilize the phone’s movement. Therefore, quasimodal gestures should be brief, should pay close attention to their relationship with the thumb, and be carefully designed to conform to the user’s mental model and expectation of physical motion as they grasp and manipulate a device. Of course, this relies on an appropriate industrial design, one that minimizes this potential tension between the thumb and fingers when performing gestures.

5.2 Hardware Design Guidelines

5.2.1 Dimensions

The first consideration is device size. The device must fit between the thenar eminence and the opposing fingertips. Ideally, small enough so that the metacarpophalangeal joint and both interphalangeal joints can be moderately flexed with the fingertips resting comfortably along the edge. If the device is too small, a user’s thumb will not have the range of motion to make
effective use of the screen. If it is too large, the user will have difficulty in the application of force with their fingertips, however, positioning the device higher or lower against the thenar eminence accommodates for a relatively large range of sizes. For the 1 percentile small woman, a comfortable range would be between 36 and 65 mm. For the 99 percentile large man, it would be between 51 and 94 mm [47].

5.2.2 Pressure ranges

Previous work has determined that human fingers are capable of delivering a great deal of force, but that force differs significantly between an individual’s fingers. As an absolute upper bound for force sensing requirements the index finger is capable of delivering 149 N of force, the middle 192 N, the ring 151 N, and the little 95 N [33]. These forces are, however, far too great for comfortable use and could in fact damage a device. As an absolute lower bound, the smallest forces that can be detected by the fingertip can be as low as 0.0005 N for men and 0.0002 N for women [53] (it should be noted that these values were determined by applying force to subjects’ fingers with fixed diameter filaments, and there is some controversy as to whether force per area might be a more useful measure [19]).

To refine the above ranges, we must consider the force required to maintain a stable grip on a device. In our studies we found that user’s rarely grasped the device with more and 1.5 N of force per finger when not asked to apply pressure for interaction. This number will vary between devices depending on their mass (our prototypes weighed between 150 and 200 grams) and the coefficient of friction between the fingers and the material used on the contacted sites of the device. The opposing grasp used on a mobile also appears to change the relative force applied by each finger from their maximum values as reported in [33]. Although precise force measurement was not a primary goal of the Isophone study, the average sensor value per finger is available in figure 22. We found that in our configuration, the index finger was able to produce the most force
on average, a finding that is in line with the results of a study by Kinoshita et al. [22], where participants moved a device weighing 400 grams in a five finger precision opposing grasp. Kinoshita et al. found also found an average maximum force of 1.5 N was produced by the index finger as stabilized their device. We found 1.5 N to be suitable as a minimum activation force in several of our interaction prototypes. It is easy to sustain, approximatly the force needed to activate a very stiff keyboard. If errors are costly or the device is to be used in an error-prone environment such as a pocket, a force between 5-10 N is still appropriate for most people to sustain. iPhone volume control buttons, for example, have an activation force of 7.8 N.

Previous work has shown, in agreement with our own study, that due to enslavement effects there is no direct correspondence between neural commands to individual fingers and the resultant finger force. Neighboring fingers have been shown to produce forces up to 67.5% of the force they would produce in a single digit task [56]. This suggests that modifying threshold values based on the set of fingers used in a multi-finger gesture may lead to improved task performance.

### 5.2.3 Sensor Position

In our early iterations of the Unifone prototype, we placed the pressure sensors at points on the device that corresponded to finger placement in a typical one-handed cellphone grasp. One sensor placed on the top left hand side to be activated by the thumb, one on the back face near the top right to be activated by the index finger, and one on the left hand side to be activated by the middle finger. Our initial thought was that if we could minimize the distance between a digit and its target location it would necessarily decrease task completion time.

During the study it became clear that use of the thumb sensor interfered with the primary use of the thumb: operating the touch screen. Repositioning time could always be reduced by using an on screen soft control instead moving the thumb all the way to the side of the device to activate the sensor and then move back to the screen to continue the task.
The index sensor on back of the prototype also performed poorly, even though it appeared to be a natural placement. We now believe that the index finger rests on the back to provide stability against the thumb moving on the front. When a force was applied to activate a control it would tend rotate the screen around the axis between the thenar eminence and the grasping fingers. It became clear to us that the supporting fingers played an active and dynamic roll in positioning the thumb on the screen. A mobile is not held statically while the thumb moves from point to point but rather the relative position of the device and thumb changes through an interplay of all parts of the hand.

The pressure sensor on the opposing right hand side of the device proved to be much more useful. Like the other two sensors, it was positioned near a natural points of contact, in this case positioned for the middle finger, but did not overload the thumb or destabilize the users grasp because the device remains stable due to the opposing force provided by the thenar eminence.

