



**University of Copenhagen
Master thesis dissertation
Fat Finger**

Name:

Evangelos Tzemis : khb107@alumni.ku.dk

Supervisor:

Sebastian Boring

November 27, 2014

Abstract

Dedication

To my golden fish..

Acknowledgements

I want to thank...

Contents

1	Introduction	17
1.1	Introduction	17
1.2	2D Interaction Techniques Evolution	18
1.3	Fat Finger & 3D Interaction Techniques	19
1.4	Thesis Structure	20
2	Related Work	23
2.1	Pressure on Mobile Devices	24
2.2	Contact Shapes and Simulated Pressure as source of input	26
2.3	New Mobile Interaction approaches	27
2.4	Fat Finger establishment	28
3	Fat Finger Concept	29
3.1	Idea - Target	29
3.2	Scientific Research Questions addressed by Fat Finger	31
4	Implementation	35
4.1	Fat Finger Application Description	36
4.1.1	Basic WorkFlow	37
4.1.2	Basic Design of a Trial	38
4.2	Trial Categories	40
4.2.1	Feedback & Discrete Targeting	42
4.2.2	Feedback & Continuous Targeting	42

4.2.3	No Feedback & Discrete Targeting	43
4.2.4	No Feedback & Continuous Targeting	44
4.3	Final Sequence of Trials	44
4.4	Data Manipulation	46
4.4.1	Monitoring	46
4.4.2	Exporting	49
5	Experiment - User Study	51
5.1	Experiment's Procedure	51
5.1.1	Verbal Instructions	51
5.1.2	Demographic Information	52
5.1.3	Calibration	53
5.1.4	Assessment	54
5.2	Participants	55
5.3	Hypotheses	58
6	Results	59
6.1	Task Completion Time	59
6.2	Offsets	62
6.3	Learning Curve	64
6.3.1	Total Time	64
6.3.2	Offset	65
6.4	Re-Entries	67
6.5	Re-Touches	68
6.6	Subjective Preferences	69
7	Discussion	71
8	Conclusion	75
A	Verbal Instructions of Experiment	81

B Demographic Information Form	84
---------------------------------------	-----------

C Technique Assessment Form	86
------------------------------------	-----------

List of Figures

4.1	Fat Finger - Abstract Flowchart of Basic Modules	38
4.2	Fat Finger - Basic Trial Interface	38
4.3	Fat Finger - Basic interface during operation	39
4.4	Feedback & Discrete Targeting Interface	42
4.5	Feedback & Continuous Targeting Interface	43
4.6	No Feedback & Discrete Targeting and Confirmation Interface	43
4.7	No Feedback & Continuous Targeting Interface	44
4.8	Repetition Calculation	45
4.9	Fat finger - Generalized experiment flow of repetitions	46
5.1	Demonstration of a minimum and a maximum contact size hand position	54
5.2	Distribution of participant ages	56
5.3	Distribution of participants Gender	56
5.4	Level of Experience with Touch-Based Devices	57
5.5	Level of Experience with Tablet Devices	57
6.1	Total Time panelled by TypeID - 95% Confidence Interval	60
6.2	Offset aggregated - 95% Confidence Interval	63
6.3	Learning Curve - Task Completion Time	65
6.4	Learning Curve - Offset	66
6.5	Target Re-Entries panelled by Type of Trial	68
6.6	Target Re-Touches for Feedback Trials	69

6.7 Mean Values assessed by participants for each trial category 70

List of Tables

4.1	Apple iPad mini with Retina Display Characteristics	35
4.2	Fat Finger - Basic Interface Regions	39
4.3	Fat Finger - Trial Categories	40
4.4	Fat Finger - Universal Parameters Monitored	47
4.5	Fat Finger - Feedback & Continuous Targeting additional parameters monitored	48
4.6	Fat Finger - No Feedback & Discrete Targeting additional parameters monitored	49
4.7	Fat Finger - No Feedback & Continuous Targeting additional parameters monitored	49
6.1	Code names for the 4 categories of trials	59
6.2	Task Completion Time: Mean differences among the different types of Trials	61
6.3	No Feedback: Percentage of successful hits	64
6.4	Fat Finger - No Feedback & Discrete Targeting affordable error	67

Chapter 1

Introduction

1.1 Introduction

Over the last few years, mobile and tablet devices, have become an indispensable part of our every day life. We mainly use them to communicate with others (Skype, FaceTime, Facebook etc.), surf the web [41], read documents, and play video games. Tablet devices have also been assimilated in various working environments, providing a simpler, more accurate and direct way to monitor results, present graphs, and manipulate data by utilizing the touch motion and the device's portability. Due to this variety of usage, most manufacturers (Apple, Samsung, Google, LG etc.), continuously update their models with improved hardware and software to perfect match the user needs. Since tablets devices and smartphones do not include any physical keyboard, numerous new techniques have been introduced to overcome this issue. The idea behind those approaches can be described with the phrase: "*Give Meaning to Touch*". Introduction of gestures and multi-finger interaction [30] has partly solved this problem. However, the need for a more compact and robust approach will always exist. This need will lead to the establishment of tools for seamless and fluent interaction with our mobile devices. That lead us to the following (rhetorical) question:

"Is the way we interact with mobile devices the most effective one? If not, how can it be improved?"

Modern mobile devices are touch-based, allowing users to navigate through the interface using their finger, and type-in using a virtual keyboard. In the very beginning, when the first commercial tablets were born, interaction was based on single-finger input. User interface was based on buttons and users had to select the corresponding buttons to perform discrete actions. Recently, Multi-Finger interaction has been introduced, trying to emulate and simulate the behaviour of objects. This intends to make the user-interface more natural and improve users experience. Styluses have also been used to provide input on tablet devices. They are meant to assist users in performing more complex and accurate tasks (as drawing, designing), while some of

them are even equipped with pressure sensitive sensors that detect the pressure applied and can be utilized by drawing applications (e.g. Photoshop) to control stroke width, or similar. Application-wise there are several applications in the App Store, Google Play etc. that use all those different kind of gestures to make complex tasks seem less confused. However users have to keep track of all those different gestures, and in case the application is not carefully designed, gestures will become cumbersome. For example, one finger for panning, blending into a two-finger pinch gesture for zooming, or even a three-finger drag to change modes in specific applications. From now on we will refer to interaction techniques that use only the position (x-y coordinates) of the finger(s) for providing basic input to tablet devices as **2D interaction techniques**, and to those that might use 2D interaction plus an extra parameter (pressure, simulated pressure, vibration absorption) as **3D interaction techniques**. Taking everything into consideration, I strongly believe that tablet devices are capable of providing even more natural ways for manipulating the content on them, and this is what I am seeking for. I am proposing the Fat Finger interaction technique which exploits the contact size of the finger touching the screen, and thus belongs to the 3D interaction techniques. It adds a degree of freedom, which could later be used to integrate multi-finger drag gestures into a single fluid interaction, or just enhance the current ways we interact with tablet or mobile devices.

1.2 2D Interaction Techniques Evolution

During the past years, the ways we use and provide input to our mobile devices have been vastly changed and evolved. The first input interface we meet is the simple 10 button keypad. Each button represents a number and a set of characters. Help buttons (as “*”, “#”, “accept call” and “end call”) were also provided. Whilst typing in such devices was cumbersome, continuous training and exposure to the interface made certain users develop high level technique, resulting in super-fast typing. In other, more recent mobile devices we meet a mobile keyboard with an increased number of buttons, called QWERTY [31], with which users had one-to-one correspondence between buttons and letters. To increase usability also in old-style phones, a new typing technique was developed. Using a dictionary-based typing system, the necessary buttons clicks needed to type any kind of text, were vastly decreased. Thus, typing speed improved, even with a very simple number-based keypad.

With the release of the first iPhone [21] in 2007, the smartphones industry, gained massive adoption. Smartphones can be thought as augmented featured normal phones. The iPhone was one of the first mobile phones to use a multi-touch interface, and the finger as the main source of input. also, modern smartphones does not include a physical keyboard. So now, instead of providing input through the keypad, virtual keyboards and touch events were introduced.

Tablets followed the release of smartphones. A tablet is a portable computer with a touch sensitive display. It is usually equipped with cameras (front & back), microphones, accelerometer, gyroscope, Touch ID [19] etc. It provides touch input capa-

bilities that can either be utilized using finger or a stylus. For text-entry, apart from handwriting recognition they also offer virtual on-screen keyboards. Finally their screen size is typically between 7" and 10.1". The first tablet device was commercially available in 1989 from GRiD systems and was named GRiDPad. Until 2010, many companies have released their own version of tablet, most of them using resistive stylus driven screens. Resistive screens allow a high level of precision and are definitely preferred when used with a stylus. After 2010 tablets use capacitive touchscreens, which allows multi-finger interaction, and avoid the use of styluses. This allows integrated hand & eye operation, since there is neither a stylus or a mouse to interfere. They also use ARM processors from improved battery life. Apple iPad was the product that defined the class of tablets and shaped the commercial market when launched, back in 2010. It runs iOS, a mobile version of MacOS, specifically designed for finger use, avoiding stylus requirements.

Above we observed that many things changed through the past years in the commercial products, both addressing the mobile phones and the tablet devices. However the principle we use to interact with all those mobile or tablet devices, did not really evolved, remaining stable to 2 degrees of interaction (2D). In older keypads, the dimensions are the buttons and a boolean value which represents if a specific button is being touched or not. When operating on touch screens, we only get the x-y (2D) coordinates of the point(s) we are touching on the screen. Using multi-touch, we can now combine multiple fingers to perform more complex operations, which proved to be very effective. This is mainly because most of the gestures and movements we can do with physical objects, just transferred to the virtual environment. The most commonly used gestures are: Tap, Double Tap, Long Press, Scroll, Pan, Flick Two Finger Tap, Two Finger Scroll, Pinch, Zoom and Rotate. Despite the flexibility multi-finger interaction provide us, 2D principle still holds because we only utilize the position of each finger touching the screen, and not other parameters as contact size, orientation, tilt, etc. Finally, taking everything into consideration, I believe that we need to exploit the capabilities of modern devices and enhance the current ways of interaction, by introducing a third input dimension, as section 1.3 explains and analyses.

1.3 Fat Finger & 3D Interaction Techniques

When referring to *3D Interaction Techniques* we refer to all those methods that use three-dimensional (3D) input to navigate into a two-dimensional (2D) environment. For the purposes of this study, by referring to a 2D environment we mean the interface provided by modern mobile and tablet devices. These interfaces are becoming more and more complex as user needs and application functionalities increase. As a result, we need to find a way to respond to this interface complexity and be able to provide as precise input as we want. Multi-touching has been proposed for this exact reason; to make interaction with applications less constrained and more fluid. However, the need to provide even better and coherent input to tablet devices, only through our finger(s) still exists and holds. For instance, using a stylus, this can be achieved by

exploiting the pressure sensing capabilities they provide. Different actions can be mapped to different pressure levels. But what can be achieved without using any external equipment?

In this study, we are studying the capabilities of our finger(s) to provide three-dimensional (3D) input on a tablet device. The three dimensions we propose are the position on the x axis, the position on the y axis and finally the size of the contact area touching the screen. The first two dimensions (x-y position) have been already extensively studied and used by all current mobile and tablet devices. However there is insufficient research on which are the capabilities of the contact size ability to provide accurate input on mobile devices. We propose Fat Finger, which investigates and extensively studies, the limits on using contact size for input on a tablet device. We settle an experiment which is in the form of discrete target selection tasks. We then analyse the results we obtained and try to reason on the actual capabilities of this interaction technique.

Getting to understand how touch works, is the only and necessary step towards applying a three-dimensional (3D) input method on tablet devices. Once we know, the limits of contact size capabilities as a source of input, then we can combine it with the on-screen position of the finger, or multiple ones, and benefit from this new way of interaction. However, apart from that, applications should be designed and implemented, to take advantage of this input method. That way complex actions will be easily performed by mapping actions to different contact size levels. Contact size can also be used to simplify current interfaces. For example zooming, or sound level are features that can be controlled by alternating the contact size of our finger, which will lead to the removal of their corresponding buttons, and thus the simplification of the interface. Applications as Adobe Photoshop, will now be able to control the stroke width without using any stylus augmented approaches. Concluding, Fat Finger will study the capabilities of using contact size and simulated pressure as an additional parameter for input on tablet devices, while integration with applications and test of possible usages will be a part of another study.

1.4 Thesis Structure

The structure and the content of the thesis is shortly described below. This work is consisted of 7 Chapter, self-including, which are structured as follows:

- **Chapter 2:** Publications related to pressure or contact size detection and monitoring on Mobile devices, are presented. We quote papers that are highly relevant to our work, and also others that are just in the same field. They are meant to help you, the reader, to better comprehend current environment and advances in the area, familiarize with this field of studies and finally realize the uniqueness of this study.
- **Chapter 3:** We present the concept and analyse the idea behind Fat Finger interaction technique. We then state the scientific question that this study tries

to respond to, and also give some details on the methodology we will use to give answers to each of those questions.

- ***Chapter 4:*** We present the Implementation of the software required to test Fat Finger. We give every possible detail on the way it works, how it is designed and on the way the interface is being built.
- ***Chapter 5:*** All the information regarding the User Study we performed, are included. We present and thoroughly explain each of the phases –steps– we followed for each participant that took part in our study. Finally we set our hypotheses on the results.
- ***Chapter 6:*** We present the results of the User Study, separated in corresponding categories. Each category represents a specific metric we tested. We also provide visualization through graphs and finally we comment on the results.
- ***Chapter 7:*** We discuss the results, and comment on whether the overall concept and the hypotheses placed on Chapter 5 hold.
- ***Chapter 8:*** Conclusion and future work suggestions are included.

Chapter 2

Related Work

In this section relevant work in related fields is presented, analysed and discussed. We first present some summarized information on relevant papers, to give an idea of what is going to follow. Afterwards, we separate relevant literature in publications that are related with pressure on mobile devices (section 2.1), those related with measuring the contact size or investigating simulated pressure techniques (section 2.2), and finally we present some new approaches on finding alternative ways to interact with our mobile devices. Below we give a short introduction on the relevant literature, by outlining some significant papers and giving a hastily intro to the research field. Part of this short introduction was also included in my thesis synopsis-proposal.

Fat Finger was mainly influenced by "*Fat Thumb: Using the Thumb's Contact Size for Single-Handed Mobile Interaction*" [4] which deals with a very similar problem. Boring et al. presents *Fat Thumb* as an alternative technique that adds a dimension to touch input on a mobile device, thus giving different meanings to touch motion. It was done by making use of thumb's contact size allowing seamless mode switching. Testing supported the hypothesis that *Fat Thumb* is at least as fast as other related techniques. In literature we find examples of efforts towards enhancing the current ways of interacting with mobile devices. FingerSkate ergonomic based study [36], introduces a variation to the current multi-touch operations. It aims to make this operations less constrained and more continuous. With FingerSkate, once one starts a multi-touch operation, he can continue the operation without having to maintain both fingers on the screen. It also addresses to major ergonomic issues, that simple multi-touch operations have. Thumb Rock [3] is another approach in interacting with mobile devices taking advantage of the contact size. In particular it presents an in-drag gesture that consists of rolling the thumb back and forth on a touch surface. Taking advantage of this, many operations on mobile devices are simplified. It can be used as a supplement to tapping, allowing editing or zooming depending on the application. Fat Finger is mainly focused on the use of our index finger and the exploitation of it's contact area to better interact with a tablet-mobile device.

"Contact size as an input parameter is closely related to pressure (i.e., more pressure

suggests a larger contact size due to flattening of the finger" [4]. Ramos et al. [33] found that only a level of six different pressure values is actually optimal and distinguishable by the users. However his study has been done using a stylus as input to a tablet device. In GraspZoom [29] a different approach was chosen integrating pressure in the interaction with the mobile device. Users could press the back of the device, allowing them to temporally switch modes (e.g. from panning to zooming). All the above systems are investigating the different ways of pressure as an input parameter, but none of these systems explore the limitations and the detail we can acquire when we use the contact size of our finger as the main source of input.

2.1 Pressure on Mobile Devices

Using pressure to interact with a mobile device has been a great field of research almost since the first computers were launched. Using pressure for controlling specific application or widgets, where the use of mouse is not behaving naturally has been previously investigated and tested.

One of the first researches, on using pressure in UI (User Interface) was done by Herot and Weinzapfel [13] back in 1978. They explored the ability of the human finger to apply pressure and torque to a computer screen. They describe a PSD, which is able to accept input of direction and torque, produced when our finger is touching the computer screen. They conclude that "*touch and pressure sensing open a rich channel for immediate and multi-dimensional interaction*" ([13]). Buxton et al. [6], in a later study, explored touch-sensitive tablet input and at that time suggested that, control when pressure is the only source of input, can be difficult, especially in the absence of buttons. However as we will see, this opinion alternated through the years, and control through various pressure techniques has been vastly improved and evolved. Ramos and Balakrishnan [32], introduced a concept prototype called LEAN. It was designed to provide navigation, segmentation and linking capabilities on digital videos, and targeted to be used with pressure-sensitive tablets, as it contains widgets that can be controlled through the pen's pressure. By utilizing pressure sensitive pens, they increased the available ways of providing input to a tablet device.

