Menu designs for note-taking applications on tablet devices

Atsushi Kitani

2018

筑 波 大 学 審 査 学 位 論 文（博 士）

Doctoral Program in Systems Management and Business Law
(Systems ManagementCourse)
Graduate School of Business Sciences
University of Tsukuba, Tokyo,

ABSTRACT

This research is a study on menu and palm rejection algorithm for note-taking applications. The purpose of this research is to expand the choices of menus that can be used with note-taking applications and to propose practical palm rejection algorithm. This research proposes three approaches to achieve the purpose and to make closer to smooth note taking.

The first approach is reconsideration of menu design. From the view point of note-taking, general linear menus on note-taking applications have drawbacks, one of which is the difficulty of rapid manipulation without occupying the writing space. To solve this issue, we developed an arc-shaped menu and compared the menu to general linear menus that were located on the left of the screen edge. As a result, the manipulation speed of our menu was faster than that of one of the linear menus that popped up on the left of the screen edge.

In the second approach, we propose a palm rejection algorithm. Some existing palm rejection algorithms accidentally remove correct ink strokes. To reduce the frequency of removing correct ink strokes, we developed a palm rejection algorithm and, conducted an experiment that found that our algorithm had fewer occurrences of removing correctly drawn strokes than other applications.

The last approach is, developing a menu invocation method. After conducting an experiment with regard to menu design, we noticed that menu invocation methods and invocation positions also have influence on smooth note-taking. In order to propose a novel menu invocation method, we applied our palm rejection algorithm to invoke context menus. Our invocation method invokes a context menu when the user lifts the writing hand from the screen. Then, we compared our invocation method with other invocation methods used in past research and existing applications, and evaluated which menu invocation methods are manipulated rapidly and where menus should appear. The results showed that when the menu position was set at the left edge of the screen, our menu invocation method was significantly faster than the bezel swipe and long press.

CONTENTS

LIST of FIGURE

LIST of TABLE

Chapter.1 Introduction

It was possible to take notes using applications of laptop computer before the tablet became popular. There were research on how to take notes by means of those applications [10, 11, 12]. However, with the spread of tablets, the real notebook applications [1, 13, 14, 15], that is, applications that can take notes by handwriting has increased.

Such applications have the potential to provide benefits for users. The users of these applications will be able to take notes by using various useful functions that can usually be used on computers, such as changing the colors of strokes, or, copy and paste. Because of such benefits, we expect these applications to begin to be used in classes or meetings. However, when we use existing note-taking applications, we find some difficulties of use. To increase the opportunities of using the applications in classes or meetings, we are sure that the applications need to improve their usability.

1.1 Our viewpoint

To clarify the reasons for difficulties of use, we considered the scenario of using note-taking applications. Note-taking applications are used within specific scenarios in classes or meetings.

In the scenario of using a real pen and notebook, when users need to change the drawing color, they exchange the writing pen with one from a pen case. They want to be able to switch the writing pens rapidly, because they need concentrate on taking notes.

In the scenario of using a note-taking application, menus play the same role as the pen case. A menu is a list of functions displayed on an application. Users need to be able to switch the writing functions rapidly through menus. Menus should be designed to appropriately suit the requirement. However, the designs of most menus for note-taking applications do not appear to have considered the scenario and the requirement. This viewpoint is the beginning of our research.

1.1.1 Menu criteria for note-taking applications

There is various research on menus and various handwriting applications. Past research and applications include key factors to improve the usability of note-taking applications. Therefore, we survey past research and develop criteria that illustrate the factors that menus for note-taking applications should cover. We evaluate past research and menus on current handwriting applications from the viewpoint of these criteria.

1.1.2 Menu design for note-taking applications

According to the criteria, we hypothesize that menu design has an influence on rapid menu manipulation. While taking notes, users should be able to select items on a menu rapidly because they want to concentrate on what they need to write.

Most traditional applications [16, 17] have their menus placed permanently at either side, or the top or bottom edge of the screen. These menus are called permanent menus [18]. When menus are displayed at all times, they occupy the writing space of the screen. Tablet devices have a relatively smaller screen than that of a general computer, and thus, for note-taking applications, it should avert the situations where screens are occupied by menus.

To set the entire screen as the writing space, context menus, which pop up near a manipulating finger or stylus, have become one of the solutions for menus for note-taking applications. Whereas, context menus still have some drawbacks. Their designs are rectangular, square, or sometimes round. When such context menus pop up, they hinder the content underneath and may disturb note-taking activity. Furthermore, because of the finger or stylus, some parts on the menu are hindered, and thus, the selection speed of items in the hindered part will be delayed.

If the items on the menu can be selected rapidly, then the writing procedure is smooth and the usability of these applications increases. To achieve rapid menu selection, we propose the arc menu. The arc menu is an arc-shaped context menu and is intended for selecting items rapidly. To evaluate the design of the arc menu, we compared it with general linear menus.

After the evaluation of the arc menu, we found that there were mainly two other problems for taking notes using applications: a manner of taking notes and a way of menu manipulation.

1.1.3 Reducing the occurrence of removing correct ink strokes

When we undertook the experiment of the arc menu, we instructed subjects to write some characters without putting their hand on the tablet screen. This was because the experimental application was not equipped with palm rejection. Palm rejection is a function that allows users to put a palm on the screen while using the note-taking application. It distinguishes which touches draw strokes and which touches should not draw strokes.

When users are taking notes by means of a tablet device with a note-taking application that does not have palm rejection, multi-touch displays compel users to float the hand. This writing posture is a cause of fatigue [19]. Palm rejection allows users to put their hand on the screen, which helps to reduce fatigue. Users can take notes using the same posture that they would adopt when using a real pen and paper notebook. Therefore, palm rejection is assumed to be important to take notes smoothly.

During our investigation of existing note-taking applications, which are equipped with palm rejection, we found a strange behavior. When users are writing something on the screen, some applications remove correct ink strokes, which is counter to the users intention. Moreover, this occurs after the stroke has been drawn. Users are compelled to rewrite the removed strokes, and this has a crucial impact when trying to take notes rapidly. Such an unusual interaction provides to be awkward for users.

Classification between the hand and the pen nib is performed while the hand and the pen nib are in contact with the screen. The unusual interaction occurs when a touch point once classified as a pen nib and then the touch point reclassified as a hand while drawing strokes. Through the adjustment of parameters, it is possible to make it easier to classify a touch point as a pen nib. If touch points are loosely classified as pen nibs, there will be a lot of unintentional strokes. Conversely, if touch points are too strictly classified as pen nibs, strokes may not be drawn correctly or they may be removed afterward.

To reduce the frequent removal of correct ink strokes after a stroke has been intentionally drawn, we developed a palm rejection algorithm, which is a combination of two logical processes: a machine learning technique and an occlusion area of protection. The algorithm of the machine learning positively classifies a touch point as a pen nib. Then, an invisible occlusion area protects against unintentional strokes. The algorithm was compared with other palm rejection techniques through existing applications.

1.1.4 Approaches for menu invocation and invocation position

The experiment of the arc menu focused on the menu designs, and thus we left menu invocation methods out of consideration. Whereas, before selecting items from a context menu, users need to invoke the menu. The invocation speed and the invocation accuracy are crucial to the usability of note-taking applications. Additionally, menu invocation positions also affect the usability.

Menus can be invoked in various ways and from various positions. For instance, most applications adopt long press, which is also called long tap, to pop up context menus. Although long press is a common way to invoke menus, users need to press a screen for a certain amount of time, and it is contrary to smooth note taking. Some multi-touch interaction [2, 20] approaches require users to manipulate complicated gestures, and users need to change their writing posture. When menus are invoked far from the writing finger or the writing stylus, it takes time to move the finger or the stylus to the menu position. Therefore, suitable menu invocation methods and menu invocation positions should be considered to improve the usability of note-taking applications.

When we developed the palm rejection algorithm, we found that the algorithm can be applied to menu invocation. Thus we developed a menu invocation method and named it palm lift invocation (PLI). PLI does not require any specific invocation interaction, except lifting the manipulating hand. When users want to change functions using menus, they lift their hand from the touch screen, so that a context menu appears under the hand. To specify a suitable menu invocation method and menu invocation position, we developed an experimental application and compared it with several menu invocation methods and menu invocation positions.

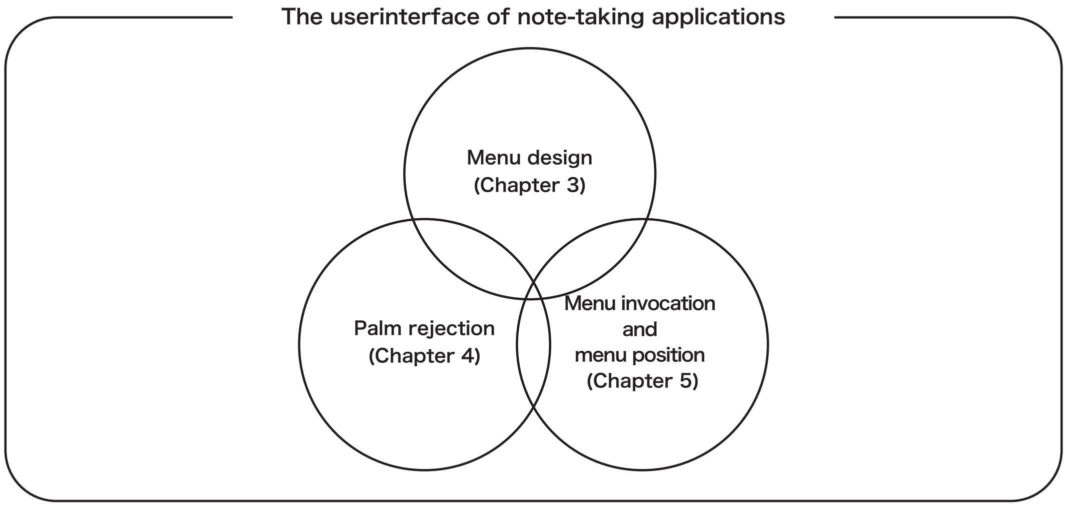


Figure. 1.1: Image of the research coverage and corresponding chapters

1.2 Research questions

For each research point, we asked the following research questions, and we answer them in each chapter.

1. Does the menu design we propose make smoother the task of taking notes compared to a general menu design?
2. How can palm rejection reduce the frequency of accidentally removing correct ink strokes?
3. Which menu invocation methods are manipulated rapidly, and where should menus appear?

Figure. 1.1 shows the schematic diagram of the research coverage and corresponding for chapters.

1.3 Purpose of this research

This research covers menu design, the palm rejection algorithm, and the menu invocation method and invocation positions. The purpose of this research is to expand the choices of menus that can be used with note-taking applications and to propose practical palm rejection algorithm.

1.4 Structure of this research

This paper is structured as follows:

In Chapter 1, we introduce the background and purpose of our research.

In Chapter 2, we examine traditional menus and menus that are included in current applications to discuss which of them are suitable for note-taking applications.

In Chapter 3, we describe the development process of the arc menu and evaluate it. The experimental processes and methods are revised for the following experiments.

In Chapter 4, we propose a new palm rejection algorithm that reduces the frequency of removing correct ink strokes. Based on the algorithm, we develop an experimental application and compare it with other consumer applications.

In Chapter 5, we propose palm lift manipulation (PLM) as a suitable design for natural menu manipulation for taking notes on tablet devices. Also, we develop an experimental application with several menu manipulation techniques, including PLM. We conduct two evaluations: one for menu invocation and the other for the menu position to show the effectiveness of PLM.

In Chapter 6, we discuss the results and conclude remarks.

Chapter.2 Related work

2.1 Introduction

The purpose of this chapter is developing criteria that menus for note-taking applications should satisfy, and discuss which menus satisfy the criteria. To develop the criteria, we investigate research on past menus, menus of existing handwriting applications, and touch interactions.

Note-taking applications have been used since tablet devices were introduced. There is various research on menus, and some menus are considered to be suitable for note-taking applications. However, it is not clear which menus are suitable for note-taking applications. We consider that, this is because there are no concrete criteria. Handwriting applications can be roughly divided into two types. One is applications for drawing pictures and another is applications for taking notes. Since each purpose is different, required functions will also be different. When drawing pictures, users do not need to manipulate quickly, and the menus should be able to load many colors and brush styles. On the other hand, when taking notes, it is possible to assume that note-taking applications will be used in classes or meetings, where users need to take notes quickly with intense concentration. It is not necessary that menus have many colors and brush styles, but they should be able to be manipulated quickly.

In order to develop the criteria, it is important to consider how any given menu is manipulated. Most of the research that has been done on menus were operations that were manipulated through the use of a pointing device rather than by multiple fingers. However, in a note-taking application using a tablet device, it is conceivable to take notes using a finger or a stylus, or even manipulate menus with a plurality of fingers. Furthermore, when we think of specific scenarios such as dotting or making strokes, some of the interactions like double taps or dragging, may conflict with those actions of taking notes. Thus, we should take all such scenarios into consideration and investigate the suitable interactions for menu manipulation.

Table. 2.1: Requirements of menus for handwriting applications

Purpose	Number of items	Rapidity	changing items
Artwork	Many	Not required	Many
Note-taking	Few	Required	Few

This chapter is structured as follows: in Section 2.1, we present the necessity to give the menu criteria for note-taking applications and then, in Section 2.2, we show our criteria, in Section 2.3, before researching menus, we review touch interactions, in Section 2.4, we research menu invocation, in Section 2.5, we survey menu selection, in Section 2.6, we discuss each menu and clarify the suitability for note-taking applications, and in Section 2.7, we provide conclusions.

2.2 Menu criteria for note-taking applications

In this section, we first classify the purpose of handwriting and the difference in the screen size. Then, we give the menu criteria for note-taking applications.

2.2.1 Purpose of handwriting

We classify handwriting applications based on their purpose. One purpose is for creating artwork, while the other is for note-taking.

Menus for artwork applications require various colors, brushes and functions. In the artwork applications, users frequently change colors and brushes using menus.

By contrast, Kim et al. [21] found that users of note-taking applications in class did not require many colors, brushes and functions. Note-taking is a complex activity, which imposes a heavy load on the device's working memory [22, 23]. While taking notes, users want to concentrate on what they are writing and do not want to spend time selecting items from menus. Therefore, the least number of items and functions possible is required. We summarize the requirements of menus for handwriting applications in Table 2.1.

2.2.2 Difference in the screen size

The difference in the screen size on tablet devices affects the usability of the application's menu. Therefore, we need to understand the difference in screen size and its effect. Recently there are various sizes of tablet devices available. We classify devices into the following three categories according to their screen size:

Large tablets : have a screen greater than 12 inches. For examples, iPad Pro and Surface pro 4 are included in them. This type of tablet is suitable for not only handwritten notes, but also professional uses. Most of these tablets can have an optional embedded hardware keyboard, thus it can be used as a substitute for PC.

Medium-sized tablets : have an approximately 10 inches screen. For examples, iPad and Xpedia Z4 are included in them. Medium-sized tablets are the most popular tablet and are suitable for a variety of uses. However, when compared with large tablets, the drawing area is small. Therefore, users would feel that it is unsatisfactory to take notes.

