# Ink-based Note Taking On Mobile Devices

by

Yi Ren

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

**Abstract**

Although touchscreen mobile phones are widely used for recording informal text notes (e.g., grocery lists, reminders and directions), the lack of efficient mechanisms for combining informal graphical content with text is a persistent challenge.

In the first part of the thesis, we present InkAnchor, a digital ink editor that allows users to easily create ink-based notes by finger sketching on a mobile phone touchscreen. InkAnchor incorporates flexible anchoring, focus-plus-context input, content chunking, and lightweight editing mechanisms to support the capture of informal notes and annotations. We describe the design and evaluation of InkAnchor through a series of user studies, which revealed that the integrated support enabled by InkAnchor is a significant improvement over current mobile note taking applications on a range of mobile note-taking tasks.

The thesis also introduces FingerTip, a shift-targeting solution to facilitate detailed drawings. Occlusion caused by users' finger on the screen and users' uncertainty of the pixel they interact with are resolved in FingerTip via shifting the actual point where inking occurs beyond the end of the user's finger. However, despite a positive first impression on the part of prospective end users, fingertip turned out only passable on the drawing experience for non-text content.

Combining the results of InkAnchor and FigerTip, this thesis does demonstrate that a significant subset of mobile note taking tasks can be supported using focus+context input, and that tuning for hand drawn text input has significant value in the mobile smartphone note taking and sketch input domain.

**Acknowledgments**

I would like to express the deepest appreciation to my supervisor, Professor Edward Lank.

I would like to thank the readers of my thesis, Professor Dan Vogel and Professor George Labahn, for being willing to spend time and efforts on reviewing my work. Their advice is very important to me.

Also, thank my parents, girlfriend and all friends for always supporting me.

## Dedication

This is dedicated to my friend Patrick and my girl friend Payson.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Open Issues

Taking notes has been a common and important technique for keeping oneself informed in daily life. Despite the widespread of mobile devices such as smartphones and tablets, physical paper such as Post-Its remains a dominant medium for note taking within a large population due to the inefficient support of note taking by current mobile applications. However, with the rapid adoption of touchscreen mobile phones and the benefits of better accessibility, we believe that note taking on mobile phones has potential. However, an important challenge with existing mobile note-taking tools is the lack of efficient methods for note entry[21, 5]. Touch keypads and technologies such as handwriting and speech recognition have made steady progress, but low-level interaction details such as tapping on correct keys or verifying recognition results still tend to distract users from taking notes.

In contrast, extensive work has shown that hand-drawn ink, without recognition, is often preferred for note taking, e.g., [17, 26, 30, 37]. Using ink allows arbitrary content, such as diagrams and symbols, and offers a close analogy to the paper-based approach. Because mobile notes are consumed mostly by the note takers themselves, the informal, unrecognized look of ink is often less of an issue. An interior design architect, presented us with a photo (Figure 1.1) annotated by sketching on his smartphone. He needed to annotate the photo so that he could explain to his team in the office of the situations on site.

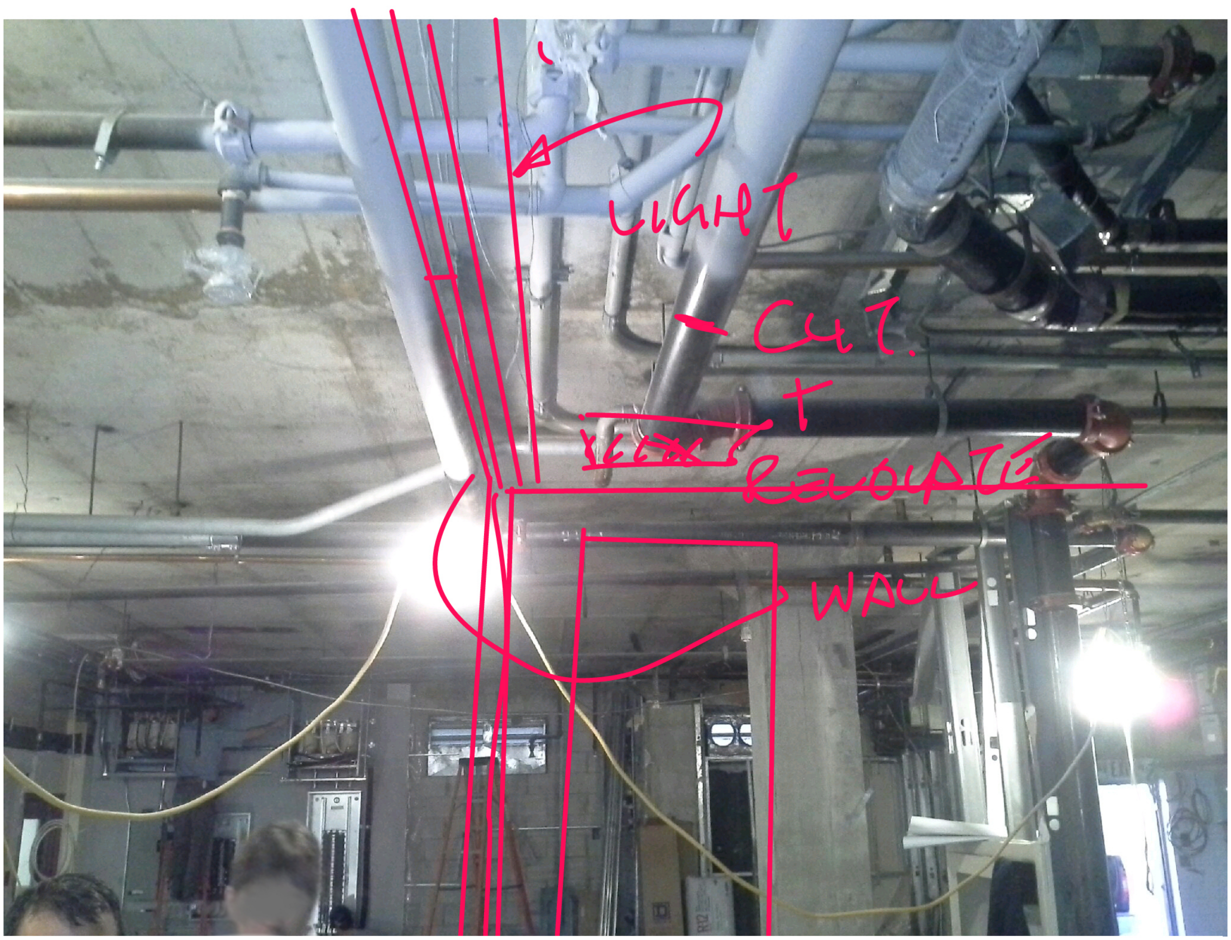



Figure 1.1: An interior design architect sketched on a photo to illustrate a light valance for a new restaurant.[8]

However, little work has been devoted to ink editing on touchscreen mobile phones, which incur different design constraints and raise new research challenges. As a result, many end users therefore purchase specialized devices such as phablets (phones with 5- to 6.9-inch screens [25]) or tablets that better support sketching. The store designer director who used to sketch on smartphones said:

> To be honest, I don't really sketch on my phone anymore. I bought the Samsung Note 10.1 specifically to make my sketching easier and more accurate.

## 1.2 InkAnchor

Despite the increasing popularity of large screen mobile devices, the access to such devices is still limited. Another option to resolve this problem, and the one we want to explore, is whether or not we can enhance the sketching experience sufficiently on small-screen smartphones so that people are not left with the need to purchase external devices.

In the first part of this document, we describe InkAnchor, an ink editor for creating finger-drawn ink-based notes on a touchscreen mobile phone (see Figure 1.2). InkAnchor addresses two major problems. The first is interaction space. Unlike pens and tablets, which prior work focused on [3, 1, 37], the small form factor of mobile phones and the occlusion caused by finger-based touch input make the interaction space clumsy. Touch-based input also has low precision because of the large contact area of the finger on a touchscreen (the fat finger problem [33]). The second problem is deducing high-level semantic units (i.e., meaningful groups of ink strokes), such as words from ink so that users can interact directly with those units. Although automatic approaches have made solid progress [30], there is often a mismatch between the structures that the machine infers and those that the users expect.

To address these problems, we propose anchoring, an approach that combines traditional text editors with the advantages of note taking on paper. Similar to writing on paper, users can write anywhere on the touchscreen at arbitrary orientations. Through an anchor, an interactive widget, ink strokes drawn by the user's finger are appropriately scaled down, automatically grouped, and laid out in a line-based flow layout (similar to a text editor). The anchor captures the user's intended layout and explicitly conveys the system's understanding of that back to the user.


Figure 1.2: InkAnchor allows users to easily create ink-based notes on a touchscreen mobile phone. In this case, the user is annotating a function image with texts in different directions.

## 1.3 FingerTip

Although text is the dominant form of information in informal notes, sketches may also contain detailed non-textual content. Current applications' support for free-form sketching has typically involved computer-style paint program tools, for example various paint brushes or rendering effects, but very little work has done dealing with detailed finger-based input. To address this oversight, alongside our work on anchoring, we explored FingerTip, an enhanced sketching technique to facilitate fine drawing by shifting the input location. (Figure 1.3) FingerTip allows users to draw with a simulated fingertip at the touch point, such that it appears that their finger is elongated by a "nib" measuring approximatively 1cm.

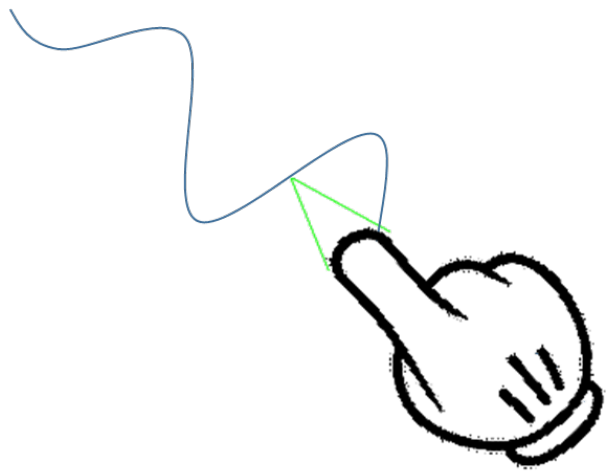


Figure 1.3: long

## 1.4 Contribution

There are two high level contributions described in this theses. The first is the evaluation of InkAnchor as a tool for hand drawn note taking on smartphones. The second is the design and evaluation of FingerTip as a tool for detailed drawing.

InkAnchor was evaluated by two studies, a comparative study with existing techniques and a first-use study. We compared InkAnchor with soft keyboard typing and Zoom, an implementation of focus+context [F+C] technique on a series of simple note taking tasks, and confirmed that the performance of InkAnchor was no worse at basic tasks. Next, to explore the usability of InkAnchor on more integrated tasks that were more close to notes in real life, we conducted a first-use study to better understand how individual features contribute to the entire system. The results convinced us of the success of InkAnchor as an integrated note taking system.

FingerTip was integrated into InkAnchor and was compared with the original InkAnchor. Despite a lengthy iterative design process, we find that FingerTip provides only marginal benefit to the task of detailed drawing on smartphones. In particular, despite overcoming the fat finger problem, the relative imprecision of finger-based drawing remains a problem for FingerTip. This causes the act of drawing to be slow, and the output produced to contain significant noise.

We believe that our designs highlight an important design constraint for multi-touch drawing. Specifically, while it is important to provide visibility to the drawn content, i.e. to solve the fat finger problem, this problem is secondary to the precision problem associated with drawing with one's fingers. By providing a focus+context input region, we overcome many of the problems of precision in drawing. Specifically, because users can draw in a larger canvas which is then scaled down to an appropriate size on the display, users are able to produce highly precise, neat drawings. This argues that any general solutions for detailed drawing and note taking on touchscreen smartphones must make use of focus+context input.

## 1.5 Outline

The document is divided into two parts as we depict in the contribution section. Part one of this thesis demonstrates the design process of InkAnchor and the study we performed to assess its efficacy for smartphone based multi-touch note taking. We examine existing informal note-taking practices. Next, we present InkAnchor, a prototype that demonstrates the anchoring approach, and describe how users employ it to create an ink-based note. We then elaborate on the underlying features that enable InkAnchor. We also report on how InkAnchor performed in both of the user studies.

The second part of the thesis focuses on supporting fine drawings. Our approach uses a shifted input point to allow users to see and precisely control inking. We call this technique FingerTip. We present aspects of the design of FingerTip, focusing specifically on techniques for precisely starting and stopping inking. Finally, we describe a user study that compares InkAnchor plus FingerTip to InkAnchor alone and argue that more work needs to be done to support detailed free-form drawing on multi-touch smartphones.

# Chapter 2

# Related Work

People frequently have the need to take notes. These informal notes often contain small but critical pieces of information, such as ideas, memos, or reminders for future actions. In this section, we will describe research into the practice of note taking and existing system for informal note taking.

## 2.1 Informal Note Taking Practice

Generally, people take notes for two reasons, to keep as storage for notable information or as a reminder for future events. [21] As a medium for keeping information, notes could either be long-term, such as for life-long learning or short-term, like personal jottings. Lecture notes that students take in classroom are very common long-term notes. People are often well prepared to take notes in classroom, so heavyweight medium for note taking such as pen and paper or notepads are more preferred in such situations. When one is in a hurry and away from his or her infrastructure, however, notes needed to be taken before information is forgotten. In these cases, notes serve as temporary storage, and are consumed instantly or transferred to a more permanent information repository. Lightweight systems such as post-its or smartphones are more suitable to carry when people are mobile and are thus a convenient medium for capturing these serendipitous pieces of information in the form of notes.

Another form of serendipitous note taking is the micronote [21], i.e. small notes that serve as pointers to future events. Micronotes exist in a variety of environments – as annotations on calendars in the home, as post-its or to-do lists on scraps of paper on a desk, or as quickly captured notes on a smartphone [21].

## 2.2 Informal Note Taking Systems

With the widespread availability of electronic devices, people seek alternative tools other than physical medium such as Post-Its for taking notes. Text content such as jottings or graffiti is widely supported by electronic systems via soft or physical keyboard. Note taking however, consists of sketching and drawings as well, so stylus devices (e.g. Tablet PC, phablets like the Galaxy Note) provide excellent platforms for note-taking since they mimic pen-paper system. A large body of research in stylus-based electronic note-taking stresses the activities such as active reading [1, 16, 24] or classroom note taking [10, 16], where users are more prepared to take notes. Under other circumstances, the requirement of a larger surface and the presence of stylus challenge the practicality of such note-taking activity. Thus, multi-touch sketching system may be more convenient. Furthermore, the widespread availability of multi-touch input devices argues for the exploration of multi-touch as an input modality for note taking.

Some recent research has explored the use of [22] multi-touch as a modality for inputting hand drawn content. For example, the MathBrush [18] project compared the recognition rate of math equations drawn with finger on iPads and with stylus on tablet PCs. Sketching on small screens, however, presents additional challenges, specifically with respect to screen size.

A series of additional interfaces exist to support the input of hand drawn text. TreadMill Ink [31] introduce a handwriting user interface that makes arbitrarily long streams of text input possible within limited space and without any interruptions. Unlike traditional pen-and-paper or write-anywhere interfaces (e.g., Transcriber [4]), the writing surface of TreadMill Ink is not fixed, but instead, scrolls from right to left while users are writing. An area on the screen, in the TreadMill user interface, behaves as a treadmill when dealing with users' handwritten input, such that the electronic ink input is moved from the middle of screen to the left of screen automatically as soon as it is entered. Users would have a feeling of writing text on a piece of virtual ticker-tape and a fluent, uninterrupted writing experience. However, TreadMill Ink requires the written ink to be recognized as characters to be added into a text editor.