Ramos et al. on pressure widgets [33], performs a high related to ours study. However, instead of using the finger as the source of input, he explores the pressure-sensing capabilities of styluses, when used to provide input on a tablet device. He reasoned on the following research questions, which also have many common aspects with our prospective for Fat Finger. *How many different pressure levels can a user distinguish; what is the impact of visual feedback on participants performance; what is training impact on user performance; what mechanisms can be used to confirm target selection?* Human ability on performing discrete selection tasks (Continuous targets are not included) by using a pressure sensitive stylus, is also investigated. He is investigating, among others, Dwell and Quick Release techniques for confirming target selection. Dwell is about maintaining the cursor inside the target region for 1 second; while Quick Release involves the quick lifting of the stylus from the tablet's surface. As

we will see later, we will also use the two aforementioned techniques, but not for comparing or even evaluating purposes. He concludes that 6 levels is the optimal division of the pressure space, because errors are still affordable. He also points out that, Quick Release is the preferred selection technique.

In GraspZoom [29], Miyaki and Rekimoto propose a multi-state input model for mobile devices that is controlled through pressure and thumb gesture. User can apply pressure to the back of the mobile device, which switches from zooming to panning mode. However this approach required an extra pressure sensor (FSR), attached on the back of each device. User studies have also been conducted to better understand the fundamental traits of pressure, in UI for general mobile devices. Stewart et al. in *Characteristics of pressure-based input for mobile devices* [38], tried to understand and reason on the mapping functions for pressure input. They investigate the results of applying front, back, and both sided pressure on a mobile device, while performing a pinch movement. They conclude that input form both sides outperforms the single-sided one and is competitive with techniques that apply pressure against solid surfaces.

McCallum et al. in PressureText [28], suggested a way of using pressure in old style - numeric keypad phones. Using an pressure-sensitive keypad, and mapping multiple taps to different levels of pressure found that, PressureText performs equally well with other existing techniques, especially after repeated and continuous exposure and training. On the other hand, Clarkson et al. proposed the addition of a simple pressure sensors under the keypad buttons, and not using an external one [8]. They conduct a study in which they facilitate using pressure, already existed interaction techniques. Brewster et al. also investigates possible ways to use pressure for text input on mobile devices [5]. They map soft presses to lower-case letters, and hard presses to upper-case, trying to boost mixed-case text typing. They conclude that pressure-based input can outperform that shift-based standard, when we focus on mobile devices.

There are also researches, which investigated different ways of identifying the levels of pressure applied on a mobile device, without using hardware augmented approaches. They rely on software to estimate-calculate the pressure applied, from sensors that are commonly available on mobile devices (e.g., accelerometer). [9, 12, 17, 18]. GripSense [9] uses the built-in sensors of mobile devices to infer hand postures. ForceTap [12], combines location data from the screen and data form the accelerometer to distinguish strong versus gentle taps. Hwang et al. in MicPen [17] with a microphone equipped stylus pen tried to estimate the pressure level applied on the screen. This is done by analysing the acoustic signal of the interference between the screen and the stylus. Also, in PseudoButton [18], Hwang et al. proposes another inexpensive way to emulate pressure sensitive touch sensor, again by utilizing and re-purposing the built-in microphone on mobile devices.

In a more recent study, VibPress [16] implements a technique to detect pressure " *by measuring the level of vibration absorption with the built-in accelerometer when the device is in contact with a damping surface (e.g., user's hands)*" ([16]). They support that this technique is at least as accurate as hardware supported approaches. The

maximum number of pressure levels that are distinguishable by users is also broached and studied. Low et al. investigated the ability to detect the pressure applied on the screen through the camera and the flash of a mobile phone [27]. This is accomplished by measuring the light that is reflected through our finger, from the flash to the camera. The more pressure the less this light is. Finally, Arif et al. investigated the pseudo-pressure detection on standard touch screens and concludes that with this technique we are only able to identify two different pressure levels [1]. He then utilizes this technique to present a pressure-based predictive text entry technique, in which extra pressure is used to omit changes from unwanted predictions.

2.2 Contact Shapes and Simulated Pressure as source of input

This section presents publications that investigate the use of Contact Shapes, Simulated Pressure - Contact Size as the source for input mainly on mobile devices. Please note that relevant studies that presented in the introduction of Section 2 are not included here.

In 1985, Lee et al. [24] presented a touch-sensitive tablet prototype. It was capable to concurrently detect the on-screen position of multiple fingers, and also estimate the contact size for each of those fingers. It was one of the first three dimensional approaches in interacting with a tablet device. There have also been approaches on applying multi-touch sensing on rear projected interactive surfaces [11]. The touch sensing was achieved by utilizing the total internal reflection.

Benko et al. proposed "Dual Finger Selections" techniques which, are designed to support and assist users in selecting very small targets on touch-sensitive displays [2]. They are capable in providing pixel-accurate targeting, but only if the tablet is also equipped with computer vision-based tracking. The user study showed the superiority of "Dual Finger Selections" compared to the standard techniques. ShapeTouch [7] tried to fully utilize the contact shape of the fingers touching the screen, by using interactive surfaces to manipulate various objects. ShapeTouch tries to simulate real object interaction by inferring virtual contact forces from the contact regions and use them to enable interaction with virtual objects. In AnglePose [34] we observe another approach to track the position and angle of the fingers touching the screen. In *Detecting and leveraging finger orientation for interaction with direct-touch surfaces* [40] we encounter yet another approach to move the interaction from 2D (only x-y coordination information), to a 3D one, by exploring the role of finger orientation.

Holz et al. in *Understanding Touch* [14], revisits the assumption that users acquire target with the centre point of the contact area between their finger and the screen. He supports that it is subject to systematic error offsets in our interactions. Their study drops the error rates from $4mm$ to $1.6mm$, which give evidence that users do align visual features with the target, and this is indeed the most possible way they are thinking of touch input. In *MicroRolls* [35] we encounter a study, in which after

having understood the already existing touching capabilities, they try to enhance the interaction between the thumb and a mobile device by detecting and discriminating those thumb gestures that present zero tangential velocity. They name them Micro-Rolls. Finally, in [10] contact area has been utilized to provide text entry capabilities to visual impaired people.

2.3 New Mobile Interaction approaches

The past few years there is an increasing demand on seeking for alternative ways on using the mobile and electronic devices. Users are willing to experience new more sophisticated ways, that might also feel more natural in mobile device interaction, than the already existed ones. In this section we present relevant work which is heading towards the fulfilment of this ideal; find alternative sophisticated ways in HCI. However we only include approaches that are relevant with pressure, or finger interaction.

Pointpose [22] proposes a way to increase the expressiveness of touch input, by adding a 3rd dimension variable on the way we interact with mobile devices; finger rotation and finger tilt. To perform this operation, Kratz et al. uses a short range depth sensor which, scans the touch screen of the mobile device, and using their proposed algorithm, finger rotation and tilt can be precisely detected and calculated. Point pose can settle the foundations to build upon and create many kinds of applications that will interact with the user by taking advantage of the finger expressiveness. Spinder et al. [37] reconsiders zoom and pan in mobile devices. They present a study in which they thoroughly compare pinch and drag gestures with their proposed technique, that relies on spatial manipulation. An example of spatial interaction can be to move a tablet device up and down to zoom in and out respectively. They conclude that their proposed technique is performing better than the Pinch-Drag-Flick mainstream one.

Lochtefeld et al. [26] addresses the problems we face when we want to use a large screen mobile device with only one hand. Operations is then limited since we do not have full control by both holding and interacting the device. They evaluate a back-front device touch input mechanism, which allows accurate handling but sacrifices the performance. TouchShield [15], tries to manipulate the one-hand limitations on mobile devices. It is done through a visual control that can be activated when large contact size is detected on screen for a specified amount of time. Then it is able to provide you thumb-access to some frequently used commands, while retaining the ability to observe the underneath interface. Finally, Li et al. [25] proposes another way for single-handed mobile interaction. It allows the user to select on-screen objects with a simple fluid-action consisted of two parts. Firstly, bezel swiping to invoke the tool and afterwards virtual pointing which is used to select objects that are beyond your thumbs access area.

2.4 Fat Finger establishment

In this section we met many different publications, all trying to find, investigate and test, alternative ways of using pressure or contact size as an input parameter on mobile devices. We encountered approaches which use pressure sensitive screens to measure the pressure applied on them. Others with external devices attached on the mobile devices, used to detect pressure or similar by scanning and tracking the movement of the finger. We also observed efforts in trying to infer pressure from sensors that are already available inside the mobile devices. What all those approaches have in common is that they try to improve the user interface and the experience we gain from using our mobile devices.

All the aforementioned systems examine alternative ways of using pressure or simulated pressure as an input parameter on mobile devices. We went through studies that extensively investigate the usage of thumb as the main source of input; especially when we use our mobile device in single-hand operation. Ramos et al. in Pressure Widgets [33] investigated the pressure as raw data, and did a significant effort to understand what are its limits. He tried to understand how many distinguishable pressure levels we can obtain when using a pressure sensitive pen on a tablet device. However none of the aforementioned papers address our research question.

In Fat Finger we are trying to achieve significant improvements, by utilizing the contact area of our index finger. Our ultimate target is to develop new ways of interaction with the iPad, which will enhance the current ones. We have to deal with the finger's contact size area and more specifically, with how much detail we can obtain from this area, to give us the ability of mapping many different actions to different contact sizes. Specifically, we are trying to exploit the capabilities of the touch area when using our index finger, in the best possible detailed way. Finally, we do not accomplish this study having in mind to use Fat Finger technique as an alternative to the already used ones, but we want to be able to use it in parallel with them, to further augment users experience. In the next chapter we explain our logic and thinking behind this application, and we analyse what we will try to achieve from this study.

Chapter 3

Fat Finger Concept

This fairly short chapter presents the original idea behind Fat Finger interaction technique, along with the way we conceptualized its usage; finally it states the main research questions that this study will try to respond to. Being motivated from the current condition, environment and facts as described in Chapter 1, we searched and then presented the relevant bibliography in Chapter 2. This helped us to understand what other researchers have already achieved in the same, or closely related fields. Moreover, among the papers, we encountered many different approaches on how to implement, test and evaluate an idea, which we used to build a robust and coherent experiment capable to provide us with the necessary tools to study Fat Finger interaction technique. The following sections present the theoretical part of this study, and more specifically, the original idea and the theoretical questions we will try to debate and reason on.

3.1 Idea - Target

In my thesis proposal, I stated the following (Quoting):

I am proposing the *Fat Finger interaction technique* which uses a finger's contact size as a form of simulated pressure. This adds a degree of freedom, which could later be used to integrate multi-finger drag gestures into a single fluid interaction, or just enhance the current ways we interact with tablet or mobile devices. In order to achieve that, we should find a way of interacting with the iPad using only our finger, while its contact size is going to give us the flexibility we want. In terms of goals, we want to exploit the expressiveness of touch, by giving more meanings to it apart from just being a boolean motion — touch or no-touch. Therefore the main question that our thesis will try to answer is:

"To which extend are we able to distinguish the different simulated pressure levels

produced by our fingers using a tablet device?"

In terms of methodology I am going to specify my approach on dealing with the above research question. We should also consider that we are not limiting the research on one finger interaction, but we also want to study multi-finger gestures, always combined with the finger contact size.

- ***How many contact-size levels are we able to determine?***

We know that one level is straightforward, but what happens then? First we need some kind of calibration with the user's finger —possibly one calibration per different finger— to determine the smallest and the biggest possible contact size. Then we want to divide this difference into N segments. Sub-target of this study is to determine how much this N can grow up, while keeping its expressiveness. The upper goal is to reach 16 different distinguishable pressure levels.

- ***How can we test that the N levels we managed to determine are useful and distinguishable by the user?***

To determine whether the levels are all distinguishable by the users we are going to use the following test structure (Figure 1), with possible variations. We divide the contact size area in N segments and then force the user to match a specific —levelled— contact size. Two N-scaled columns are provided; one with the target contact-size (left) and another with our current contact-size (right). The contact size feedback indicator is growing up meaning that bigger contact sizes are higher in the column.

- ***What about a multi finger combination?***

We would like to increase the expressiveness of a multi finger gesture using each finger's contact size. Testing can be performed in the same way. The aim, again, is reaching the specified contact-size, either by taking the average contact area of the fingers used or by introducing a new column for each finger.

As we can see, the main target of this study is to understand how touch and simulated pressure works, under our specific point of view. We want to investigate the capabilities and the precision when we use the contact area of the index finger of our dominant hand to interact with a tablet, and after that utilize and combine Fat Finger interaction technique with already existing gestures used for current interaction with mobile devices. That way, it will become possible to exploit its capabilities and see how it performs in real life use and in combination with the already existed gestures currently provided by Apple iOS, Google Android and Microsoft Windows mobile operating system. Also, of great importance is the use of multi-finger combination alongside with the use of the contact size provided by those operating systems. For example, a different action could be performed in a three finger drag gesture depending on how hard we touch the screen. Of course, the learning curve of such

techniques would be steeper, which can be leveraged by the advantages we will gain. Despite how tempting this technique might seem, it goes beyond the sphere of this study and maybe it can be explored in the future. It is important to understand "one-finger interaction", which we decided to be our **index finger** of our **dominant hand**, and then move to higher complexity levels (more than one finger at the same time). Thus, this study focuses only in studying how index-finger interaction through contact-size or simulated pressure behaves on mobile tablet devices. Multi finger interaction technique would be a great attribute that would enhance the current ways we use to interact with tablet devices, but as we already mentioned it is beyond the scope of this study. We focus on understanding the capabilities of users in using the contact size area to perform target acquisition tasks. In section 3.2 we present the basic research questions that we will try to reason on in this study.

3.2 Scientific Research Questions addressed by Fat Finger

This sections presents the scientific research questions addressed by Fat Finger. For each of them, we shortly explain the concept and a glance of the methodology we will use in our effort to provide clues, facts and results in the corresponding questions. To study and test Fat Finger interaction technique we designed an experiment in the form of target selection trials. Implementation of the corresponding software is described in Chapter 4, while the user study in Chapter 5. Below we present the research questions that this study will try to respond to.

- *Q1: How many discrete pressure levels are we able to distinguish when using the index finger to interact with a tablet device?*

It is really important to define the limits for the maximum number of discrete pressure levels that users can distinguish. It is a metric of the quality of the source of input. We therefore need to test a wide range of pressure levels and check in which of them, the error rates are small enough to allow fluid operations. In this user study we will be investigating a range from 2 to 16 pressure levels. Our target is to define an upper limit for pressure levels, after which, input will be prone to errors. Higher values for pressure levels, infer that Fat Finger interaction technique is accurate enough, meaning that users are able to provide precise n-level pressure input on the tablet device.

- *Q2: How the size of the target region influences overall performance?*

This research question is highly related to **Q1**, since the number of Pressure levels varies inversely as the size of the target. When the number of pressure levels increases, each level represents fewer pressure values (shorter range). When aggregated pressure levels should add up to the total pressure range. In other words, we try to fit different number of pressure levels in the same pressure range.

- *Q3: In which region is our finger more capable to operate on? Smaller or larger contact sizes?*

We will try to reason in which level of pressure we are more capable to operate. Since pressure is related to contact size, and contact size can mainly altered by changing the posture-position of our hand; we need to investigate whether the position and physical ergonomic construction of our hand has anything to do with precision and so on.

- *Q4: Which is the role visual Feedback? To which extend it affects the performance?*

Feedback is an inseparable aspect of most (if not all) interaction techniques in HCI (Human Computer Interaction). For instance, when we move the mouse we get feedback through the movement of the cursor on the screen. What would happen if there were no cursor indicating the position of the mouse or even if the cursor would only appear frequently (partial feedback), and not always (full feedback). Our interaction with PCs would be much more difficult, right? We do need to investigate this behaviour in Fat Finger and observe how feedback influences users performance on relevant selection tasks.

- *Q5: What is the differences among the target selection techniques? How they affect performance?*

Fat Finger is designed to enhance the interaction with tablet devices. Thus, it is very important to be able to perform target selection tasks. For instance, when we use a simple mouse device, we perform button selections by clicking. However, when we use touch as the main source of input we need to find an appropriate target selection mechanism. We will be selecting targets either using delay or by lifting off the screen.

- *Q6: Does training in Fat Finger affect performance? Which is the Learning Curve?*

We would like to measure the effort required to learn and get used to Fat Finger interaction technique. To achieve that, training face can not be separated from the actual experiment. If we train users during the experiment procedure, we should be able to apply metrics to capture and monitor their performance. A technique whose learning curve is not steep, would be definitely preferred by users. As we will see in Chapters 4, 5 and 6, we will combine training phase with the rest of the experiment, which will give us the ability to study and then export results regarding the learning behaviour of users.

- *Q7: In which detail are users able to develop haptic memory on various pressure levels? Is it even possible?*

Haptic memory is introduced mostly when there is not visual feedback. In that occasion users tend to develop haptic memory, which assists them in selecting the desired targets. Haptic memory is of major importance and can vastly improve user performance; especially when they have already memorized the movements and they are also provided with feedback.

Taking into account the above research questions, we build a User study that will help us to provide evidence and appropriate data which will give answers to them. Specifically, on chapter 6 we present the results we collected, while on Chapter 7 we discuss the results and we try to respond to the aforementioned questions.