Small tablets : have an approximately 8 inches screen. For example, Nexus 9 is included in them. Because of the small screen size, some works that require a wider screen size are not be suitable for this type of tablet. By contrast, these tablets are rather utilized for browsing websites or using social network services because of the handy but limited screen size.

For modern applications on tablet devices, most menus are set at either side, or the top or bottom of the screen. When menus appear all the time, they occupy a certain writing space on the screen. Users can accidentally touch menus with their hands or fingers when they put their hands or fingers on the screen. Typically, small tablets are influenced by the occupation problem. Nishimura et al. [24] showed the minimum size of buttons for touch devices that are manipulated by fingers. The minimum size show that menus and buttons should have enough sizes even if the screen size is small.

On the other hand, large tablets have less influence for the occupation problem, because it can have relatively larger writing space. However, users sometimes have to move their hands to the menu from the current writing position and return them to the menu again because of the largeness. Users often repeat this behavior whenever they change colors, pen weights or other tools from the menu. This series of time-consuming activities is called “tool palette round trips [6].”

Medium-sized tablets are influenced by both occupation and tool palette round trips; however, the degree of influence of occupation is less than that one for smaller tablets and the degree of influence of tool palette round trips is less than that one for large tablets.

2.2.3 Problems and drawbacks

When menus are set at either the left, right, top or bottom of the screen, the occupation problem and the “tool palette round trips” influence to the menu manipulation. If there is a button for the purpose of controlling the visibility of menus, then this compensates for the occupation of menus and/or reducing the accidental menu touches; however, more time is required to select menu items.

Introducing context menus solves the problems. Rather than appearing at the edge of the screen, context menus pop-up near a manipulating finger or stylus. When users select items from a context menu, the menu disappears and does not occupy the space.

However, context menus also have some drawbacks, which occur in the process of touch interactions through the context menus. Most typical touch interactions are already assigned as writing activities rather than menu invocation. Additionally, when a pop-up menu hides the content underneath, and disturbs note-taking activity. Furthermore, because of finger or stylus manipulation, a user’s hand or fingers tend to hinder some items on the menu. The last two drawbacks typically influence the note-taking activity when users operate small or medium-sized tablets.

To clarify the problems and drawbacks, we list the following qualitative criteria that the menus of note-taking applications to be satisfied:

1. Menu invocation should not conflict with note-taking activity.
2. The more efficient menu invocation is, the better it is.
3. The faster menu selection is, the better it is.
4. All items on menus should be recognized at a glance.
5. Note-taking activity should not be disturbed by menus and their manipulation.

6. Menus should not occupy the writing space.

Criteria 1 and 2 are for menu invocation. Criteria 3 and 4 are for menu selection. Criteria 5 and 6 influence note-taking activity.

2.3 Touch interactions

In this section, we first briefly look back on the transitions of user interfaces. Second, we clarify the difference between an active stylus and passive stylus. Third, we briefly classify touch interaction. Fourth, we show the relation of touch interaction and active stylus interaction. Then, we discuss the difficulty of distinguishing between intended touches and unintended touches.

2.3.1 The transitions of user interfaces

Currently, most standard operating systems provide the graphical user interface (GUI). Before appearing the GUI, people needed to use a character user interface (CUI), and it was difficult to novice users. One of the first personal computers (PCs) that had the GUI was the Xerox Alto and it was designed during early 1973. The Xerox Alto had the GUI, keyboard and pointing device [25]. Xerox Star, Apple Lisa and Macintosh OS were early computers which have menu systems [26, 18]. Gradually, people were getting used to using PCs. After a time, a touch interface appeared. We did not think that the touch interface spread to the general public, however, once Apple's iPhone was released, a multi-touch interface became popular. Furthermore, iPad and other tablet devices extended the capability of the multi-touch interface.

2.3.2 Active stylus and passive stylus

Currently tablet devices embed multi-touch interface. Most people manipulate applications by fingers, however, in note-taking applications, the point to be especially taken into consideration is that a stylus is frequently used. There are several types of styluses, of which two types this paper covers. One is a passive capacitive stylus, which we simply call a stylus. These styluses typically have a large tip made of rubber and work in the same manner as finger touches.

The other type is an active stylus. Active styluses include electronic components and allow users to write directly onto a touch screen. Electromagnetic induction enables the sensing of pen pressure and tilt. Some active styluses have input buttons. Therefore, active styluses enable users to have various interactions. However, an active stylus is generally larger than a passive capacitive stylus, and tends to be expensive.

2.3.3 Classification of touch interactions

We consulted the following guidelines: Apple iOS Human Interface Guidelines (Apple iOS HIG) [27], Google Material Design (Google MT) [28] and Microsoft Universal Windows Platform touch interactions (MS UWP) [29]. There are small differences of some interaction names in mentioned guidelines. For instance, Apple and Microsoft use "tap," while Google uses "touch." We mainly use the simplest keywords from Apple iOS HIG in this paper. As an exception, we use "long press" rather than "touch & hold" because "long press" is more appropriate as the expression of the touch interaction.

From the guidelines, we consider touch interactions in Table 2.2 that are available on most tablets and familiar to tablet users. We classified those common interactions as standard touch interactions (STIs).

Whereas, the other interactions are classified as custom touch interactions (CTIs). CTIs include multi-touch interactions and the combination of two or more touch interactions. We add active stylus manipulation into the set of CTIs.

2.3.4 Variation of CTIs

Although CTIs are unfamiliar, they, which are regarded as extensions of the multi-touch interactions, have widely expanded the purpose of manipulation. Harrison et al. [30] proposed unique CTIs that were inspired by the manipulation of physical tools from the real world, such as a tape measure, a rubber eraser etc. They simulated the way of using those physical tools and attempted to simplify difficult touch interactions. We can also use both hands for manipulation [31, 32].

Novel multi-touch interactions based on recognition of hand gestures have been proposed in the field of multi-touch tabletop devices [33, 34, 35, 36, 37]. Most research focuses on how to distinguish hands on large screens, and how to display and manipulate menus near the hand positions. However, it may not be feasible to adopt them

Table. 2.2: The names of STIs by each guideline and the descriptions

Apple iOS HIG	Google MT	MS UWP	Description
Tap	Touch	Tap	lightly hitting a screen with a finger or stylus
Double tap	Double touch	-	to tap twice quickly
-	Double touch drag	-	a combination of double tap and drag, to keep a finger or stylus on a screen after the last tap and drag
Drag	Drag	Slide	to move a finger or stylus while keeping the finger or stylus on the screen
Flick	Fling	Slide	to move a finger or stylus with a short sudden movement on a screen
Swipe	Swipe	Swipe	to slide a finger or stylus quickly on a screen
Touch & hold	Long press	Press & hold	to press a screen and hold for an arbitrary amount of time
-	Long press drag	-	a combination of long press and drag
Pinch in/out	Pinch open / closed	Pinch / stretch	to place two fingers on a screen and move them together or apart to make an object on the screen appear smaller or larger
-	Rotate	Turn	to place two fingers on a screen and turn them together

for menu manipulation on note-taking applications. Because for the tablet devices the whole screen will be used and, it is inefficient to spread hands to manipulate the menu while holding a stylus and writing letters or figures.

Active stylus interaction and pressure sensitive interaction are currently available for some touch devices. Combinations of multiple touches and stylus manipulation have also been proposed [38, 39, 40, 31]. Yang et al. [41] proposed pen-based user interfaces and showed five mode switching techniques:

- press barrel button
- press and hold
- use non-preferred hand
- pressure-based mode switching
- use the eraser end of the pen

These techniques are proposed as mode switching techniques rather than menu manipulation, though we can manipulate menus by means of these techniques. Users are very familiar with using pens, and therefore, stylus manipulation is intuitive for most users. A negative point regarding use of an active stylus is that, for most tablet devices, it is optional and is still expensive.

2.3.5 Intended touches and unintended touches

CTI does not only have advantages; it especially has negative influence for handwriting applications as well as multi-touch interfaces. When a user tries to write something on a tablet using a general handwriting application, multi-touch interaction compels the user to float the hand above the screen to avoid accidental inking. Because this unnatural way of writing produces difficulties for handwriting applications [19], a function called "palm rejection " is crucial. Palm rejection is a function that distinguishes intended touches from unintended touches, and prevents accidental inking [42].

Generally, there are two approaches to avoid accidental inking: one is to positively use various functions that are embedded into hardware, such as using an active stylus, which is distinguished as a specific touch from other general finger touches. The second approach is to focus on multi-touch interaction itself, and solve the problem using software algorithms, which does not depend on specific hardware or devices. Current capacitive touch devices can correctly receive all touches through the use of software algorithms; however, they still have difficulty in distinguishing which touches are intended.

We implemented several approaches for palm rejection using software algorithms. One well-known approach is to prepare a specific region in which applications ignore all touches. This approach allows a user to put and/or place the hand on that region of the screen. All touches outside that region are recognized as intended touches, so that ink strokes are made there.

Although this is a simple and reliable solution [43], users need to decide the position of the specific region manually according to where they want to write.

Daniel et al. [44] presented how to handle the occlusion area of a palm, forearm and pen nib while using a tablet device. They shows that using a scalable circle and pivoting rectangle can distinguish the pen nib from a palm or forearm.

Schwarz et al. [45] proposed a novel solution using spatiotemporal touch features.

Every 50 ms up to 500 ms, their method iteratively votes on whether all touches are intended or unintended using a decision tree, which is a kind of machine learning technique. Their solution is currently the basis of palm rejection.

Using specific active styluses to prevent unintended touches is also a feasible approach. Annett et al. [46, 42] considered the problem of classification speed for spatiotemporal touch features, and emphasized the effectiveness of distinction when using active styluses. Whether using software algorithms or active styluses to distinguish unintended touches, palm rejection is one of important issues for handwriting applications.

2.4 Menu invocation

In this section, we describe menu invocation methods using STIs and CTIs.

2.4.1 STIs for menu invocation

Particularly while taking notes using note-taking applications, we assume that it is difficult to make use of most STIs for menu invocation. To clarify our assumption, we investigate and discuss general applications.

Investigation

In general applications on smartphones or tablets, double tap is commonly assigned to invoking context menus. Also, we often use long press to invoke context menus. Bamboo Paper, which is a handwriting application for the iPad, has the long press operation to invoke context menus [1]. Figure 2.1 shows the context menu of Bamboo Paper.

WriteOn INKredible [47] is an application that incorporates an interaction, which is swiping from a bezel of a screen, to invoke menus. The interaction is especially named bezel swipe. Users of INKredible invoke a menu by bezel swipe from both sides of the screen.

Discussion

In order to discuss which STIs are reasonable for menu invocation, we clarify what actions are needed for taking notes. Writing requires two basic actions. One is drawing strokes, and another is making dots.

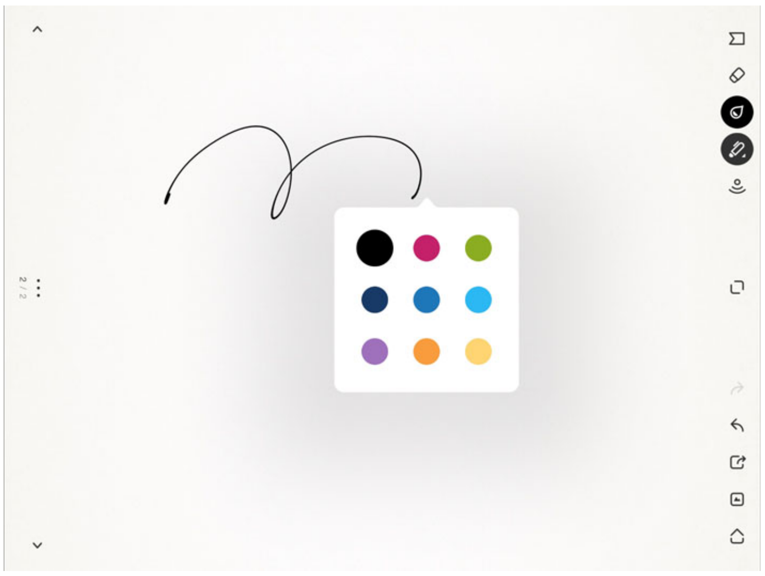


Figure. 2.1: Context menu of Bamboo Paper [1]

While using a note-taking application, users drag a screen by their figure or a stylus to draw a stroke. To brush with a short stroke, they swipe or flick the screen. To make a dot, they tap the screen. To make two dots, they tap the screen two times. These STIs have already been assigned to the basic writing actions, and hence, they conflict with some parts of writing characters. Figure 2.2 shows the image of STIs with the corresponding basic writing actions and the conflicting part in characters.

Zooming in/out an art board and turning the art board are not writing actions, though, they are basic manipulations for note taking. Pinch in/out are generally assigned to zooming in/out on an art board. Turn is generally assigned to turning an art board. Figure 2.3 shows the image of STIs with the corresponding basic manipulations.

If we assign one of these STIs to menu invocation, then, menu invocation will conflict with the corresponding basic actions or manipulations. For instance, if we assign the tap interaction to menu invocation, it conflicts with making a dot. In this case, an application will not be able to distinguish between invoking menus or making a dot. And hence, we cannot assign these STIs to menu invocation.

Although long press is inconsistent with our criterion 2, it is a reasonable interaction for menu invocation by process of elimination. Bezel swipe is also reasonable for menu invocation. The application can distinguish between a bezel swipe interaction and an ordinary swipe interaction. Thus bezel swipe does not conflict with a writing action.