Another typical approach for ink editing on mobile phones is to apply focus+context to a drawing canvas [20, 2]. In this approach, a high detail focus area and a low detail periphery are integrated in one user interface. User can write comfortably with a finger in the zoomed area, and at the same time, the surrounding environment remains visible for them to get the context of the entire note. The strokes of ink user draw with finger are scaled down corresponding to the ration of zoom focus size relative to the entire canvas. When user continue writing, the focus area, a rectangular region, has to be moved or reselected by the user so that they can write in another area.

Various ways for moving the zoom focus are widely implemented in electronic note taking tools, e.g. by panning with two fingers [2] or switching to the panning mode [20]. Users could dismiss the focus+context mode and select another area of the canvas as the new focus (e.g. DIZI [3]).

Besides manual focus shifting approaches, in Note Taker, a commercial iPhone application [2], a semi-automatic approach is implemented by providing several hint areas on the screen for users to shift the zoom focus either horizontally (similar to a space) or vertically to the next line (similar to an enter). This design serves perfectly for entering text in a horizontal, lined-based layout.

While focus+context works well for writing in a relatively small interaction space and is adopted in many applications, existing work rarely supports structuring digital ink. In most of the focus+context-based system, ink is treated as pixels grouped in strokes. A stroke itself however, is not semantically meaningful, thus making it insufficient for advanced editing, e.g., selecting, deleting or inserting a word. High-level structuring of ink (e.g. chunks, lines, or blocks) is necessary for text entry to support formatting (e.g., wrapping) and editing (e.g., selection, insertion and deletion). Moran et al. [24] designed an interaction where free-form handwritten texts are manipulated according to the implicit structures, i.e. strokes are grouped and organized based on their semantic meanings. The pen-based system Moran et al. implemented support largely list-related structures, e.g. text (horizontally aligned sequences of inks) and lists (vertically aligned sequences of items, which are horizontally clustered sets of inks). Other researchers have also designed algorithms to automatically group ink strokes and infer layouts [26, 30, 32], but these techniques are rarely incorporated to support mobile note-taking because of the poor recognition accuracy.

## 2.3 Conclusion

Physical paper such as Post-Its, remains dominant against electronic supports as a traditional medium for note taking. It allows freeform text entry and freehand sketching with reasonable input speed and accuracy. However, lack of accessibility makes physical paper less useful for serendipitous messages captured when a user in public contexts far from a desk. Despite challenges with the efficiency and accuracy of input, people are using digital tools to take notes [5]. Multitouch smartphones are often more accessible than pen and paper. The primacy of multi-touch in mobile contexts leads to the question: Can we support multi-touch note taking on small screen devices?

# Chapter 3

# InkAnchor: Ink-based Text Input on Mobile Devices

In this chapter, we review research into InkAnchor, a tool for mobile users to create ink-based notes on touchscreens with finger. We present how InkAnchor is designed based on the previous research and preliminary studies and give an overview of how users would use this tool.

## 3.1 Designing InkAnchor

Based on the previous understanding of informal note taking, our collaborator Dr. Yang Li at Google Research conducted an online survey to refresh our knowledge and capture the potential changes caused by the rapid evolution of mobile devices. The survey focused on the situations where informal notes were taken, what types of content were dominant in participants notes, what tools they used and the difficulties in taking notes. The survey was sent to a mailing list of an IT company, and 138 (a 9% response rate) people responded.

We found in the survey that informal note taking is equally distributed between at-work and on-the-go situations. When people are on the go, they often have their mobile phones with them. Participants complained most about how cumbersome to enter notes. We also found that the primary use of notes was targeted for note creators themselves, implying that the informal look of ink strokes should not be a major concern. With respect to the content in their notes, 99% of the participants (137 out of 138) stated that typed text was involved in their notes, while only 43% of the participants used drawings (which could include text as well). This result has informed us that text entry should be well supported by our application, while at the same time, drawings could not be excluded as well.

Based on the understanding of existing practice of note taking, Dr. Li proposed an initial design for an application he called InkAnchor, a multitouch note taking smartphone app, where users draw with finger on the canvas, leaving a stroke according to the series of points which the finger traverses. Informed by Dr. Li's study of informal note taking, the goal of InkAnchor is to efficiently support text input while not excluding drawings from hand drawn content produced via multitouch on a smartphone's display. To efficiently support text, InkAnchor performs high-level clustering of strokes into various high-level structures. We structure a group of ink strokes as presented in Figure 3.1. A *block* of ink, content within the red boundary, largely serves as a paragraph, i.e. a distinct section of text, usually dealing with a single theme. An ink block can be at different size, position, and orientation on the canvas in InkAnchor, and is constructed by a single or multiple *lines*, which is underlined with blue lines in Figure 3.1. Each area shadowed in grey indicates a *chunk*, which is usually a word or other sort of primary semantic unit. A chunk is a cluster of ink strokes and can be manipulated as a whole.

A significant body of work has been done to enhance the design of InkAnchor, such as enhancing visualization, tuning and refining design details. In order to make users enjoy a fluent experience while inputting text, one of the most important design decision we make is to follow a flow layout to manage the input, i.e. the sequence of chunks displayed on the canvas either from left to right horizontally to form a line or from top to bottom vertically to form a block. Despite the constraints in forming a block of ink, InkAnchor provide an absolutely free-form mechanism for user to devise the layout of blocks. This is achieved by using the anchor (Figure 3.2 (b)) as an indicator for the size of font, position and orientation of the upcoming block. The anchor is composed by a short vertical line, which is twinkling like the cursor in a text editor, and a long horizontal line indicating the baseline of the text. In our design, focus+context is also implemented around the anchor. The glass panel in the foreground is the focus area and the anchor in the background tells user the size, position and orientation of the current focus area. To give a better feedback on the size of the focus area, the anchor is displayed on the glass panel as well.

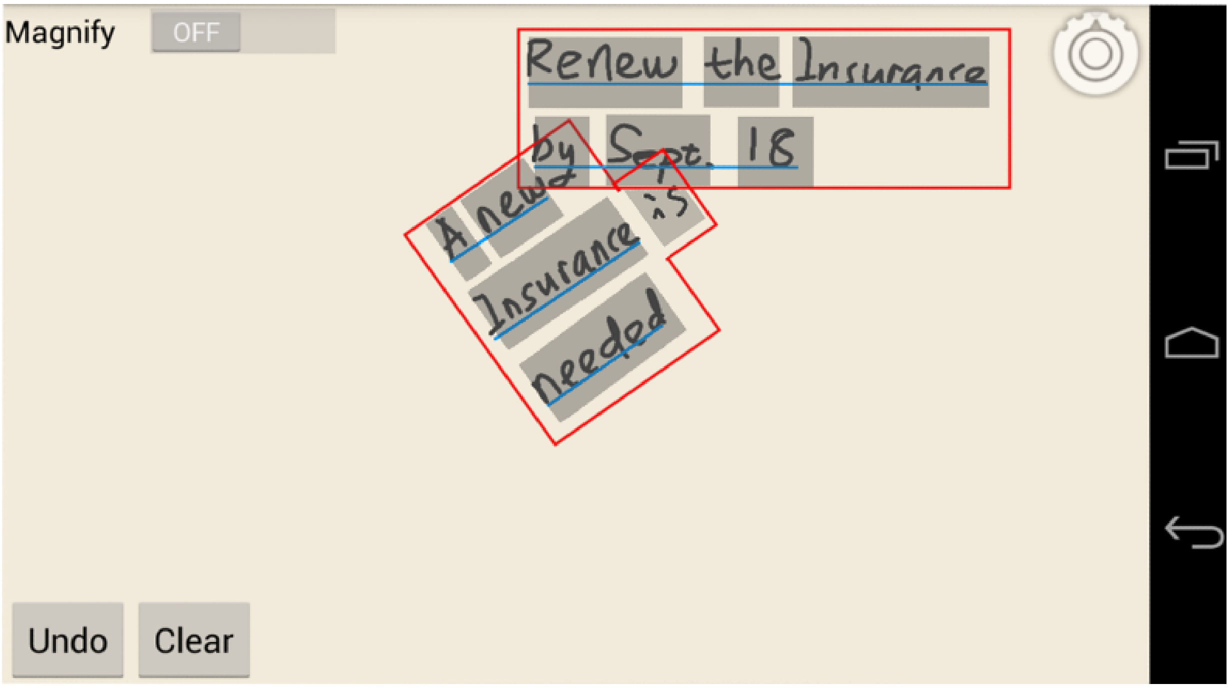
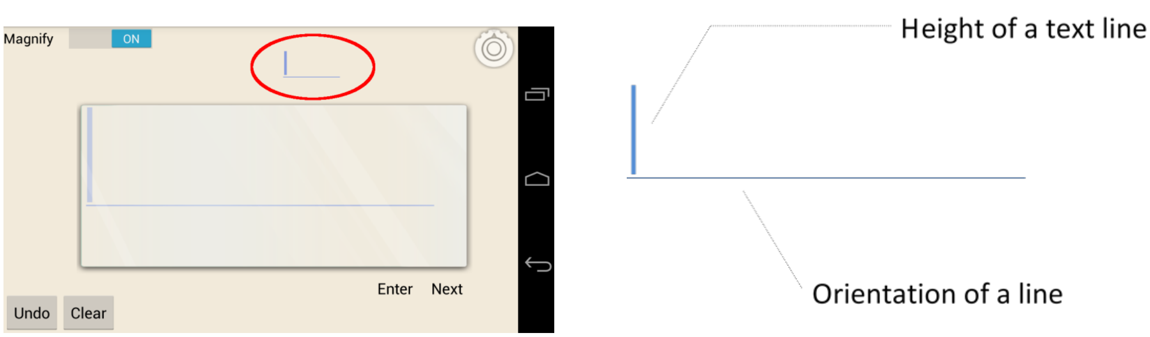


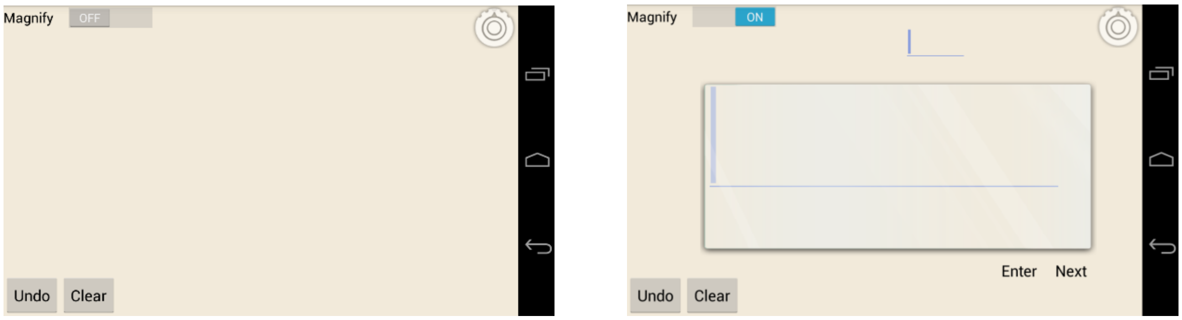

Figure 3.1: With InkAnchor, users can create a *block* of text, and arrange them conveniently




(a) InkAnchor launches with the anchor (circled in the figure) and a rectangular magnifying lens.

(b) The anchor defines height, position, and orientation of the ink block.

Figure 3.2: The interface of InkAnchor




(a) Non-magnified mode (b) Magnified mode

Figure 3.3: Different Modes of InkAnchor

## 3.2 Creating notes with InkAnchor

In this section, we discuss how a mobile phone user would use InkAnchor to enter and edit an ink-based note. We implement two modes in InkAnchor to support all-purpose note taking, the conventional mode (Figure 3.3 (a)) and the magnified mode (Finger 3.3 (b)). By toggling the Magnify button at the top left corner, user can turn on/off the magnified mode easily. The animation of the glass panel dropping down from the top of the screen is played when the user turn on the magnified mode, and is withdrawn otherwise. In the conventional mode, users sketch and draw with their finger touching the screen, leaving a trace of ink under their finger. While the conventional mode supports free-form drawings, the magnified mode is designed to facilitate text entry. The semitransparent glass panel in the middle of the screen is the primary workspace for users. While user is writing on the glass panel, their raw handwriting strokes are displayed at the original size on the glass panel and at the same time, are anchored into a scaled-down, well-arranged from on the background (Figure 3.4). The anchor defines how the strokes will be geometrically transformed.

Users pause when they finish a semantic unit (e.g., a word) or their writing uses up the entire touchscreen. There are two ways to proceed on to the next entry, clicking the Next button bottom right to the glass panel or wait for a short timeout (650ms in our current design). No matter which one of the two fashions is triggered, the collection of strokes on the glass panel is then wiped out. At the same time, the transformed strokes on the background is wrapped up as a chunk and the anchor will be repositioned to the right of this chunk while keeping a line-based layout (similar to a text editor) according to the anchor's size, position, and orientation (see Figure 3.5). However, when the line reaches the boundary of the screen or collides with any of the other existing blocks, it automatically wraps and starts a new line, while the beginning of the new line align with the beginning of the last line vertically to form a block. For example, the block Insurance in Figure 3.5 automatically wrapped when it collided with the block by Sept. User can also start a new line explicitly by clicking the Enter button, left to the Next button. Consequently, a user can keep writing without having to pay attention to how the strokes are organized.

A user can start a new ink block by repositioning the anchor with a long finger press at the target position anywhere on the touchscreen. Once a long press is detected, a vibration feedback is given as a notification, and the anchor jumps to where the finger is positioned, then following the finger while moving around. After the user lifts up the finger, the position and orientation of the anchor is set. In addition, to avoid the occlusion by the user's finger, a big red cross centered at the anchor is shown in the adjustment stage (see Figure 3.6) so that the user can easily see where the anchor is positioned before dropping it.

The anchor's orientation can be changed by rotating the physical screen (see Figure 3.6). This design was inspired by the observation that people tend to keep their writing posture unchanged while rotating physical paper to write in different orientations. Orientation is changed simultaneously with position in the adjustment stage, so a user can accomplish reposition and rotation of the anchor within one operation. While the user rotates the screen, the anchor and the cross remain horizontal relative to users. However, to preclude unintended change in orientation caused by subtle oscillation by the users, we fix the anchor horizontal while the angle is within the range from -10 degrees to +10 degrees. Besides the characteristic operations above, InkAnchor also provides conventional editing functionalities. Undo the last step and clear the whole canvas can be achieved by clicking the corresponding buttons at the bottom left corner, while other settings like font size could be configured in the setting menu at the top right corner.

## 3.3 Tuning InkAnchor

Informed by Dr. Li's survey of note-taking practices, and specifically by the importance of text in ink note-taking, InkAnchor incorporates four features to support ink entry on touch-


Figure 3.4: A user can write on the glass screen, and the ink will be scaled down automatically and show up on the background in real time.

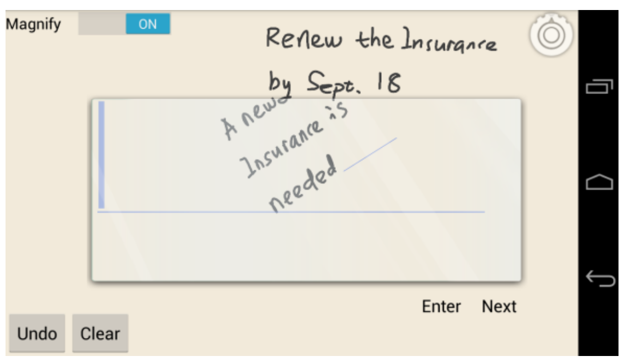



Figure 3.5: The ink strokes written by the user's finger are automatically scaled down and organized according to anchor size, position, and orientation. The line in a block automatically wraps when it reaches the boundary of the canvas or collides with another ink block.