Chapter 4

Implementation

Fat Finger is an application that is developed and targeted for Apple tablets devices. It runs on a standard Apple iPad mini with Retina Display device, which runs the iOS 7 (version 7.1.1) operating system. The device has the following specifications:

Specification	Description
Display	7.9-inch (diagonal) LED-back-lit Multi-Touch
Display resolution	2048-by-1536 resolution at 326 ppi
Chip:	A7 chip with 64-bit architecture

Table 4.1: Apple iPad mini with Retina Display Characteristics

Fat Finger can also be ported on any other iPad devices that run iOS version 7 or higher of the Apple's mobile operating system (iOS). It should be mentioned that while iPad devices come in different screen sizes and resolutions, porting an App from one device to another is not an issue anymore. This is mainly feasible due to the innovative way of thinking in terms of pixels and points, Apple proposed for using graphics on an iOS device.

In particular, since iOS 4, dimensions are measured in “**points**” instead of pixels. All coordinate values and distances are specified using points, which are floating-point values. One point can refer to one or more pixels. The measurable size of a point varies from device to device and is largely irrelevant. Points provide a fixed frame of reference for drawing graphics on screen. To have an inside look on how points correspond to pixels, imagine that an iPad screen 768 x 1024 points and is size independent. Thus, this point resolution scales to the following pixel resolution for some referential models:

- **iPad 1** - 768 x 1024 pixels (i.e. a scale factor of 1.0)
- **iPad Air** - 1536 x 2048 pixels (i.e. a scale factor of 2.0)
- **iPad Mini - Retina Display** - 1536 x 2048 pixels (scale factor of 2.0)

When performing graphical operations, you should specify your measurements in points for all devices, and iOS automatically draws everything to the right proportion on the screen. The outcome is that when you draw the same content on two similar devices, it has the same scale on both.

The code has been developed in Objective C language using Xcode. Xcode is an Integrated Development Environment (IDE) created by Apple for developing software for both OS X and iOS. Full Documentation for developing iOS applications is also included, alongside with many other utilities that are designed to enhance your coding experience (Tester, Debugger, etc.). Apple along with the release of iOS 8, last June, also released a new programming language for coding Apple software, **Swift**.

"Swift is a multi-paradigm, compiled programming language created by Apple for iOS and OS X development. Introduced at Apple's 2014 Worldwide Developers Conference, Swift is designed to work with Apple's Cocoa and Cocoa Touch frameworks and the large body of existing Objective-C code written for Apple products. Swift is intended to be more resilient to erroneous code ("safer") than Objective-C. It is built with the LLVM compiler included in Xcode 6, and uses the Objective-C runtime, allowing C, Objective-C, Objective-C++ and Swift code to run within a single program."

[23]

4.1 Fat Finger Application Description

Fat Finger is an application which has been built to test the capabilities and the limits of the index finger's operational skills, on using the contact area touching the screen for integrating with a mobile device. The interaction with the finger requires an interface that will always listen to touch events on the screen. Thus, Fat Finger, whilst not being a complete application, it embeds all the infrastructure needed to support the new interaction we are proposing. The basic scenario is the following: *Whenever a finger is touching the screen we should be able to calculate the contact area between the finger and the screen.* After obtaining that raw data, we should use that to interact with the interface provided. Despite how simple it may seem at a first glance, Apple does not provide a way to calculate this contact area, and neither does with the shape of the area covered by the finger. The iPad's API only provides the major radius in pixels of a contact points' ellipse, which can be in rough lines obtained as follows:

```
NSValue *val = [touch valueForKey:@"_pathMajorRadius"];
float size = [val floatValue]; //size in pixels
```

As a result, we can obtain only the major radius of the area covered by the finger. Moreover, we can not have continuous observation of the area but we can calculate this size only when a new touch event has happened. Generally in iOS, touch is event driven. Whenever a finger is touching the screen a relevant function is triggered to serve this event. iOS provides us the following 4 basic functions for handling touch events:

- **TouchesBegan** – This is called when the screen is first touched.
- **TouchesMoved** – This is called when the location of the touch changes as the user is sliding their fingers around the screen.
- **TouchesEnded** – TouchesEnded is called when the users' fingers are lifted off the screen.
- **TouchesCancelled** - It is called if the system cancels the sequence of touch events.

Thus, whenever each of the first three functions are getting called, we can measure the size of the area covered by the finger. The problem arises exactly when we realize that each of the above functions is called when the location of the touch changes, and not when the contact area changes. In other words, a new event is triggered only when the center point of the touching area changes. Indeed, there are occasions that the contact area is changing without altering the center of the ellipsis. It results in non continuous observation of the contact area touching the screen, and there is not an existing way of alternating this behaviour ourselves. We are constrained in measuring the area if the location of the finger has been changed. Changes in contact area that do not alter the location in the screen, can not be measured. However, after investigation we realized that this circumstances are extremely rare under normal use so, despite the limitations we faced, our methodology works fine.

4.1.1 Basic WorkFlow

Fat Finger application can be divided on the following modules:

- **Start App** - Represents the launching of the application.
- **Register Participant** - In this module, we assign a new unique ID to each participant through a custom form.
- **Calibrate** - Participant can calibrate his finger.
- **Visualize** - Participant can visualize the previously obtained calibrated values. If he is not satisfied with the outcome, he can jump back to Calibration mode and calibrate again.
- **Trials - Experiment** - This is the main part of the experiment, in which all trials are included.
- **End of Experiment** - After having finished the experiment trials, experiment is finished.

The flowchart of those modules can be observed on Figure 4.1. We notice that there is a circular connection between Calibrate & Visualize. This is due to the fact that we can calibrate and then visualize as many times as we need or want to. When we are satisfied with the calibrated values, we may proceed to the Experiment Trials.

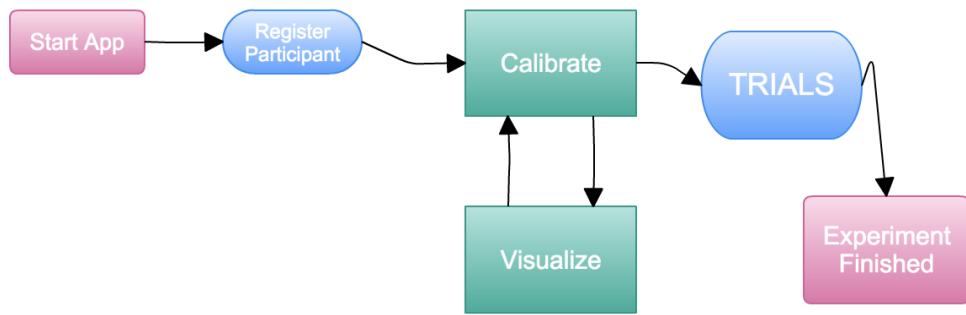


Figure 4.1: Fat Finger - Abstract Flowchart of Basic Modules

4.1.2 Basic Design of a Trial

Each Trial has one basic design and interface. Our target was to find the most obvious and physical way to be able to on-screen present all the aspects we wanted. I ended up on using the design, illustrated in Figure 4.2. The challenging part was in finding a competitive alternative to the simple bar design. The basic usage scenario is:

The user should move his finger, in a way to alternate the contact area between the screen and his finger. There should be an indicator that will give him feedback. He should know the amount of pressure his is applying to the screen, through a visual representation.

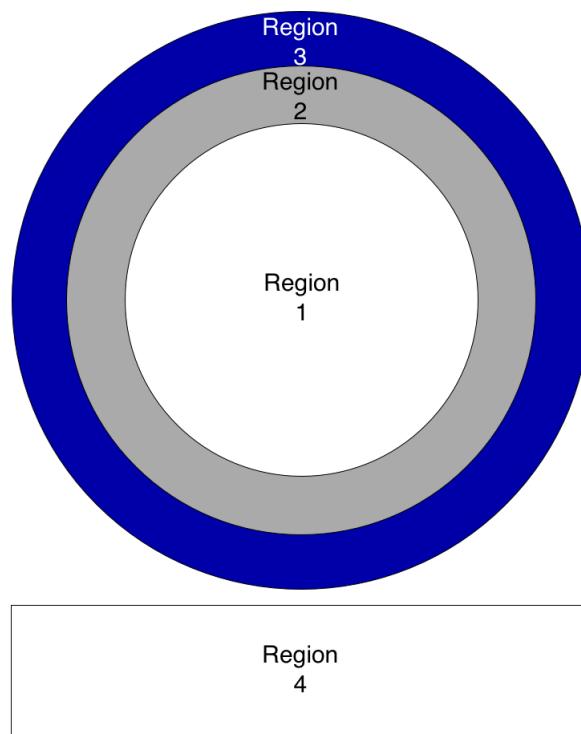


Figure 4.2: Fat Finger - Basic Trial Interface

In Figure 4.2 we can observe that our interface has 4 regions, all of them are placed on an iPad screen, and that the whole interface is based on overlapping circles with common center. Table 4.2 illustrates the correspondence between Numbers and regions.

Region No	Description
1	Non interactive region. Is placed there, simply in terms of design.
2	Red coloured targets are placed here.
3	FeedBack region.
4	Touch Region. Touch is enabled everywhere though.

Table 4.2: Fat Finger - Basic Interface Regions

To better assimilate this design lets use a driving example. *Imagine that there is a bar (similar to a progress bar), which is empty when we are barely touching the screen and full when we apply full pressure on the screen. Moreover in each trial, our mission is to achieve a specific level of filling it (e.g. 40%). Since the bar depicts the movement of our finger, this can be achieved by alternating the area covered by our finger.*

Now we are ready to move to the next level. Instead of familiarizing with this simple and purely designed idea, I decided to use the circle as the fundamental design shape. Region 3 (Table 4.2) will be alternated while the participant is moving his finger, so as to perfectly map his movement. Therefore, we will have a partially filled circle for the different contact areas (Figure 4.3).

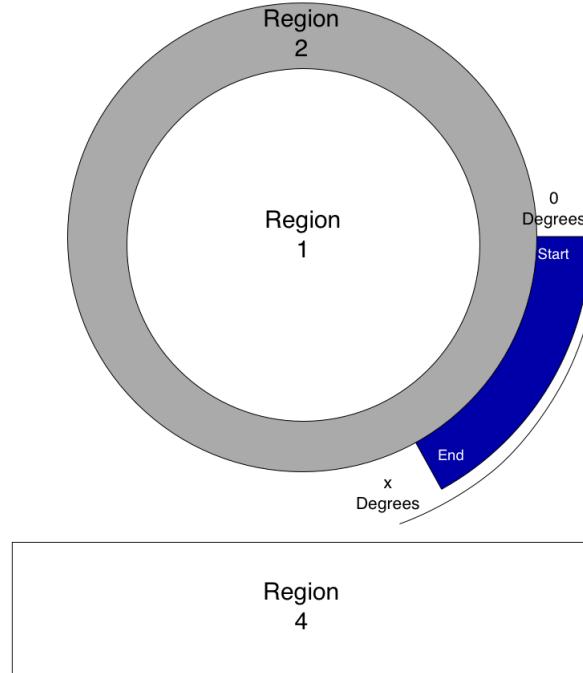


Figure 4.3: Fat Finger - Basic interface during operation

From Figure 4.3 we can point out many significant characteristics for our design. The starting point of measurement is x-positive axis, and degrees grow with a clockwise orientation. Thus when the contact area is minimum $x = 0$, when maximum $x = 360$ and $0 < x < 360$ for all other values. Finally there is a linear correspondence between contact size and degrees x , which is calculated through the type: $x = \frac{\text{currentSize} - \text{minSize}}{\text{maxSize} - \text{minSize}} * 360$, in degrees.

4.2 Trial Categories

Fat Finger consist of four (4) Trial categories. Those 4 categories are based on the values of 2 binary variables **TARGET & FEEDBACK**. The values of this variables can be:

- **Target = Discrete or Continuous**
- **Feedback = be Feedback or No Feedback**

Producing all possible combinations of values from **Target** and **Feedback** we end up having the 4 types we have already mentioned. All possible trials one can encountered belong to one of the those types, which are:

Targeting	Feedback	No Feedback
Discrete	Feedback & Discrete	No Feedback & Discrete
Continuous	Feedback & Continuous	No Feedback & Continuous

Table 4.3: Fat Finger - Trial Categories

During the experiment, those 4 types (Table 4.3) will occur multiple times with different variable difficulties. Difficulty can be defined as "*The combination of the position and the size - width of the target*". The smaller the target the more difficult one trial is. To be able to measure the difficulty of each Trial we use a new variable **N** which indicates how many buckets each trial has. So First we need to define what a bucket is.

Bucket is a part of the targets region (Region 2 in Figure 4.2). The target region can be divided in N equal parts. Each of those parts is called Bucket. For example if N = 2 then the first bucket is the "down" semicircle of the target region (0 – 180 degrees), and the second is the "upper" one (180 – 360 degrees)

N varies inversely as the value of **Buckets**. In an inverse variation, the values of the two variables change in an opposite manner - as one value decreases, the other increases. Therefore, as **N** increases, the range of each bucket decreases. Moreover the ranges of all buckets should add up to 360 degrees.

At this point, the values for the two basic variables **Target** and **Feedback** will be thoroughly analyzed.

- **Discrete.** This only relates to Region 2 (Figure 4.2), so the formatting of the targets. In this type, targets are discrete, meaning that Region 2 is divided in N buckets, but only one of them is activated. The activated bucket is coloured in red and acts as the target for this trial. All other inactive buckets are coloured in gray.
- **Continuous.** This also relates to Region 2 (Figure 4.2) exclusively. In this type, Region 2 is not divided into buckets, at least visible. We think in term of targets, but we do not present them graphically. The target is just one small red line (a bucket with 1 degree range). However it might be impossible for the user to select such a tiny target, so we offer an offset area which is delimited with two yellow lines. The position of the target is not random. As I mentioned before, we still think in term of buckets. But instead of considering the whole bucket as a target, we just randomly select a small region inside that bucket, and draw the red line there. Thus, we have the same amount of trials as in the Discrete type.
- **Feedback** is related with Region 3 (Figure 4.2). In this type, Region 3 is visible, meaning that we have continuous feedback while we are moving our finger. In order to select a target the user has to perform the Dwell technique. Thus, keep the edge of the blue line (Region 3) inside the red bucket for at least one second. Then a short sound confirms the selection and the trial is dismissed. Summarizing, the mission is:

Keep the edge of the blue "line" inside the red target for 1 second.

- **No FeedBack** is also related with region 3 (Figure 4.2). In this type, Region 3 is now invisible. Therefore we do not have feedback indicator while we are alternating the contact area between our finger and the screen. Thus we need to memorize the movement and be able to predict the position of the edge of the "blue" region. To confirm selection, user has to lift his finger from the screen, or perform the QuickRelease selection technique. The software will then choose the last contact area size, before the lift as the one the user implied. As a consequence we can only lift the finger once. We are not allowed to perform retouches, because on the first lift, target selection will be performed.

It should also be mentioned that, in this type of trials, we are not forced to hit the target successfully. We just make a prediction and then lift our hand. Our input will be recorded and then displayed to us through a graphical Confirmation Interface. Summarizing, in this type, our mission is:

Touch the screen, alternate the contact area and make a prediction on where the blue line should be to hit the target, lift the finger when you are on the preferred contact size, and finally observe the outcome - feedback.

We have now explained and introduced all the necessary background, and we are ready to present the Interface of the 4 categories of trials as stated in Table 4.3.

4.2.1 Feedback & Discrete Targeting

In this type of trials, we can understand that the targets will be discrete and there will also be feedback on our input. Figure 4.4 presents the interface of this type. As we can see both Region 2 and 3 (Figure 4.2) exist. Since we have *Discrete* targets, Region 2 is separated in n buckets, all with equal size. The target is the one coloured in red, and all others are inactive and coloured in gray. On the other hand, *Feedback* keyword is mentioned too, which means that the blue circle (Region 3) exists. The movement of the blue region corresponds to the movement of our hand. To accomplish this trial, the participant needs to perform the Dwell selection technique.

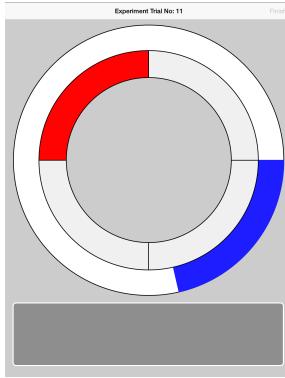


Figure 4.4: Feedback & Discrete Targeting Interface

4.2.2 Feedback & Continuous Targeting

In this occasion, we are still provided with feedback on our input, but the target will now be of type Continuous. Figure 4.5 presents the interface of this category. *Continuous* targets mean that only one thin red line exist in Region 2 (Figure 4.2). While the position of the target is related with buckets, the actual buckets are just invisible. As we previously mentioned, the red line is randomly positioned within one of the N buckets. The two yellow lines, surrounding the red one are the offset limits and their distance is the offset range. User can confirm a target by staying inside that range, however he is instructed to target as close and accurate to the target as possible. Finally, *Feedback* keyword specifies that Region 3 exists. The process for confirming the selection of a target, is to keep the moving edge of the blue region inside the range of the target (yellow lines), for at least 1 second.