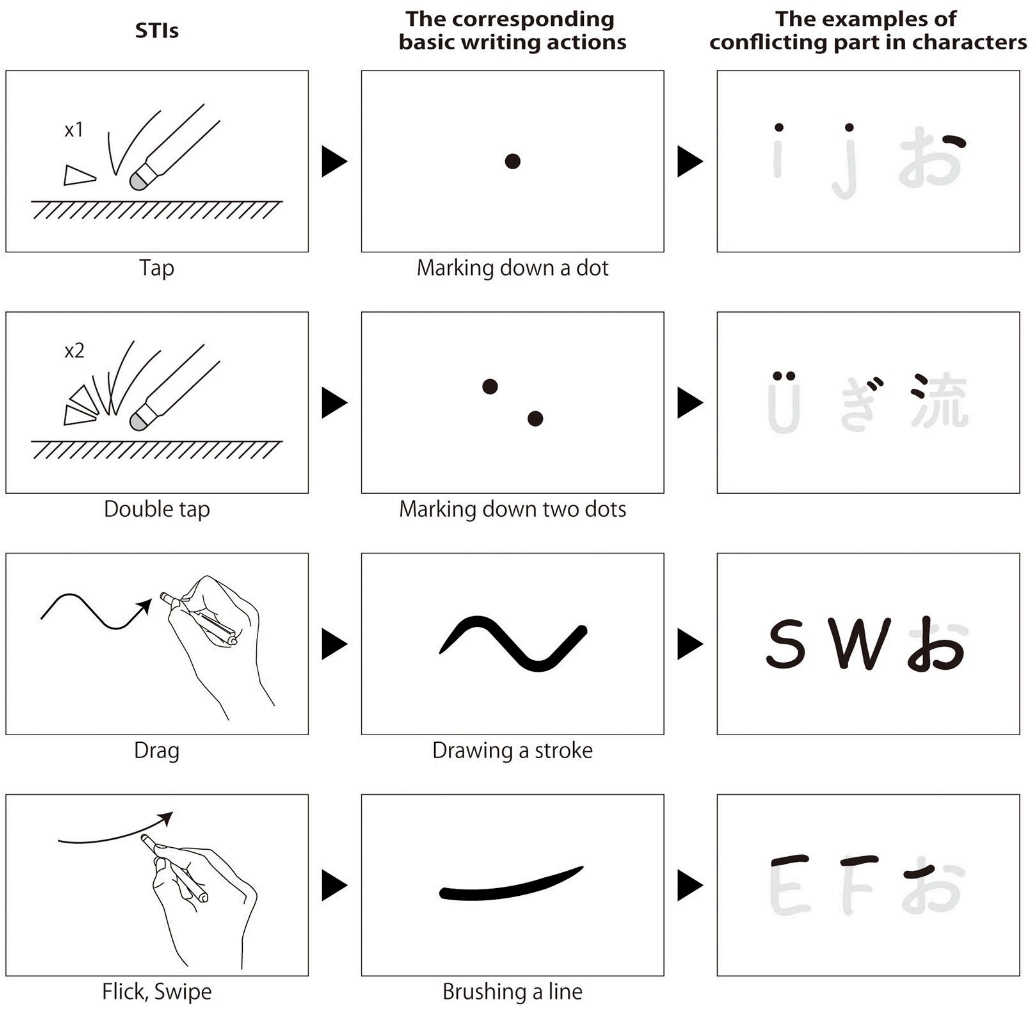


Figure. 2.2: The image of STIs with the corresponding basic writing actions and conflicting part in characters

The negative point of bezel swipe is that users are required to move their finger or a stylus to the edge of the screen. This is inconsistent with our criterion 5.

2.4.2 CTIs for menu invocation

CTIs are more abundant in variation than STIs and therefore, are potential candidates for menu invocation in note-taking applications.

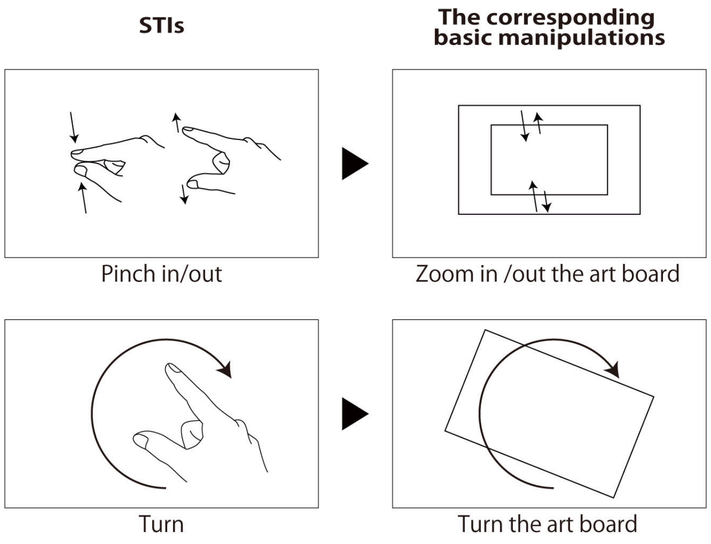

STIs
The corresponding basic manipulations
Pinch in/out
Zoom in /out the art board
Turn
Turn the art board

Figure. 2.3: The image of STIs with the corresponding basic manipulations

Investigation

Lepinski et al. proposed menu invocation method for a touch screen with multi-touch interaction [2] called the multi-touch marking menu (MMM), which is an enhanced menu of the “marking menu” [7], where touch screen apparatus recognizes how many fingers touch the screen and how the fingers are ordered. Some combination patterns of the number and order of touched fingers can be used as instructions for operations, each instruction of which is assigned to a specific menu. Each code is assigned to a specific menu. For instance, an instruction with a thumb and middle finger corresponds to a menu for changing color, and an instruction with a middle finger and little finger corresponds to a menu for changing the pen weight. Figure 2.4 illustrates the use of MMM.

Wagner et al. proposed bimanual interactions for hand-held tablets [20]. They focused on users who hold tablets with a non-dominant hand. We normally operate tablet devices with a dominant hand, whereas the non-dominant hand is often used to hold the device. The idea of their approach was to add small shortcut buttons near the holding hand. The interactions can also be applied to menu invocation. Kurtenbach et al. [48] proposed bimanual interactions for menu manipulation that was manipulated through customized input devices rather than touch devices.

Similar menus called HandMark Menus are preposed by Uddin et al [3]. They propose

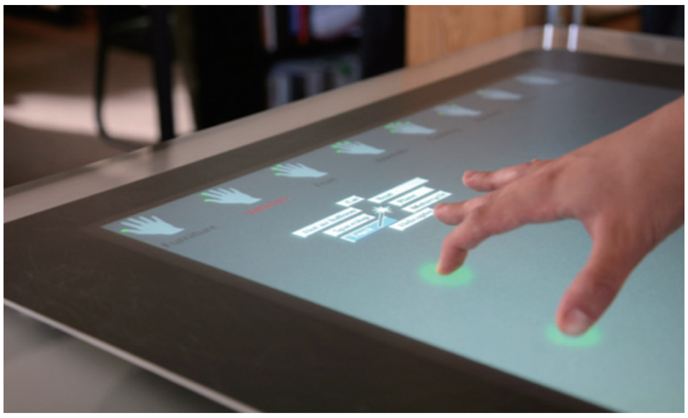


Figure. 2.4: Multi-touch marking menu [2]

two types of HandMark Menus, one is named HandMark-Finger and another is named HandMark-Multi. HandMark-Finger invokes subdivided menus between each finger. HandMark-Multi invokes larger grid menu between thumb and index finger. To invoke menus, the five fingers of the left or right hand are required to touch down the tabletop device. When users use HandMark-Finger, they need to spread the hand to provide space between the fingers. Figure 2.5 shows HandMark-Finger.

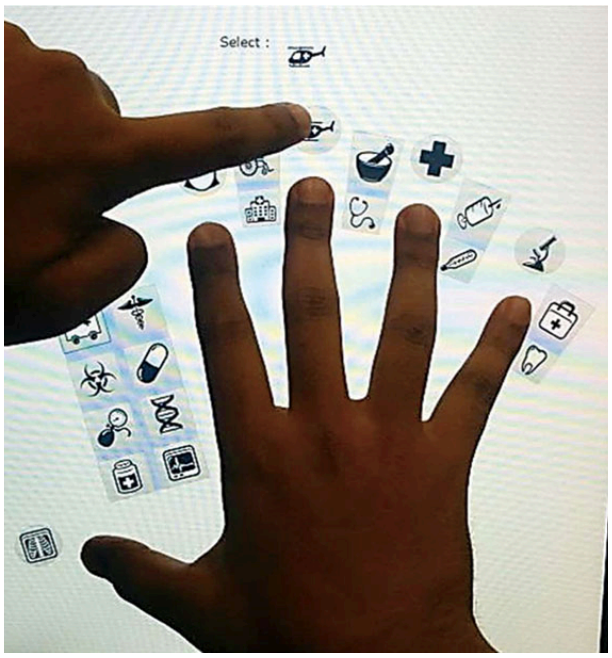


Figure. 2.5: Making a selection with HandMark-Finger [3]

There are other approaches [49, 50] in which sensors are embedded into the barrel, although they are not used for menu invocation. It is possible to make use of those functions with customized styluses for menu invocation.

Discussion

CTIs tend to require relatively complicated manipulation and need a certain manipulation time, and hence, they are inconsistent with criterion 2. It is unlikely that CTIs conflict with note-taking activity. However, if multi-touch interactions are assigned to a menu invocation function for an application which equips palm rejection, multi-touch interactions will be recognized as unintended touches. Therefore, multi-touch interactions are not suitable options to menu invocation.

Active stylus interactions do not conflict with any other note-taking activity.

2.5 Menu selection

For general PCs, we use pointing devices, and are accustomed to selecting items by clicking it. For touch devices, we select menus using our fingers or a stylus. The differences affect the menu selection methods and also influence menu design. Researchers have recognized the differences of them and have been attempting to adjust the design of touch interactions.

First, we survey the menu selection methods using STIs, including pointing devices. Then, we describe the menu selection methods using CTIs.

2.5.1 STIs for menu selection

Menu selection methods have been researched since the beginning of GUI development[51, 52, 53, 54]. Most menus with the methods have been manipulated using pointing devices, which we can adopt in STIs. We focus more on handwriting and menus can be used on tablet devices, and discuss these menus.

Investigation

Traditional linear menus list all items from the top to the bottom of the window, and the selection speed for each item is unequal. To equalize the selection speed, Hopkins et al. designed a unique round menu called the “pie menu” [4]. This menu pops up around the tip of a pointing device. Because the menus is round, each item is located at the same distance from the pointing device, and thus, can be selected with equal rapidity. Items on the menu are selected by dragging a mouse and releasing the mouse button. However, they conducted experiments using a pen-shaped pointing device rather than touch interaction. Figure 2.6 shows the pie menu.

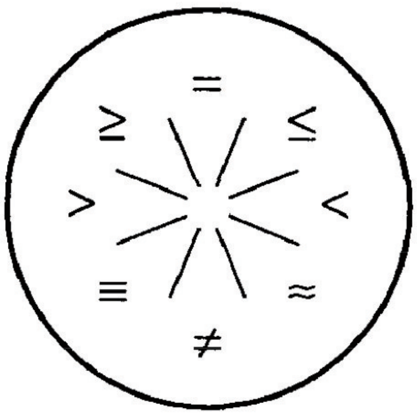


Figure. 2.6: Pie menu [4]

The pie menu has had a great influence on other research, and a large number of menus derived from it have been developed [55, 56, 57, 58], for example, the “occlusion-aware menu” by Brandl et al. [5]. The occlusion-aware menu was designed for large digital tabletops; however, the concept of the menu can be applied to tablet devices. The main concept of the menu is that of taking hand-occlusion areas into account. When a user holds a special stylus, both positions of the stylus and hand are recognized by its system. Then a round menu with items placed only in areas that are not occluded by the hand appears. The angle and position of the occluded area are adjustable. Items on the menu are selected using STIs. Figure 2.7 shows the occlusion-aware menu.

Generally, round menus, such as the pie menu, have two advantages. One is the selection speed. When the pie menu pops up, the pointing device has equal distance to each item. Because the menu is round, users can select items rapidly with small movements, whereas a traditional linear menu lists the items in a line, and some of them are located far from the pointer, such as the items at the bottom of the line, which take

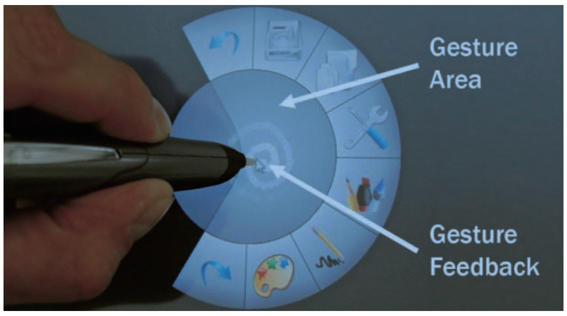


Figure. 2.7: Occlusion-aware menu[5]

more time to be selected according to Fitts's law [59]. Hopkins et al. compared the pie menu with the linear menu and indicated the advantage of the pie menu. Another advantage of round menus is to save space. Because of the small design derived from roundness and popping up on demand, these menus do not occupy a certain space in a screen.

The disadvantage is the limited number of items; because of the design, it is difficult to include many items in the menu. The maximum number of items for round menus is eight. For instance, there are more than 20 items in the File tab on the menu bar of Mac OS X. Practically, the maximum number that we can normally remember at a glance is approximately seven [60].

Fitzmaurice et al. [6] designed an applicable menu specifically for a handwriting application on a tablet device called the " tracking menu." Figure 2.8 shows tracking menu in a pen-based environment. Although the selection method is to simply touch items with a pen nib, the tracking menu has a unique characteristic user interface. The menu has an edge around itself, and when a cursor that is within the boundary touches the edge from inside the menu, the menu moves to keep it under the cursor. This interaction enables the menu to track the cursor and stay close to the writing stylus.

The marking menu, which Kurtenbach et al. [7] designed, also has unique selection methods. There are two selection modes: novice mode and expert mode. When a novice clicks a screen, the menu appears a short time later. Then, the novice drags the cursor and selects one of the items on the menu. However, an expert manipulates the device without showing the menu; the expert already remembers the location of each item. And therefore, when the expert selects items, through rapidly drags in the direction corresponding to the target item. Marking menu items are selected using dragging

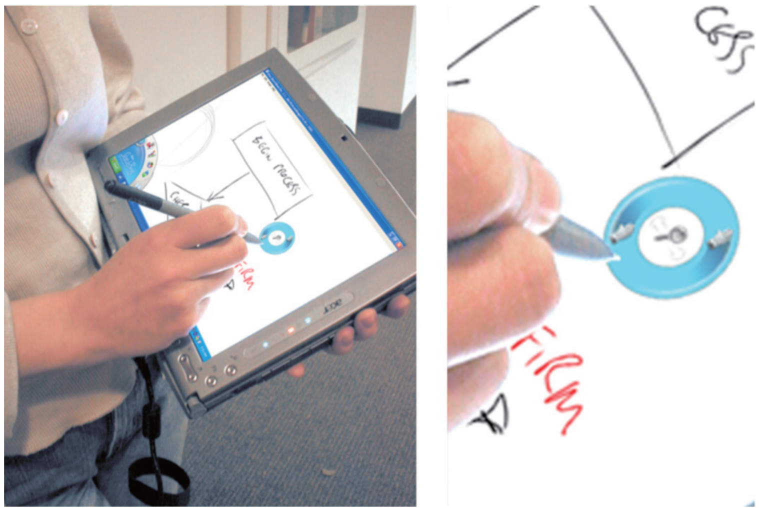


Figure. 2.8: Tracking menu [6]

directions. Figure 2.9 shows the marking menu.