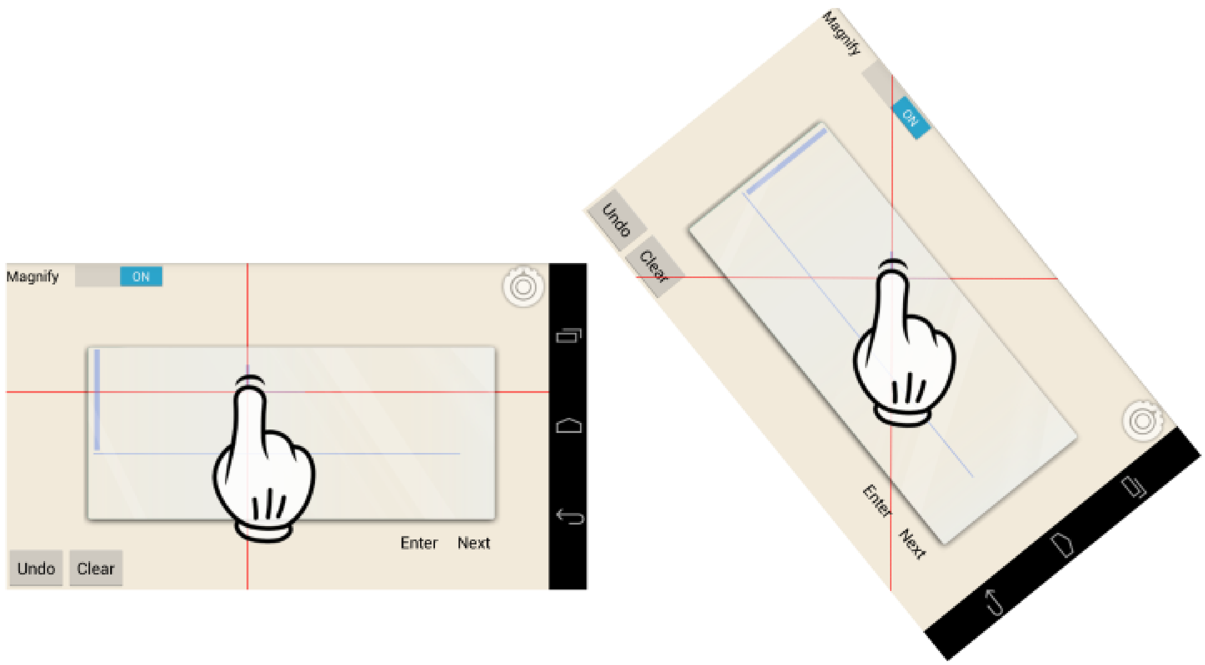


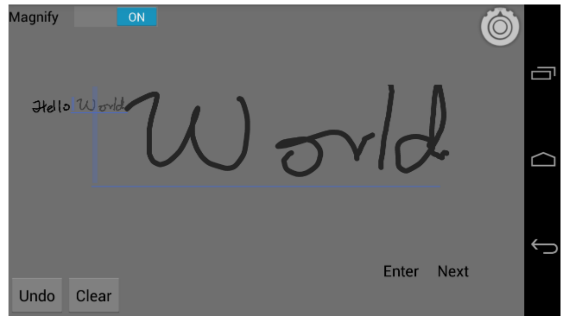

Figure 3.6: A user can reposition the anchor by pressing on the touchscreen, and the anchor follows the finger movement until the finger is lifted. A user can change the orientation of the anchor by rotating the physical screen. The red cross lines allow the user to precisely adjust the cursor position even when the finger occludes the anchor.

screens: focus+context views, text anchoring, free-form orientation, and auto-delimiting of text. There are many different ways that these features can be realized in an application, so we conducted a series of four design studies to refine the design of these features in InkAnchor. The design studies used a thinkaloud protocol to assess participants' perspectives on how well InkAnchor supported a set of simple tasks. The tasks in our design studies included asking the participants to write some basic text, to draw a simple node-link diagram, or to draw a graph and annotate the graph. We provided participants with exemplars of the tasks, but also allowed participants to specify their own content and to play with the InkAnchor interface. Finally, we walked through individual features being explored in the design study with each participant and collected their thoughts on how the features could be improved. A total of seven participants participated in our design studies. Because our design studies incorporated an aspect of participatory design, we considered it important that participants be skilled users of touchscreens. As a result, our participants were all chosen from technical disciplines within our university community, and all were owners of touchscreen smartphones. While scheduling constraints did not permit all participants to participate in all design studies, at least four participants participated in each design study, and all participants participated in at least two design studies.

### 3.3.1 Focus + Context

As discussed in Related Work, note taking on small touch screen devices benefits from the focus + context approach. However, to create sundry notes, sketches and text often require input at different scales, which implies that magnify scale should be adjustable in our system. From the result of previous research and the observations from the pilot studies, it seems clear to us that the magnifying scales of sketches and text are distinguishable, which means by switching off the focus+context, a user should be able to master most of the sketches. While researchers have developed some automatic methods to classify ink strokes into text and drawings [32], automatic segmentation remains challenging, particularly in our system, where there is often insufficient information for inference since transformation is done in real time. Therefore, we decided to active focus+context manually in InkAnchor.


(a) A glass penal is drawn down from the top of the screen upon entering maginied mode.




(b) The canvas will grey out in the magnified mode, giving users an impression that they are sketching on a layer above the canvas

Figure 3.7: Different visualizations of an alternative layer to sketch on

### Mode Switch

As depicted in the previous chapter, mode switch is done by toggling the Magnify button in our design. This action seems somewhat inelegant and may cost a user significant amount of extra time if they switch mode a lot. In our design studies, however, we observed that users would draw in the non-magnified mode, then spend significant time drawing text in the magnified mode, then revert to large scale drawing briefly, then return to text for another block of time. Essentially, our mode-based approach seems analogous to the behavior of our participants, where text entry was segmented in time from large-scale drawing. However, in our early design where there was no visual feedback on the mode switch, we found that users sometimes lost awareness of which mode they were in. As a result, it happened a lot when users wrote a word in the non-magnifying mode and waited for the scaling down. To resolve this problem, we decided to provide different visualizations for different modes. Both of the ideas of magnifying lens and background grey-out popped up in brainstorm. However, controversy on which type of visualization should be used was not settled by discussion, so a pilot collecting feedback on two types of visualizations was conducted.

4 participants, all experienced in User Interface designing, were asked about their opinions on both types of visualization. In particular, for the magnifying lens, participants were asked to evaluate the sizes of the lens, and how they liked the animations of the lens sliding in on mode switch. The sizes of the lens included a lens that covered the whole

canvas, a lens that fitted the size of the anchor on the foreground, and a lens belt that went through the left to the right of the screen with the height of the anchors height. For these three types of lenses, participants voted the most for the second one. Animations of the lens included sliding in and out from the top, the bottom, the left and the right, and participants liked for the one from the top most.

We also made great efforts in tuning the grey-scale to make the visualization of grey-out to be in good shape. A series of grey-out visualizations of different grey-scales were provided for the participants to evaluate. Finally the color of the grey-out was set to 0xff6f6f6f (in Android color notation).

After we were confident enough that both types of the visualization were most satisfying, we conducted another pilot and asked participants of their opinions on the visualizations. Summarized by the feedback, they thought the magnifying lens gave them an impression that they could draw on a panel on the foreground, instead of messing around with the contents on the background. On the other hand, they also argued that the magnifying lens was distracting, and they loved the simplicity of the grey-out. However, we finally decided to stay with the glass screen, since participants shared a more positive feedback on it.

**Font Size**

Besides the different in scales between text and sketches entry, we are also aware of the need of inputting text of various size. In our preliminary design, to change the size of the scaled-down strokes or the height of a line, a user can adjust anchor height. A long press on the anchor brings up a slider for adjusting the height of the anchor. However, according to the result of the pilot study, this gesture always confused the participants, since they wanted to long press on the anchor before move it, in order to form a dragging gesture.

Interestingly, in studying the re-sizing of ink we found that participants had rarely any need to adjust the height of a line frequently. If a good default scaling is provided according to the size and resolution of the device screen, users would leave this default unchanged. If there's a need of creating larger or smaller text, e.g. a title or an annotation, they would just simply adjust the size of their input. This behavior seems too mimic what is observed when people are sketching on a whiteboard: There is an expected viewing distance from the eyes to a given medium, and writing is often scaled based on that expected viewing distance. We depicts our default scaling for the hardware we used in our studies (Samsung

Galaxy Nexus phones) in Figure 3.4. We tuned the default font size through a series of initial sessions, and, once set, we found that new participants seldom adjust the default scaling of ink.

As a result, we eliminate ink resizing via long-press on the anchor, and provide radio options (small, medium, large) of text size in the setting menu in our current design. Long-press on the anchor would then allow participants to drag the anchor around, while a long press at any other location would cause the anchor to teleport to that location. Thus the gesture of long press in InkAnchor is consistent regardless of the pressing location.

### 3.3.2 Text Anchoring

To facilitate text input, we developed text anchoring as a novel way of moving the zoom focus, where users handwritten ink is scaled down automatically to a specific size at a specific position. We made lots of efforts to help users better understanding what they are doing. Perhaps one of the most important features added through our design studies, to support inking accuracy, was real-time visual feedback of ink scaling when users are entering text.

#### Real-time Visual Feedback of Ink Content Scaling

With the real-time visual feedback, users do not have to wait until they finish the whole word to see their handwritings on the background. It not only gives users a more comfortable and intuitive working flow, but also benefits when writing something according to the context on the background. For example, if users want to write along a line on the background, he can tell immediately when he start writing, and begins to adjust during the entire process, instead of deleting the whole word and writing again if the word turns out not align with the line on the background. The immediacy of the visual feedback allowed users to intuitively adjust size and position to produce a more accurate drawing.

#### Fitting Ink Strokes into a Line-Based Flow Layout

An important step in transforming large, finger-drawn strokes into a line-based layout is to scale and translate them appropriately so that the chunks in the block align with

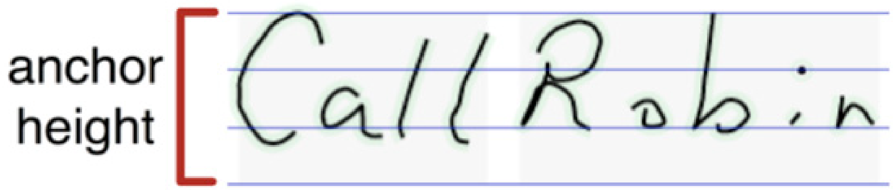
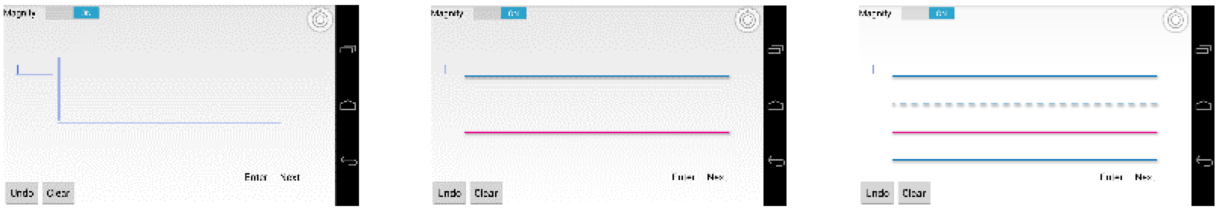



Figure 3.8: InkAnchor fits a chunk into a three-row template.

each other. We experimented extensively with baseline detection techniques. Because the anchor defines the height of a line in a block, we aim to fit each chunk of strokes into a three-row template that is typical for the Latin alphabet (see Figure 3.8). An important task in this process is to identify the baseline of the chunk; combining this with the height of the anchor determines both the scaling factor and the translation for the vertical alignment in a line.

However, there are two facts that impair the worth of baseline-detecting technique. First, it is almost impossible to find the baseline of an English word without handwriting recognition. Second, since the real-time visual feedback of ink scaling is an important part of our design, we need to decide the absolute position of the transformed inks right after the users start writing. As a result, it conflicts our original design to wait for the users to finish writing an ink chunk to calculate the baseline or to dynamically scale. Such tow rationale convince us of the need of some other feedback mechanisms to allow users to adjust the scale.

When young aged children begin learning to write, they are asked to write on notebooks with guidelines which facilitate children to figure out where they should write. Inspired by the way of teaching children alphabets, our solution to this is providing virtual guidelines to format users' handwriting. We summarized the popular forms of alphabet guidelines, and developed 3 different ways of guidelines as shown in Figure 3.9. Since the 2-line and 4-line guidelines have the shortcomings like overlapping background inks and lack of intuition, we decide to keep Figure 3.9 (a) as our final design.



(a) A larger anchor mapped to the anchor on the background.

(b) 2-line guideline.

(c) 4-line guideline.

Figure 3.9: Visualizations of guidelines

### 3.3.3 Auto-Delineation of Chunks

We provide two kinds of delimiters to ink chunk in InkAnchor which lean up the magnifying lens and advances the anchor for subsequent input, i.e. repositioning the anchor to the right edge of the ink chunk that was created. The first is an auto-delimiter, which wraps up users' handwritten ink automatically after a certain period of time. We ran a user study in order to find an optimal timeout for the auto-delimiter. We asked participants to accomplish several basic tasks with different timeouts (420ms, 650ms and 1000ms) and we selected 650ms as the timeout in our final design. The second one is by clicking the next button or the enter button at the bottom right of the magnifying lens. At the same time, the anchor is moved to the beginning of the next line automatically, leaving an interval between the two lines and the magnifying lens is cleared if it is not empty.

This auto-delineation was the most controversial feature in our design, since we noticed that in our pilot studies, participants disliked the feature in the beginning. We experimented frequently with manual advance or with longer or shorter timeouts, trying to figure out if there is any perfect solution. And we found that, in the case where there was no timeouts, as participants became more comfortable with the interface they would typically want the auto-advance/timeout re-enabled to support the entry of longer blocks of text. If, instead, we simply left it enabled at the beginning of the studies, participants seemed to quickly acclimatize to the timeout. As a result, we take auto-advance with timeouts as a useful feature and keep it in our system. By testing with different timeouts, we finalize it to be 650ms since after users become familiar with the tool, this timeout worked the best out of the three timeouts in our experiment, and participants were most comfortable with the this timeout setting.

### 3.3.4 Determining Anchor Orientation via Phone Orientation

In contrast to the horizontal layout of a text editor, users frequently need to write text along an arbitrary orientation as on physical paper. For example, while users are annotating the vertical axis of a graph in a mathematical document or labeling a road in a hand-drawn map, they would create text blocks along the function image or the trend of the road. As a result, InkAnchor must be able to detect the intended orientation of users so as to determine how ink strokes should be geometrically transformed. Since multi-touch rotations (e.g. two-finger rotation) have the disadvantages of ergonomic failures [15], and are inefficient on small screen devices, we decided to look for alternative approaches for rotation. When people write on a sheet of paper, they often rotate it to write at a different orientation, so the direction in which their hand moves remains the same relative to their body. Inspired by this observation, we designed InkAnchor to capture the user's intended orientation by detecting the mobile phone's screen orientation. In other words, no matter how the phone is rotated, InkAnchor keeps the anchor's orientation horizontal and rightward relative to the user's coordinate system. To avoid unintended rotation due to the vibration caused by writing, we only enable rotation when users trigger the adjustment stage by long pressing on the screen. Our approach for detecting screen orientation is similar to that used by existing mobile devices such as Android phones. The basic idea is first to extract the acceleration caused by gravity from readings of the phone's 3D accelerometer using 1 € Filter [7] and then to find out how gravity is distributed along the device's three axes. Gravity distribution determines the angle between the device orientation and gravity. Consequently, this technique works when the user is in a normal upright posture, such as sitting or standing. When gravity is evenly distributed along the three axes, e.g., when the phone lies flat on a table, it is not feasible to determine the device's orientation by using acceleration alone. As a result, the anchor orientation in such a situation is unchanged. We attempted to resolve this defect by combining the data from geomagnetic field sensor as well. However, since the geomagnetic field sensor is not sufficiently accurate, we abandoned this idea and sacrifice the usibility when the the phone is placed horizontally, in order to remain a better performance. Since a mobile phone is in constant motion when the user rotates the phone to find a target orientation, InkAnchor stabilizes anchor orientation by snapping to the horizontal and the vertical orientation. As a result, users can easily rotate to these to orientations which are the most commonly used ones.