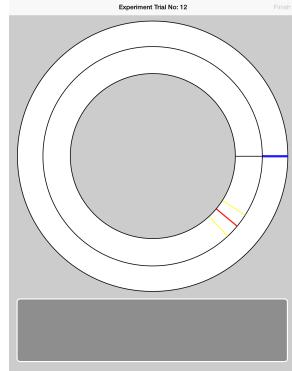
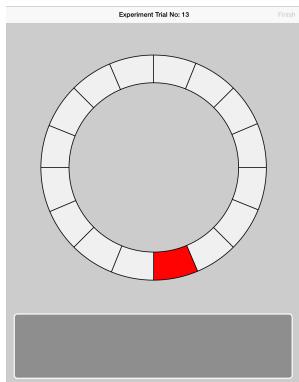


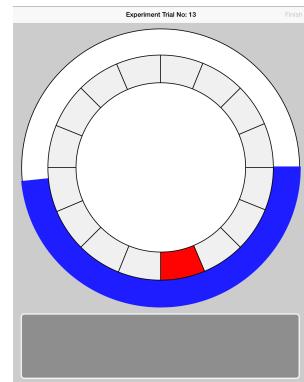
Figure 4.5: Feedback & Continuous Targeting Interface

4.2.3 No Feedback & Discrete Targeting

The rules for *Discrete* targets apply here. Thus, Region 2 (Figure 4.2) is separated in N buckets, all with equal size. Variable N mostly specifies the difficulty of the trials. The target is the one coloured in red, and all others are inactive and coloured in gray. This category differs from the aforementioned because it is of type *No Feedback*. That means that Region 3 (Figure 4.2) does not exist. So, in the beginning we encounter the interface illustrated in Figure 4.6a. Our task is to **a**) touch the screen, **b**) predict the contact area needed to select the target and **c**), confirm selection with QuickRelease. After confirming selection, we face the **Confirming Interface** shown in Figure 4.6b, in which we can observe our performance in the trials. Did we hit that target or not? This input can be used as a learning parameter which will help us to improve our performance as we move through the trials.



(a) Targeting Interface



(b) Confirmation Interface

Figure 4.6: No Feedback & Discrete Targeting and Confirmation Interface

4.2.4 No Feedback & Continuous Targeting

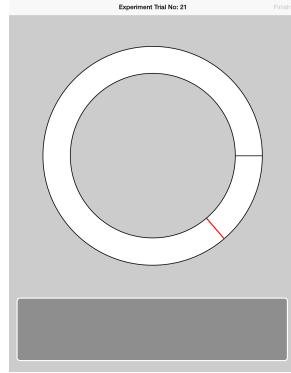


Figure 4.7: No Feedback & Continuous Targeting Interface

Figure 4.7 presents the interface of the *No Feedback & Continuous Targeting* category. Targets are *Continuous*, meaning that only one thin red line exist in Region 2 (Figure 4.2). The two yellow lines, surrounding the red one still exist. They represent the offset limits and their distance is the offset range. It also of type *No Feedback*. Thus, Region 3 (Figure 4.2) does not exist. Our task is again to **a**) touch the screen, **b**) predict the contact area needed to select the target and **c**), confirm selection by lifting the finger form the screen. After confirming selection, we once more face the **Confirming Interface** shown in Figure 4.6b, in which we can observe our performance in the trials.

4.3 Final Sequence of Trials

Thus far we have explained all the main characteristics of our experiment. Now we should focus on the final sequence of the trials and the whole structure of the experiment. In this experiment we want to investigate 7 different N's, or bucket sizes, which can have the following values:

$$N = [2, 3, 4, 6, 8, 12, 16]$$

For each N we have the same amount of related targets. For instance if $N = 2$, then we have two available targets, one for each bucket. If $N = 4$, the we have 4 possible targets, etc. As a result for all the above mentioned values for N we can gave the following number of possible targets:

$$2(N = 2) + 3(N = 3) + 4(N = 4) + 6(N = 6) + 8(N = 8) + 12(N = 12) + 16(N = 16) = 51 \text{ targets}$$

Then we have 4 different types of Trials: Feedback & Discrete, No Feedback & Discrete, Feedback & Continuous, No Feedback & Continuous.

For each of the above categories we conduct all the aforementioned targets. So the

total number of Trials is: $51 * 4 = 204$. Furthermore we need to randomize their order just to make sure that it will not have any effect in our study. After the randomization takes place we say that we have 1 **Repetition** of the Trials. Figure 4.8 graphically illustrates this procedure.

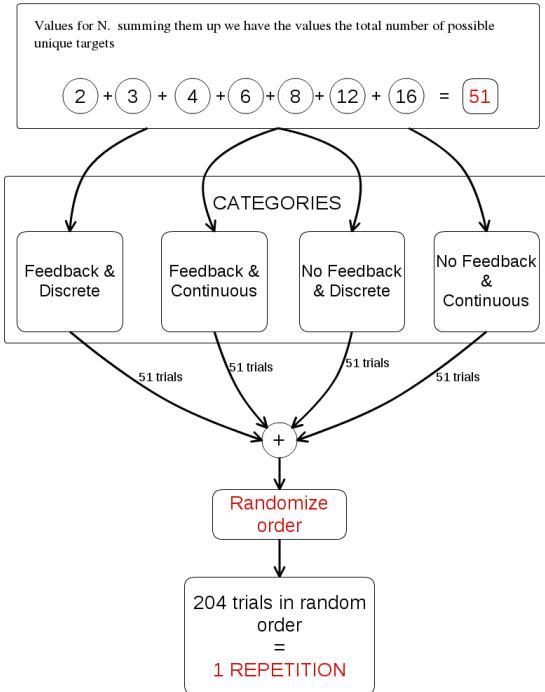


Figure 4.8: Repetition Calculation

However, this experiment contains 3 repetitions of those trials and not only 1. So in total we will have $203 * 3 \text{ repetitions} = 612 \text{ trials}$. We chose to have 3 Repetitions for the following 2 main reasons.

- There is no warm-up session for this experiment. Instead of having separate trials to train users, we chose to combine the Learning phase with the 1st repetition of the Trials. That way we will be able to observe and study the pace with which users are learning this new interaction technique.
- The experiment should last enough so as to give users the time to absorb the information given, familiarize with the environment and technique and also explore whether side effects as distraction of attention and fatigue have an impact on their performance.

Figure 4.8 graphically illustrates how the repetitions are combined together. It also gives an overview of the work-flow of Fat Finger.

Finally it should be mentioned that between each trial of this application there exist a button that should be pressed in order to proceed to the next trial. It is annotated with the text: "Next Trial" and the reason it exists is just for allowing users to take smalls breaks whenever needed. While this button is visible, no counter or other

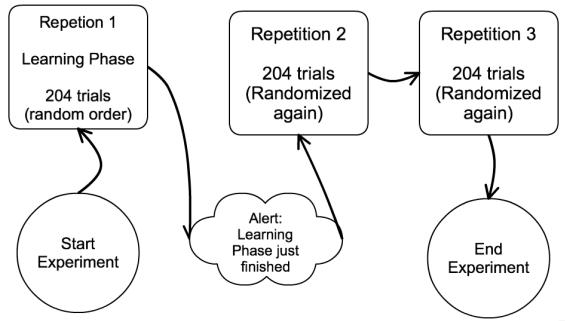


Figure 4.9: Fat finger - Generalized experiment flow of repetitions

mechanisms are fired. Finally it has the same shape, size and position as Region 1 in Figure 4.2

4.4 Data Manipulation

Data Monitoring and Exporting are of significant importance when we have to deal with the accomplishment of an experiment. When performing a user study you need to decide which parameters you are going to monitor, and to find a way to measure them. Code-wise, you need to implement a software that is able to calculate all the metrics you want to observe, and then store them somewhere consistently. After the experiment is finished, it is truly crucial to be able to export those data from the place they are stored into a common format that is accessible from another program. Summarizing we have a three-step process when we conduct an experiment.

1. Decide which parameters to measure.
2. Store the observed values in a consistent database.
3. Export the data to appropriate format (.xls, .csv, etc.)

Our approach for the first two points is analyzed on section 4.4.1, while for the latter one on section 4.4.2.

4.4.1 Monitoring

Table 4.4 presents the basic parameters we measure that are common for all types of trials. The first column contains the name of the corresponding variable, the second a short description of its usage, and on the third we can observe which is the field of values for each variable.

Name	Description	Field of Values
trialID	Incremental id of this trial	[1, 2, ..., 612]
typeID	Id of the type of the trial	[1,2,3,4]
N	Total number of Buckets	[2,3,4,6,8,12,16]
min	Minimum calibrated radius	<i>Float > 0</i>
max	Maximum calibrated radius	<i>Float > 0</i>
rawInputValue	Raw Value between Min and Max	Float number
reEntries	Number of Target Re-entries	<i>Decimal >= 0</i>
repetitionID	Repetition id they belong to	[1,2,3]
reTouches	Number of Target Re-Touches	<i>Decimal >= 0</i>
target	Which of the buckets was the target	[1,2,...,n]
totalTime	Total time to accomplish the Trial	Float number

Table 4.4: Fat Finger - Universal Parameters Monitored

Combining the above values, apart from giving us the possibility to uniquely identify the trials of the experiment, we can also reconstruct the whole experiment. This can be achieved, because we are storing the outcome of each trials, the rawData we collected form the user, the total time, etc. Now, we should further analyse each of those parameters:

- **trialID.** Unique, auto-incremented identifier for each trial. Most importantly it specifies the order of appearance of the experiments.
- **N.** Specifies the number of buckets. **N** is an indicator that is related, at least in the Feedback trials, with the difficulty. This study is mainly concerned in specifying an upper limit for the number of identifiable levels in the contact area range. Thus, N is the sorting parameter for being able to distinguish the upper limit for those levels.
- **min.** The minimum value for the PathMajorRadius we collected from the calibration. During the calibration process, user can calibrate his finger as many times as he want. When he proceeds to the experiment, we collect the lastly calibrated values and store them accordingly.
- **max.** The same applies here too, except that we are referring to the maximum value of the PathMajorRadius.
- **rawInputValue.** This is the PathMajorRadius value, for the contact area that hit the target. For Feedback trials, it is the last touch input size exactly after the 1 second delay has passed. For No feedback, it is the area just before lifting our finger.
- **reEntries.** Counts the times we went outside the target region. It starts counting after the initial entry to the target.
- **repetitionID.** Identifies in which of the three repetitions this trial belongs to.

- **reTouches.** Counts the times we lifted our finger off the screen. Of course for *No Feedback* trials this parameter is always zero since we confirm target selection by lifting our finger from the screen.
- **target.** Specifies which of the buckets is the target. Thus it can take values from 1 to N.
- **totalTime** Total completion time of the trial. The timer fires when the first touch is performed and stops when the confirming sound is being played. As a result *Feedback* trials will have values ≥ 1 second. This is because one second is the duration someone has to be inside the target region to confirm target selection. On the other hand, on *No feedback* trials, durations are generally shorter.

The above parameters are measured for all 4 types of trials. However, all except for *Feedback & Discrete* required some extra monitoring. **Feedback & Continuous** has the following two extra measured parameters that are also presented in Table 4.5:

- **targetPosition.** Defines the exact position of the Target. In this type, target is represented as a thin red line. The position will always be inside the **target**-th bucket out of the **n** in total. However since it is randomly positioned inside this bucket, we needed to record its accurate position. Value monitored is between **min** and **max**. To transform the position into circle degrees we can use the type $PositionInDegrees = \frac{targetPosition - min}{max - min} * 360$.
- **offset** Indicates how far we were from the target. It can be thought as an error rate too. The higher the offset the higher the error we performed on our selection. Positive values mean that we hit somewhere after the target. Negative values mean that we hit before it. Also offset is measured in percentage. If we want to find how many degrees we were off the target then we can just use the type: $DegreesOff = offset * 360$

Name	Description	Field of Values
targetPosition	Position of the Target	min \leq Float \leq max
offset	Distance from target in percentage	float :[-1 to +1]

Table 4.5: Fat Finger - Feedback & Continuous Targeting additional parameters monitored

No Feedback & Discrete targeting requires a different set of extra parameters to be measured. Feedback region does not exist in this type, which means that users have to predict where it is. This type of trial requires longer learning time. Thus for the initial trials we expect that participants predictions will not be very accurate. As a result, offset will be high. To monitor that we use the following two parameters as shown in Table 4.6:

- **hitInsideTarget** As it can be inferred from its name, it indicates whether we actually hit inside the target or outside.
- **offset** mostly represents the error in percentage over the whole range. In *No Feedback* trials error can be extremely high as there is no limitation in performing in-target selection. Thus positive values mean that the selection was performed at a point after the center of the target, while negative values stand for before.

Name	Description	Field of Values
offset	Distance from target in percentage	float :[-1 to +1]
hitInsideTarget	Indicates if we successfully hit the Target	True, False

Table 4.6: Fat Finger - No Feedback & Discrete Targeting additional parameters monitored

Finally, in **No Feedback & Continuous Targeting** we meet a combination of the extra parameters we have already mentioned. To better monitor this type we use the extra supplementary parameters, illustrated in Table 4.7. Each of them has been already explained and analyzed in the aforementioned types.

Name	Description	Field of Values
offset	Distance from target in percentage	float :[-1 to +1]
targetPosition	Position of the Target	min <=Float<= max
hitInsideTarget	Indicates if we successfully hit the Target	True, False

Table 4.7: Fat Finger - No Feedback & Continuous Targeting additional parameters monitored

4.4.2 Exporting

Monitoring the results and saving them into a database was half the way towards the fulfilment of our goal. There is a need to export them into a reasonable format so that they can be imported into a statical analysis tool. Fat Finger used Core Data for storing all the aforementioned parameters. *"Core Data is an object graph and persistence framework provided by Apple. It allows data organized by the relational entity–attribute model to be serialized into XML, binary, or SQLite stores. The data can be manipulated using higher level objects representing entities and their relationships. Core Data manages the serialized version, providing object life-cycle and object graph management, including persistence. Core Data interfaces directly with SQLite, insulating the developer from the underlying SQL."* [20]

Core Data does not provide a specific output data format, neither has an automatic exporting tool. However, since we have full access on the data stored in database we can implement our own customized export tool. For the purposes of this study, I

used the Comma Separated Values (**.csv**) format. The output file has the following format:

- Each line represents a case. In other words, each line represents the data collected for a specific trial of a user.
- Each case can be uniquely identified by the combination of the usedID and trialID parameters.
- Each user performed 612 trials. Those he will be presented by 612 unique cases.
- Each variable (n, targets, userID, totalTime etc.) is represented by a column.

Chapter 5

Experiment - User Study

5.1 Experiment's Procedure

In this chapter, the procedure of the experiment is discussed and thoroughly analysed. We conducted a user study to test and analyse Fat Finger. Each user participating in this study, should follow the instructions given to him, complete the experimental part and finally provide us an assessment. The procedure followed for each participant consist of the following parts.

Welcoming - Verbal instructions - Demographic information - Calibration - Experiment - Assessment

The elements for each of them will be discussed, interpreted, and visualized with descriptive figures. Duration of each experiment procedure was calculated to last around 50 minutes. Adding to it the time needed for the instruction and the final assessment we end up having a study that requires about one hour to be fully completed. Apart from that, we should mention that, whilst being an anonymous user study a camera was used to record each participant during the experiment. To keep anonymity, the camera was only focusing on the participants hand and the iPad of course. The basic aim behind using a camera is that this will very likely make it possible to observe many interesting traits on the way participants use their finger to interact with the UI (User Interface). We might be able to detect different techniques or ways of interaction used among the users. Finally it can be a vital tool on reasoning on weirdly acquired or ambiguous collected results.

5.1.1 Verbal Instructions

In the beginning of the experiment a set of instructions (see Appendix A) was given to the participants. Instructions are divided into an introductory and an explanatory part. On the first part, explanatory, the experimenter introduces himself and welcomes the participant. At this point, the participants rights should be explained to

the participant. He is allowed to withdraw from the experiment, any time he desires if he feels uncomfortable. Also it is mentioned that this study is anonymous so we will keep no personal information on him. After that we thank him for participating and helping us in this study. Finally we introduce him to the purpose of this study and what we are trying to achieve.

At this point we assign him a **user id** and pass him the Demographic information form, which explained in detail in Section 5.1.2.

After having filled in the Demographic information form, we are ready to move to the explanatory part. Here, we start by explaining the various aspects we will encounter through this experiment. Firstly the 4 different types of trials are presented and thoroughly explained to him. Before proceeding to the next step we need to make sure that they understood the differences on the targets, the feedback, and also the way of confirming target selection on each type. Then we advise them that they should be as fast and accurate as possible during the trials. Finally, in case they experience fatigue or they just need a break during the experiment, they are welcome to do so, but only when the "Next Trial" button is visible on the screen. That way the parameters of the experiment (e.g., time, re-touches etc.) will be not affected at all.

5.1.2 Demographic Information

Demographic Information form is used to collect several basic information about the participant of this study. However, to ensure confidentiality we do not ask them to fill in their name, surname or address. The fields that need to be filled in by the user are:

- **Age.** Age is based on the year of birth, and not on the actual date.
- **Gender.** Male or Female.
- **Optical Aid.** None, Glasses, or contact lenses.
- **Handedness.** Right, Left or Ambidextrous.

According to these fields we can easily categorize the participants into corresponding groups, and export meaningful results (see Chapter 6).

Finally we ask users to rate their experience in using touch-based mobile devices, and also in using tablet devices. Ranking is based on a 5 point rate scale; very inexperienced (1) to very experienced (5). We also query them if they own, or at least have used a tablet devices and a touch-based device before. In particular:

- **Do you own a tablet device (e.g. iPad)?**

Yes - No

- **Have you used a tablet device before?**

Yes - No

- **How do you rate your experience with touch-based mobile device?**

Highly inexperienced (1) - Highly experienced (5)

- **How do you rate your experience with tablet devices?**

Highly inexperienced (1) - Highly experienced (5)

The printed version of the Demographic information form can be found in Appendix B, with all the information included.