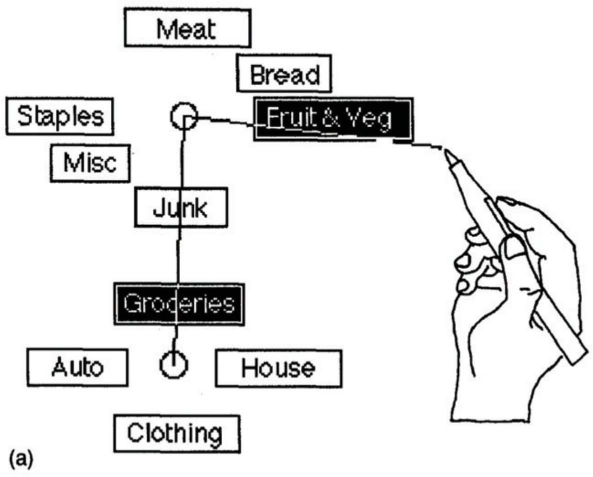


Figure. 2.9: Marking menu [7]

The marking menu has a hierarchical structure. Once a user selects one of items from the menu including a sub-menu, the user needs to drag the cursor in another direction to select the item on the sub-menu. The name "marking" comes from this manipulation. Although the selection speed from sub-menus is slower than that from non-hierarchical menus, the hierarchical structure enables each menu to have more items than those in non-hierarchical menus. Notice that maximum number of items on one menu is eight, whereas the maximum number of sub-menus within a hierarchical structure is two

[61]. More items or more menus cause inaccurate operations. The marking menu was continuously researched [62] and has also given a remarkable influence on other menu approaches [63, 64, 65, 66, 67, 68].

One of the unique menus derived from the marking menu, is the “flower menu.” Bailly et la. [8], which invented this novel menu, mentioned that the flower menu makes it possible to put many items at each menu level. The key point is that they introduced curved gestures into menu selection. The flower menu has marking directions vertically and horizontally, each of which is curved and branched. This characteristic selection method enables each menu to have more items without sub-menus. A single flower menu can theoretically contain a maximum of 56 items. Figure 2.10 shows the flower menu.

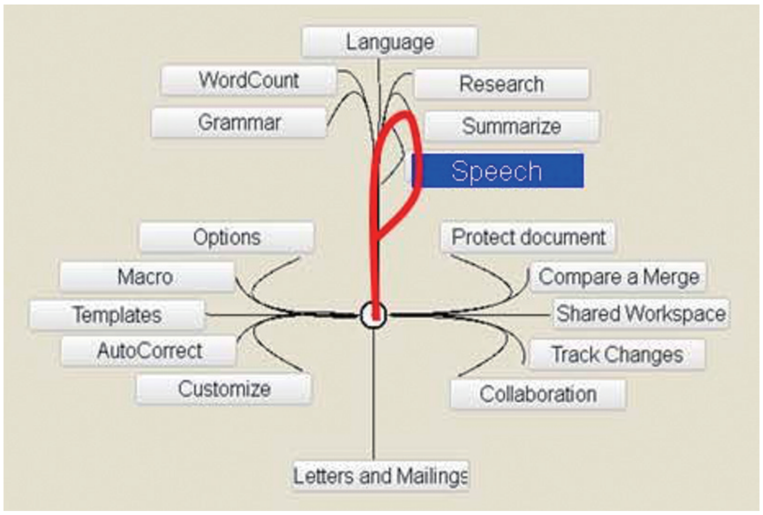


Figure. 2.10: Flower menu [8]

The advantages of the marking menu and other derivative menus are rapidity and the number of items. In particular, expert mode makes a significant contribution to the rapidity of manipulation. The unique design enables the menu to contain a sufficient number of items.

Discussion

The marking menu and other derivative menus have expert mode. Expert mode enables users to select items rapidly; however, in terms of note-taking applications, it is difficult to use expert mode as menus because there are no graphical menus and users cannot distinguish menu selection from general application use.

The occlusion-aware menu takes into consideration touch interaction. The tracking menu is designed typically for handwriting applications, and hence it is also necessary to consider how the menu will be used. Therefore, these menus are feasible options as menus for note-taking applications.

2.5.2 CTIs for menu selection

CTIs can be assigned to menu selection. We investigate trials of CTIs for menu selection.

Investigation

There are menu selection methods using multi-touch interaction [69, 70, 71, 72]. Banovic et al. developed the uni-manual multi-finger pie menu (UMPM) [9]. Figure 2.11 shows the UMPM.

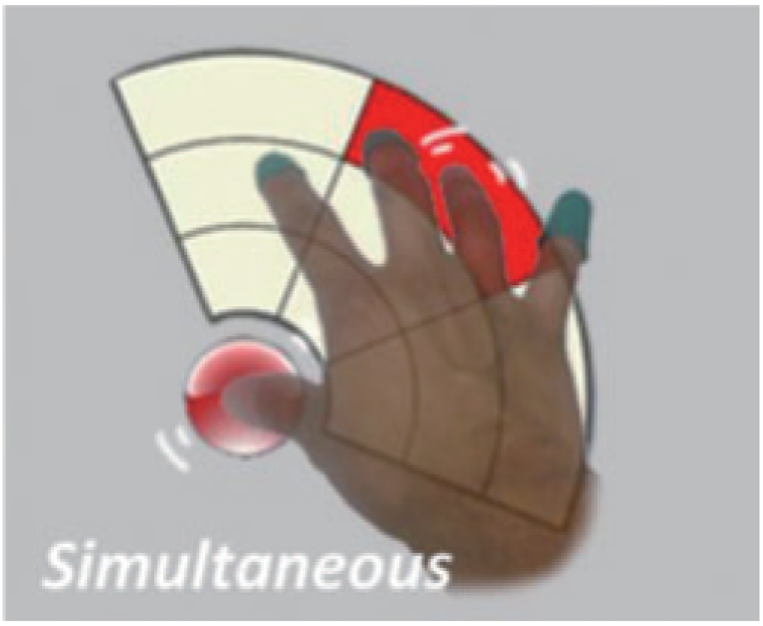


Figure. 2.11: Uni-manual multi-finger pie menu [9]

Roudaut et al. proposed the "RollMark menu" for handheld devices [73]. To select items, users roll a thumb on a touch device. Although this interaction may be useful for devices that are mainly manipulated by a thumb, it is not suitable for note-taking applications because note-taking activity is disturbed by this interaction.

The "layer pie menu" by Xianghi et al. [74] and "tilt menu" by Tian et al. [75] are pen-based interactions. The layer pie menu has four layers at most, and can set more items than the original pie menu. The menu is operated using a specific pen-based computer that can detect pen pressure, and each layer is switched by the pressure. Items on the layers are selected using flick interactions after adding pen pressure. The tilt menu also depends on a specialized active stylus, which can detect three-dimensional orientation information. Tilt menu items can be selected by tilting the stylus.

There are other unique approaches [76, 77] that using 3D sensors to distinguish the hand movements and apply the movements to menu operations. Whereas these approaches require additional sensor equipment such as camera devices and hence, they are not feasible to be used in scenarios of classes or meetings.

Discussion

As mentioned, CTIs require complicated manipulation, and for the same reason, multi-touch interactions for menu invocation may disturb note-taking activity and take a certain amount of time. Thus, criteria 5 and 3 are not satisfied. The RollMark menu has a unique menu selection method, which is a roll of a thumb on a touch device. It might be useful for small smartphones that are mainly manipulated by thumbs, but is not suitable for menu selection for note-taking applications on tablet devices.

Using pen pressure for menu selection also needs a certain time, whereas using pen tilt interaction may be feasible because it does not take time and does not conflict with the note-taking activity.

2.6 Evaluation

We evaluated the following representative items with our criteria in order to clarify the suitability for note-taking applications.

- The context menu of Bamboo Paper (BAMBOO)
- The context menu of INKredible (INKredible)

- Marking menu
- Pie menu
- Tracking menu
- MMM
- Occlusion-aware menu
- UMPM

The results are listed in Table 2.3. "∘" indicates that the menu satisfied the criterion, whereas "✗" indicates that it did not. Some menus could not be evaluated for specific criteria that were relevant to menu invocation, and we indicated this with a "-" mark in the criteria column.

Table. 2.3: Evaluation of STIs and CTIs for menu manipulation

Menu	STI/CTI	Criteria					
		1	2	3	4	5	6
BAMBOO	STI	∘	✗	∘	✗	✗	∘
INKredible	STI	∘	∘	∘	∘	✗	∘
Marking Menu	STI	-	-	∘	✗	✗	∘
Pie Menu	STI	-	-	∘	✗	✗	∘
Tracking Menu	STI	-	-	∘	∘	✗	∘
MMM	CTI	∘	✗	✗	∘	✗	∘
Occlusion-Aware Menu	CTI	∘	∘	∘	∘	✗	∘
UMPM	CTI	-	-	✗	∘	✗	∘

Criteria: 1 Menu invocation should not conflict with note-taking activity. 2 The more efficient menu invocation is, the better it is. 3 The faster menu selection is, the better it is. 4 All items on menus should be recognized at a glance. 5 Note-taking activity should not be disturbed by menus and their manipulation. 6 Menus should not occupy the writing space.

BAMBOO assigns long press to menu invocation and, it takes time to invoke menus. When the context menu pops up, it covers contents underneath and disturbs note-taking activities. Therefore, BAMBOO does not satisfy criteria 2, 4 and 5.

While users are taking notes and want to invoke menus by INKredible, they need to move their hand to the edge of the screen. "Tool palette round trips" require a certain amount of time to invoke menus, and hence, INKredible does not satisfy criterion 5.

When users invoke marking menu as a novice mode, the menu spreads out in all directions and covers the contents underneath. Furthermore, a manipulating finger or stylus will cover the items. Hence, marking menu does not satisfy criteria 4 and 5.

Because of the same reasons as marking menu, pie menu also fails to satisfy criteria 4 and 5.

When users use tracking menu, they can move the menu manually and, the distance of "tool palette round trips" will shorten. However, considering the frequency of moving the menu manually for note-taking applications, it will disturb the smoothness of note-taking activities. Thus, tracking menu does not satisfy criterion 5.

MMM assigns multi-touch interaction to menu invocation. Although the interaction does not conflict with drawing actions, users need to change their manipulating hand posture, and as a result, it takes time to invoke menus. In order to select items, users need to move touching fingers simultaneously in the same direction. Taking the scenario of note-taking by an application into consideration, the way of menu invocation and menu selection will disturb the smoothness of the note-taking activity. Therefore, MMM does not satisfy criteria 2, 3 and 5.

Occlusion-Aware Menu equips an arc shaped menu, though the center is not hollow, and the gesture area provided covers the contents underneath, hence, criterion 5 is not satisfied.

2.7 Conclusion

In this chapter, we investigated touch interactions, menu invocation and menu selection. We developed qualitative criteria for menus of note-taking applications. Then we followed the criteria and clarified which menus in research or applications are suitable for note-taking applications. Investigations and discussions in this chapter contribute to more deeply understanding menus and help to develop menus for note-taking applications.

Chapter.3 Arc menus for note-taking applications

3.1 Introduction

We consider the scenario of taking notes using applications, where note-taking occurs in classes or meetings. In this scenario, users concentrate on what they need to write. Under the situation of concentration, the entire activity of taking notes, which includes menu manipulation, requires rapidity. Therefore, the application menu should allow rapid manipulation. Furthermore, because the size of the display of tablet devices is smaller than that of paper notes, users require as much writing space as possible to take notes.

We surveyed the menus of traditional applications including note-taking applications. We found that most existing applications constantly set a linear menu at either the upper, lower, left, or right side of the screen. However, to take notes, that position of menu mainly causes two problems. The first problem is called "tool palette round trips [6]," in which application users need to move a hand to the menu from the writing position, and after having selected an item from the menu, they need to return their hand to the original position. The series of activity would need to be done iteratively, and thus the activity would disturb rapid manipulation. The second problem is caused by the occupation by the menu of the screen; an area of the display is occupied by the menu itself, so the writing space is reduced.

The context menu is one of the solutions for cited problems. Context menus pop up near the manipulating finger or stylus so that items on the menus can be selected rapidly. Context menus pop up only when they are needed, and hence they do not occupy the screen the entire time.

However, in the case of note-taking applications, it is thought that when a context menu is displayed, it causes two problems. First, when the context menu is displayed, the contents written under it are obscured and disappear. It is also conceivable that a part of the menu is hidden by a finger or a stylus that is operating the tablet, and compared with other items on the menu, thus, hidden items are late to be selected.

The purpose of this chapter is to clarify whether or not the menu design we propose makes smoother the task of taking notes, as compared to a general menu. To evaluate our menu design, we design the "arc menu." The arc menu is an arc-shaped context menu that pops up near a manipulating finger. As a general menu to be compared, we selected linear menus, which are widely used and are adopted in many applications. Then, we compare the manipulation times of the arc menu with the manipulation times of the linear menu displayed at the left end of the display.

The rest of this chapter is structured as follows. In Section 3.2, we describe related work. In Section 3.3, we explain the design process and provide a detailed account of the arc menu. In Section 3.4, we describe the experiments. In Section 3.5, we present our results. In Section 3.6, we discuss the results of the experiment and review our experimental techniques. In Section 3.7, we provide conclusions.

3.2 Related work

For ordinary linear menus, items farther away from a cursor are more cumbersome to choose and the action is delayed due to distance. Hopkins et al. designed a unique round context menu called the "pie menu [4]," for which items appear around the cursor. The pie menu enables users to select items with very short movements of the mouse cursor. All items on the pie menu can be selected at a uniform speed.

In general applications, menus should be able to contain many items. However, because the pie menu is a circular design, it can only contain at most 8 items. This is a drawback of the pie menu, and to overcome this drawback, other menus such as a "marking menu [7]" or a "layer pie menu [74]" were proposed.

On the other hand, in terms of note-taking applications, a number of items are not necessarily required. Kim et al. [21] investigated the requirements for electronic note-taking systems. They found that students want to have simple and quick note-taking systems rather than many optional functions that disturb the basic task of note-taking. Therefore, it seems that the pie menu is suitable for note-taking applications. That said, we recognize that there are two other drawbacks of the pie menu for note-taking by manipulating a finger or stylus. One problem is that when the pie menu pops up, it covers the content underneath. The other problem is that the manipulating finger or stylus hinders part of the menu and some items cannot be seen by the user.

Brandl et al. [5] proposed the “occlusion-aware menu,” which is one of the menus derived from the pie menu. The occlusion-aware menu is designed to be used by a stylus on a large touch screen board on a desk. This menu has an arc-shaped design with items placed only in areas that are not occluded by the hand. The menu can be determined by a system that tracks the position of the hand or the position of the stylus, and can adjust the angle of the occlusion area to the hand.

3.3 Arc menu

While considering a new menu design for note-taking applications, we set the basic design concepts of the menu as follows:

- Users can take notes efficiently using the menu.
- The note-taking area should be as wide as possible.
- The menu should not hide written content.

Based on these concepts, we designed the “arc menu,” which has an arc-shaped design and appears dynamically near a manipulating finger. Thus, when users are taking notes, the arc menu is not shown on the screen. There are two important factors for the characteristics of the arc-shaped design. One is that the menu avoids covering the content underneath or near the finger. Another is that the menu avoids being covered by the hand so that all items on the menu can be clearly recognized at a glance.

We develop a handwriting application using JavaScript and HTML. The application runs as a web application so that most tablet devices connected to the Internet can run the application.

3.3.1 Prototype of the arc menu

Following our concepts, we develop a prototype of the arc menu, which had a gradual arc shape. Figure 3.2 shows the prototype of the arc menu.