## 3.4 Synopsis of Design

As we note in this chapter, a significant amount of research was done to produce a final design for the InkAnchor system. Initial design work was performed by Dr. Li. In particular, Dr. Li conducted an initial study of note taking practices and proposed the use of anchoring to support informal note taking on multi-touch devices.

To further this research, we invested a significant amount of time refining the design of InkAnchor. The specifics of this design refinement are described in detail in the preceding section. In the following chapter, we describe our work evaluating InkAnchor's utility for supporting multi-touch notetaking on small screen smartphones.

# Chapter 4

# Evaluation of InkAnchor

## 4.1 Comparing InkAnchor with Existing Techniques

To test the efficiency of InkAnchor, our collaborator, Dr. Yang Li, conducted a small scale user study by comparing InkAnchor with two existing techniques, software keyboard and Zoom, an alternative implementation of focus+context (Figure 4.1). The study was conducted not to prove that InkAnchor was faster, but instead to ensure that it was not worse at basic text or drawing tasks after combing the massive set of features.

Dr. Li asked 6 participants to draw six different diagrams, an email contact (name plus email address), a reminder, an address, a grocery shopping list with nine items, a simple graph of data with axes labels and three annotations, and a simple map of driving directions showing three streets with names. As depicted in Figure 4.2, the first four diagrams were created using all three techniques, where InkAnchor has similar performance as Zoom, and slightly better than typing in the task of shopping list. The remaining two, where graphics were involved, were only performed by InkAnchor and Zoom. Again, InkAnchor performed better than Zoom. Based on this study, we found InkAnchor had similar overall time performance to both Focus+Context drawing and typing on the set of tasks in the experiment.

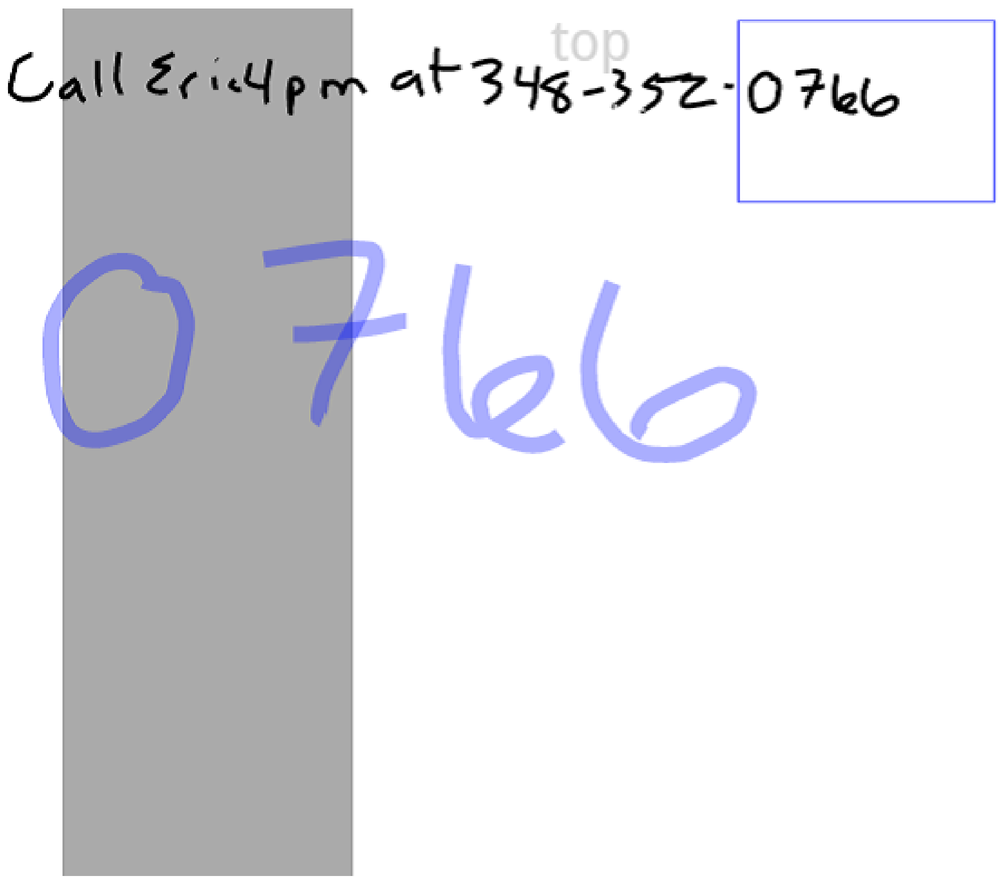



Figure 4.1: An alternative Focus+context implementation that Dr. Li used for comparative study

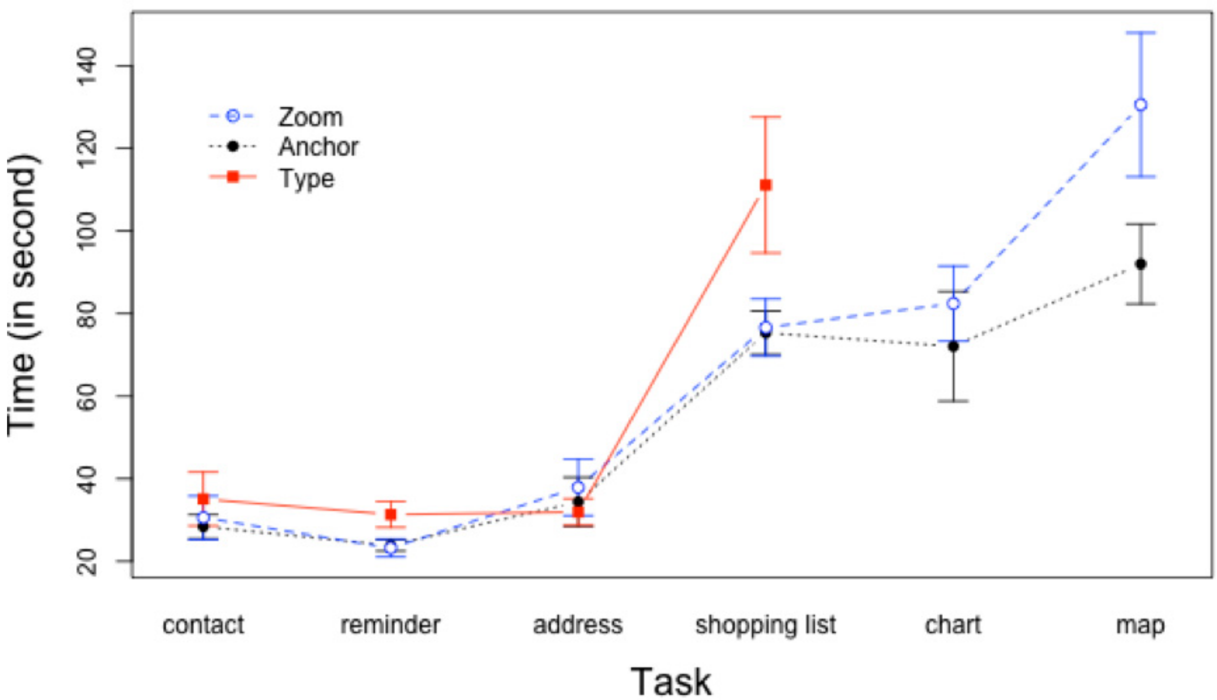



Figure 4.2: Performance on 6 tasks of different techniques in Dr Yang Li's study

## 4.2 Experiment Settings

Convinced by Dr Li's study, where InkAnchor showed its potential advantages against existing techniques, we are more interested in how each of the features in InkAnchor, as an integrated solution to note taking, contributes to the system. As a result, a first-use study [38] was conducted to capture users' ability to perform a series of diagram entry tasks with InkAnchor. We recruited participants from within our university community, where both students and professionals were included. The participants were recruited under the constraint that they had some experience with multi-touch sketching, since we felt that the need for preconceived impressions of multi-touch sketching was particularly important for a first-use study of a novel sketching app. 16 participants (thirteen male; three female) were recruited to participate in a study of approximate 1 hour (we gave each of them a $10 Tim Horton's Gift Card). Nine of the participants had experience with using sketching applications on smartphones; three on tablets; one on both; the rest have experience on other platforms. A semi-structured interview was conducted at the start of the study. The interview focused on participants impressions of their experience interacting with the sketching applications they used before. Participants were asked about their feelings when sketching, what they like/dislike about the sketching application, what features they are expecting to see in a sketching application, and so on. Next, we showed participants InkAnchor on a Samsung Galaxy Nexus phone, running Android 4.2.1, which has a 4.65-inch multi-touch display with a resolution of $720 \times 1280$. After being introduced to the basic interactions of the application, participants were then given the device and started sketching randomly with InkAnchor while expressing their first impressions of the application to the researcher. We asked participants to perform 3 tasks with InkAnchor when participants felt they were comfortable with the interactions. The tasks consisted drafting a map, labeling a room diagram, and annotating a course note.

## 4.3 Guidelines for Pre-study Interview

The goal of our semi-structured interview was to capture our participants' perspectives on taking notes and interacting with current note taking tools. At the beginning of the interview, participants were asked to describe their recent note taking, focusing specifically on multi-touch note taking. We asked each participant to walk through their most recent experience taking informal notes. Specifically, we were looking for both the context of the note taking task and an understanding of the content that they wished to capture in an informal note. In particular, we made a particular effort to focus on the tools they used to take notes. We were curious of the functionalities or key features of the tools they used, and asked them to demonstrate tools used if possible. The interview often ended with participants attitude toward the tools, say, the tasks that were best supported by the tools, the functionalities that were the most useful or any functionalities that they were expecting but did not exist in the tools. We tried to follow this guideline in every pre-study interview.

## 4.4 Experiment Tasks

As we noted earlier, after the initial interview, we allowed participants to experiment with InkAnchor. Once they were comfortable, we then asked them to complete three different experimental tasks. In the design of these tasks, we addressed common seen break downs and potential defects of sketching on small screens. We invested significant effort in piloting each of the tasks to balance the workload of the three tasks, so that participants would not be biased when they were evaluating the performance of InkAnchor on the three tasks. Finally, a short interview focusing on their opinions of InkAnchor was conducted after they finish all the tasks, and participants completed a questionnaire in order to evaluate the features of InkAnchor.

#### Map Drafting

The map sketching task mimics the creation of an informal note with both drawings and tasks. Participants were asked to draw a drafted local map with InkAnchor, as shown in Figure 4.3. The idea of this task came from the fact that one of the participants in the pilot study mentioned he sketched a map for his friend, indicating the location of his house. We included an extra-long street name "Fisherhallman" in the task intentionally, as it is hard to handle sketching such long word on a small screen, and we wanted to see how participants could resolve this: Would they notice the need to only enter half of the word at a time before advancing the anchor? Would auto-advance cause any particular


Figure 4.3: Map drafting tasks

problems? Would they adjust their behavior by writing smaller? Would they short-circuit the timeout?

### Room diagram labeling

Participants were shown a room diagram with a list of room names. The background of InkAnchor was replaced by a picture of the room diagram and we asked participants to label each room with the corresponding name in the list as in Figure 4.4. In this task, we are looking into how accurate InkAnchor can perform for novice users. Participants were asked to fit each word into the corresponding blank space, which is very limited in size.

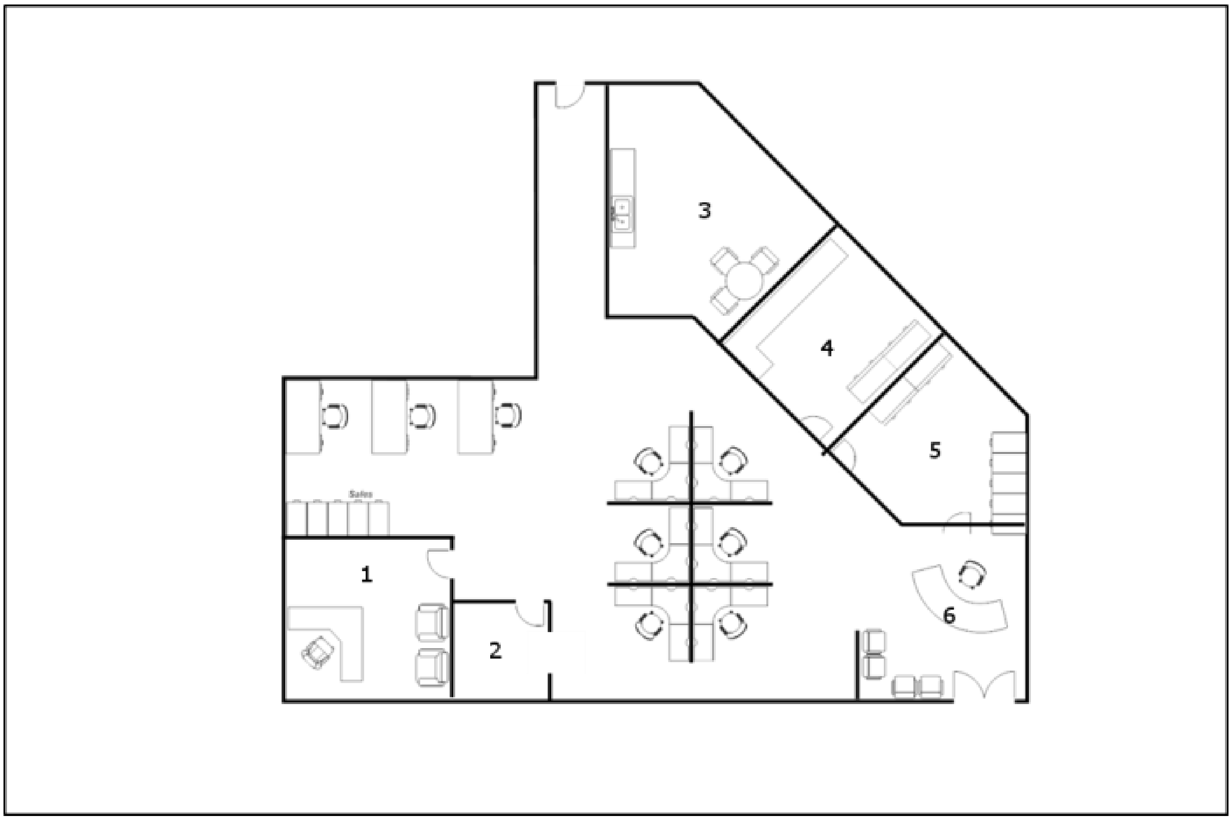



Figure 4.4: Room diagram labeling task

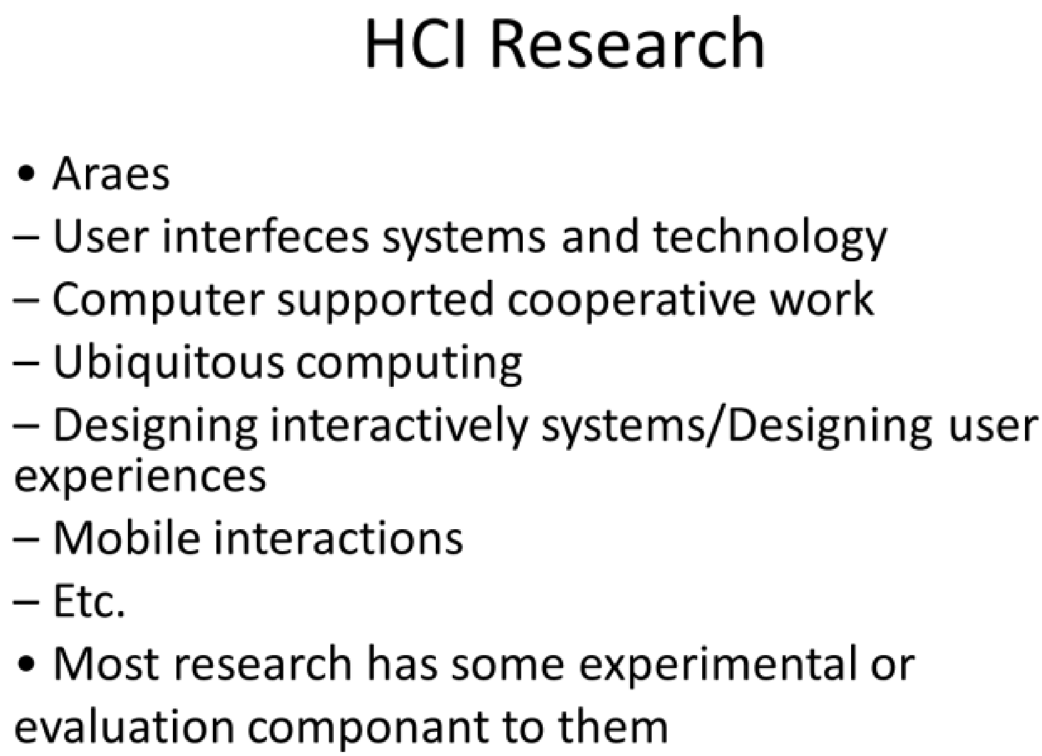



Figure 4.5: Course note annotating task

#### Text Editing and Annotation

Since many of the participants mentioned they needed mobile sketching tools to annotate other text content, for example during active reading, proof reading, or note taking in a course or classroom environment, we designed a task (Figure 4.5) in order to simulate such situations. We gave participants a prepared course notes as the background of InkAnchor, with 3 typos to correct and 1 definition to annotate. We didn't specify a way to annotate, so participants could do whatever makes sense in their point of view to finish this task. As a result, evaluate InkAnchor as an annotation tool for active reading tasks.