5.1.3 Calibration

This section is devoted on the finger calibration process. Calibration is a very important parameter, and necessary too. Since fingers sizes vary among people. The application should be recalibrated for each participant, so as to better match his unique hand dimensions. Thus, it consist an indivisible part of our study, which takes place just before the start of the experiment. Calibration means that we simply let the system measure the dimensions of our index finger. That way it will be trained to understand if we are barely, merely or fully touching the screen.

There are many possible ways a user can calibrate his finger, or restated, each finger can be calibrated with more than one ways. Calibration is a very subjective parameter. Each participant is allowed to calibrate his finger the way he feels more comfortable. During the calibration process we simply let the system know which are our desired minimum and maximum contact area values.

There are two (2) basic techniques for alternating the contact size of the on-screen covered area.

- **Alternate finger's angle.** For this approach, you need to touch your finger almost vertically to the screen to set the minimum contact area size, and move towards a horizontal position for the maximum desired value. Please note that the maximum and minimum values are not feasible ones, but just the preferred ones. If a user wants to perform smaller movements with his hand then he needs to adjusts the values to have a shorter range. However if someone wants to perform longer and bigger movements, they he is welcome to set the minimum and maximum values accordingly.
- **alternate the pressure applied.** Apply no pressure to set the minimum desired contact area and enough pressure the set the maximum desired point. The same rules for the desired and possible values apply here too.

However some guidelines were given to them, to help them perform it properly.

1. Touch the screen softly with a medium contact size. This movement should be as natural as possible. No need to apply pressure or tension at all.
2. Move your finger to the desired minimum level. Again there is no need to apply any pressure on the device. The participant should be as relaxed as possible.
3. Move your finger to the desired maximum level. The allowed part of the finger that can be used as contact area is delimited by first joint point.
4. From this maximum position please lift your finger.

After performing those steps, the participant may proceed to the visualization screen. In this view, he is able to play around with the interface which is calibrated according to the previously acquired input values. When he feels confident that the calibration is well responding to his finger movement then he may proceed to the experiment. Otherwise, he can calibrate again as many times he wants, until he is completely satisfied with the visualization outcome.

Finally, there are two basic techniques users follow for alternating the contact area. In the first we keep all our fingers tight together, and only the index finger is pointing. To change the contact size we need to move our palm and arm. This causes increased fatigue after long term use. On the second one, we simply relax our palm on the table, in a position that only the index finger will be touching the iPad screen. This way alternation of the contact size is effortless and produces the minimum amount of fatigue to the finger.



(a) Minimum contact size position



(b) Maximum contact size position

Figure 5.1: Demonstration of a minimum and a maximum contact size hand position

5.1.4 Assessment

After having completed all the trials, participants instructed to fill a questionnaire to access, rate and finally comment on each of the type of trials they encountered. The questionnaire (Appendix C) was divided into 4 parts, one for each targeting technique. So, participants had to provide an assessment for all the following type of trials:

1) Feedback & Discrete Targeting **2)** Feedback & Continuous Targeting **3)** No Feedback & Discrete Targeting **4)** No Feedback & Continuous Targeting

The assessments had the same structure, and were independent of the type of the

trials. Each assessment consists of 6 categories, all on a five point Likert scale. The categories were:

- **Smoothness during operation.** The user should rate the smoothness during the operation of the specified Trial type. Smoothness behaves as a general factor that is totally subjective. Signs of roughness can be a slow rate frame in the graphics, or lack on the accuracy of the feedback line.

Scale: Very rough (1) - Very smooth (5).

- **Operation speed.** If participants had the feeling the everything flows in a normal rate then they should rate it with 3 which is the normal speed. In this index the best rating is considered 3.

Scale: Too fast (1) - 2 - Normal (3) - 4 - Too slow (5).

- **Finger fatigue.** Level of finger fatigue experienced by users through the corresponding type of trial in this experiment.

Scale: None (1) - Very high (5).

- **Wrist fatigue.** Level of wrist fatigue experienced by users through the corresponding type of trial in experiment.

Scale: None (1) - Very high (5).

- **General comfort.** The meaning is inferred from its name. What we ask for, is a general impression on how comfortable this type of trial was. Of course there is interference between this factor and the wrist and finger fatigue ones. However, a type of trial might be comfortable under normal use, but it may cause fatigue in longer use.

Scale: Very uncomfortable (1) - Very comfortable (5).

- **Overall assessment.** It measures the general impression of this type. It is measured in usability factors. Thus rating 1 means that it was easy to use and in turn, rating 5 that it was really tough to use.

Scale: Very difficult to use (1) - Very easy to use (5).

Finally, at the bottom of each page there was a comment section, in which participants were asked to comment on any advantages or disadvantages they might have encountered during the operation of each type of trial. It was not a required section, however many participants shared their thoughts with us, and we did extract some really important results out of them.

5.2 Participants

We recruited 26 participants to participate in our study ranging in age from 19 to 52 years old. They had to fill in a demographic information form, from which we collected the following info.

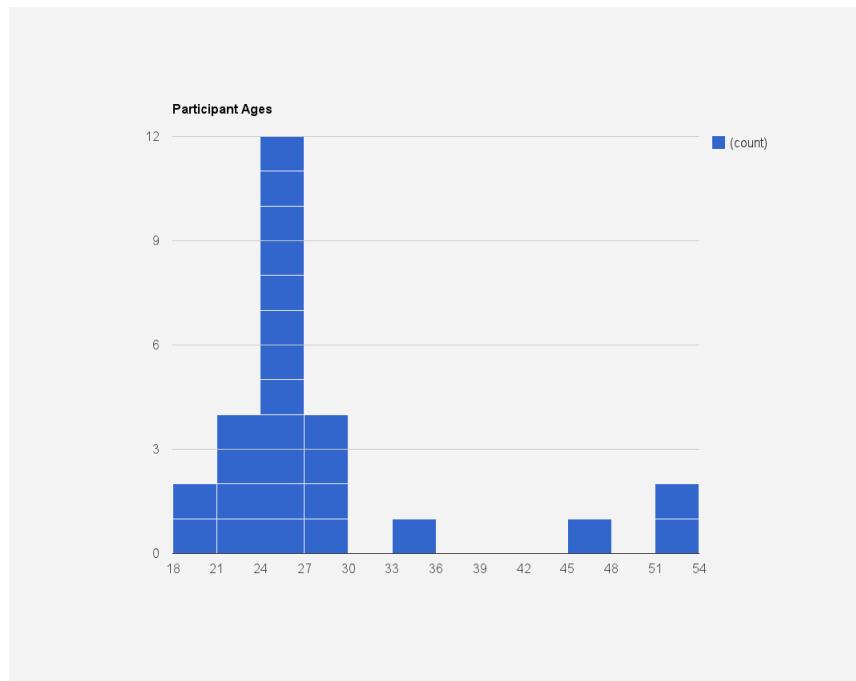


Figure 5.2: Distribution of participant ages

As we can see from Figure 5.2 participants are mostly ranged between 21 and 30 years old. We also tried to recruit people from older ages, that would hopefully behave interestingly different on the experiment. We managed to recruit 2 participants on the 33-48 range and other 2 above 50 years old.

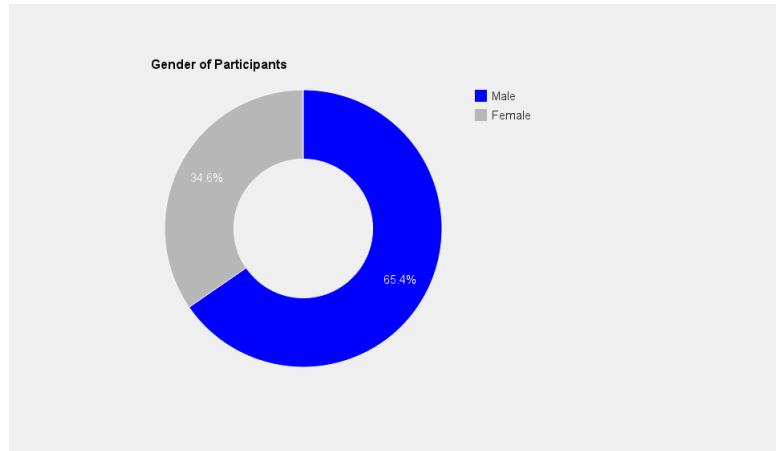


Figure 5.3: Distribution of participants Gender

Participants were mostly male (65.4%) as it can be observed on Figure 5.3. To be precise, we had 17 participants that were male and 9 female (34.6%). Also 24 of them were right-handed while only 2 used their left hand for the experiment. Moreover only 21 of them owns a touch based device of any possible kind, and from those just 12 own an actual tablet device. A mobile phone, a GPS etc. can be regarded as touch

based devices. We can then conclude that less than half of the users owns a Tablet device, but not specifically an Apple iPad one.

Figures 5.4, 5.5 illustrate how users rated their level of experience on using Touch-Based and Tablet devices respectively.

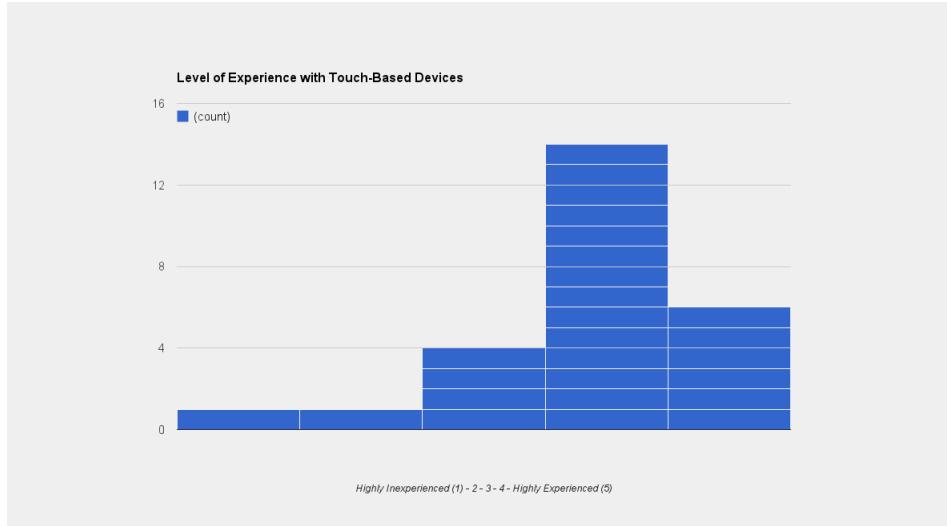


Figure 5.4: Level of Experience with Touch-Based Devices

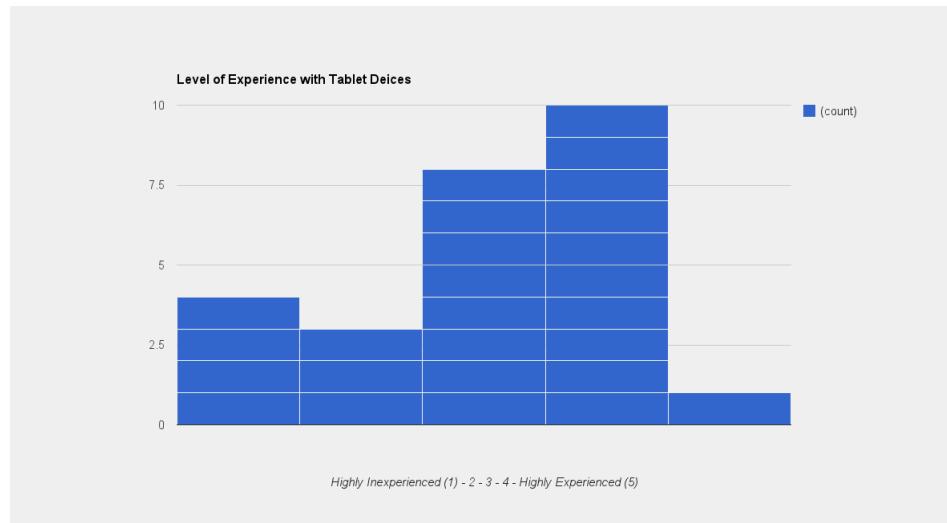


Figure 5.5: Level of Experience with Tablet Devices

In the above charts, each buckets represents a participant. The more on the right the buckets is (only 5 levels), the more experience this participant has, and vice versa. Thus, we observe that most of them were experienced using touch-based mobile devices, while fewer were high experienced addressing the tablet devices. This can also be inferred from the fact that less than half the users own a tablet device. As a result they have less experience on using one. Finally, it should be mentioned that participants received a small gift, as compensation for their time and effort.

5.3 Hypotheses

According to and based on our understanding of Fat Finger and on using contact size as the main source for input on tablet devices, we hypothesize the following:

- (**H1**). Feedback supplied trials (Discrete or Continuous) outperform No Feedback ones in terms of offset.
- (**H2**). No Feedback & Discrete outperforms No Feedback & Continuous in terms of offset.
- (**H3**). Task completion time will be gradually decreased over time.
- (**H4**). Error rates will be gradually decreased over time.
- (**H5**). Task completion time, when feedback is provided, is dependent on the number of elements.
- (**H6**). Average contact areas will be subconsciously preferred by users, as this is the natural position of the finger.
- (**H7**). Feedback & Discrete is the most preferred type of trial, as it best combines speed and accuracy.

Chapter 6

Results

This chapter presents the results we collected after performing the user study on Fat Finger interaction technique. We decided to focus on 5 basic categories, which include the most significant information we collected and analysed. We present the results for Task Completion Time, Offsets, Learning Curve, Re-Entries - Re-Touches and Subjective Preferences. We also refer to the categories of the trials with the shortages presented in Table 6.1. This will help the reader to better understand and interpret the various graphs presented onwards.

Name	Code-Name	TypeID
Feedback & Discrete	FD	1
Feedback & Continuous	FC	2
No Feedback & Discrete	NFD	3
No Feedback & Continuous	NFC	4

Table 6.1: Code names for the 4 categories of trials

6.1 Task Completion Time

Task completion time is a parameter that counts the total time the participant spent on a specific trial. As mentioned in section 4.4.1, the timer fires when the first touch is performed and stops counting when the confirming sound is being played, which indicates that we successfully hit the target. Also in Feedback trials we use the Dwell selection technique which, requires a user to stay inside the target for 1 second. In No-Feedback trials there is no such time constraint, since we use the QuickRelease method (no obligatory delay) to perform target selection.

We performed a $4 * 7$ (TypeID * N) within subjects ANalysis Of VAriance (ANOVA) [39], using the aggregated values of total time. We use ANOVA to determine if the mean values for offset are statistically different. Of course, we can export basic information about the means by just comparing them, but we want to know what do

these directional differences in the means infer about our results. ANOVA is able to determine if the differences between condition means are significant.

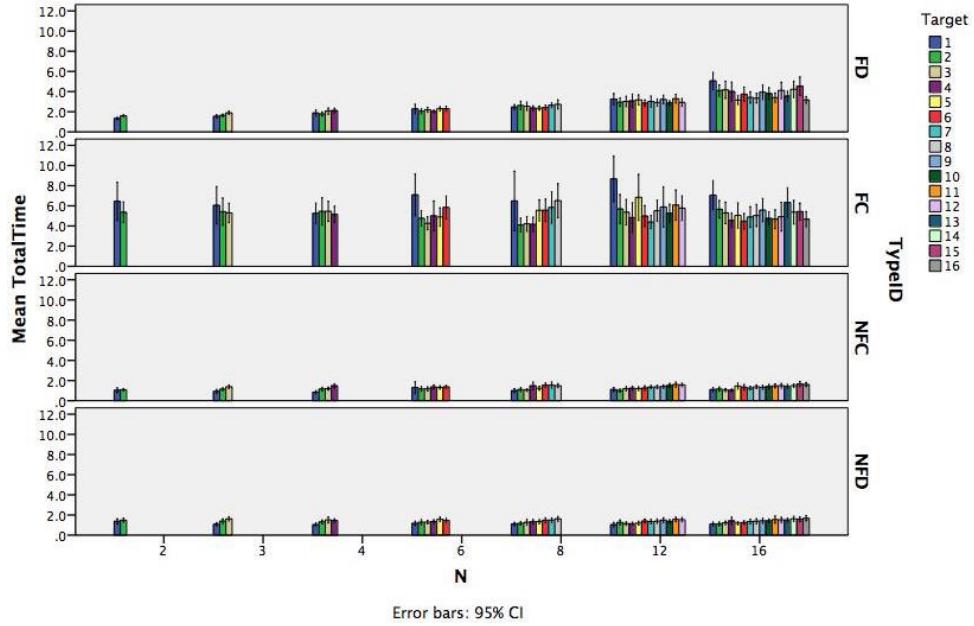


Figure 6.1: Total Time panelled by TypeID - 95% Confidence Interval

Figure 6.1 illustrates the aggregated task completion time values, panel-by typeID and cluster-by the number of buckets (N). We found that Mauchly's Test of Sphericity is violated for all Type of Trial (TypeID), number of elements (N), and also their combination (TypeID*N). That means that the type of the trial, the number of elements and their combination have a significant effect on completion time. We found that Grand Mean is 2.619 seconds (STD = $0.142s$). Feedback & Discrete ($M = 2.387s$ STD = $0.084s$) was really fast, taking into consideration that it expects $1s$ delay to confirm target selection. No Feedback & Discrete ($M = 1.237s$ and STD = $0.093s$) and No Feedback & Continuous ($M = 1.365s$ and STD = $0.013s$) were ranked almost equally and were the fastest. Feedback & Continuous ($M = 5.485$ and STD = 0.457) was the slowest one, but this can be partly explained from the fact that on the one hand, it contained the smallest targets, and on the other, the user had to pass the $1s$ inside target delay. Stay for $1s$ inside a designated small area can be really difficult under certain conditions.