The prototype of the arc menu has the following functions: “undo,” “redo,” “red,” “blue,” “black,” “green,” “highlight marker,” “normal marker,” “thick line,” “thin line”

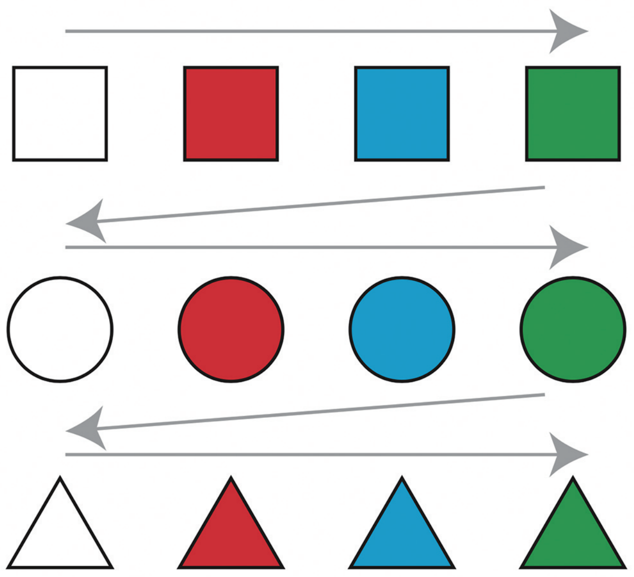


Figure. 3.1: Image for the subjects to draw

and “eraser.” Additionally, “remove menu” and “handle” are added at the top of the menu, where “remove menu” hides the menu and “handle” moves the menu. The other icons on the menu were dummies and did not work.

Preliminary experiment

To confirm the usability of the prototype of the arc menu and to obtain feedback, we conduct a preliminary experiment.

There are four subjects in the preliminary experiment: three males and one female, all aged from 30 to 50 years old. Three subjects are right-handed, and one subject is left-handed. They use an iPad as a tablet device.

We show them an image to draw. Figure 3.1 shows the image for the subjects to draw.

There are three types of shape, squares, circles and triangles, and each is colored white, red, blue or green.

Each picture has arrows so that the subjects can follow the given order to complete the drawing. We instruct the subjects to draw the picture using two types of menus: the prototype of the arc menu and a menu called the two-column linear menu (TCLM).

Figure 3.3 shows the image of the TCLM.

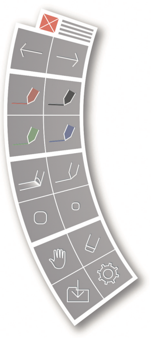


Figure. 3.2: Prototype of the arc menu

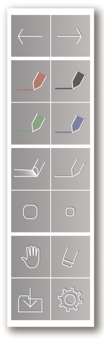


Figure. 3.3: The two-column linear menu

The TCLM is set on the left-hand side of the screen. The TCLM cannot be moved and is displayed on the screen permanently.

Each of the menus has two columns. The height and the width of each item are set in a same size. To invoke the arc menu, subjects need to double tap the screen. Then, the arc menu is displayed in the double tap position.

After a briefing the subjects on the preliminary experiment and its purpose, they spend approximately 10 minutes on a practice trial to become accustomed to the tablet device and the two types of menus. The prototype application records all of the drawing processes and the time for the following actions:

1. leaving a finger after drawing a stroke
2. invoking the menu
3. selecting an item from the menu
4. putting a finger down to draw a new stroke

Results of the preliminary experiment

After the preliminary experiment, we evaluated the recorded data and calculated the set of actions to obtain average scores for each menu.

The arc menu appears near the finger, and the distance between the writing point and items are always the same. Although the TCLM is positioned at the same location, the distance between the writing position and items is not constant, and is mostly further than the distance for the arc menu. Thus, we expected that the arc menu would be manipulated faster than the TCLM. However, all the subjects produced the opposite results; the manipulation speed of the TCLM was faster than that of the arc menu.

Evaluation of the preliminary experiment

We conducted a group discussion and found the causes for the delay in using the arc menu. One reason was that the TCLM was shown all the time in the same location, so it was easier to remember where the items were. The arc menu was only shown using the double tap, and accordingly, it was more difficult to remember the items' positions. Fourteen items were on each menu and this was too many for the subjects to remember all of them.

Another reason for the slowness of the arc menu manipulation was the lack of usability. After choosing an item, the arc menu disappeared after subjects tapped a white space, otherwise, without doing so, the arc menu would stay in the same position.

The subjects commented that this was really stressful because sometimes the arc menu was in the place where they wanted to start drawing and they needed to remove it. Although the arc menu had items that enabled it to move and remove itself, the subjects wanted to remove the menu after choosing the color or line weight, so that they could continue writing smoothly.

Additionally, we found that the double tap for menu invocation took time, and the double tap was difficult for users who were not familiar with tablet devices.

3.3.2 Basic design of the arc menu

According to feedback from the prototype, we design a simpler arc menu. Figure 3.4 shows the image of the arc menu.

The prototype had 14 items; however, we reduce the number to eight. It has the following functions: "undo," "redo," "red," "blue," "black," "green," and "white as an eraser." Additionally, there is a half-sized "remove menu" button. When subjects select items (other than the undo and redo buttons), the menu disappear so that subjects can easily restart writing regardless of the new stroke location. When a user is left-handed,

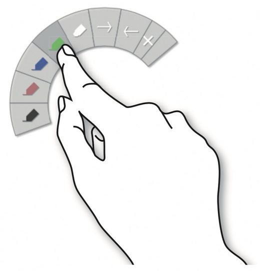


Figure. 3.4: Image of the arc menu

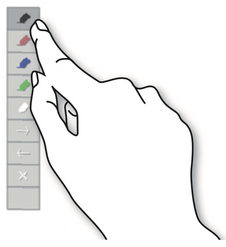


Figure. 3.5: Image of the PopupLM and LM

the angle of the arc menu adjusts to be on the opposite side. Figure 3.6 shows the image of the writing process using the arc menu.

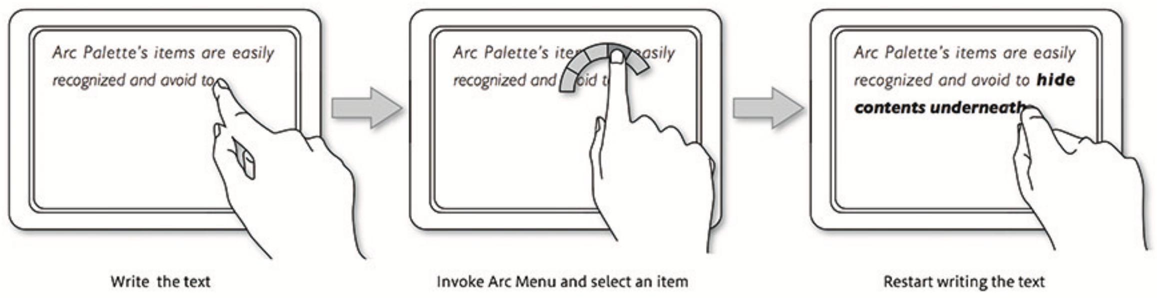

Arc Palette's items are easily
recognized and avoid to
Write the text
Arc Palette's ite
recognized and
asily
oid t
Invoke Arc Menu and select an item
Arc Palette's items are easily
recognized and avoid to **hide**
contents underneath
Restart writing the text

Figure. 3.6: Writing process using the arc menu

3.4 Experiment for the menu manipulation speed of different menu designs

To conduct research for suitable target users and to evaluate the arc menu, first, we develop an application for the experiment. Second, we select menus for the experiment, and then establish an experimental design. Finally, we conduct the experiment.

3.4.1 Application and device for the experiment

We develop an experimental application using JavaScript and HTML. The application runs as a web application so that it can run on most tablet devices. The data from writing strokes, writing timing and menu manipulation are collected by the application. All data is sent to a web server.

Considering the reproducibility of the experiment, we think that the device we use should be popular. We select an iPad for the experiment because it is one of the most popular devices, and thus, using the device will contribute to being easy to reproduce the experiment. The screen is set to be horizontal because the writing direction moves from left to right and the horizontal layout fits this movement.

3.4.2 Menus for the experiment

We select linear menus for comparison with the arc menu because linear menus are the most popular menus for note-taking applications.

Typical note-taking applications set their menus on the left-hand side permanently. However, this occupies the writing space of the screen. To avoid occupying the writing space, context menus are equipped in some applications.

Therefore, we set two types of linear menus. One is called the pop-up linear menu (PopupLM), which appear and disappear (as do the arc menu) and the other is the linear menu(LM), which is shown on the screen the entire time. Both menus are located on the left-hand side of the screen.

All menus have the same items and size. The PopupLM and LM have the same design. Figure 3.5 shows the image of the PopupLM and LM.

3.4.3 Experimental design

For the experiment, we define the following concepts:

- menu manipulation for note-taking applications
- writing object and the manner of writing
- menu invocation method
- subjects of the experiment

Menu manipulation time for note-taking applications

We consider that the manipulation time is not just time for invoking a menu and select items from the menu, but also includes time in taking notes. If only menu invocation and menu selection are recorded as the menu manipulation time for note-taking applications, then the connection between note-taking and menu manipulation is lost.

One easy way to calculate the menu manipulation time for note-taking applications is to record the time of the entire note-taking process without menu manipulation and the time of the entire note-taking process with menu manipulation. Then, the menu manipulation time is calculated by the difference between these times. However, there can be other factors, such as rewriting a spelling mistake, and these other factors should not be included in the menu manipulation time.

To clarify the connection between the note-taking process and menu manipulation without other factors, it is necessary to record the time between the end of drawing a stroke and menu invocation, and the time between menu selection and the start of drawing a stroke. Therefore, we define the menu manipulation time for note-taking applications as follows:

1. leaving a finger after drawing a stroke
2. invoking a menu
3. selecting an item from the menu
4. putting finger down for drawing a new stroke

Figure 3.7 shows the range of the menu manipulation times for note-taking applications.

Writing object and the manner of writing

If an actual note-taking scenario needs to be reproduced, for example, the subjects need to take notes from something written on a blackboard, it is possible to disturb the manipulation, for example, by looking away or experiencing difficulties in recognizing what is written on the blackboard. These disturbances are not relevant to menu manipulation.

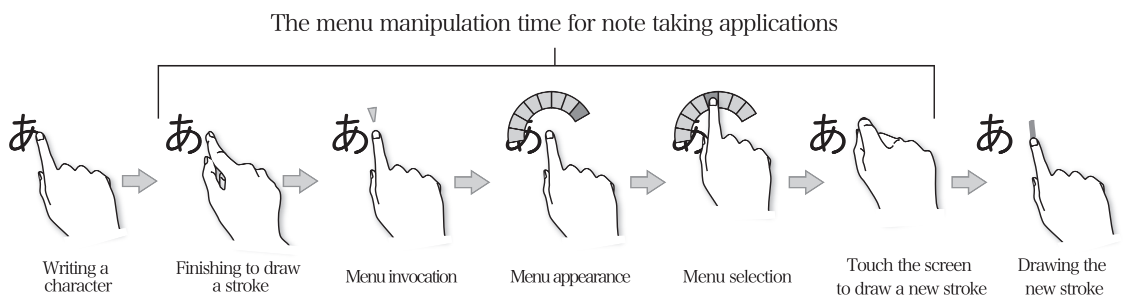

The menu manipulation time for note taking applications
Writing a character
Finishing to draw a stroke
Menu invocation
Menu appearance
Menu selection
Touch the screen to draw a new stroke
Drawing the new stroke

Figure. 3.7: Range of the menu manipulation time for note-taking applications

To reduce the possibility of disturbance, the writing object is set at the background of the screen. This is not same as the note-taking scenario, but represents the process of taking notes. Figure 3.8 shows the image of the writing object. We prepared a Japanese character object because some of the subjects may not have understood alphabetic symbols.

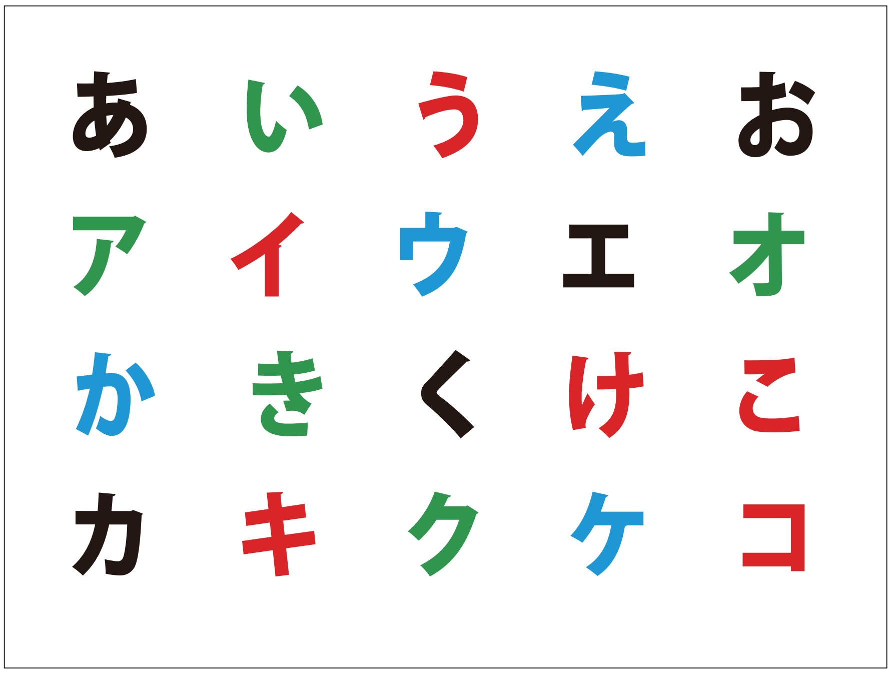


Figure. 3.8: Image of the writing object

Menu invocation method

We focused on the menu design for note-taking applications. Additionally, LM did not need to be invoked. Thus, while manipulating menus, the invocation time should be as short as possible

There are several interactions to invoke pop-up menus. A long press is one of the most common menu invocation interactions, whereas it requires certain time.

In general note-taking applications, a single tap or double tap will conflict with a dotting interaction. However, only in the writing object of this experiment, the Japanese characters did not contain dots, and thus, we set a single tap as the interaction of menu invocation for the arc menu and PopupLM.

Subjects of the experiment

Students need to take notes more frequently than adults, especially when they are in a school. Considering the frequency of taking notes, we set students as a main target. Additionally, if there are differences between students and adults, we will need to consider a suitable menu design for each group. Thus, we also set adults as a supplemental target for this experiment.

Six subjects, all of which study at a local elementary school, are selected at random. They are named as subjects S. Three of them are male and the others are female. One is left-handed. The average age is 11.8.

Three adult subjects are gathered from a local area at random. They are named subjects A. One is male and two of them are female. All of subjects A are right-handed. The average age is 41.6.

3.4.4 Experimental procedure and method

We explain the purpose of the experiment to the subjects and how to use the three types of menu. After the experiment, the subjects are requested to complete a questionnaire.