## 4.5 Experiment Results

In this section, we highlight several outcomes of the user study, discussing how they impact users' attitudes and behaviors in mobile note-taking. We organize the results around InkAnchor's key features, analyze the usability and explore the potential usage of InkAnchor, referring to specific observations, questionnaire results, and coded responses.

### Evaluation of Key Features

We summarized five features of InkAnchor that we found important in the progress of design and pilot study: real-time feedback, auto-scaling, ink-block collision detecting, ink rotation, and auto-delimiter via timeout. While these features were derived from our iterative design studies, we are interested in how they would be evaluated in combination in InkAnchor. The raw data of the questionnaires can be found in the appendices, where each question is graded on a -3 to +3 scale, as well as the mode, the average, the median and the number of participants rating a specific score (-3 to +3). We depict median values and 95% confidence intervals for the median in Figure 4.6. The confidence intervals extend significantly beyond +3 in many cases because of extremely high median and mode values for InkAnchor's features. Particularly, real-time visualization, auto-scaling, ink rotation and block collision/word wrapping were all highly rated, with median and mode values of +3 for usefulness. Even the timeout auto-delimiter had median and mode values of +2, affirming the significance of the linked feature set in InkAnchor.

**Rotation of Ink** The way of rotating the ink, not a self-revealing feature, was hard to perceive in the beginning of each study, but could be learned instantly by the participants once we told them how it worked. One participant used both a finger rotation and a device rotation when he or she was trying to re-orienting ink:

> Yeah, I'm not quite sure what kind of tech you are using for rotation; maybe it's multi-touch, or the gravity, so I tried to use both of them to make it work. (Participant is rotating the device in one direction, while rotating his finger in the opposite direction) [P1]

Participants pointed out that rotating their finger instead of the device would be more intuitive when asked about which way they prefer.

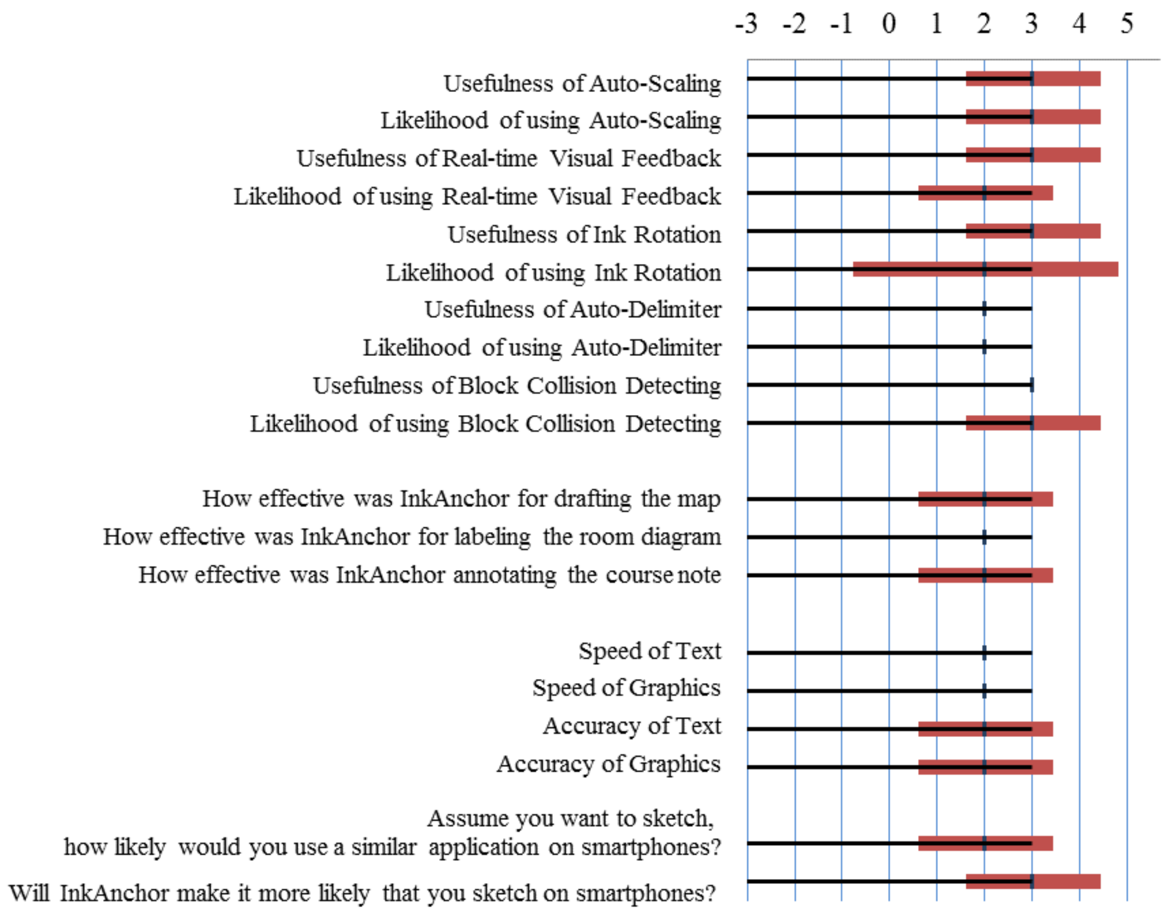



Figure 4.6: Each question was graded on a -3 (not useful/highly unlikely) to +3 (extremely useful/extremely likely) scale. The bars show the 95% confidence interval of the median.

> Just the rotation, I'm not sure which direction I'm actually writing. I think if you can, instead of rotating the phone, maybe rotate your two fingers to address the text alignment. It may be better. [P2]

In our observation, except for one participant, P1, participants eventually realized that rotation was controlled by the accelerometer, and they had few problems with rotation in InkAnchor. As we mentioned in the previous chapter, our system works particularly well when users were cradling the phone, rather than setting it on a desk. However, participants noted that it is more comfortable to draw while holding the phone, so this did not seem to impede usability in our studies.

**Delimiters of Ink Chunks** Not surprisingly, the other feature that we found participants had some difficult was the auto-delimiter timeout of 650ms. Even though we spent significant time during pilot studies adjusting the time-out for auto-delimiter, participants complained at the beginning of the studies, and argued for a longer time-out.

> So the pause time is a bit is too short. When I'm writing "Fisherhallman", the first words I wrote was "Fissher", and I'm considering how much space I need to write the "hallman", but I automatically wrapped it up. [P2]

However, as they got used to the interface, the time-out became quite comfortable for them, and perform efficiently in most of the cases.

> Yeah, I like it. It's professional. If I want to move to the next word, it helps a lot. I think it's professional... But for me, I think it's ideal. The time is perfect. I like it. [P3]

Based on the process of participants' habituating to the timeout, one straightforward solution is to explore gradually tightening the time out during first use. Perhaps the application could start with a timeout of approximately 1.5s and the timeout is shorten over the first ten or twelve timeouts down to the current 650ms. This may help users to perform better initially, but it might also cause problem since users are hard to perceive a definite timeout in the system, and they have to adjust themselves over time in the initial use. Considering the time participants spent on adapting to the current timeout, it is questionable whether a looser initial timeout is necessary.

**Other features** Participants understood the association between the foreground and the background rapidly after they drew the first character on the magnifying lens, and made rarely any effort to get used to the auto-scaling down function. In some extreme cases (e.g. contents on the background is very detailed), participants were looking for functions like zoom-in, but most of the times they agreed that auto-scaling down could help them achieve their goals. Real-time visual feedback was considered to be useful by the participants to create accurate text entry, and ink-blocking collision detecting was a good add-on in terms of building a sophisticated system, since it reduced interruptions during text inputting greatly.

#### Overall Assessment of InkAnchor

Most of the participants had a negative feeling about sketching on smartphones, no matter whether they had ever sketched on smartphones or not. One of the participants, who is a civil engineer, he goes to construction site a lot, where he needs to draw figures to describe his ideas to his colleagues. In such cases, he always draws with an application on his smartphone while explaining. However, he still complained about the performance of sketching on smartphones,

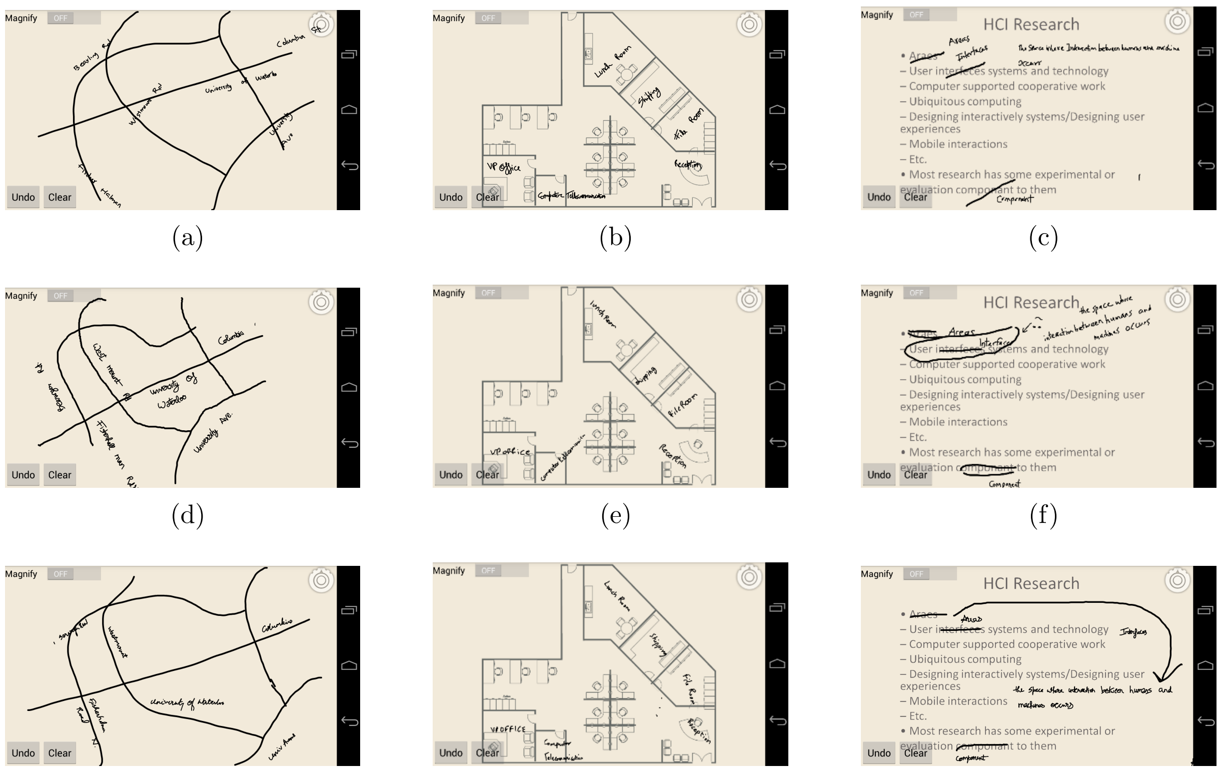
> I would rather draw with my fingers on the dusted windshield of my car than draw with the phone. [P8]

Other participants from P4, a graduate student in computer science, to P2, a graduate student in engineering, all expressed similar sentiments:

> I think, hmm, I think a mouse gives more control to a certain extent, I'm pretty good at mouse. I'm not quite used to this, you know, drawing with your finger things. [P4]

> On a smartphone, sometimes I feel like my fingers are blocking my vision, so I can't really see what I'm drawing on the graph, as my finger blocks most of the things on the screen. ... [P2]

However, at the end of the study, participants held positive attitudes towards sketching with InkAnchor. Most of the participants showed their interest in using the application in real life, and some even asked if we could put InkAnchor on Android Market.




(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.7: Sketches drawn by three different participants in our study

**Speed** In the pilot study, we tried our best to call for a balance between speed and controllability, by adjusting the time-out the speed of auto-delimiter. It turns out to work quite well in the user study, since most of the participants were satisfied with the input speed of InkAnchor, and each of the tasks was accomplished within 5 minutes.

**Accuracy** Handwritten ink has poor recognizability, no matter whether it's on paper or on touch-screen devices, so accuracy is very important in sketching applications. Many efforts has been made to enhance the accuracy of InkAnchor, such as real-time visualization, stroke curving, and default scale optimization. As a result, we received positive feedback on accuracy from the results of both questionnaires and the sketching of participants during the experimental the tasks (see Figure 4.7).

## 4.6 Conclusion

Most of the participants expressed a negative feeling about taking notes on touchscreen phones at the beginning of the study. One of our participants, a site coordinator (civil engineer) who frequently sketched on his smartphone at construction sites, stated:

> I would rather draw with my fingers on the dusted windshield of my car than draw with [on] the phone. [P8]

Other participants from P4, a graduate student in computer science, to P2, a graduate student in engineering, all expressed similar sentiments:

> I think, hmm, I think a mouse gives more control to a certain extent, I'm pretty good at mouse. I'm not quite used to this, you know, drawing with your finger things. [P4]

> On a smartphone, sometimes I feel like my fingers are blocking my vision, so I can't really see what I'm drawing on the graph, as my finger blocks most of the things on the screen. ... [P2]

However, we found at the end of the study, participants held positive attitudes towards sketching with InkAnchor. Thirteen out of sixteen participants expressed a desire or a strong desire to use InkAnchor, and one of our participants even requested a download link for the app from the Android Market.

Results from the experiments confirmed the performance of InkAnchor, particularly for drawing text-based content. Information one wishes to capture in the form of short notes however, is not always text-based. Drawings, equations and graphs are all examples of content that is not one-dimensional text consisting of words and phrases but where detail may be important. Given that InkAnchor effectively supports text input, one remaining question to answer is whether or not it is possible to support other detailed drawing tasks on multi-touch smartphones.