Table 6.2 presents the mean differences between all possible combination of the type of Trials (TypeID). As we can see, and as already stated, Mean differences are really significant for all possible combinations.

TypeID (i)	TypeID (j)	Mean Difference (i-j)	Sig.
FD	FC	-3.098	0.000
FD	NFD	1.151	0.000
FD	NFC	1.022	0.000
FC	NFD	4.249	0.000
FC	NFC	4.120	0.000
NFD	NFC	-0.129	0.002

Table 6.2: Task Completion Time: Mean differences among the different types of Trials

Type of Elements (N), is a parameter that specifies the size of the target for Discrete trials and also the position of it, depending on which is the target bucket out of the N specified. In Continuous trials the size of each trial is independent of the number of elements (N), and is stable. However, N limits the position of the target under a specific region. Lower values for N result in a wider space range ($N=2$ and $TargetBucket=1$ means that the target can be anywhere inside the lower semicircle), while higher values significantly narrow this range ($N=16$ and $TargetBucket=1$ means that the target will be somewhere really close to the minimum value).

We found that task completion time does not statistically differ for $N = 2, 3, 4, 6, 8$. For all those different type of elements the Mean differences are not statistically different, thus we can conclude that they behave quite similarly. However, as expected, the scenario is different for 12 and 16 Elements. For $N=12$ we detect significant differences with $N=2$ (Sig. = 0.024) and $N=3$ (Sig. = 0.017), while for the rest we have no significant differences (Sig. values > 0.05). However $N=16$ differs significantly with all the aforementioned N 's, apart from $N=12$.

The combination of the number of Elements (N) and the type of Trials, in turn points out some other important results too. For FC, NFD and NFC we observe that the number of Elements does not influence that task completion time. We have very similar completion times which are independent of N. Specifically, for *Feedback & Continuous* time for all possible N's is around 5.485 seconds, for *No Feedback & Discrete* around 1.237 seconds and for *No Feedback & Continuous* around 1.365. However, interest is really triggered when we check the values for *Feedback & Discrete* targeting. The completion time is significantly increased as the number of Elements increases. We start with 1.460 s for $N=2$ and we end up with 3.858 s for $N=16$. Completion time accordingly gradually increases for all intermediate N values, which supports H5. For $N=16$ we have that bucket size is still bigger than the size of a continuous target (offset range including). We expect that for even smaller targets, as in *Feedback & Continuous*, the completion time will be even higher, as the target range is smaller. This claim indeed holds. We saw above that for FC the completion time is stable and independent on the number of elements (N), which makes sense because in FC target size does not even change. Summarizing we can conclude that:

- When feedback is provided, completion time is highly dependent on the size of

the target, increasing as the target size decreases.

- *When feedback is provided, completion time is statistically independent for N values up to 8 buckets.*
- *In No Feedback trials, completion time is independent of the size of the buckets.*

6.2 Offsets

Offset is a parameter that generally represents the error for each trial. Error is measured by the percentage we were off the target. For instance, imagine that we have a target of type Continuous, placed on x degrees, but we confirm selection at y degrees, with $x \neq y$. Then the error is the difference of x and y , $x - y$. In other words it is the distance from the point we selected to the center of the target, compared to the whole range (*360 degrees*). This difference produces a positive number if the confirmed contact size is bigger than the target's ($y > x$), and negative otherwise ($y < x$). Thus error rates can either be negative or positive depending on whether we performed selection before or after the target respectively. We also decided not to measure offset for FD trials, as there is no way that users can perform errors in those trials. Users were instructed to hit the target at whichever point inside its range, and not implicitly at the center of it. However, in No-Feedback trials the participant can confirm selection even if he is outside the target. Thus, error measurement is a much significant parameter, we need to analyse and export for those trials.

We aggregated the Offset values for each user, and with the aggregated offsets we performed a $3 * 7$ (*TypeID * N*) within subjects ANalysis Of VAriance. We found that Grand Offset Mean is *11%*, which can be translated into 39.6 degrees (*11% * 360 degrees*). It is vital to mention that grand mean is much influenced from No Feedback trial values since, Feedback & Continuous does not allow high values for error; simply because you can not select a target if you are outside the yellow lines. The range of the yellow lines is *4%*, thus errors are limited in the *+2%* region. For Feedback & Continuous trials, Mean offset is *1,1%* and STandard Deviation (STD) *0,1%*. On the other hand, offset values are significantly increased in No Feedback trials. Well interestingly, both Feedback & Discrete and No-Feedback & Continuous share the same Mean offset values: *Mean offset = 16,1% and STD=1,1%*.

Applying ANOVA and Post Hoc multiple means comparisons we found that **none** of the number of buckets (N), the type of the Trials and the combination of them have an effect on the error rate - offset. Moreover we discovered that (including all possible N values) there is no significant difference between NFD and NFC ($Sig = 1$). We can conclude that the differences between those offset Means are likely due to chance and probably not due to the type of the trials. Actually it is very interesting that NFD and NFC behave exactly the same along the experiment, which might infer that success rate of no feedback supported trials is not dependent on the size of the target. Also, as expected FC behave differently ($Sig = 0.000 \leq 0.05$) with both NFD and NFC.

Figure 6.2 illustrates the aggregated Offset values, panel-by typeID and cluster-by the number of buckets (N). Observing the graph we can extract some meaningful results:

1. **In FC error rates are always very small apart from a sudden increase they have for the last bucket of each N .** We attribute that to the fact that when the target was close to the maximum point (*360 degrees*) users used to apply full pressure, ans so reach the maximum level, than trying to achieve high precision on the target by hitting it accurately.
2. **In NFD and NFC error rates follow a common pattern.** For the first buckets of each N they are positive, and they start going gradually negative as we move to the last buckets. This can be partly explained from the fact that average contact sizes are much easier to get achieved. When our finger simply touches the screen with, it is more likely to be somewhere between the minimum and the maximum area, that close to the limits. This is the natural position of our finger when are simply touching the screen.

We realize then, that participants tend to go closer to this normalized area. They seem to have better control there, and more comfort too. This is probably why we observe that pattern on the graphs. When target are really close to the limits we tend to select areas that are closer to the normalized region than the correct ones.

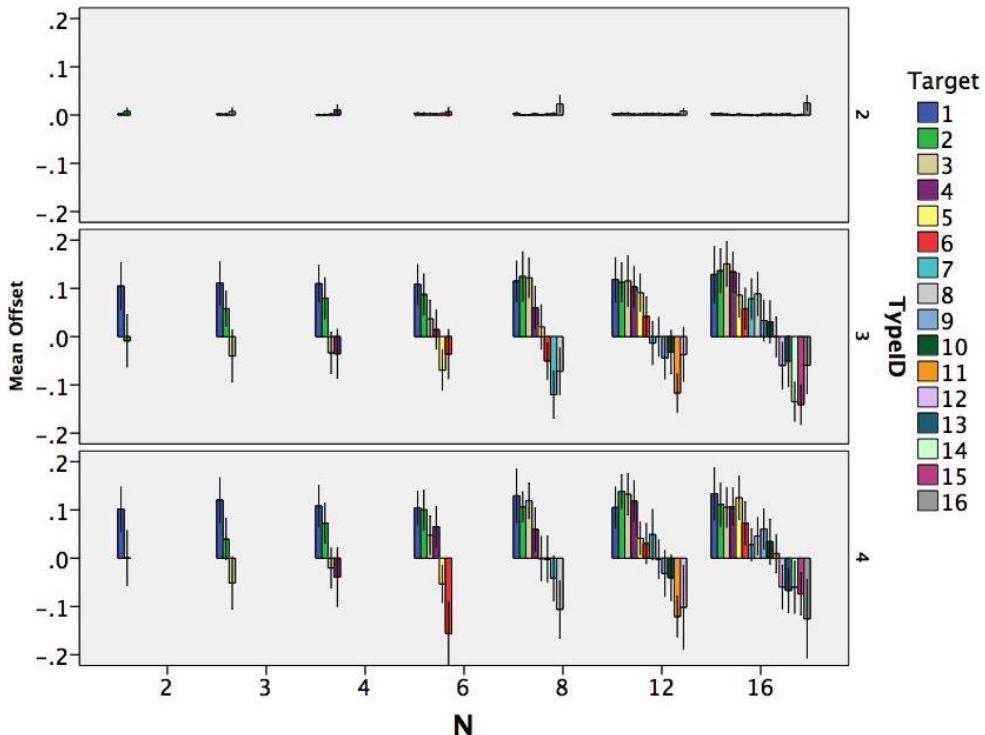


Figure 6.2: Offset aggregated - 95% Confidence Interval

After observing the error rates, it would be also nice to take a look on the No Feedback

trials success rate. Table 6.3 presents the percentage of successful selections for both No Feedback & Discrete and No Feedback & Continuous. For Discrete targets the successful trials (33%) are much more than for the Continuous targets (11%). However they share the same error rates as mentioned above. How is that even possible?

Well, for Discrete targets we count the offset from the middle of each target. For instance, if $N = 2$ then the middle of the first bucket (down semicircle) is at *90 degrees*. Even if the offset is 25% (= 90*degrees*) then we are still inside the target, at the edge of it though. The same applies for higher N values, but the 'permitted' offset is then narrower. Summarizing:

*Offset values for Discrete targeting are not always considered as errors, as we measure offset from the center of the targets range. The real error can be calculated by the type:
 $RealErrorDiscrete = OffsetValue - (TargetRange * 0.5)$*

	Successful	Unsuccessful
NFD	33.76%	66.16%
NFC	11.25%	88.75%

Table 6.3: No Feedback: Percentage of successful hits

6.3 Learning Curve

In section 4.3 we mentioned that training phase is not isolated from the rest of the experiment. Our approach is to use to use a 3-repetition design (Figure 4.9), in which the first repetition will also act as the Learning phase, while the remaining 2 would just be the normal experiment. This way we are not eliminating learning from performing phase and most importantly, this give us the possibility to monitor the way and the pace the participants are getting used to this new interaction technique and observe the improvements of performance over time. In other words, this design provide us the appropriate tools for building the Learning Curve of Fat Finger.

6.3.1 Total Time

We hypothesized (H3) that the average total time needed for the completion of each trial will be decreased over time. In other words we hypothesized that:

$$AverageTimeRepetition1 > AverageTimeRepetition2 > AverageTimeRepetition3$$

Figure 6.3a presents the average completion time learning curve in each repetition. We can observe that for repetition *No 1* the mean time is just above *3 seconds*. In Repetitions *No 2* and *No 3* the time decreased down to around *2.5 seconds*. Thus we have a learning factor over time, especially between repetition *No 1* and repetition *No 2*. Repetition *3* whilst not obviously improving the average trial time, is still contributing in the Learning effect in regards of specific trials; as I will thoroughly

explain onwards.

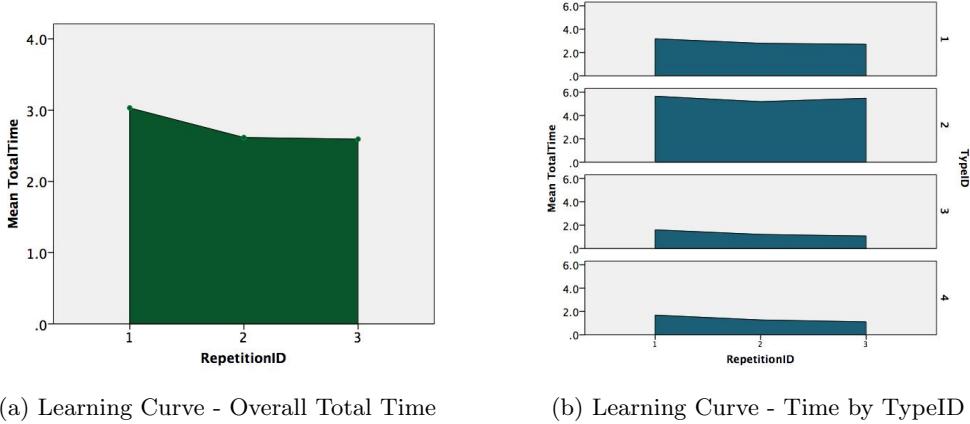


Figure 6.3: Learning Curve - Task Completion Time

Figure 6.3b presents the average completion time learning curve in each repetition, panelled by Trial category ($TypeID$). Learning curve for NFD ($TypeID = 3$) and NFC ($TypeID = 4$) depicts a continuous improvement of the average completion time among repetitions. However, the error rates learning curve as we point out in section 6.3.2, have a completely reversed behaviour over time. Feedback & Discrete learning curve follows the same behaviour as the overall time learning curve in 6.3a. On the other hand, Feedback & continuous is the only type of trials for which, we do not observe continuous improvement over time. However, whilst being slower his error rates constantly decrease over time (section 6.3.2). Thus, higher time values does not necessarily mean that our performance is being degraded.

Taking into account the aforementioned graphs, we realize that there exist obvious learning improvements over time. However while for repetition *No 1* to repetition *No 2* we do observe significant learning factors, this is not that obvious for repetition *No 3*. Mental and physical fatigue might have a share in this behaviour. While we do not have any measurement for fatigue to validate this claim, we can however extract some meaningful results from the assessments participants provided us at the end of each experiment. A short conclusion would be: if it was not for fatigue (mental and physical), learning curve might have been further decreased on repetition 3.

6.3.2 Offset

In Figure 6.4 we can observe the behaviour of Error rates along the repetitions, and use that knowledge to extract results regarding the error rates learning curve for all Feedback & Continuous (blue), No Feedback & Discrete (green) and No Feedback & Continuous (red) trials.

As we can notice, there is no significant improvement of the error rates over time, neither for FC (1,1% - 1,1% - 1%), nor for NFC (16,3% - 15,9% - 16%). The differences

are on the 0,1% scale which is not even noticeable. However in No Feedback & Discrete trials (green) we do observe a drop of 2% on the second repetition, but a slight increase (0,5%) on the third one. According to the aforementioned values, we realize that error rates are not subject to the same learning factor we experienced for Task Completion time in section 6.3.1. However, the fact that we did not meet increasing values along the repetitions, is very promising. It infers that users are capable to maintain the same error rates, which are relatively low, along by decreasing their competition time.

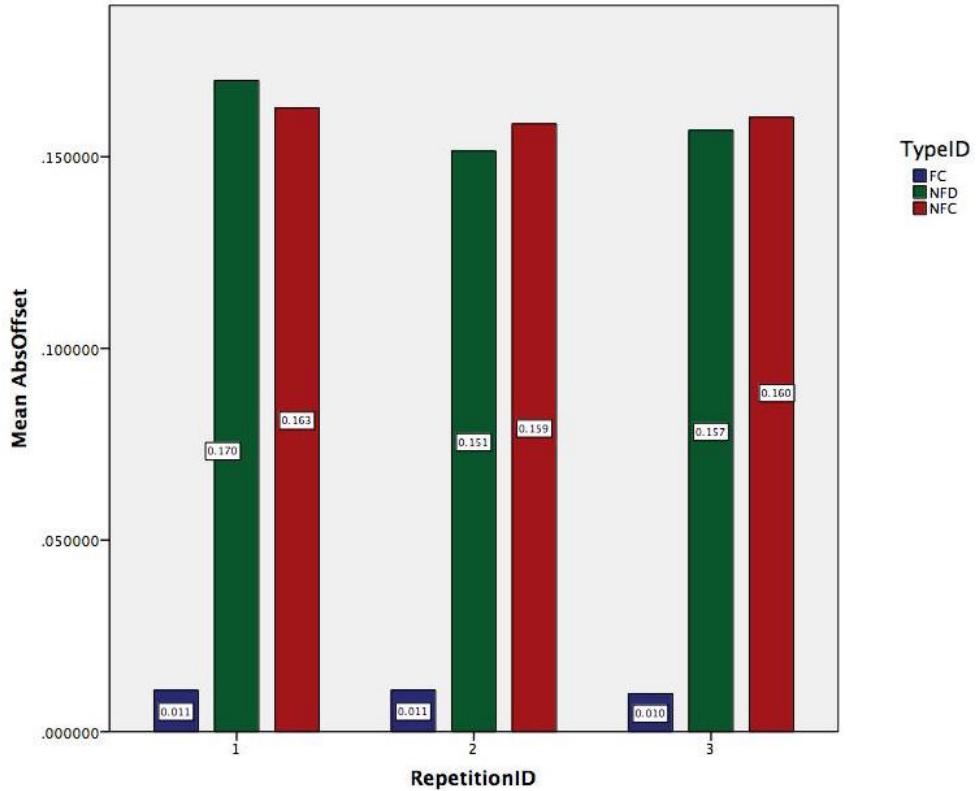


Figure 6.4: Learning Curve - Offset

One important thing to note is, that certain values for error are affordable in No Feedback trials. Especially for No Feedback & Discrete Targeting that maximum affordable error is presented on Table 6.4. It can be calculated using the type:

$$AffordableError = \left(\frac{100}{N} = BucketSize \right) * \frac{1}{2}$$

Affordable error is half of the size of its bucket for a specific N . Now, by combining the data from Table 6.4 and Figure 6.4 we can extract even more meaningful and interesting results. With the given error rates for NFD, we find that for $N = 2$ or $N = 3$, we will always perform selection inside the target area. That happens because the average error rates are lower than the affordable error, meaning that this error is indeed acceptable and thus we successfully performed an in-target selection. We also realize that hitting smaller targets ($N > 4$), is not trivial in No Feedback trials since average error rates are much higher compared to the maximum affordable error rates

for those N values.