All the subjects practice all menus once. The order of the experiments is random. Figure 3.9 shows scenes from the experiment.

Figure. 3.9: Scenes from the experiment

3.5 Result

We recorded all manipulation processes and calculated the average time for each menu. Furthermore, we calculated the average time for subjects S and A separately.

Figure 3.10 shows the average manipulation time for subjects S and A. The results for subjects S show that the manipulation time for the arc menu (mean 1.77 seconds, s.d. 0.51) was faster than that for the LM (1.96 seconds, 0.94) and the *t*-test result was $t(153) = 1.816, p = 0.0356(< 0.05)$. Compared with the PopupLM (2.31 seconds, 0.89), the *t*-test result was $t(171) = 5.562, p = 0.0001(< 0.05)$.

Subjects A showed different results. For the manipulation time for the arc menu (mean 1.85 seconds, s.d. 0.50) and the manipulation time for the LM (1.66 seconds, 0.89), the *t*-test result was $t(85) = 1.344, p = 0.0911(< 0.05)$ and there was no significant effect compared with the PopupLM (2.22 seconds, 0.51), which had a significant effect; the StS-test result was $t(109) = 3.921, p = 0.0001(< 0.05)$.

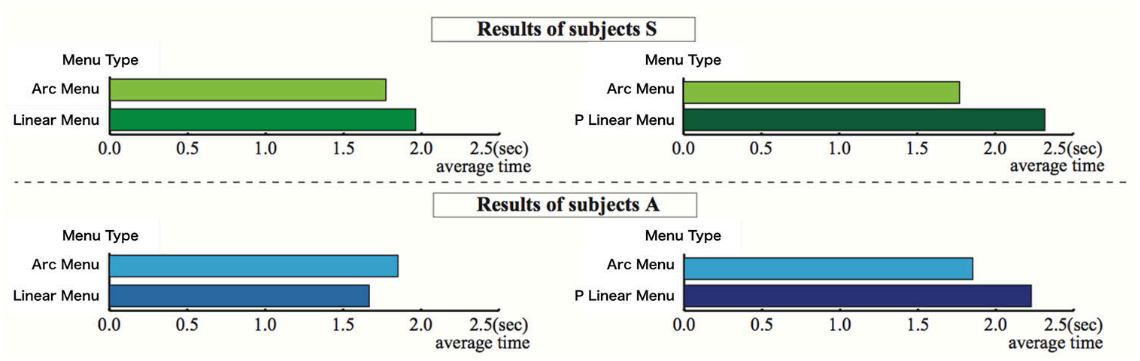


Figure. 3.10: Average manipulation time for subjects S and A

3.6 Discussion

In this section, we discuss the results of the experiment and reflect on the experimental design.

3.6.1 Discussion of the results of the experiment

In the case of subjects S, we found that even though the LM was shown the entire time, the manipulation time of the arc menu was faster. However, in the case of subjects A, there was not a significant difference between the arc menu and LM. Both subjects S and A showed that the manipulation time of the arc menu was significantly faster than that of the PopupLM. We examined some possible reasons why difference occurred in results between the subjects' manipulation times for the arc menu and LM. One possibility is that subjects A had already become accustomed to an ordinary menu design like that of the LM and could not become accustomed to using the arc menu. By contrast, subjects S had become accustomed to both the LM and arc menu; therefore, they easily adapted to the arc menu. They were not biased to either menu.

Another possibility is that the size of the arc menu fit the hand size of subjects S, whereas it did not fit that of subjects A.

3.6.2 Reflection on the experimental design

Throughout the experiment, we found that some experimental techniques were useful. We also found that there were several points that should be revised. We discuss these

Table. 3.1: The number of writing materials in a pen case

	Subject S1	Subject S2	Subject S3	Subject S4	Subject S5	Subject S6	Average
Writing materials	6	14	6	5	11	6	8
Frequently used materials	4	6	3	2	4	5	4

pros and cons so that we can take advantage of this experience and knowledge in our future work.

Outline of the experimental application

We focused on menu design, thus this experimental application did not consider multi touch. In other words the application did not have any palm rejection functions, and hence, the subjects were forced to keep the writing hand floating. Practically, if the note-taking application does not have palm rejection, then users will feel fatigue, which will give a negative influence to them while taking notes.

The number of items

For the menus in the experiment, we only set four colors in the items. The other items had different functions, such as undo and redo. There may have been few items on the menu. In order to find whether the number of menu items is sufficient, we prepared a questionnaire, which asked subjects S about writing materials typically found in the pen case that they usually use. Although the average number of writing materials used was approximately eight, the subjects answered that only four materials are frequently used, on average. The number of writing materials is shown in Table 3.1.

We set eight items in each of the menus and this seemed to be too few, whereas according to the questionnaire answers, it was sufficient for taking notes while using an application for educational purposes in the classroom. The number of frequently used writing materials was four.

However, when we consider practical use, there should be more useful functions, such as save, copy and share. Thus, the number of items or number of layers of the menu will increase.

Menu invocation method

In this experiment, we equipped a single tap for menu invocation. It was faster than a double tap or long press. However, for practical use, a single tap conflicts with pointing a dot, and cannot be equipped for the menu invocation method. Therefore, we need to consider suitable menu invocation methods while taking notes.

Note-taking using a finger

The subjects could draw strokes using their forefinger rather than a stylus. When we consider the general note-taking scenario, we are familiar with using a pen for taking notes. Thus, in future work, we should consider using a stylus as a writing device.

Target user

We set students as the main target user because we thought that note-taking activity was commonly performed by students at school; however, in settings such as meetings or seminars, there are several scenarios for taking notes for every generation. Therefore, the target user should not be limited to students.

3.7 Conclusion

Throughout the experiment, we found that the manipulation time of the arc menu was significantly faster than that of the PopupLM, which represents the general menu design. Additionally, when subjects S used the arc menu, the manipulation time was significantly faster than that for the LM. Therefore, it has been concluded that if it is in a same condition, the menu design we proposed makes smooth the task of taking notes compared to the general menu design.

There are various types of menu designs. Other types of menus, such as the pie menu, were not compared with the arc menu in this experiment. Therefore, comparison with other context menus remains as future work.

Menu design is one of the elements that affects the smooth note-taking activity in applications. Through this experiment, we found that the difficulty of taking notes without palm rejection. We also found that there was difficulty when applying a rapid menu invocation method to note-taking applications. These two problems are important elements that have influence on smooth note taking.

Chapter.4 Toward the reduction of incorrect drawn ink retrieval

4.1 Introduction

When a user writes something on a tablet using a note-taking application, multi-touch interaction compels the user to float the hand above the screen to avoid accidental inking. Because this unnatural way of writing produces difficulties [19] for digital note-taking applications, "palm rejection" becomes crucial. Palm rejection distinguishes intended touches from unintended touches and prevents accidental inking.

Annett et al. [42] classified all intended and unintended touches into the following four categories: true positive, true negative, false positive and false negative. We added results of each touch interaction and compiled a list of the categories. The list of touch interactions are shown in Table 4.1.

Table. 4.1: Classification of touch interactions

Touch interactions	Classification	Result
Intended touch	True Positive	Correctly ink stroke is drawn
Intended touch	False Negative	Incorrectly ink stroke is not drawn
Unintended touch	True Negative	Correctly ink stroke is not drawn
Unintended touch	False Positive	Incorrectly ink stroke is drawn

When palm rejection correctly distinguishes an intended touch, it results in a true positive and the touch draws a correct ink stroke. When palm rejection correctly distinguishes an unintended touch, it results in a true negative and the touch does not draw an ink stroke.

By contrast, when an unintended touch is recognized as an intended touch, it is a false positive and accidental inking occurs. When an intended touch is not recognized correctly, the touch is a false negative and it is incorrectly rejected.

Making distinctions between intended touches and unintended touches is complicated

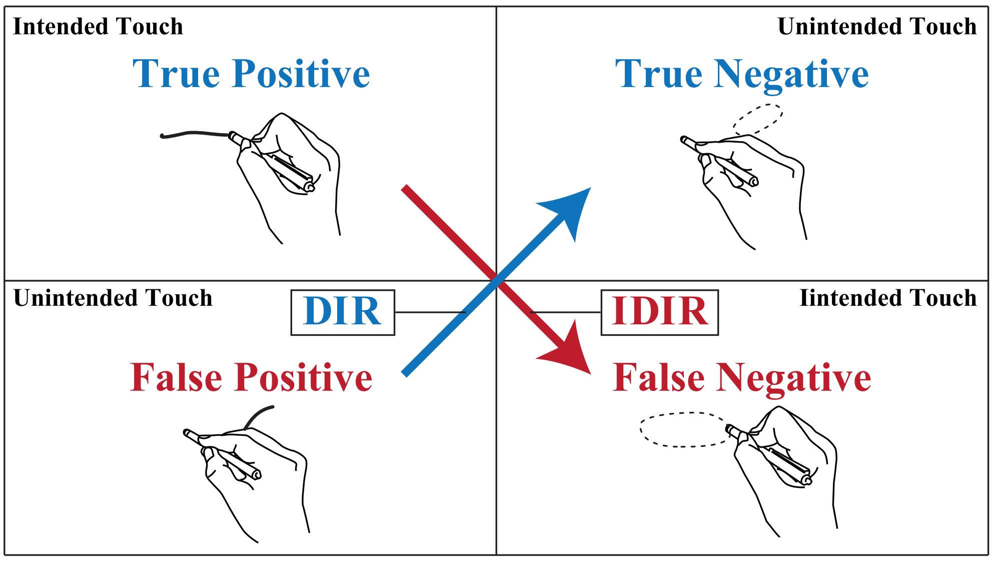


Figure. 4.1: Image of DIR and IDIR

because most touches move rapidly and are not stable. It is important to analyze all touch data within a short amount of time and make an immediate distinction.

There are general applications [14, 1, 15] that have already equipped palm rejection. While investigating palm rejection for these note-taking applications, a curious interaction was detected. When a user attempts to write something on a touch screen using these note-taking applications and an intended touch draws an ink stroke correctly, what occasionally occurs afterward is that the stroke is removed in a very short time against the user's will. From this interaction, we infer that some palm rejection algorithms iteratively classify intended touches and unintended touches, and switch the distinction afterward. This is reasonable when an interaction occurs for a touch that should have been classified as a true negative but is classified as a false positive, so that draws an accidental ink stroke is drawn under the palm.

However, in some cases, a true positive touch is switched to a false negative touch afterward. This is rather perplexing for users when the interaction incorrectly switches the classification and removes correct ink strokes. We call the correct interaction "drawn ink retrieval (DIR)" and the incorrect interaction "incorrect DIR (IDIR)." Figure 4.1 shows the image of DIR and IDIR occurrences.

Note-taking applications are used in classes or meetings, and hence, users need to be able to take notes rapidly. In this scenario, IDIR results in a negative impression for users. When IDIR occurs, users need to rewrite the retrieved strokes. This takes time and runs counter to the requirements of rapid note-taking activity.

The purpose of this chapter is to propose a palm rejection algorithm that reduces the frequency of IDIR. To realize the algorithm, we take an algorithm that uses multi-touch interaction, and then we combine two logical processes: a machine learning technique and occlusion area protection. For convenience, we call our palm rejection algorithm "machine learning and occlusion area protection (MLOAP)."

This chapter is structured as follows. In Section 4.1, we present the problem of palm rejection called IDIR. In Section 4.2, we briefly categorize two types of approaches for considering palm rejection. In Section 4.3, we introduce the combination of two logical processes. In Section 4.4, we describe a process for developing a note-taking application and explain the experiment. In Section 4.5, we discuss the results and any remaining problems, and in Section 4.6, we provide conclusions.

4.2 Related work

In this section, the existing approaches are surveyed and categorized into the following two types: active stylus interaction and multi-touch interaction.

There are several approaches for classifying intended touches and unintended touches. Annett et al. [42] researched such approaches and compared them. They categorized the various approaches into four types: user adaptations, firmware approaches, operating system approaches and application-specific approaches. As a simpler categorized method, Schwarz et al. [45] categorized existing approaches into hardware solutions and software solutions. One is to positively use various functions that are equipped in hardware, for example, an active stylus, which is distinguished as a specific touch from other general finger touches. This is the current main approach of palm rejection. The other approach is to focus on multi-touch interaction and solve the problem using software algorithms, which does not depend on specific hardware or devices.

The approach of multi-touch interaction is less precise than the approach of using active styluses. Even though the above statement is the case, researching and developing the approach of multi-touch interaction is meaningful, because not every standard capacitive touch device is equipped with and active stylus.

4.2.1 Active stylus interaction

Various prototypes have been researched as novel interaction devices [40, 49, 78]. Such research can be used as the base technology of future products.

Several devices have already had active stylus interaction equipped. Samsung Galaxy's S Pen [79], the WACOM digitizer [80] and Microsoft Surface's Pro pen [81] have a similar function to palm rejection, and furthermore, they can even recognize pen pressure. Additionally, some of them manipulate the touch device without physical touches on the screen. Using these active styluses enables the application to simplify touch classification. Although they are a reliable solution for accidental inking, these approaches depend on specific hardware. For instance, the S Pen depends on Samsung Galaxy, and the Pro pen depends on Microsoft Surface.

Various active styluses that use Bluetooth technology, which frees their dependence on specific touch devices, are available for standard capacitive touch devices, such as the iPad. BambooPaper by WACOM [1], GoodNotes by Time Base Technology Limited [15] and Penultimate by Evernote [14] have the option of connecting these styluses via Bluetooth. If we use the active stylus, these applications display more accurate palm rejection results.

FiftyThree provides both an original active stylus called Pencil [82] and an application called Paper [83]. This application supplies palm rejection, but it only works with the original active stylus.

4.2.2 Multi-touch interaction

In terms of precision, multi-touch interaction approaches are inferior to active stylus interaction approaches. By contrast, in terms of versatility, multi-touch interaction approaches are more adaptable for several operating systems and multi-touch devices than active stylus interaction approaches.

Several studies for multi-touch interactions [20, 33, 34] have attempted to recognize finger touches. Current capacitive touch devices can correctly receive all touches, but still have difficulty in classifying which touches are intended.

One well-known approach is to have a specific region in which applications ignore all touches. Users can put a hand onto the region without ink strokes. All touches outside the region are recognized as intended touches, and so, draw ink strokes. NoteAnytime, which is a note-taking application by MetaMoJi, takes this approach [43]. An advantage of this simple approach is that while a user is putting a hand on the region, accidental inking does not occur. Thus, DIR and IDIR also do not occur. A disadvantage is that users need to move the region manually according to the hand position and where they want to write. In terms of usability, this uncomfortable way of writing results in negative user experiences.

Vogel et al. [44] collected the pen and hand position data, which is called occlusion silhouettes, by capturing images from a head-mounted camera. Then, they presented a scalable circle and pivoting rectangle geometric model, which detects the position of the hand and forearm from pen nib coordinates. If the pen nib coordinates are clearly pinpointed, then the model can use palm rejection.