# Chapter 5

# FingerTip: Fine Sketching with Finger on Touchscreens

We address the challenges of text entry in InkAnchor. However, we also noticed in our studies that participants were looking for more detail when drawing. As one of the participants noted:

> I wanted to add a road beside its road name, but it's really hard. It either overlap or too far away from the text. [P7]

The problem that this participant is describing is related to the fact that finger occlusion impedes the performance of participants' fine drawing in InkAnchor, and previous research also implies similar results. Two fundamental problems impede direct-touch finger input on touchscreen devices [36], the occlusion problem, where the users' finger occludes the intended target when users are touching the display, and the fat finger problem [33], where users' fingers are significantly larger than a pixel of the display such that users have no idea which pixel under the finger is taken as the actual touch point by the system. Both problems negatively impact the user experience when participants sketch in our studies. In some cases, for example in a flowchart sketching task from one of our initial design studies, where participants tried to add a box around a block of text, they needed to be extremely careful since they lost awareness of the accurate position of the block boundary which was covered by their finger. Even with great caution, participants sometimes needed to undo and retry several times to accomplish the task.

The fat finger problem is another challenge associated with finger drawing on touch screen devices. Users know that the stroke will be positioned somewhere under the finger, but they are not sure of the exact position before the stroke is created and displayed on the screen.

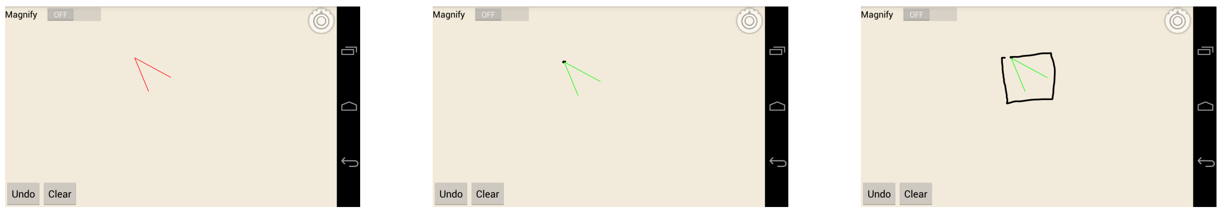
Unlike drawings, text entry does not suffer as significantly from these input challenges.In particular, the mechanisms of people writing and drawing are significantly different. When writing, the geometry of strokes in a word are already well shaped in the users' mind, such that they can complete the word even without looking at the canvas. Drawing however, requires users to read the context simultaneously while making decisions in positioning each stroke. It seems clear to us that supporting fine sketching and drawing in InkAnchor requires that we solve both of the above problems.

Research exists addressing those two problems. Potter et al. first designed and developed the idea of Offset Cursor [27], which was a cursor about half inch above the finger. With Offset Cursor, users can easily determine the accurate position of targets since they can easily look over their finger and see the target. Unsurprisingly, Offset Cursor enhances the performance of direct-touch finger input, and serves as a guideline for related research. Shift [35], combining Offset Cursor and focus+context, provide a more sophisticated and accurate approach for target selecting.

Inspired by the performance enhancements from Offset Cursor, we designed FingerTip, an offset targeting technique where users can sketch using an offset cursor like an extension of their finger (Figure 5.1 (a)). When users place their finger on the screen, a red cursor is displayed. The tip of the cursor indicates the point where ink appears, while the base of the cursor sits under the user's finger. The user triggers inking either by tapping or by a hard press, and the cursor turns green, indicating the user can begin drawing the stroke. (Figure 5.1 (c)) A stroke is terminated on finger lift.

## 5.1 Designing Inking Trigger

Unlike InkAnchor, the design of FingerTip eliminates scaling and focuses only solving the fat-finger problem [33]. We performed a pilot study comparing different offsets for fingertip, and as a result, we fixed the shape to a component of two lines, point (-100, 100) to point (30, 30) (10.5 mm) and point (-100, 100) to point (-50, -20) (12 mm)relative to the




(a) (b) (c)

Figure 5.1: Process of generating an ink stroke with FingerTip

touch point. One big challenge in designing FingerTip however, is how to trigger inking when users decide to start drawing from a certain point. We examined three alternative techniques: a heavy-press trigger, a second tap trigger, and a volume button trigger.

### Evaluating Pressure Sensing

The pressure trigger, starting inking when a heavy press on the screen is detected, was the first option examined. To begin drawing, a user presses down on the screen. The contact area is analyzed to determine size, and a size threshold initiates inking. Users in FingerTip do not need to hold the hard press throughout one stroke of drawing. Once inking is turned on, users can drawing at a comfortable pressure level, to avoid fatigue when drawing with FingerTip.

We collected initial feedback of the heavy-press trigger we developed from a pilot studies with four participants, whom we presented FingerTip with heavy-press trigger to and asked to complete a few basic drawing tasks. For three of the participants who could trigger inking with a heavy press, participants considered it an efficient mechanism to initiate inking. Given two or three attempts, these participants adapted to the heavy-press trigger, and had few errors.

Since all our participants were recruited from a group of people with high technical background, we believe however, the one case where heavy-press trigger failed would be far more common in the general population. We found in this case the participant could not trigger inking at all and was frustrated at the end of the study, convincing us of the need to better understand the practice of heavy-press.

**Device perspective** Capacitive screens, where pressure on the screen is unavailable, is dominant in the field of touchscreen devices. In order to get a relative degree of pressure, most of the mobile operating systems (e.g. Android and IOS) measure pressure based on the size of the contact area This measure is both inaccurate and noisy. For example, in our Android application we found that the value of inferred pressure varied across different devices preventing the use of single pressure threshold.

**User perspective** Regardless of device dependency, users' behavior affects the design of the heavy-pressure trigger as well. Pressure level varies from different users due to the difference in finger size, the way they press their finger on the screen and which finger they use to draw with. Specifically, the participant who failed to activate the heavy-pressure trigger press in our pilot study, pressed with the index finger perpendicular to the screen such that the contact area of their finger basically remained the same during the study. Thus, no significant fluctuation was observed from the pressure reading. Participants also noted in the pilot studies that they were holding the heavy press throughout the process of drawing a stroke even after being instructed that they did not have to. We understand that it might be difficult for users to perceive the concept of trigger, because to them, the heavy press was perceived as a mode switch rather than a button.

### Designing Intelligent Heavy-Press Trigger

The failure of a single pressure threshold lead to the exploration of a design of a more intelligent heavy-press trigger. A large body of previous work exists on pressure detection on smartphones and on endpoint prediction, providing guidelines for a more intelligent heavy-press trigger. There are two techniques for pressure detection via internal mobile sensors, contact-area based technique leveraging the contact area size interface (e.g. getPressure() in Android system) provided by the mobile software development kit (SDK), and an accelerator-based technique reading data from the internal accelerometer. Essl et al. [12] compared both contact size and accelerometer pressure sensing to an external force sensing resistor, and showed that both contact area and accelerometer techniques performed poorly compared to the external sensor. Strachan and Murray-Smith [34] enhanced the accelerometer-based approach by observing and leveraging the characteristics of muscle tremor to better analyze the oscillation caused by squeezing the screen. Furthermore, recent research [13] has raised the accuracy of three-level pressure detection to around 95% by turning on the vibration motor in the mobile device. With the vibration on, Mayank Goel et al. [13] uses the measurement of a low frequency signal from the accelerometer as a result of vibration being damped by a heavy press. In our initial work on intelligent heavy-press trigger, we tried to replicate the work of Mayank Goel et al. However, we could not accurately measure variations in pressure, a result that seemed tied to a vibration motor with lower intensity output.

**Endpoint Prediction via Motion Kinematics** When we examined the drawing task in a user's input stream, we noted that users would position the endpoint carefully before beginning to draw. More particularly, users would target a specific location on the screen and would move using pointing to that location, stop, activate inking, and then begin to draw. A significant body of research work has examined endpoint prediction [19, 14, 29] and the speed accuracy tradeoff in pointing tasks [23]. Understanding the constraints of mobile interaction, we believed a basic model of endpoint identification for drawing could be developed based on the following assumption: Motion slows down while curvature of trajectory increases when approaching the initial point in a drawing task.

In order to verify our hypothesis, we conducted a study collecting data where we asked participants to perform 2 sections of stroke drawing tasks, with 50 repeated tests of connecting two randomly generated points on a circle. We mimicked the ISO Pointing task [11] (Figure 5.2), where a large circle (diameter = 48 mm) is displayed on the canvas, with 16 small circles (diameter = 1.6 mm) evenly distributed on it. In each repeated test, one of the small circles will be chosen as the starting point, highlighted with thicker black stroke, while one of the other circles, with the thinner black stroke, serves as the ending point. When the fingertip pointer is placed in the range of starting point, the stroke of the corresponding circle will become even thicker, indicating the entry of the fingertip pointer. Participants were then asked to perform a heavy press before drawing a line to the end point. 8 participants (3 females and 5 males) were recruited, all right handed due to the restriction that the fingertip direction is fixed in our test application. Each session took the participants around 10 minutes to complete, and based on the outcome of pilot studies, 5 minutes of break were arranged before the second section to eliminate potential fatigue.

Throughout the test, we kept track of the pressure, position and time for each touch event. In analyzing the characteristics of motion, we performed a cubic spline along both x axis and y axis with the help of the Apache Commons Mathematics Library [39], an open

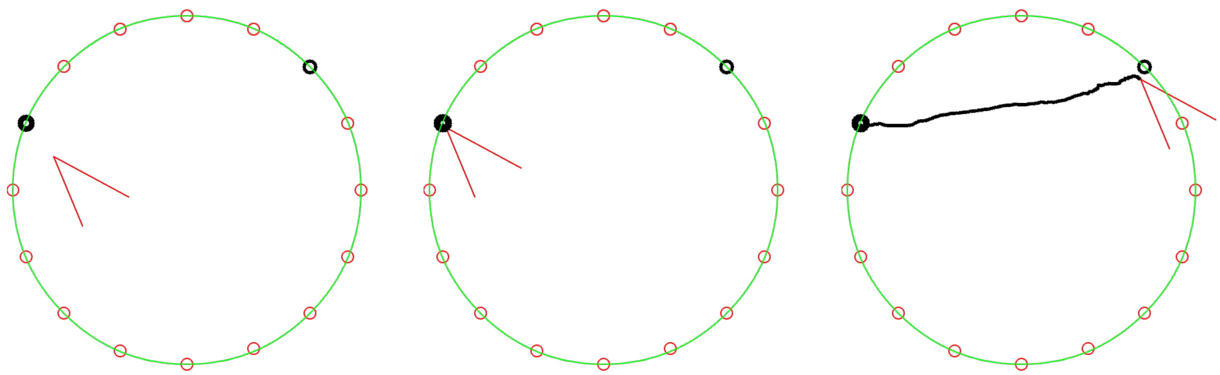


Figure 5.2: Users were asked to connect two randomly generated points on the large circle

source library:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{5.1}$$

$$y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 \tag{5.2}$$

As presented above, we fit x and y independently as a function of time (t), so the constant term of the first derivative $a_1$ and $b_1$ would be the velocity along x axis and y axis, while the constant term of the second derivative $a_2$ and $b_2$ would be the acceleration along the two axes. Thus, we calculated the speed at each touch event:

$$s(t) = \sqrt{v_x(t)^2 + v_y(t)^2} \tag{5.3}$$

We filtered the speed at each point through a low pass filter [27], and the result of speed in sketching motion is quite promising, revealed in Figuref-pressure, where the first 50 tests of a random selected participants are displayed. The speed when approaching the target is low and we can see an instant raise in speed after the participants were notified of the target selection. Participants slowed down for two probable rationale: They were concentrating on the target so they slowed down to be more accurate, or they were performing the heavy press. Either of these rationale however, could make the speed in motion to be a reasonable support for triggering inking. Curvature was calculated via standard curvature formula:

$$\kappa = \frac{|v_x a_y - v_y a_x|}{(v_x^2 + v_y^2)^{\frac{3}{2}}} \tag{5.4}$$

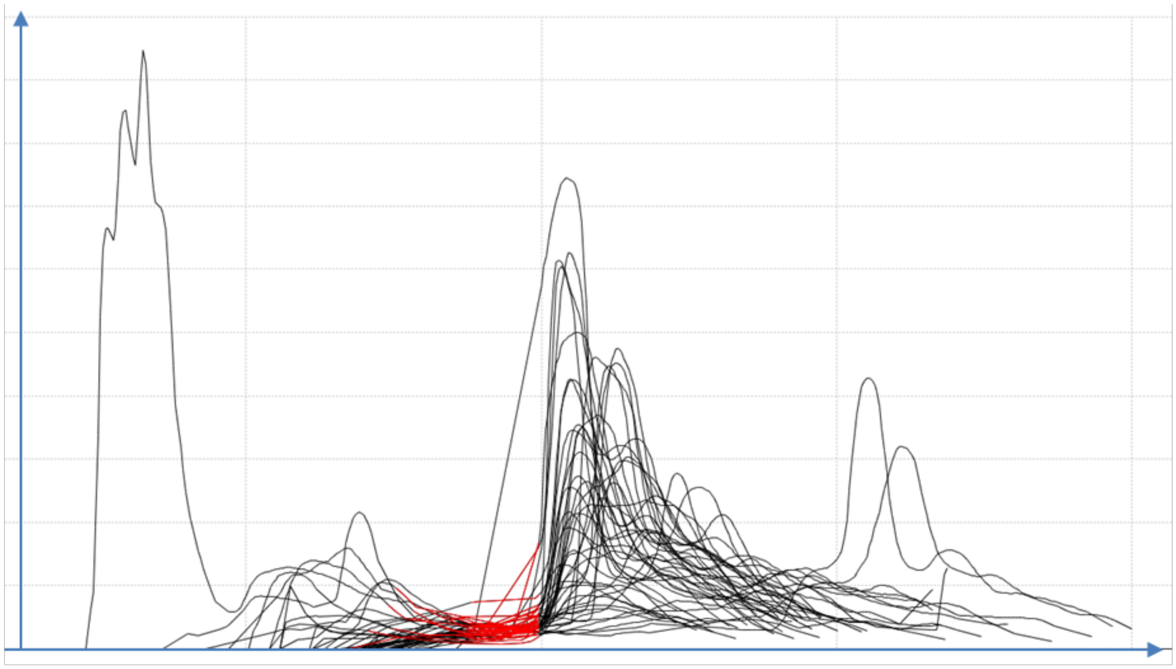


Figure 5.3: X: time; Y: speed

The outcome of the curvature (Figure 5.3) however, failed to confirm our hypothesis. Of the fifty tests, less than 10 present a peek in curvature around the point of entering the target. Moreover, pressure was insignificant, partially an artifact of participants indolent in performing heavy press, as well as inaccuracy of hardware.

Another observation that impedes endpoint prediction via kinematics is that we found participants were adapting to the offset throughout the tests, thus decreasing the time spent in the pre-stroke to a period too short to be detected at the end of the experimental sessions. As presented in Figure 5.4, participants sometimes even start exactly at the target point instantly when pressing on the screen.

### Pilot Study: Comparing Different Triggers

**Final Design** Given the challenges associated with pressure and movement to predict the beginning of an ink gesture, we decided to activate inking through an explicit user action.