After observing the time advantage of using the opposing fingers to provide a stable internal force, we built the next prototypes with the intention of maximizing this effect. With the final Unifone prototype, we made the entire opposing edge of the device pressure sensitive to further reduce targeting time. This led to significant reductions in task time, however, it could only detect force at one position at once, limiting the expressiveness of individual finger motions. The Isophone prototype uses a high spatial resolution array of pressure sensors so the position of force can be determined more accurately, and can be varied at multiple positions at the same time.

Ideally, a device would have both sides entirely covered with such arrays to allow use with either hand as well as creating an opportunity for interesting two-handed interactions. For single-handed use, it is important that there be enough force on the lower side of the phone to keep the force balanced against the opposing thenar eminence. Higher forces create a rotational moment that can destabilize grip. We believe that knowing the exact position that the thenar eminence is
pressed against allows novel interactions such as a reach-to-scroll where one simply reaches with the thumb and the device scrolls content toward it.

Figure 22. Graphical representations of data collected during the isophone study. Averages across participants are shown for a) screen area reached by the thumb b) TLX score (reminder: a lower TLX score indicates a more favorable result) c) inter-response interval d) force per finger.

5.3 Software Design Guidelines: Formulating an effective gestural language

The index finger is very capable of delivering pressure relatively fast, but as the results of our third study indicate, it limits the thumb’s range of motion and should be used for confirmations or cancelations but not for actions to be performed in conjunction with the thumb. During continuous gestures of the thumb, the index finger is better placed against the back of the device, where it can provide attitudinal stability.
The middle finger should activate the highest priority functions and those functions that are needed in conjunction with the thumb as it has the best combination of thumb range of motion and timing.

Force provided from the ring finger allows the most stable grasp, improves the thumb’s ability to move around the screen, and decreases the inter-response interval when combined with the other three fingers. However, a grasp becomes destabilized without it and should not be used to activate controls. Gestures should be tolerant of force from the ring finger but not rely on it for interaction.

The little finger, on its own, has a very long self-paced inter-response interval and does not provide a very stable grasp. However, similar to the middle finger, when combined with the ring finger, the little finger’s inter-response interval drops and the ring finger stabilizes the device, increasing the thumb’s range of motion around the screen.

We relied on activation from three fingers, but different sensor configurations such as a hybrid capacitance/force or extremely sensitive pressure sensor may present the user with options upon detecting presence of the appropriate fingers. The squeeze with the middle ring and little fingers allows the positions of the fingers to be determined. Further, false detection of the gesture is unlikely, as three different points would have to produce a significant force to appear as fingers, making this gesture a suitable home button replacement.

During the course of our first study, we found the use of continuous force-modulated input to be detrimental to task completion time. Precise control of isometric force with the fingers is difficult and complicated when the same finger also has to stabilize and coordinate the device with the thumb. Continuous input can, however, be used effectively to signal an early warning that the activation threshold is about to be reached. This gives the user time to correct their grip before inadvertently activating the control. As well as having the potential for lower error rates, it makes the device feel more responsive when activation is intended.
Data collected in the Isophone study suggest that it is most difficult to maintain a consistent force on the side of the device when the thumb is either reaching for a distant area of the screen and extra force is needed to push the device into the thenar eminence, or the contact area is too close to the base of the thumb and force needs to be decreased. If possible, on-screen elements that need to be manipulated by the thumb should be arranged in an arch centered at the contact point between the device and the base of the thumb and with a radius of the user's slightly flexed thumb. This allows the grasping fingers a greater degree of freedom for communication.

If constraining the screen layout is not desirable and the thumb is to perform a continuous gesture such as dragging, scrolling or drawing, the force from the thenar eminence can be still be taken into account. As long as the screen position of the thumb is known, it can be used to dynamically scale the activation threshold for finger-based controls. This would allow a relatively high activation force, and reduce unintentional activations at distant screen areas while maintaining a comfortable level of force closer to the thumb.
Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we presented two hardware prototypes: Unifone, and Isophone. Each investigated how auxiliary finger input enables novel one-handed gestures using a mobile device. Early iterations of the prototype helped determine which areas of a device would be suitable for auxiliary finger input. Results indicate that placements that allowed simultaneous action of the thumb combined with pressure from an opposing finger can be highly usable, as demonstrated in the final Unifone prototype. This prototype used a force-distributed pressure sensing metal attachment, one capable of sensing input on the top, middle, and bottom of its outside edges. This configuration allows course input controls that are calibrated for a user’s auxiliary fingers and affords imprecise targeting.

We evaluated how one-handed isometric gestures perform on four common mobile tasks compared to thumb-only input based on input time. The task completion time of five tasks that correspond to Unifone’s use as a direct, supportive, continuous, and discrete input device were measured. We showed that gestures that augment the behavior of the thumb perform best.

For the last study, a prototype with a high-resolution linear array of force sensors that could detect force from any finger down its entire length. This design eliminated any need for targeting a specific area and can support input from any combination of fingers.