N	Maximum affordable error in %
2	25,00%
3	16,66%
4	12,50%
6	08,33%
8	06,25%
12	04,16%
16	03,12%

Table 6.4: Fat Finger - No Feedback & Discrete Targeting affordable error

6.4 Re-Entries

Analysis of the target Re-Entries that occurred during the whole experiment will help us to better understand where users faced difficulties, caused by tremolo or general stabilization parameters. We meet higher values for target re-entries, when the target's range is smaller. As we will see onwards, when the finger of participant was not completely accurate and stable, the tremolo produced was enough, to cause small and usually fast movement of the feedback region. If the feedback edge was close enough to the edge of the target, then this slight movement ("in" and "out" of the target region) was encountered as 1 re-entry.

Figure 6.5 presents the average target re-entries for each of the trial categories and all possible N's. **Feedback & Discrete** has almost zero re-entries per trial especially for low values of N. When N increases re-entries follow along and we reach a maximum of around 1.5 target re-entries per trial ($N=16$). This enhances and further supports the claim we made before about the connection between target size and target re-entries. **Feedback & Continuous** have almost the same number of target re-entries as the size of the target is independent of the value of N . The range of the target is even smaller than the FD one of $N=16$, so the target re-entries per trial are even higher than the maximum value for Feedback & Discrete targeting. We have an average of around 2 target re-entries for all Feedback & Continuous trials. **No Feedback & Discrete** and **No Feedback & Continuous** have almost the same reaction regarding target re-entries. Since users are not aware of the position of the feedback region, and they need to stay inside the target for a specific time; they just move their finger in the desired position and then simply lift it. Thus, movements such as trembling around a specific area for a long time to stay accurate did not encountered. As a result target re-entries are too low; 0.5 per trial or differently expressed, one every two trials. So for NF trials, target re-entries are independent both from the target's size and from the of trial (NFD or NFC).

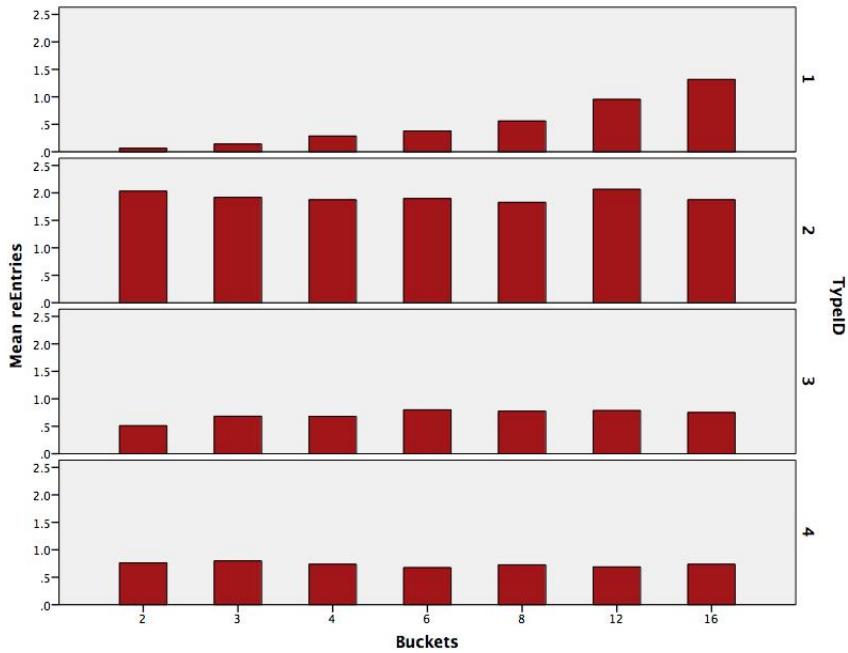


Figure 6.5: Target Re-Entries panelled by Type of Trial

6.5 Re-Touches

Re-touches is a parameter which is meaningful only for Feedback trials. It measures the times users re-positioned their finger by lifting it during one trial. Re-touches are not measured for No Feedback trials, because in these trials lifting is the way to perform target selection. So, users are not allowed to reposition their finger by lifting it from the screen, because that way they will falsely select the target. As Figure 6.6 illustrates, Feedback & Discrete retouches are really low, with an average of around *0.6* per trial. However, for Feedback & Continuous targeting we have a completely different pattern. Re-touches are a lot higher (around *2.0* per trial) with small diversities for *N=2* and *N=12*. However, I point out two significant observations that were extracted from the comments section of the assessment participants filled in after the experiment.

1. Users had many difficulties to hit continuous targets that were really close to the minimum point. Because the red line is randomly positioned within each bucket, in case it is really close to the minimum point, selection of the target becomes a cumbersome task. They usually had to perform several re-touches and re-entries to be able to select the target.
2. The above behaviour was also encountered when the target was close to the maximum area but not exactly on it. Then participants faced the same difficulties, to avoid tremolo and stabilize their finger.

We conclude that for targets placed really close to the minimum or the maximum calibrated ones, the ability to control the feedback line is significantly decreased. This is probably not happening due to software reasons, but possible because of other ergonomic issues of the finger.

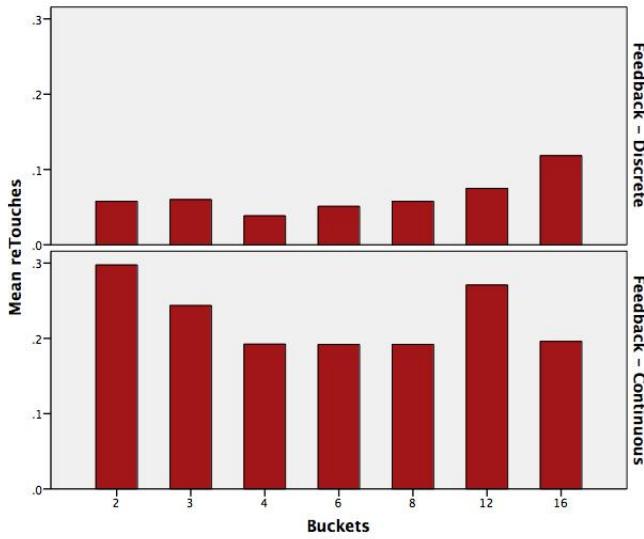


Figure 6.6: Target Re-Touches for Feedback Trials

6.6 Subjective Preferences

In Figure 6.7, the mean values for all the parts of the assessment presented in section 5.1.4 are illustrated and presented. On each diagram the vertical axis represents the mean values of the corresponding category, which are ranged from 1 to 5 (5 point Likert scale). Generally, higher values mean that participants positively rated the specified category. The horizontal axis corresponds to the type of Trial. Each graph has 4 columns, identified with the numbers 1-4, which correspond to the respective type of trials.

FD(1) scored consistently well across all the above categories. It scored mean values that were always improved than the ones of other trials. Precisely, Feedback & Discrete Targeting (1) was ranked higher concerning the Overall Performance ($mean = 3.46 / 5$), Figure 6.7f. It can be observed that overall performance is decreased as the identification of the type of trial is increasing. The same behavior is also noticed for Mean Smoothness ($mean = 4.03 / 5$), Figure 6.7a. However Most importantly, all four techniques ranked amazingly concerning the Fatigue of both the finger and the wrist. Even being an extensive and long experiment, which might not well correspond to everyday use scenario, the fatigue participants experienced was in extremely low levels. Finger fatigue had $mean = 1.86$, as shown in Figure 6.7c, and Wrist fatigue also had $mean = 1.58$, as shown in Figure 6.7d. Concerning Speed ($mean = 2.82$), Figure 6.7b, all techniques ranged quite uniformly, outlining that the operational speed of the experiment and of each technique in particular was mostly normal. Finally, Smooth-

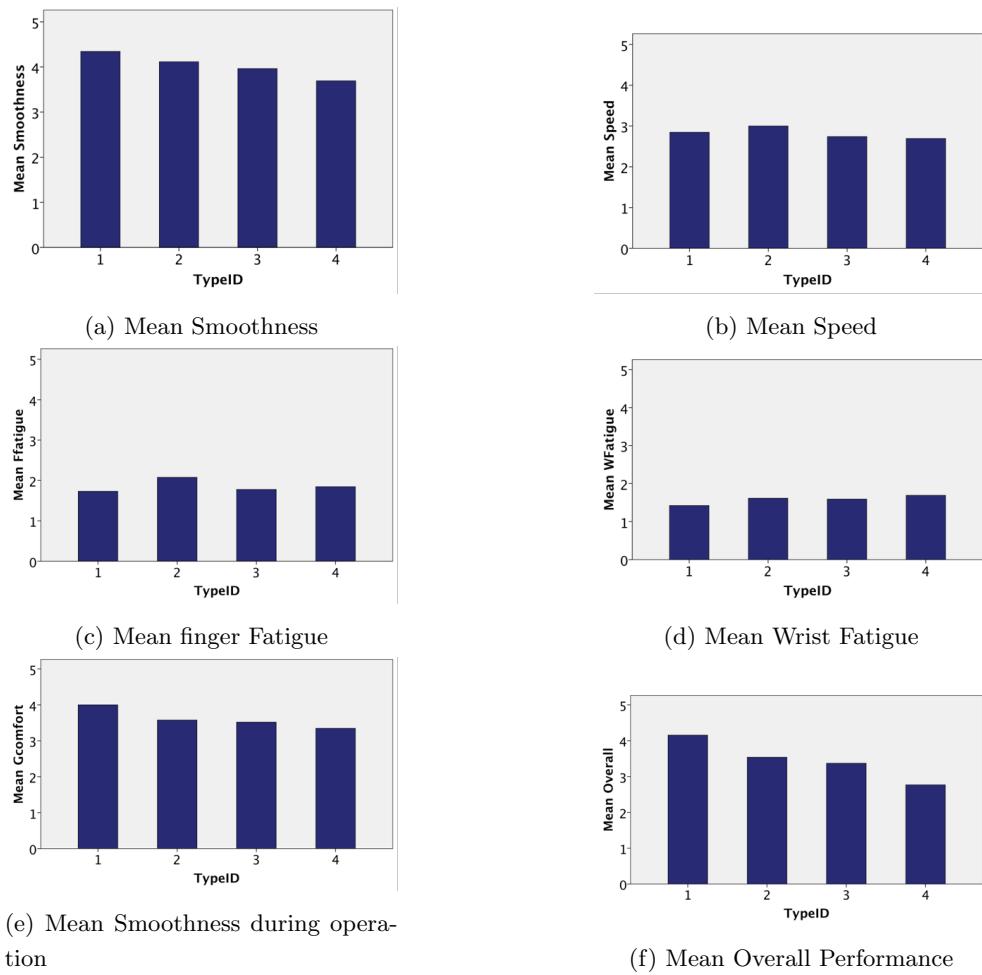


Figure 6.7: Mean Values assessed by participants for each trial category

ness ($mean = 4.03 / 5$), Figure 6.7f, ranged also very high for all techniques, which proves that our application is well structured and well corresponds to the movement of the finger.

Chapter 7

Discussion

Conducting a study, without commenting and discussing on the results produced, does not make much sense. On Chapter 5 (section 5.3) we stated the hypotheses we made regarding this study. Then, on Chapter 6 we presented, analysed and commented on all collected results, after separating them in six groups (Task Completion Time, Offsets, etc.). Now, we need to provide the necessary infrastructure to glue them together. That can be done by commenting and discussing if the hypotheses hold, taking into account the results we calculated. This is what this chapter is about; discuss, comment and reason whether the hypotheses stated, hold or not. We commence by restating each of the hypotheses, along with discussing on their validity.

- **(H1). Feedback supplied trials (Discrete or Continuous) outperform No Feedback ones in terms of offset.**

In section 6.2, we mentioned that offset represents the error for each trial. The offset represents the distance (in percentage) between the point we selected and the center of the target. While No Feedback trials have no limitations on the value of the error, in Feedback & Continuous trials the maximum allowed offset is $\pm 2\%$. In chapter 6 we noted that, in No Feedback trials error rates were huge compared to the FeedBack Trials (only Feedback & Continuous being considered). This was expected as in No Feedback trials, users have no indication on where is the feedback line; being also free, due to the QuickRelease technique, to lift their finger at whichever point (even if they are outside the target). Their ability to develop memory and predict correctly, was not enough to outperform Feedback trials. We can then conclude that, H1 holds.

- **(H2). No Feedback & Discrete outperforms No Feedback & Continuous in terms of offset.**

In section 6.2 we mention that: *Offset values for Discrete targeting are not always considered as errors, as we measure offset from the center of the targets range. The real error can be calculated by the type: $RealErrorDiscrete = |OffsetValue| - (TargetRange * 0.5)$*

On the results section, we noticed, much of our interest, that both NFD and NFC share the same error rates. Taking into account the aforementioned statement, that the real error in NFD trials is always smaller than offset we are provided with, we indeed realize that the real error in NFD trials is smaller than in NFC ones. As a result, H2 holds.

- **(H3). Task completion time will be gradually decreased over time.**

In other words, our hypothesis states that:

$$\text{AverageTimeRep1} > \text{AverageTimeRep2} > \text{AverageTimeRep3}$$

In section 6.3.1 we presented and analysed the Task Completion Time Learning Curve. Learning curve is directly connected and related with the task completion time improvements over time. Taking into account the graphs presented in figure 6.3, we realize that there exist obvious learning improvements from repetition 1 to repetition 3, thus task completion time gradually decreases over time. The only type of trials that we do not observe continuous improvement over time, is Feedback & Continuous, for which in the 3rd repetition we have a slight increase of the average task completion time. As we also mentioned in section 6.3.1, mental and physical fatigue might have a share in this behaviour, concluding that if it was not for fatigue (mental and physical), task completion time might have been further decreased on repetition 3. Taking everything into consideration we can argue that H3 holds, even after experiencing that minor increase of FC on the 3rd repetition.

- **(H4). Error rates will be gradually decreased over time.**

In section 6.3.2, Figure 6.4, we observed that for all FC, NFD, NFC trials there exists a minor but almost unobservable error improvement over time, except probably for NFD. We saw that error rates slightly dropped on the second repetition, and remained stable or decreased in some cases on the third one. We can then conclude that, H4 is supported from the results, and thus it holds in general.

- **(H5). Task competition time, when feedback is provided, is dependent on the number of elements.**

In section 6.1 we made the following observations. All type of trials concerned, task completion time does not statistically differ for $N = 2, 3, 4, 6, 8$. However we do observe a difference for $N = 12$ and $N = 16$. However, addressing Feedback trials only, the scenery is alternated. For instance, for Feedback & Discrete, completion time increases as the number of Elements increases. we observe 1.460s for $N = 2$, ending up with 3.858 s for $N = 16$. As we have already mentioned, in Discrete trials, the number of elements (N) is highly related with the size of the target. However this does not hold for Continuous trials since there, N specifies the range of target's position. The size of a continuous target is fixed, and smaller than $N = 16$ targets for Discrete trials. We indeed observed that in FC, completion time is stable and independent on the number of elements (N), which makes sense exactly because in FC target size does not even change. Thus for FC trials, time is independent on the number of the

elements, but is dependent of the size of the elements. Thus H5 holds only for FD trials, while if we modify it a bit it can also hold for FC trials. A modified version of H5 would be:

Task competition time, when feedback is provided, is dependent on the size of the target.

- **(H6).** *Average contact areas will be subconsciously preferred by users, as this is the natural position of the finger.*

If we observe Figure 6.2 in section 6.2 we notice that in No Feedback trials, offset rates follow a common pattern. We experienced high positive offsets for targets located close to minimum contact area, almost zero offset when the target was located closed to the average contact area, and high negative offsets when target was located close to the maximum contact area. That, in other words means that, participants when not limited by the feedback line indicator, tend to subconsciously prefer and select contact areas that are closer to the average - median contact area. However when feedback is provided, they are forced by the application to precisely select the predefined target. That makes it impossible to study what H6 hypothesizes when we deal with Feedback trials. We then conclude, that (when Feedback is not provided) H6 holds.

- **(H7).** *Feedback & Discrete is the most preferred type of trial, as it best combines speed and accuracy.*

In the additional assessment users provided us with (Appendix C), we asked them to put the 4 different techniques in ascending order depending on which was the easiest to use. Users dominantly preferred Feedback&Discrete trials (as the easiest to use), with Feedback&Continuous, No Feedback&Discrete and Feedback&Continuous following on the line. This definitely depicts user preferences, and if we also observe the results we got from the offset and completion time parameters, we realize that Feedback&Discrete outperforms all other techniques regarding all measured parameters. Even for Task Completion time, in which due to the Dwell technique it requires an 1s delay to confirm target selection, the results are comparable to No Feedback trials, which in turn do not require any extra delays. As a result we can positively support that H7 holds.