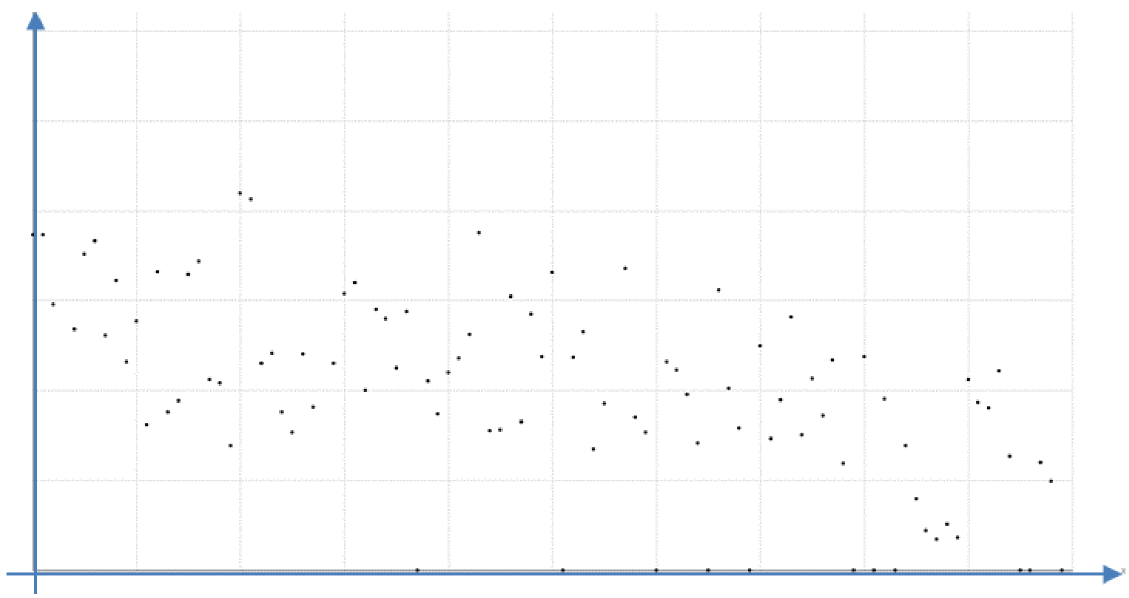


Figure 5.4: X: serial number of trials; Y: time spent on corresponding trial

The two options we considered were an on-screen tap and triggering using a physical device button, the volume button.

With on-screen tap, the user leverages a free finger to tap on the display to toggle inking. We borrow this interaction from Wagner et al.'s work on BiTouch and BiPad. Because modern touch devices support multiple touch points, once can leverage this second contact point as a mode switch while continuing to contact the screen with the primary touch point, the cursor. We contrast this with physical button because of the potential benefits of the use of a physical button. It is frequently the case the multitouch devices experience spurious taps, and our desire was to partially address these by requiring a physical button press.

We conducted a pilot study with 5 participants testing these two different activations. Overall, our participants preferred the on-screen tap because the volume button was both poorly placed and difficult to press while simultaneously interacting with the screen. It is, however, important to stress that the negative results from use of the volume button were a function of the nature of the volume button itself, and not physical buttons in general. However, for the purposes of evaluating Fingertip, we chose to use the on-screen tap activation, a technique that is explicit (addressing challenges with implicit mechanisms such as contact area or pressure) and ergonomically preferred on current hardware.

**Initial Impressions of Fingertip** During our many pilot studies, participants seemed to appreciate the precision provided by Fingertip. Given the solution to the fat finger problem, we noted that many participants found Fingertip both intuitive and effective at enabling accurate inking. This, then, begged the question of whether Fingertip would prove to provide any quantitative benefit over basic touch-based inking. We address this question in the next chapter.

# Chapter 6

# Evaluation of FingerTip

In order to understand whether only eliminating occlusion and fat finger is sufficient to improve accurate finger-based sketching on mobile devices, we integrated InkAnchor with FingerTip, the fingertip taking the place of the non-magnified mode. A study was conducted comparing InkAnchor with and without fingertip to see how fingertip could change participants' behavior, whether shifted pointer supports fine sketching or whether magnified mode is still needed. Participants, both students and professionals, from within our university community were recruited. We specified the restriction of being right-handed due to the constraints of our application. 4 participants (three male; one female) were recruited to participate in an initial 30 minute study and were offered a $5 Tim Horton's Gift Card.

Participants used both the original InkAnchor and InkAnchor with Fingertip on a Samsung Galaxy Nexus phone, running Android 4.2.1, which has a 4.65-inch multi-touch display with a resolution of $720 \times 1280$. The former was named Application 1 (app1) and the latter Application 2 (app2) in our study to eliminate potential bias raised by application names. After being introduced to the basic interactions of both applications, participants were then given the device and allowed to experiment with the applications. Participants were asked to perform the tasks with both applications in the order presented in Table 6.1, again to avoid bias caused by being acquainted with either of the application. After all the tasks were completed, questionnaires were provided for participants collecting their experience using both applications.

| | Map | Flow chart | Math equation |
|---|---|---|---|
| Participants 1 & 3 | app1 ->app2 | app2 ->app1 | app1 ->app2 |
| Participants 2 & 4 | app2 ->app1 | app1 ->app2 | app2 ->app1 |

Table 6.1: Sequence of tasks that different participants performing

## 6.1 Experimental Tasks

The study focused on fine sketching, so the map task in the previous study was kept along with two other newly designed tasks, a flow chart and a math equation. The latter two tasks mimic the behavior of student taking note in class, and require drawings with relative accuracy such that text in the flow chart was kept in the boundary and the math equation was readable.

## 6.2 Experimental Results

All of the four participants had little trouble in finishing the tasks, and gave initial feedback through the questionnaire, where we collected their attitude about performance (how successful they were in accomplishing the tasks with given application), speed (how fast they were in accomplishing the tasks), effort (how hard they had to work to accomplish their level of performance) and frustration (how insecure, discouraged, irritated, stressed, and annoyed they were) for both applications on three tasks independently. Each question was rated on a scale of -3 to +3, and Table 6.2 shows the answer from the participants. Depicted in Figure 6.3, participants' responses to fingertip are unexpectedly negative, where performance and speed are worse than InkAnchor without fingertip while consuming more effort and raising more frustration in most of the cases.

The result not only surprised us, but one of the participants also express his puzzle about fingertip at the end of the study:

> I thought it (fingertip) would be pretty useful when I heard the idea, but it just didn't work out somehow. [p2]

To figure out what impedes the performance of fingertip, we analyzed participants' behavior based on the observation during the study and a close look at their drawings. Map drafting

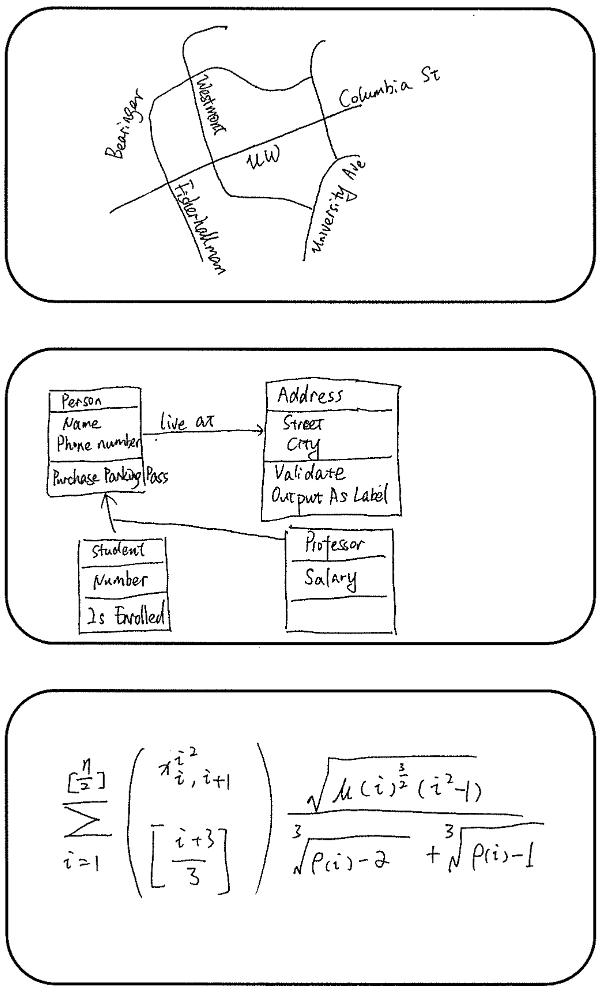
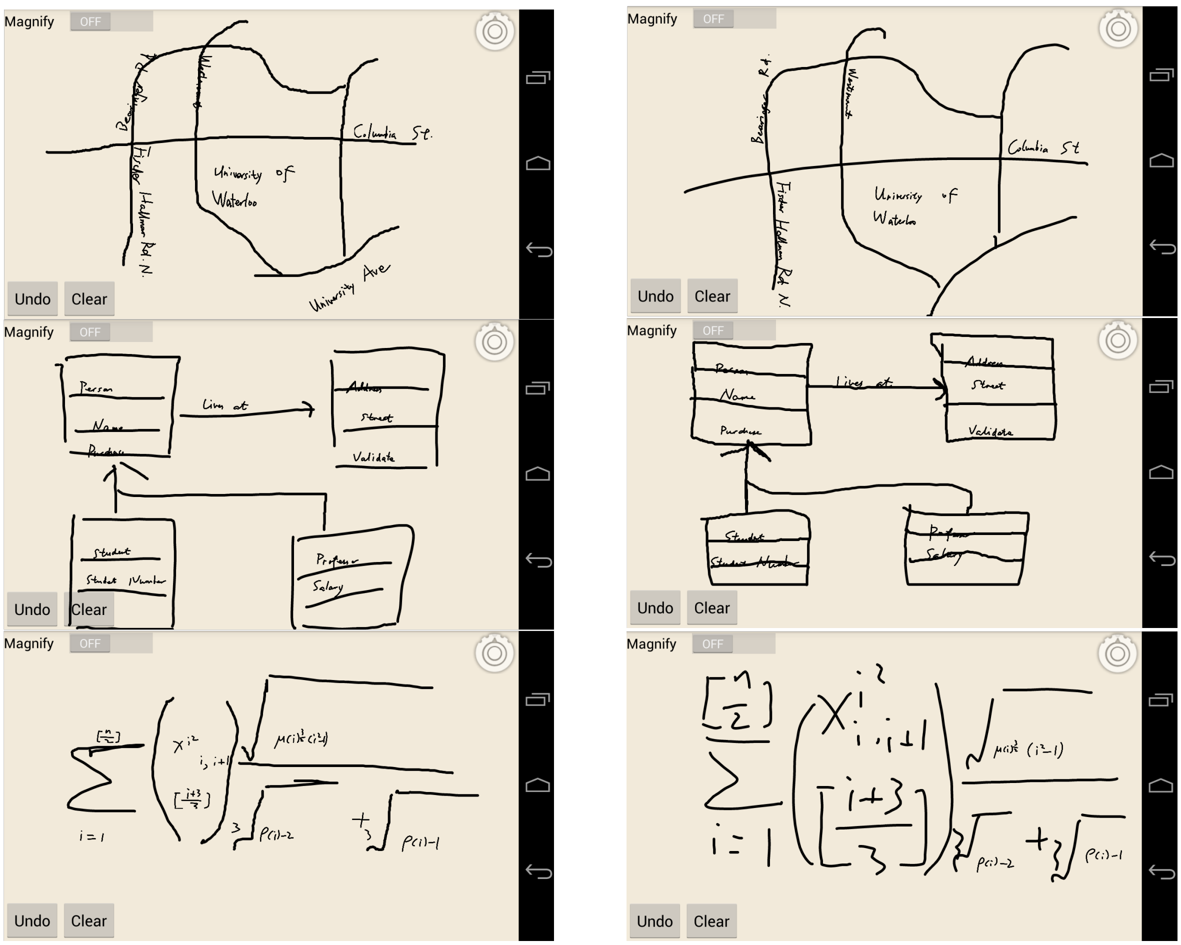



Figure 6.1: Tasks from top to bottom are a map, a flow chart and a math equation




(a) InkAnchor with FingerTip

(b) InkAnchor without FingerTip

Figure 6.2: FingerTip experiment sketches of participants

| Question # | Participants' Ratings | | | |
|---|---|---|---|---|
| 1 | 3 | 1 | 1 | 1 |
| 2 | 2 | 3 | 2 | 1 |
| 3 | 2 | 0 | 0 | 0 |
| 4 | 3 | 2 | 2 | 2 |
| 5 | -1 | 2 | 1 | 0 |
| 6 | 1 | 2 | 1 | 2 |
| 7 | 1 | -2 | 0 | 0 |
| 8 | 1 | 3 | 3 | 1 |
| 9 | 1 | 3 | 0 | 0 |
| 10 | 2 | 3 | 1 | 3 |
| 11 | -1 | 0 | -1 | 0 |
| 12 | 1 | 1 | -1 | 0 |
| 13 | 1 | 2 | 0 | 1 |
| 14 | 2 | -1 | -2 | 0 |
| 15 | 2 | -1 | 0 | 2 |
| 16 | 1 | 1 | -1 | 1 |
| 17 | 3 | 2 | 1 | 2 |
| 18 | 0 | 1 | -2 | 1 |
| 19 | 0 | 2 | -2 | 1 |
| 20 | 1 | -3 | -3 | 2 |
| 21 | 1 | -1 | -1 | 2 |
| 22 | 0 | 0 | -3 | 0 |
| 23 | 3 | 1 -1 | -1 | 2 |

Table 6.2: Ratings (-3 to +3) on questions of each participants

exhibits very few issues with fingertip, since it requires least drawings of the three tasks. We noticed however, that strokes drawn with fingertip are significantly less smooth, and that movement is characterized by an increased number of stop-and-go actions, i.e. a 'jerkiness'. Based on our observations of participant behaviour, it appears that participants slowed down their motion unconsciously when they could see the inking point, resulting in oscillation of their finger and jump in pixels due to the discrete movement on the touchscreen. That could also explain why participants felt fingertip significantly slowed drawing speed.

> I felt it very unnatural to me, it's not the way I'm used to drawing. [P4]

Participants also noted that it was impossible to create ink around the right bottom corner since fingertip cannot reach that region. This problem could be easily resolved if the canvas is well designed, for example with a set of buttons or scrollbars, but it serves as one of the reasons that participants dislike fingertip in its current design. Even though participants expressed negative attitude toward fingertip, we found that in the math equation task, participants did reasonably better than without fingertip (Figure 6.2). The pay-off however, is possibly minor, as, with such negative user-impression scores, we did not pursue accurate scoring of the 'neatness' of the math equations by outside, independent evaluators.

## 6.3 Conclusion

Despite some initial promising commentary from our participants, it seems that Fingertip provided virtually not benefit in the drawing process. The poor usability of Fingertip surprised both us and participants in our pilot study, leading us to believe that it was not implementation tuning that caused the problems. Instead, the idea of an offset cursor seems questionable, causing participants some distraction and failing to address the primary challenges associated with drawing on small screen multi-touch devices. As a result, we believe that focus+context implementations hold significant promise over our offset cursor design.


Figure 6.3: Participants ratings from questionnaires in the study

# Chapter 7

# Conclusion

Informal note taking has been a common and important practice in everyday life [21] long before the popularization of mobile devices. Significant part of users still adapt to the original way of taking notes with paper and pens due to the inefficient support of mobile note taking tools. To address this issue, the idea of designing an improved note taking application on mobile devices emerges via investigating current practice of informal note taking. We presented our work in two parts, we first presented InkAnchor, an ink editor for informal note taking on touchscreen mobile phones. InkAnchor serves as a unique combination of key features, including text anchoring, focus+context input, free-form layouts and auto-delineation of ink groups. The design of InkAnchor evolves through an iterative design process, involving a series of pilot studies and revising repeatedly according to participants' feedback. Our user studies have shown that the anchoring approach performs on par with the state of the art and offers additional editing benefits that stem from its user-driven ink structuring. While InkAnchor addressed text entry, the second part of the thesis focused on elevating fine drawings on mobile devices, where we presented the work of fingertip, a shifted pointing technique to facilitate accurate inputting. Large body of work has been done on designing intelligent triggers, and the results showed it was not sufficient to improving drawing on small screen devices by simply shifting the focusing area.

## 7.1 Future Work

In the first-use study of InkAnchor, we found that most of the participants had the experience that trying to locate the anchor in the non-magnified mode, suggesting a better design for mode switching.

> I guess I probably need some time to get familiar with the app, but I think after I get used to that, that shouldn't be a problem, it's just what I'm thinking. [P2]

And sometimes participants suggested other alternative ways for mode switching.