The Isophone prototype was used to study how force provided by each of the fingers impacts the thumb’s ability to move around the screen, as well as the inter-response interval of each combination of fingers at performing a squeeze gesture. We found that providing force with the
ring finger increases performance in both these measures. We also found that applying force from
the index finger generally hinders the thumbs range of motion. These findings, and our
experience in building the prototypes, have been distilled in the set of hardware and software
design guidelines presented in the previous chapter.

6.2 Future Work

The research presented in this thesis has focused on the form factor of a smart phone, featuring a
flat, rectangular, screen. While this basic design does have its benefits, particularly ease of
manufacture and compatibility with the format of visual media, it is not particularly well
optimized for the mechanics of the hand. Today’s thin film screen technologies make curvature
along a single axis entirely feasible, even for mass production. The LG G-flex line of phones now
has concave screens that better fit the dynamics of the thumb. It is, however, just a small
departure from conventional design.

Inkjet screen manufacturing technology is already being used to produce FOLEDs and could
likely be adapted to print screens with arbitrary curvature along more than one axis. While
complex occlusions need to be taken into account, a screen curved along different dimensions
could be specifically optimized for the ranges of motion of the joints in the hand and thumb. Such
a device would probably not offer the best experience for viewing video clips, but may offer an
increase in task performance.

Moving away from a device made from rigid components presents many new fabrication
challenges to but also offers new opportunities for usability. In the context of auxiliary input,
rigid devices limit the auxiliary interaction to isomorphic pressure. Fingertips are most sensitive
to the force range between 0 and 5 Newtons, as that is the range at which their surface shape
deforms from semispherical to flat. Having the device give way to some degree could distribute
this force over the range of motion of the fingers and combine it with the proprioceptive senses. The studies we presented found very little success with using isometric force for precise control with the grasping hand, however, various tools and controllers very effectively use a pistol grip and variable trigger for continuous rate control. It would be very interesting to study possibilities for recreating this type of control in a generally deformable device, where the device itself replaces the variable trigger.

As materials and flexible digital technologies advance, it may be possible to realize our vision of a fully deformable device. Advances are already being made toward the development of deformable interactive electronic devices (we refer the reader to [36] for an overview). During the course of this work, the Human Media Lab has explored several of the potential uses and benefits of deformable device technology. In order to gain insight into how a typical user would expect to hold and interact with a thin film flexible display, Dijkstra et al. constructed a mockup paper display backed with a flexible pressure sensing multi-touch device known as Touchco [9]. Using the apparatus, the group was able to measure the effects of different structural holds on multi-touch performance.

Burstyn has created FlexView [5], a prototype system using an electrophoretic display backed with flex sensors and a flexible Wacom digitizer. Similar to and expanding upon the Unifone study, Burstyn uses deformation as an input channel to communicate with the device, comparing leafing (holding one side of the device rigid and bending the opposite, like leafing through the pages of a book) and squeezing to a baseline multi-touch interface.

Gomes et al. have continued to expand upon the uses of deformation by actuating a flexible screen device, and by using deformation as an output channel [12]. An electrophoretic display was backed with a custom-made actuation layer created by sewing Flexinol shape-memory wire through a cardboard substrate. They investigated the effectiveness of actuation and device animation at communicating the urgency of system notifications in a user study.
The aforementioned works have used flexible displays, flexible sensors and flexible actuators to create deformable interactive surfaces. However, in all cases, the power supply and most of the electronics driving the device have been contained in a separate housing and connected to the interactive area via cables. In order for deformable devices to achieve widespread adoption, advances materials and manufacturing procedures are needed to allow for fully flexible printed circuit boards, batteries and screens.

The company FlexEl has introduced a product known as BatteryCloth that uses ruthenium oxide in an electrochemical energy cell to produce thin film batteries that reportedly outperform conventional thin film lithium ion batteries by a factor of ten [68]. LG Chem is developing other enabling battery technology. By coating coiled copper wires with nickel-tin researchers have created a cable battery capable of forming a variety of shapes while remaining functional, even as they are repeatedly deformed [34].

An often overlooked, but critical, component needed to construct a generally deformable device is a deformable radio communication module. While the required flexible transistors have been available for some time, flexible diodes that can operate at the speeds required for Bluetooth and Wifi communication have only recently been created in a laboratory setting [57]. Combined with the abovementioned technologies, the widespread availability of these diodes will enable new mobile device designs that conform to the dynamics of the hand.
References

1. Agvard, A. Android one finger zoom tutorial.  


28. Morris, P. Unopened And Sealed First-Gen iPhone Hits eBay For A Cool $10,000. .


62. *Interlink FSR® Integration Guide*. 


65. Redpark TTL Serial Cable. 

76


Appendix A: Average Pressure at Screen Position