Taking into consideration the above discussed hypotheses, we perceive that all of them hold, with minor modifications at certain cases. We note that *Feedback & Discrete* seems to be the most effective out of all types of trials, having comparable task completion time with No Feedback trials, almost overpassing the 1s obligatory delay obstacle. As discussed in section 6.1, task completion time does not statistically differ for N = 2, 3, 4, 6, 8. Thus we can conclude that users are able to distinguish up to 8 different discrete pressure levels when feedback is supplied, without any statistical significant difference on the task completion time. H7 further supports our claim, stating that FD is the most preferred selection technique, which also combines the best values for accuracy with a correspondent speed.

We then observed that completion time for No Feedback trials is independent on the number of elements, N . Thus, performance can be mainly measured through the error rates. In section 6.3.2 we mentioned and commented on the error rates and the corresponding learning factor. We concluded that a level of 3, possibly 4, discrete levels is in turn identifiable and distinguishable by users, when feedback is not provided. H6 reveals that average contact areas are subconsciously preferred by users. This mainly happens because the natural position of the finger on the screen, requires an average contact size. This tendency is a significant parameter which contributes to the ability of No feedback trials to distinguish less discrete pressure levels; excluding the non-presence of the feedback region, which is the main and most important one. We finally saw that the ability of a person to develop haptic memory to precisely select a target when N is very high, is limited. This further explains why the performance on No Feedback trials is decreased compared to the Feedback ones.

Chapter 8

Conclusion

In this study we presented Fat Finger, an alternative interaction technique that uses the finger's contact size as the main source of input on a mobile or tablet device. The main objective of this study was to investigate how many distinguishable contact size levels can be actually achieved and perceived by users. In Chapter 1 we introduced you to the current environment and developments on the market, explaining the limitations of the 2D interaction techniques, used to interact with the current smartphones and tablet devices. Then we broached you to the 3D interaction techniques, whose principle is to use the contact size of the finger touching the screen, in addition to its position on the screen. In Chapter 2 we presented all the relevant work and researches that have been already conducted on the same or highly related fields. We separated it in pressure-based and contact-shape-based approaches, thoroughly explaining similarities and differences with Fat Finger.

In Chapter 3 we presented the original idea behind Fat Finger interaction technique, along with the way we conceptualized its usage. Finally we raised the basic research question that this study will respond to, which is: *"To which extend are we able to distinguish the different simulated pressure levels produced by our fingers using a tablet device?"* The overall concept is that we want to investigate the capabilities and precision of using the contact area of the index finger to interact with a tablet device.

Chapter 4 presents the interface the structure and the design of the application we developed to test and study Fat Finger interaction technique; it is in the form of discrete target selection trials, runs on an Apple iPad device, and consists of many consequent trials, each of them requiring the user to perform a target selection task. First we explain the basic work-flow we meet once we launch the application followed by the basic design and interface of a simple trial. Then we analyse the 4 categories of trials: Feedback & Discrete, Feedback & Continuous, No Feedback & Discrete and No Feedback & Continuous. Finally, we explain the final sequence of the trials in the user study and the procedure and methods we used to monitor and measure user performance. In Chapter 5 we continue by analysing the procedure and the context of the user study. We had 26 participants taking part in this study. Each user had to go

through the verbal instructions, fill in the Demographic information form, calibrate his finger, accomplish the six hundred and twelve trials of the experiment, and finally fill in an assessment for each of the 4 different type of trials. We then gave statistical information regarding the demographic information of the participants that took place in our study.

Finally, in Chapter 6 we presented and discussed the computed results for all the basic parameters: Task completion time, Offset, Task Completion Time Learning Curve, Offset Learning Curve, Re-Entries and Re-Touches. Discussion on the results and arguing on the validity of the hypotheses stated in Chapter 5, are all performed in Chapter 7. We conclude that for Feedback supplied trials a level of 8 distinguishable contact size levels is easily perceived by users, while for No-Feedback ones we meet tolerated error rates only until 3 or 4 contact size levels. Also, we gave evidence and supported that all of the hypotheses we have set, hold.

While this study focuses on understanding touch and contact size on mobile devices, much work has to be done in order to produce real life products that will make use of this interaction technique. It will require the redesign of current interfaces, to make the cooperation with this technique feasible. But prior to redesigning the interface, we need to find the most appropriate way(s) to interact and exploit these multiple level contact size capabilities.

Thus next steps and future work will include the seek for a way to integrate this technique into on-market products. Fat Finger is capable to become both a sufficient alternative to the current gestured based approaches and also a much intuitive way to perform operations that are infeasible at the moment. One immediate way to integrate interaction through contact size with mobile interfaces is probably feasible by allowing the control of continuous variables through the variation of the contact size. For instance, volume, brightness, exposure etc. will now be controlled without the usage of a bar, which will not occupy any on-screen space. As a result, interfaces will be become simpler and more intuitive. Another approach would be to use fingers to perform precise operation for which, currently, only pens are used. For the shake of illustration, S Pen is used by Samsung to perform operations that need augmented precision and detail. Is it possible, using Fat Finger, to achieve the same kind of detail through an interface which will provide us with the necessary infrastructure? Finally, another suggestion is to study how FatFinger can be applied and integrated with multi-finger interaction. What is the impact on the ability to perceive the contact size levels, when multiple finger are used? What are the possible ways to combine current gestures with Fat Finger interaction technique?

Bibliography

- [1] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. Pseudo-pressure detection and its use in predictive text entry on touchscreens. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, OzCHI '13*, pages 383–392, New York, NY, USA, 2013. ACM.
- [2] Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '06*, pages 1263–1272, New York, NY, USA, 2006. ACM.
- [3] David Bonnet, Caroline Appert, and M Beaudouin-Lafon. Extending the Vocabulary of Touch Events with ThumbRock. *perspective*, 2013.
- [4] Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. The fat thumb: Using the thumb's contact size for single-handed mobile interaction. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services, MobileHCI '12*, pages 39–48, New York, NY, USA, 2012. ACM.
- [5] Stephen A. Brewster and Michael Hughes. Pressure-based text entry for mobile devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '09*, pages 9:1–9:4, New York, NY, USA, 2009. ACM.
- [6] William Buxton, Ralph Hill, and Peter Rowley. Issues and techniques in touch-sensitive tablet input. *SIGGRAPH Comput. Graph.*, 19(3):215–224, July 1985.
- [7] Wilson A. Balakrishnan R. Hinckley K. Cao, X. and Hudson. Shapetouch: Leveraging contact shape on interactive surfaces. Proc. TABLETOP, IEEE, 2008.
- [8] Patel S. Pierce J. Clarkson, E.C. and Abowd. Exploring continuous pressure input for mobile phones. GVU Techreport. GIT-GVU-06-20, 2006.
- [9] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. Gripsense: Using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, UIST '12*, pages 545–554, New York, NY, USA, 2012. ACM.

- [10] Taedong Goh and Sang Woo Kim. Eyes-free text entry interface based on contact area for people with visual impairment. In *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST’14 Adjunct, pages 69–70, New York, NY, USA, 2014. ACM.
- [11] Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST ’05, pages 115–118, New York, NY, USA, 2005. ACM.
- [12] Seongkook Heo and Geehyuk Lee. Forcetap: Extending the input vocabulary of mobile touch screens by adding tap gestures. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI ’11, pages 113–122, New York, NY, USA, 2011. ACM.
- [13] Christopher F. Herot and Guy Weinzapfel. One-point touch input of vector information for computer displays. *SIGGRAPH Comput. Graph.*, 12(3):210–216, August 1978.
- [14] Christian Holz and Patrick Baudisch. Understanding touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’11, pages 2501–2510, New York, NY, USA, 2011. ACM.
- [15] Jonggi Hong and Geehyuk Lee. Touchshield: A virtual control for stable grip of a smartphone using the thumb. In *CHI ’13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’13, pages 1305–1310, New York, NY, USA, 2013. ACM.
- [16] Sungjae Hwang, Andrea Bianchi, and Kwang-yun Wohn. Vibpress: Estimating pressure input using vibration absorption on mobile devices. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI ’13, pages 31–34, New York, NY, USA, 2013. ACM.
- [17] Sungjae Hwang, Andrea Bianchi, and Kwangyun Wohn. Micpen: Pressure-sensitive pen interaction using microphone with standard touchscreen. In *CHI ’12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’12, pages 1847–1852, New York, NY, USA, 2012. ACM.
- [18] Sungjae Hwang and Kwang-yun Wohn. Pseudobutton: Enabling pressure-sensitive interaction by repurposing microphone on mobile device. In *CHI ’12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’12, pages 1565–1570, New York, NY, USA, 2012. ACM.
- [19] Apple Touch ID. <https://www.apple.com/iphone-6/touch-id/>.
- [20] Core Data iOS. http://en.wikipedia.org/wiki/Core_Data.
- [21] Apple iPhone 1st generation. [http://en.wikipedia.org/wiki/IPhone_\(1st_generation\)](http://en.wikipedia.org/wiki/IPhone_(1st_generation)).

- [22] Sven Kratz, Patrick Chiu, and Maribeth Back. Pointpose: Finger pose estimation for touch input on mobile devices using a depth sensor. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 223–230, New York, NY, USA, 2013. ACM.
- [23] Swift Programming Language. [http://en.wikipedia.org/wiki/Swift_\(programming_language\)](http://en.wikipedia.org/wiki/Swift_(programming_language)).
- [24] SK Lee, William Buxton, and K. C. Smith. A multi-touch three dimensional touch-sensitive tablet. *SIGCHI Bull.*, 16(4):21–25, April 1985.
- [25] Wing Ho Andy Li and Hongbo Fu. Bezelcursor: Bezel-initiated cursor for one-handed target acquisition on mobile touch screens. In *SIGGRAPH Asia 2013 Symposium on Mobile Graphics and Interactive Applications*, SA '13, pages 36:1–36:1, New York, NY, USA, 2013. ACM.
- [26] Markus Löchtefeld, Christoph Hirtz, and Sven Gehring. Evaluation of hybrid front- and back-of-device interaction on mobile devices. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, MUM '13, pages 17:1–17:4, New York, NY, USA, 2013. ACM.
- [27] Suzanne Low, Yuta Sugiura, Dixon Lo, and Masahiko Inami. Pressure detection on mobile phone by camera and flash. In *Proceedings of the 5th Augmented Human International Conference*, AH '14, pages 11:1–11:4, New York, NY, USA, 2014. ACM.
- [28] David C. McCallum, Edward Mak, Pourang Irani, and Sriram Subramanian. Pressuretext: Pressure input for mobile phone text entry. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 4519–4524, New York, NY, USA, 2009. ACM.
- [29] Takashi Miyaki and Jun Rekimoto. Graspzoom: Zooming and scrolling control model for single-handed mobile interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 11:1–11:4, New York, NY, USA, 2009. ACM.
- [30] Multi-Touch. <http://en.wikipedia.org/wiki/Multi-touch>.
- [31] QWERTY. <http://en.wikipedia.org/wiki/QWERTY>.
- [32] Gonzalo Ramos and Ravin Balakrishnan. Fluid interaction techniques for the control and annotation of digital video. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 105–114, New York, NY, USA, 2003. ACM.
- [33] Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. Pressure Widgets. *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*, 6(1):487–494, 2004.

- [34] Simon Rogers, John Williamson, Craig Stewart, and Roderick Murray-Smith. Anglepose: Robust, precise capacitive touch tracking via 3d orientation estimation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2575–2584, New York, NY, USA, 2011. ACM.
- [35] Anne Roudaut, Eric Lecolinet, and Yves Guiard. Microrolls: Expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 927–936, New York, NY, USA, 2009. ACM.
- [36] Jeongmin Son and Geohyuk Lee. Fingerskate: Making multi-touch operations less constrained and more continuous. In *Proceedings of the Adjunct Publication of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13 Adjunct, pages 107–108, New York, NY, USA, 2013. ACM.
- [37] Martin Spindler, Martin Schuessler, Marcel Martsch, and Raimund Dachselt. Pinch-drag-flick vs. spatial input: Rethinking zoom & pan on mobile displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1113–1122, New York, NY, USA, 2014. ACM.
- [38] Craig Stewart, Michael Rohs, Sven Kratz, and Georg Essl. Characteristics of pressure-based input for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 801–810, New York, NY, USA, 2010. ACM.
- [39] ANOVA ANalysis Of VAriance. http://en.wikipedia.org/wiki/Analysis_of_variance.
- [40] Feng Wang, Xiang Cao, Xiangshi Ren, and Pourang Irani. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, pages 23–32, New York, NY, USA, 2009. ACM.
- [41] World Wide Web. http://en.wikipedia.org/wiki/World_Wide_Web.

Appendix A

Verbal Instructions of Experiment

The following description will be read to each participant at the beginning of the study to inform participants of the procedure prior starting the experiment.

Introduction Fat Finger

Hello and welcome to the iLab, my name is <experimenter>, and I will guide you through the experiment. Feel free to ask me any question at any time.

Before we start I need to let you know about your rights as a participant.

- If you feel uncomfortable you may quit at any time. The data that we have collected up to that point may be kept if you withdraw before the end of the experiment.

Before we go on with the instructions of the experiment, I would like to let you know that we appreciate your helping us in this study.

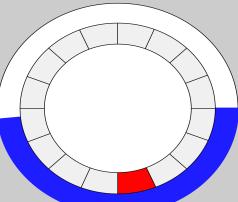
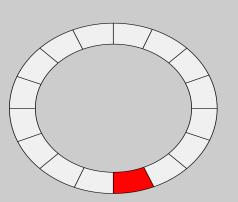
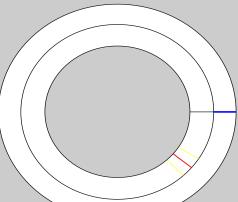
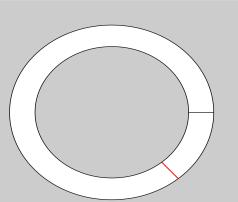
We aim to understand how we can use the index finger's contact size as additional parameter for input besides just its pure location on the device's display. This can be used to enhance the current ways of interacting with a mobile or tablet device, as it is capable of providing us more degrees of freedom when we interact with a display.

Now please read and fill in the Demographic information form. This experiment is anonymous so we will keep no personal information.

<Hand in Demographic information form>

<Continue on next page>

Now I will guide you through the experiment, and start by explaining the different type of trials it consists of.

Targeting	Feedback	No Feedback
Confirm Target	Stay inside target for at least 1 sec	Lift finger from screen
Discrete		
Continuous		

Those 4 types of trials, will occur multiple times with different variable difficulties. Your target is to hit the targets as accurate and as fast as possible. If for any reason you experience finger or wrist fatigue, or you just need a break, you are allowed to do so. Breaks can only occur when the "Start Next Trial" is on screen.

One last but important thing is that you should perform an one-time calibration of your index finger. When prompted, touch the screen with the index finger of your dominant hand. Move it in a way to achieve a minimum and a maximum contact area with the screen. You may visualize your input and calibrate again if necessary.

Remember that you are not allowed to change finger or hands during operation. You may use the index finger of your dominant hand, or the hand you feel more comfortable.

<Start of experiment>

Appendix B

Demographic Information Form

Simulated Pressure Mobile Interaction Participant ID _____

Demographic Information & Device Assessment

Demographic Information

Age:		Gender:		Optical Aid:		Handedness:	
------	--	---------	--	--------------	--	-------------	--

Do you own a touch-based mobile device? Yes No

Have you used a touch-based mobile device before? Yes No

Do you own a tablet device (e.g. iPad)? Yes No

Have you used a tablet device before? Yes No

How do you rate your experience with touch-based mobile devices?

Highly inexperienced Highly experienced

How do you rate your experience with tablet devices?

Highly inexperienced Highly experienced

Appendix C

Technique Assessment Form

Feedback & Discrete Targeting

1. Smoothness during operation was:

Very rough Very smooth

2. Operation speed was:

Too fast Too slow

3. Finger fatigue:

None Very high

4. Wrist fatigue:

None Very high

5. General comfort:

Very uncomfortable Very comfortable

6. Overall, the input method was:

Very difficult to use Very easy to use

Please state your own comments (advantages / disadvantages) below:

Feedback & Continuous Targeting

1. Smoothness during operation was:

Very rough Very smooth

2. Operation speed was:

Too fast Too slow

3. Finger fatigue:

None Very high

4. Wrist fatigue:

None Very high

5. General comfort:

Very uncomfortable Very comfortable

6. Overall, the input method was:

Very difficult to use Very easy to use

Please state your own comments (advantages / disadvantages) below:

No Feedback & Discrete Targeting

1. Smoothness during operation was:

Very rough Very smooth

2. Operation speed was:

Too fast Too slow

3. Finger fatigue:

None Very high

4. Wrist fatigue:

None Very high

5. General comfort:

Very uncomfortable Very comfortable

6. Overall, the input method was:

Very difficult to use Very easy to use

Please state your own comments (advantages / disadvantages) below:

No Feedback & Continuous Targeting

1. Smoothness during operation was:

Very rough Very smooth

2. Operation speed was:

Too fast Too slow

3. Finger fatigue:

None Very high

4. Wrist fatigue:

None Very high

5. General comfort:

Very uncomfortable Very comfortable

6. Overall, the input method was:

Very difficult to use Very easy to use

Please state your own comments (advantages / disadvantages) below:

Additional Assessment

1. Did you use your dominant hand during operation?

Yes No

2. Please order the corresponding techniques in ascending order:

Please use numbers from 1 (easiest to use) to 4 (hardest to use).

Feedback & Discrete Targeting:

Feedback & Continuous Targeting:

No Feedback & Discrete Targeting:

No Feedback & Continuous Targeting:

Additional general feedback (optional):