Yoon et al. [84] used the model of Vogel et al. to reject unintended touches while an active stylus was recognized. However, for note-taking with a general stylus, the pen nib does not always touch the screen. Thus, alternative logic needs to be applied to this model to reject unintended touches for note-taking applications.

Schwarz et al. [45] proposed a novel solution using spatiotemporal touch features. It voted for all touches iteratively each 50 ms to 500 ms regarding whether they were intended touches or unintended touches using a decision tree, which is a machine learning algorithm. Their solution is the current baseline for palm rejection. By contrast, Annett et al. [42] highlighted the problem of classification speed of the solution by Schwarz et al. In the study, DIR and IDIR were not evaluated, although the authors mentioned that false positive touches would be switched to true negative touches through iterative classification.

BambooPaper [1], GoodNotes [15] and Penultimate [14] also have the palm rejection function, which uses multi-touch interactions. To activate this function, a registration of the user's dominant hand information is required. Additionally, GoodNotes and Penultimate require a frequent hand posture. These applications are considered use machine learning techniques to classify intended touches and unintended touches. Therefore, applications need to adjust the learning data to the user's writing posture. When the writing posture fits the registered posture, palm rejection mostly works correctly. However, if the writing posture becomes different from the registered posture, then the applications tend to make incorrect rejections, and thus, IDIR also tends to occur.

4.3 Algorithm

According to past research [84] and an existing application [43], using an occlusion area is a reliable way to reduce occurrences of IDIR. It is important to note that creating a dynamic occlusion area without any pen nib information requires other information that can detect hand positions.

Using a machine learning technique becomes a standard approach to classify intended touches and unintended touches. Generally, the technique is used to reject all unintended touches that are considered as unnecessary. Most unintended touches are generated by the writing hand, and hence, those touches indicate the location of the writing hand. This means that unintended touch information can be obtained by the production of the dynamic occlusion area.

The algorithm of MLOAP is to build a touch distinction model using a support vector machine (SVM), which is a machine learning algorithm suitable for solving two-class tasks. The SVM model classifies intended touches and unintended touches. True positive intended touches are recognized as pen nib and draw strokes. True negative unintended touches do not draw strokes. Furthermore, unintended touches are taken advantage of in the sense that they produce the dynamic occlusion area.

4.3.1 Touch distinction using the SVM model

To distinguish the pen nib from all touches, the following classifier is introduced:

$$y = \sum_{i=1}^{N} w_i^\top \boldsymbol{x}_i + b, \tag{4.1}$$

where N is the number of explanatory variables. The numbers of touch coordinates and touch records, which are described in Subsection 4.3.4, are used as the explanatory variables. The value of the x-coordinate and y-coordinate is $\boldsymbol{x}_i$. The bias is b, and w_i is determined by the SVM, in which L2-regularized L2-loss SVC [85] solves the following primal problem:

$$\min_{w} \quad \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C\sum_{i=1}^{l}\left(\max\left(0, 1 - y_i \boldsymbol{w}^\top \boldsymbol{x}_i\right)\right)^2. \tag{4.2}$$

The SVM is a supervised learning method, and requires a tagged dataset. In this case, the tags are intended touches or unintended touches. To build the dataset, both w and $\boldsymbol{x}$ are matrices in the classifier (4.1), and therefore, w and $\boldsymbol{x}$ are vectorized to apply L2-regularized L2-loss SVC (4.2).

After building the SVM model using the dataset, the SVM model classifies intended touches and unintended touches. If y is a plus in the classifier (4.1), then the coordinate is classified as an intended touch, whereas, if y is a minus, then it is classified as an unintended touch.

The difficulty is when there is only the palm on a screen and there should not be intended touches, in the case of which, the classification algorithm occasionally detects intended touches incorrectly and accidental inking is produced. Iterative classification can retrieve it; however, latency and IDIR occur as side effects.

4.3.2 Touch protection using the dynamic occlusion area

To reduce the occurrence of IDIR without latency, simple and robust rejection logic is needed. Traditional approaches used pen nib information and the hand posture geometric model to detect the hand position [44]. In the case of note-taking, the pen nib does not always touch the screen. However, we create a dynamic occlusion area using the information of the hand position, which is detected using SVM model classification. Inside the occlusion area, all touches are rejected. This occlusion area supports to avoid accidental inking by false positive touches.

4.3.3 Collecting the dataset for the SVM

To collect the dataset, the technique by Schwarz et al. [45] was applied. The dataset was separately collected from 10 right-handed participants and two left-handed participants. The participants held a passive capacitive stylus, which had a simple rubber nib and is referred to as a “stylus.” The participants put a palm onto the touch screen. Touches inside the circle represents intended pen touches, whereas touches outside the

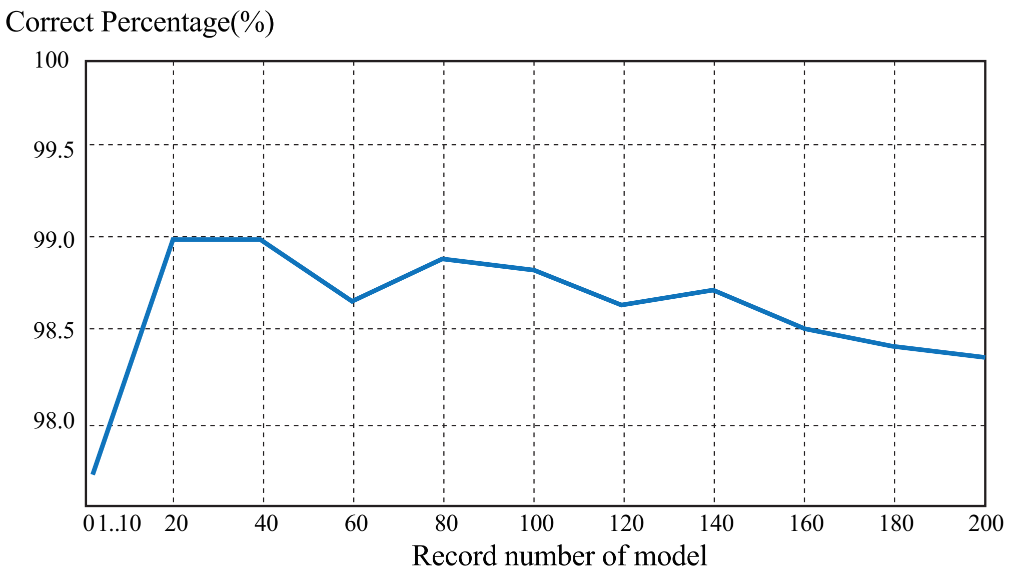


Figure. 4.2: Correct percentage for each weight size

circle are interpreted as unintended touches. We allowed the circle to dynamically follow the pen nib so that we could collect realistic note-taking data. The participants were told to put the pen nib onto the circle and make strokes on the touch screen evenly. To do so, the dataset becomes closer to real note-taking data.

When any touch event occurred on the screen, one record is produced, which includes all existing x and y touch coordinates. Approximately 250,000 records were set as the training dataset, with 5,000 records set as the validation dataset. From the dataset, a total of 20 models were produced. The record number of the model increased by one up to 10 and, after that, the record number increased by 20 up to 200. Before conducting the experiment, we applied each model to the validation dataset and examined which model was the most effective for the most precise classification. A model size with 20 records provided the highest correct percentage of 98.98% for classification. Thus, a model size with 20 records was adopted.

In this paper, we set the weight size to 20 records. Figure 4.2 indicates the correct percentage and the weight size.

4.3.4 Developing the SVM model

To distinguish between intended and unintended touches, the actual touch dataset also needs to contain the same 20 records. The 20 records are composed of 19 contextual records and one most-recent record.

We let r denote the number of records and t denote the number of touch coordinates. In this experiment, the maximum number of touch coordinates is set to 10. Thus, in the classifier 4.3.1 (4.1), N is $t \times r = 10 \times 20 = 200$. The algorithm considers the following touch events:

- "touch start:" When fingers or a stylus nib touch the screen, this event occurs.
- "touch end:" When fingers or a stylus nib leave the screen, this event occurs.
- "touch move:" When fingers or a stylus nib move on the screen, this event occurs.

Whenever touch events occur, all touch coordinates are stacked into the record. If there are fewer touches than the maximum number, then zero is stacked to fill the record. In the case in which there are no records when the touch events start, it takes approximately 5 ms to stack all the recorded 20 records to distinguish the touches. When 20 records are already stacked, it takes approximately 1 ms for the calculation. If there are no touch events, then the records are not stacked. When no touch points are detected, the stacked records expire.

4.3.5 Implementation of occlusion area protection

After the SVM model had classified intended touches and unintended touches, the dynamic occlusion area is applied to avoid accidental inking and to reduce the occurrences of IDIR. The dynamic occlusion area is an invisible circle. A round occlusion area is adopted to manage the changing writing hand angle. The x-coordinate for the center of the circle is the mean value of the x-coordinates of all unintended touches. The y-coordinate for the center of the circle is calculated in the same manner.

The circle has a radius of 230 px. The radius is derived from the dataset that was collected previously. The mean length between the intended touches and unintended touches was 390 px. We heuristically adjusted the size of the radius of the circle. The suitable radius was slightly longer than half the mean length.

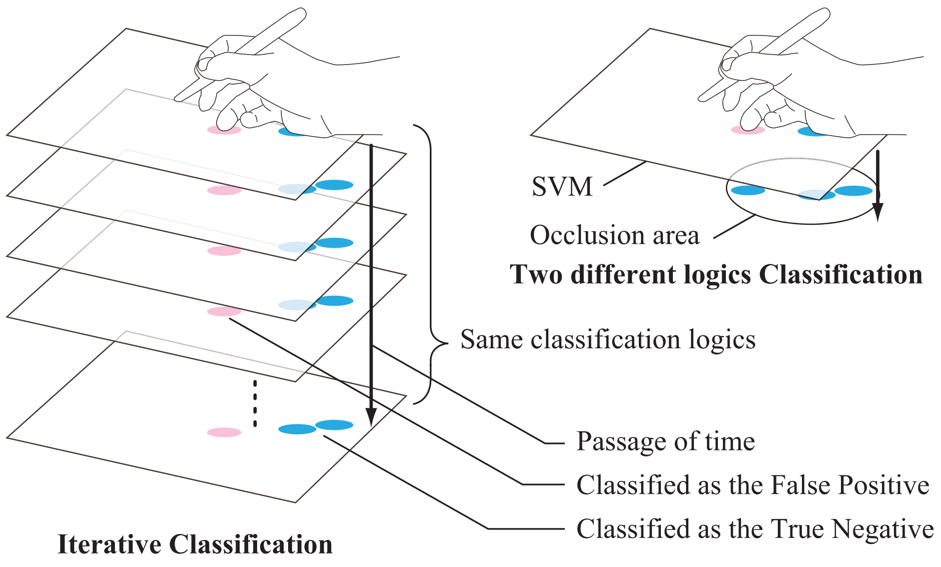

SVM
Occlusion area
Two different logics Classification
Same classification logics
Passage of time
Classified as the False Positive
Classified as the True Negative
Iterative Classification

Figure. 4.3: Concept of iterative classification and our algorithm

When the SVM model classifies intended touches and unintended touches, and if a false positive touch is inside the dynamic occlusion area, then the occlusion area will avoid accidental inking.

Figure 4.3 compares the concept of iterative classification and our algorithm.

When a user puts a hand on the screen when writing something with a stylus, it takes approximately 1 ms to generate the dynamic occlusion area. In total, it takes 2 ms for the combined classification with the inclusion of the drawing records, and up to 10 ms without the records.

In the application of this algorithm, the dynamic occlusion area cannot be applied without unintended touches. Thus, the first touch causes one particularly difficult scenario. If the first touch and second touch occur individually, and the first touch is recognized as an intended touch before the second touch occurs, then the first touch will immediately draw an ink stroke. However, there are three possible cases.

The first case is where the first touch was truly intended, and the second touch was unintended. The second case is where the first touch was unintended, and the second touch was intended. The third case is where both the first touch and second touch were unintended. In both of the second and third cases, the unintended drawn ink stroke should be removed. Therefore, to minimize the occurrences of IDIR, we equip a function that invokes DIR only for the first touch. Figure 4.4 shows the basic flow of drawing a stroke by MLOAP.

4.4 Evaluation

To evaluate MLOAP, we developed an experimental application, on which we conducted an experiment.

4.4.1 Developing a note-taking application

To realize this algorithm, an experimental note-taking application was developed as a web application using JavaScript and HTML. Because we have various types of multi-touch devices, a hardware specific application has the difficulty of extensibility for all the various devices. JavaScript and HTML can execute within most modern browsers and multi-touch devices.

Current capacitive touch devices, such as the iPad, can use a touch radius. Some general applications can take advantage of touch radius information to improve the precision of palm rejection. However, whether the touch radius can be used depends on the operating system. The present algorithm does not use touch radius information. Therefore, there is the benefit of extensibility to other browsers, operating systems and multi-touch devices.

4.4.2 Experiment

It is important to compare with applications actually used, rather than the experimental applications so that we can verify practicality of our algorithm. Thus, the note-taking application, which had equipped MLOAP, was compared with two other well-known applications: GoodNotes [15] and Penultimate [14]. Both of these applications equip palm rejection, which supposed to be based on machine learning techniques, and, both of them require setting a suitable angle for using the function of palm rejection. Figure 4.5 shows an image of setting a suitable angle for GoodNotes. Figure 4.6 shows an image of setting a suitable angle for Penultimate.