> Yeah, for the magnify mode, can we try to add some multi-touch interaction, say like slides this way (P1 shows the gesture of tow-finger pan) [P1]

Another better solution inspired by the participants' behavior is that a long press in either mode will call the function that locating the anchor, and especially, when in the non-magnified mode, the mode will be toggled automatically. In this way, users can switch to the magnified mode instantly and the long press gesture in non-magnified mode will not cause any conflict. The current design of InkAnchor can be limiting for writing long chunks. For example, a long word such as watermelon might be have to be broken into two chunks because of the phone's limited screen size. To address this issue, we explored two methods for accommodating long chunks. One approach was automatic scrolling, which shifts the original written ink away from the edge to create more space when the user writes close to the edge. This technique was found useful previously [31]. However, we discovered that auto scrolling often disrupted users. When users wrote fast, their ink often ended up at an unexpected position after auto scrolling. It was hard for them to pay attention to the scrolling while writing, most likely because of occlusion from their finger. It also takes time for users to re-target to continue writing after the scrolling. Instead, a manual approach might have better performance, where the user uses two fingers to pan the original ink (before it is transformed into a line) for more space. Requiring panning can certainly slow down writing, but the user has complete control. Moreover, words that take more than one screen to write are relatively rare for mobile note taking, since people tend to use shorthand for fast note capture. Even fingertip as an application, was not taken as an efficient approach for fine drawings, participants suggested fingertip to be a nice add-on to the current design, where they can easily turn on/off. Thus an unambiguous mode switcher and a better designed inking trigger are needed for such system.

# APPENDICES

## A Questionnaire for InkAnchor First-use Experiment

Assuming that you are going to use a sketching application on smartphones, how do you find the following techniques useful, and how likely you want the techniques integrated into the application? Please select from a scale of -3 to 3.

**Auto-scaling**

Usefulness of the technique

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not useful | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Useful |

Integrate technique into the application?

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not likely | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Likely |

**Rotation of Ink**

Usefulness of the technique

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not useful | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Useful |

Integrate technique into the application?

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not likely | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Likely |

**Automatically wrapping up a word**

Usefulness of the technique

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not useful | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Useful |

Integrate technique into the application?

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not likely | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Likely |

**Boundary checking**

Usefulness of the technique

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not useful | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Useful |

Integrate technique into the application?

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not likely | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Likely |

**Real-time feedback**

Usefulness of the technique

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not useful | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Useful |

Integrate technique into the application?

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not likely | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Likely |

Evaluate the performance of the application on the three tasks you have performed.

**Map drafting**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Poor | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Excellent |

**Room diagram labeling**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Poor | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Excellent |

**Course note annotation**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Poor | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Excellent |

How fast you can input texts/graphics with this application?

**Text**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very slow | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very fast |

**Graphics**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very slow | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very fast |

How accurate you can input texts/graphics with this application?

**Text**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very inaccurate | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very accurate |

**Graphics**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very inaccurate | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very accurate |

Assume you want to sketch, how likely would you use a similar application on a smartphone?

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not likely | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Likely |

Would this help make it more likely that you would sketch texts/graphics on smartphones?

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Not much | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very much |

| Question # | Participants' Ratings | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 3 | 2 | 3 | 2 | 3 | 1 | 2 | 2 | 3 | 2 | 2 | 3 | 3 | 3 |
| 2 | 3 | 3 | 3 | 3 | 2 | 3 | 2 | 2 | 1 | 3 | 2 | 2 | 3 | 3 | 3 |
| 3 | 3 | 3 | 2 | 1 | 2 | 3 | 0 | 2 | 3 | 3 | 3 | 2 | 3 | 3 | 3 |
| 4 | 3 | 3 | 1 | 1 | -2 | 3 | 1 | 2 | 1 | 3 | 3 | 2 | 3 | 3 | 0 |
| 5 | 2 | 2 | 3 | 2 | 2 | 3 | 2 | 3 | 2 | 2 | 3 | 1 | 2 | 2 | 3 |
| 6 | 2 | 2 | 3 | 2 | 2 | 3 | 2 | 3 | -2 | 2 | 3 | 1 | 2 | 1 | 3 |
| 7 | 3 | 2 | 3 | 2 | 1 | 3 | 2 | 2 | 3 | 3 | 3 | 2 | 3 | 1 | 3 |
| 8 | 3 | 2 | 2 | 2 | -1 | 3 | 2 | 2 | 2 | 3 | 3 | 1 | 3 | 1 | 3 |
| 9 | 3 | 1 | 2 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3 | 2 | 3 | 2 | 3 |
| 10 | 3 | 1 | 3 | 3 | 3 | 3 | 2 | 2 | 0 | 3 | 3 | 2 | 3 | 2 | 3 |
| 11 | 3 | 3 | 3 | 2 | 0 | 0 | -1 | 2 | 2 | 3 | 1 | 1 | 3 | 2 | 0 |
| 12 | 2 | 2 | 2 | 3 | 1 | 0 | -1 | 3 | 1 | 2 | 3 | 2 | 3 | 3 | 1 |
| 13 | 0 | 2 | 2 | 3 | -1 | 3 | 2 | 2 | 2 | 3 | 3 | 1 | 3 | 3 | 3 |
| 14 | 2 | 2 | 2 | 3 | -1 | 3 | 1 | 3 | 0 | 3 | 1 | 3 | 2 | 2 | 1 |
| 15 | 2 | 2 | 1 | 3 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 16 | 3 | 2 | 2 | 3 | 1 | 1 | 1 | 3 | 1 | 3 | 1 | 2 | 2 | 1 | 1 |
| 17 | 3 | 3 | 2 | 3 | 1 | 1 | 2 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 |
| 18 | 3 | 2 | 3 | 2 | 0 | 2 | 1 | 3 | 2 | 3 | 3 | 2 | 3 | 2 | 3 |
| 19 | 3 | 2 | 3 | 3 | 1 | 3 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 2 | 0 |

Table 1: Ratings (-3 to +3) on questions of each participants

| | | | | Number of Rating | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Question # | Mode | Average | Median | -3 | -2 | -1 | 0 | 1 | 2 | 3 |
| 1 | 3 | 2.466667 | 3 | 0 | 0 | 0 | 0 | 1 | 6 | 8 |
| 2 | 3 | 2.533333 | 3 | 0 | 0 | 0 | 0 | 1 | 5 | 9 |
| 3 | 3 | 2.4 | 3 | 0 | 0 | 0 | 1 | 1 | 4 | 9 |
| 4 | 3 | 1.8 | 2 | 0 | 1 | 0 | 1 | 4 | 2 | 7 |
| 5 | 2 | 2.266667 | 2 | 0 | 0 | 0 | 0 | 1 | 9 | 5 |
| 6 | 2 | 1.933333 | 2 | 0 | 1 | 0 | 0 | 2 | 7 | 5 |
| 7 | 3 | 2.4 | 3 | 0 | 0 | 0 | 0 | 2 | 5 | 8 |
| 8 | 3 | 2.066667 | 2 | 0 | 0 | 1 | 0 | 2 | 6 | 6 |
| 9 | 3 | 2.6 | 3 | 0 | 0 | 0 | 0 | 1 | 4 | 10 |
| 10 | 3 | 2.4 | 3 | 0 | 0 | 0 | 1 | 1 | 4 | 9 |
| 11 | 3 | 1.6 | 2 | 0 | 0 | 1 | 3 | 2 | 4 | 5 |
| 12 | 2 | 1.8 | 2 | 0 | 0 | 1 | 1 | 3 | 5 | 5 |
| 13 | 3 | 2.066667 | 2 | 0 | 0 | 1 | 1 | 1 | 5 | 7 |
| 14 | 2 | 1.8 | 2 | 0 | 0 | 1 | 1 | 3 | 5 | 5 |
| 15 | 2 | 2.133333 | 2 | 0 | 0 | 0 | 0 | 3 | 7 | 5 |
| 16 | 1 | 1.8 | 2 | 0 | 0 | 0 | 0 | 7 | 4 | 4 |
| 17 | 3 | 2.333333 | 2 | 0 | 0 | 0 | 0 | 2 | 6 | 7 |
| 18 | 3 | 2.266667 | 2 | 0 | 0 | 0 | 1 | 1 | 6 | 7 |
| 19 | 3 | 2.266667 | 3 | 0 | 0 | 0 | 1 | 2 | 4 | 8 |

Table 2: Statistic of each question

# B Questionnaire for FingerTip Experiment

**Performance:**

How successful were you in accomplishing the tasks with given application?

Map

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Failure | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Perfect |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Failure | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Perfect |

Flow Chart

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Failure | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Perfect |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Failure | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Perfect |

Math Equation

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Failure | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Perfect |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Failure | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Perfect |

**Speed:**

How fast were you in accomplishing the tasks with given application?

Map

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Slow | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Fast |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Slow | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Fast |

Flow Chart

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Slow | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Fast |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Slow | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Fast |

Math Equation

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Slow | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Fast |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Slow | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Fast |

**Effort:**

How hard did you have to work to accomplish your level of performance?

Map

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

Flow Chart

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

Math Equation

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

**Frustration:**

How insecure, discouraged, irritated, stressed, and annoyed were you?

Map

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

Flow Chart

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

Math Equation

Application1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

Application2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Very Low | -3 | -2 | -1 | 0 | 1 | 2 | 3 | Very High |

# References

[1] In situ search for active note taking. U.S. Classification 707/770, 382/306, 382/186, 382/187; International Classification G06F17/30; Cooperative Classification G06K9/6255, G06K9/00436, G06F17/30864; European Classification G06K9/00K3U, G06F17/30W1, G06K9/62B6.

[2] Note taker hd. https://itunes.apple.com/ca/app/note-taker-hd/id366572045?mt=8, sep 2014.

[3] Maneesh Agrawala and Michael Shilman. DIZI: A digital ink zooming interface for document annotation. In Maria Francesca Costabile and Fabio Patern, editors, *Human-Computer Interaction - INTERACT 2005*, number 3585 in Lecture Notes in Computer Science, pages 69–79. Springer Berlin Heidelberg.

[4] Claude Barras, Edouard Geoffrois, Zhibiao Wu, and Mark Liberman. Transcriber: Development and use of a tool for assisting speech corpora production. 33(1):5–22.

[5] Michael Bernstein, Max Van Kleek, David Karger, and M. C. Schraefel. Information scraps: How and why information eludes our personal information management tools. 26(4):24:1–24:46.

[6] Christopher Campbell and Paul Maglio. Supporting notable information in office work. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pages 902–903. ACM.

[7] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1 € filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2527–2530. ACM.

[8] Bryan Chartier, 2013.

[9] Luke Dahl and Ge Wang. Sound bounce : Physical metaphors in designing mobile music performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 178–181.

[10] Richard C. Davis, James A. Landay, Victor Chen, Jonathan Huang, Rebecca B. Lee, Frances C. Li, James Lin, Charles B. Morrey, III, Ben Schleimer, Morgan N. Price, and Bill N. Schilit. NotePals: Lightweight note sharing by the group, for the group. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 338–345. ACM.

[11] Sarah A. Douglas, Arthur E. Kirkpatrick, and I. Scott MacKenzie. Testing pointing device performance and user assessment with the ISO 9241, part 9 standard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 215–222. ACM.

[12] Georg Essl, Michael Rohs, and Sven Kratz. Use the force (or something)-pressure and pressure-like input for mobile music performance. In *NIME 2010 Conf. on New Interfaces for Musical Expression*, 2010.

[13] 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*, pages 545–554. ACM, 2012.

[14] N. Hogan. An organizing principle for a class of voluntary movements. 4(11):2745–2754.

[15] Eve Hoggan, John Williamson, Antti Oulasvirta, Miguel Nacenta, Per Ola Kristensson, and Anu Lehti. Multi-touch rotation gestures: Performance and ergonomics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3047–3050. ACM.

[16] Matthew Kam, Jingtao Wang, Alastair Iles, Eric Tse, Jane Chiu, Daniel Glaser, Orna Tarshish, and John Canny. Livenotes: A system for cooperative and augmented note-taking in lectures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 531–540. ACM.

[17] Wolf Kienzle and Ken Hinckley. Writing handwritten messages on a small touchscreen. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 179–182. ACM.

[18] George Labahn, Edward Lank, Scott MacLean, Mirette Marzouk, and David Tausky. Mathbrush: A system for doing math on pen-based devices. In *Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on*, pages 599–606. IEEE, 2008.

[19] Edward Lank, Yi-Chun Nikko Cheng, and Jaime Ruiz. Endpoint prediction using motion kinematics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 637–646. ACM.

[20] Edward Lank and Son Phan. Focus+context sketching on a pocket PC. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 1275–1278. ACM.

[21] Min Lin, Wayne G. Lutters, and Tina S. Kim. Understanding the micronote lifecycle: Improving mobile support for informal note taking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 687–694. ACM.

[22] S. MacLean, D. Tausky, G. Labahn, E. Lank, and M. Marzouk. Is the iPad useful for sketch input?: A comparison with the tablet PC. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '11, pages 7–14. ACM.

[23] David E. Meyer, J. E, Sylvan Kornblum, Richard A. Abrams, and Charles E. Wright. Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action. In *Attention and performance 13: Motor representation and control*, pages 173–226. Lawrence Erlbaum Associates, Inc.

[24] Thomas P. Moran, Patrick Chiu, William van Melle, and Gordon Kurtenbach. Implicit structure for pen-based systems within a freeform interaction paradigm. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 487–494. ACM Press/Addison-Wesley Publishing Co.

[25] Jared Newman. Phablets are a niche, not a fad. *Time*, apr 2013.

[26] Elin Rnby Pedersen, Kim McCall, Thomas P. Moran, and Frank G. Halasz. Tivoli: An electronic whiteboard for informal workgroup meetings. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 391–398. ACM.

[27] R. L. Potter, L. J. Weldon, and B. Shneiderman. Improving the accuracy of touch screens: An experimental evaluation of three strategies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, pages 27–32. ACM.

[28] Yi Ren, Yang Li, and Edward Lank. InkAnchor: Enhancing informal ink-based note taking on touchscreen mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1123–1132. ACM.

[29] Magnus J. E. Richardson and Tamar Flash. Comparing smooth arm movements with the two-thirds power law and the related segmented-control hypothesis. 22(18):8201–8211.

[30] Eric Saund, David Fleet, Daniel Larner, and James Mahoney. Perceptually-supported image editing of text and graphics. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 183–192. ACM.

[31] G. Seni. Treadmill ink - enabling continuous pen input on small devices. In *Eighth International Workshop on Frontiers in Handwriting Recognition, 2002. Proceedings*, pages 215–220.

[32] M. Shilman, Zile Wei, S. Raghupathy, P. Simard, and D. Jones. Discerning structure from freeform handwritten notes. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings*, pages 60–65 vol.1.

[33] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. Fat finger worries: How older and younger users physically interact with PDAs. In Maria Francesca Costabile and Fabio Patern, editors, *Human-Computer Interaction - INTERACT 2005*, number 3585 in Lecture Notes in Computer Science, pages 267–280. Springer Berlin Heidelberg.

[34] Steven Strachan and Roderick Murray-Smith. Muscle tremor as an input mechanism. *UIST*, XP002473042:1–2, oct 2004.

[35] Daniel Vogel and Patrick Baudisch. Shift: A technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 657–666. ACM.

[36] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. Lucid touch: A see-through mobile device. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 269–278. ACM.

[37] Lynn D. Wilcox, Bill N. Schilit, and Nitin Sawhney. Dynomite: A dynamically organized ink and audio notebook. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 186–193. ACM.

[38] Ron Yeh, Chunyuan Liao, Scott Klemmer, François Guimbretière, Brian Lee, Boyko Kakaradov, Jeannie Stamberger, and Andreas Paepcke. Butterflynet: a mobile capture and access system for field biology research. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 571–580. ACM, 2006.