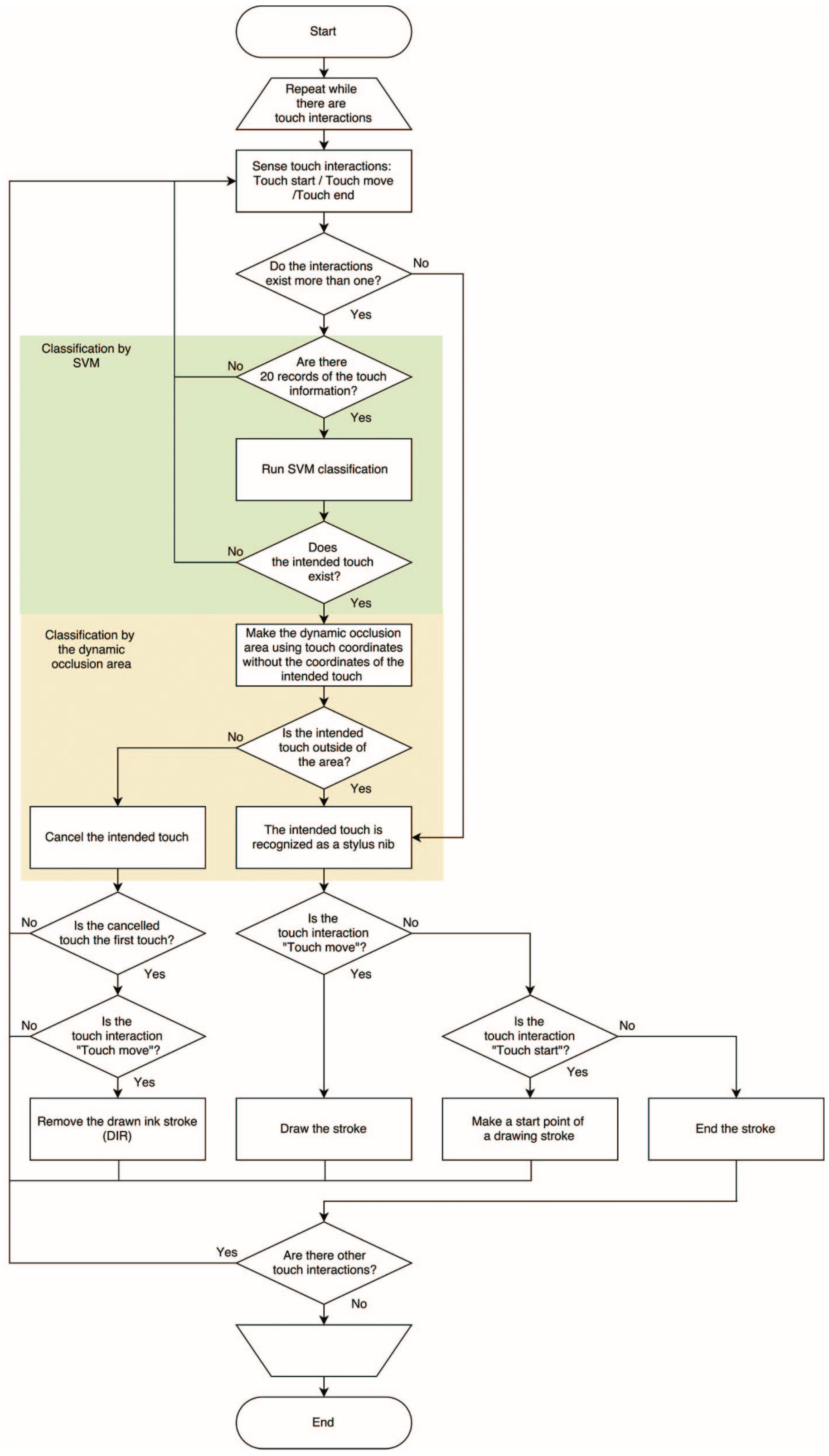
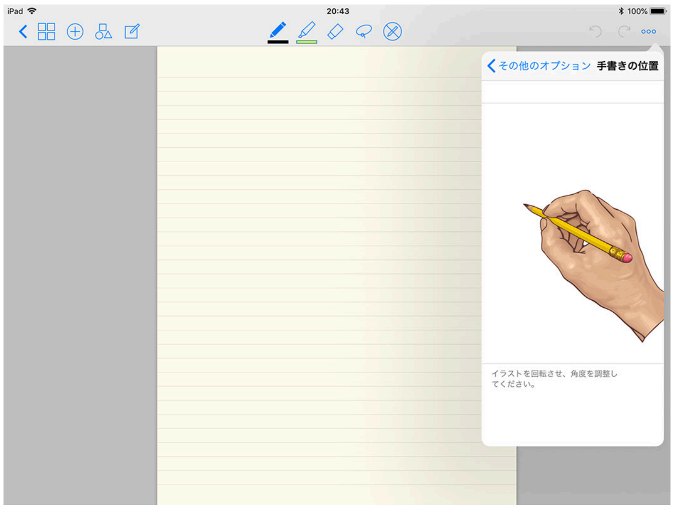
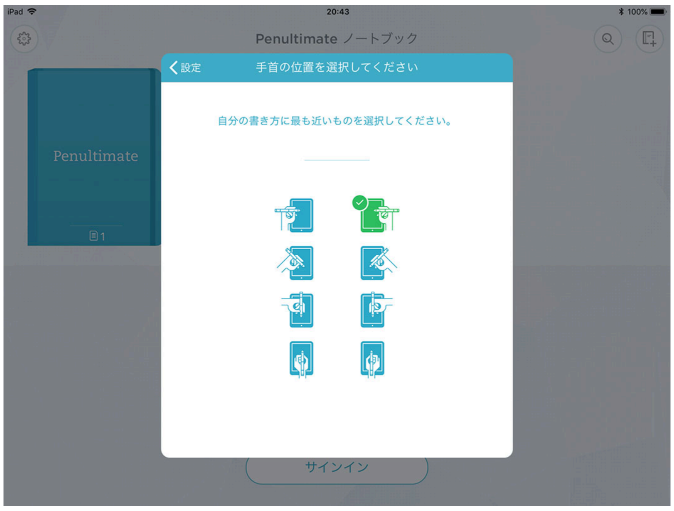

Start
Repeat while there are touch interactions
Sense touch interactions: Touch start / Touch move /Touch end
Do the interactions exist more than one?
No
Yes
Classification by SVM
Are there 20 records of the touch information?
No
Yes
Run SVM classification
Does the intended touch exist?
No
Yes
Classification by the dynamic occlusion area
Make the dynamic occlusion area using touch coordinates without the coordinates of the intended touch
Is the intended touch outside of the area?
No
Yes
Cancel the intended touch
The intended touch is recognized as a stylus nib
Is the cancelled touch the first touch?
No
Yes
Is the touch interaction "Touch move"?
No
Yes
Is the touch interaction "Touch start"?
No
Yes
Remove the drawn ink stroke (DIR)
Draw the stroke
Make a start point of a drawing stroke
End the stroke
Are there other touch interactions?
Yes
No
End

Figure. 4.4: The basic flow of drawing a stroke by MLOAP



イラストを回転させ、角度を調整してください。

Figure. 4.5: Image of setting a suitable angle for GoodNotes



手首の位置を選択してください

自分の書き方に最も近いものを選択してください。

Figure. 4.6: Image of setting a suitable angle for Penultimate

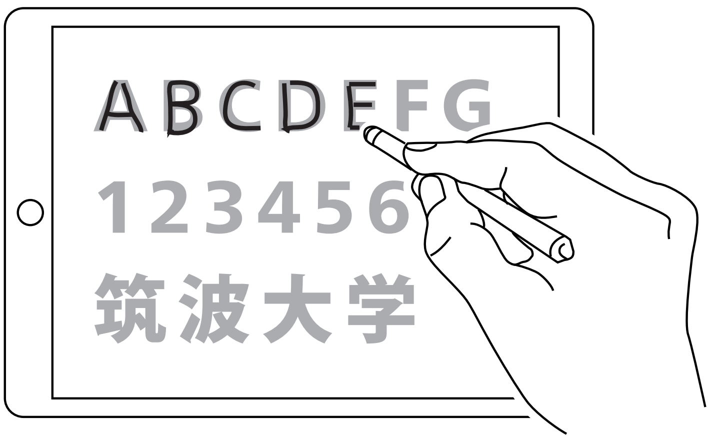


Figure. 4.7: Image of the experiment

To increase the reproducibility of the experiment, we chose the iPad Air 2, which is one of the most popular devices, in the experiment. When we collected the dataset of SVM, we set the iPad to horizontal orientation. Therefore, to fit to the dataset, we also set the iPad to horizontal orientation in the experiment.

In order to ensure that the experiment result is not affected by subjects' experience of the tablet devices, the subjects should use tablet devices or, at least they should be able to use multi touch devices. Therefore, as subjects, we gathered eight university students who are familiar with using tablet devices. Seven subjects were right-handed, and one was left-handed. All the subjects were different from the subjects who participated in the collection of the dataset.

Three types of characters were displayed in each application: the capital letters A to G, numbers one to six, and Japanese Kanji, which consisted of four characters. Figure 4.7 shows an image of the experiment.

The subjects were instructed to trace all the characters according to the following list:

- For GoodNotes and Penultimate, set a suitable angle for the dominant hand from each application's setting.
- For our application, choose the right or left hand.

- Write every character in the correct order and use the correct number of strokes.
- Do not rewrite a character, even if a stroke is not correctly drawn.
- Do not rewrite a character, even if DIR occurs.
- Writing in cursive style is prohibited.
- The writing style should be same as the style in which one writes something using a pen and notebook.
- Use a stylus, which we provide.

For each application, one practice period was provided. The experiment was set in random order. The writing process was recorded on video. We classified all strokes as the following three interactions:

- correct ink stroke (true positive: TP)
- accidental inking (false positive: FP)
- changing the classification from true positive to false negative (IDIR)

The classification of true negative was invisible and could not be evaluated. The occurrences of DIR, which is changing the classification from false positive to true negative, were not evaluated because the interaction mostly occurred under the palm and could not be confirmed.

4.4.3 Result

Table 4.2 shows the total number of interactions for each application. The difference in the number of all strokes is because some subjects wrote characters following the wrong procedure. For instance, a letter A was written with two strokes instead of the correct three strokes. The number of false positives was close for all three applications.

GoodNotes and Penultimate recorded five IDIRs, whereas our application recorded one.

Table 4.3 shows the details of the IDIR numbers for each subject. The results of our application show that subject G recorded one IDIR, whereas the other seven subjects

Table. 4.2: Total number of interactions and classifications

Application	All Strokes	TP	FP	IDIR
Application with MLOAP	437	416	36	1
GoodNotes	435	429	38	5
Penultimate	436	397	35	5

record no IDIRs. Compared with our application with MLOAP, the other applications recorded more IDIRs.

Table. 4.3: Number of FPs and IDIRs for each subject

Application	Subjects (R/L)	Strokes	FP	IDIR
Application with MLOAP	A (R)	55	0	0
	B (R)	56	4	0
	C (L)	56	0	0
	D (R)	55	9	0
	E (R)	53	0	0
	F (R)	54	8	0
	G (R)	54	8	1
	H (R)	54	7	0
GoodNotes	A (R)	54	3	1
	B (R)	56	1	2
	C (L)	56	5	0
	D (R)	55	6	1
	E (R)	53	2	0
	F (R)	54	11	0
	G (R)	53	5	0
	H (R)	54	5	1
Penultimate	A (R)	55	5	0
	B (R)	56	0	2
	C (L)	56	0	0
	D (R)	55	12	1
	E (R)	53	0	1
	F (R)	54	6	1
	G (R)	54	5	0
	H (R)	53	7	0

4.5 Discussion

In this section, we discuss the result of the experiment and threats to validity.

4.5.1 Discussion on the number of occurrences of IDIR

IDIR occurred at the application with MLOAP when a subject was writing the lower left kanji. In the case of GoodNotes and Penultimate, it was found that IDIR occurred at the time of writing of the upper right alphabet and plural Kanji. We consider that there are two factors in this result. First, both GoodNotes and Penultimate require setting the orientation of a writing hand in advance. Thus, in the case of writing at the screen edge like lower left or upper right, the possibility that the inclination deviates from the preset hand orientation may be considered. On the other hand, since the application with MLOAP correctly learns the orientation of the hand corresponding to the drawing area when we collect the learning data, even when the orientation of the hand changes at the screen edge, the algorithm could correctly distinguish the hand and the pen nib. Secondly, because compared with numbers on the middle and alphabets, especially Japanese Kanji, are complex and have more strokes, it is assumed that it is difficult to distinguish whether putting hands or pen nib.

4.5.2 Discussion on the number of occurrences of FP

There is almost no difference in the number of occurrences of FP in any application, and each occurred from 30 to 40 times. In MLOAP, FPs do not occur in the dynamic occlusion area, but if it is out of the area, FP will occur. Since the size and writing style of the hands of the subjects are different, it is considered that FPs were caused by touches out of the circle. Also, in GoodNotes and Penultimate, it seems that some subjects were prone to generate FPs due to differences in hand size and writing style.

4.5.3 Threats to validity

All the subjects were university students, who were familiar with using touch devices; thus, if the subjects had no experience in the use of touch devices, the results would change. Additionally, the size of hands influenced the results of the experiment.

The experiment of Schwarz et al. [45] defined true positive strokes as stroke recognition and false positive strokes as error strokes. Their results were 97.9% for true positive strokes and 0.016 for the false positive rate. Our algorithm resulted in 95.2% for true positive strokes and 0.082 for the false positive rate. Schwarz et al.'s experiment was to draw below six symbols: character L, S, vertical line, horizontal line, dot and circle. Considering that all these symbols are composed of a single stroke and are simpler than the characters in our experiment, our results with regard to precision were convincing. Although the entire number of IDIRs was smaller than the total stroke number, we did not achieve statistical significance.

The iPad Air 2 was used as the device for collecting the dataset and for the experiment. We set the iPad to horizontal orientation. To use vertical orientation or other devices, another dataset would be needed, because the device specification influenced the data collection and classification speed.

4.6 Conclusion

Past research and existing applications have focused on reducing the occurrence of accidental inking. We focused on IDIR, which occurs as a result of a reclassification from true positive to false negative.

Concerning the number of correct ink strokes and accidental inking, our application with MLOAP behaved on a par with the other applications.

Throughout this research, we illustrated that to find unintended touches, which used to be regarded as useless information, a dynamic occlusion area can be used. We confirmed that our application with MLOAP reduced the occurrences of IDIR throughout the experiment. In terms of reducing the occurrences of IDIR, we can say this algorithm can be one of manners for palm rejection.

Our palm rejection algorithm has many points to be improved in terms of precision and needs more experimental results. The size of the area should be adjusted according to the user's hand size. The shape and position of the area should be considered. These improvements will be reflected in our future work.

Chapter.5 Evaluation of menu manipulation using palm lift manipulation

5.1 Introduction

In the scenario of note-taking using an application, it is important to consider how users manipulate the application's menus. Further, because users switch functions using menus while concentrating on classes or meetings, rapidity is required for menu invocation.

If a menu is displayed all the time, it is possible to manipulate the menu quickly. However, occupying the screen has the problem that the area to take notes becomes narrower. To fully use the writing area, context menus can be a solution. To invoke context menus, various invocation methods and invocation positions have been proposed. For instance, multi-touch interaction and gestures can be arranged to invoke menus. However, users are holding a stylus to take notes, and hence, it is thought that they do not want to frequently change their writing posture to invoke menus. Most applications equip the long press as an invocation method for context menus. The drawback of the long press is that users need to put a stylus or finger on the screen for a while to invoke menus.

The purpose of this chapter is to clarify the following research question; which menu invocation methods are manipulated rapidly, and where should menus appear? Therefore, we propose a menu invocation method and a menu invocation position assumed to be used for a note-taking application. Then we evaluate our methods and other methods.

This chapter is structured as follows. In Section 5.1, we present the problem of menu invocation method for note-taking applications. In Section 5.2, we introduce menu invocation methods and menu invocation positions. In Section 5.3, we propose our menu invocation method and menu invocation position. In Section 5.4, we describe the experiment for menu invocation methods and positions. In Section 5.5, we present the results of the experiment. In Section 5.6, we discuss the results and any remaining problems, and in Section 5.7, we provide conclusions.

5.2 Related work

In this section, we survey past research on menus from the viewpoints of menu invocation methods and menu invocation positions. We also investigate palm rejection, which is equipped into applications on multi-touch tablet devices.

5.2.1 Menu invocation methods

Various menu invocation methods, such as tap, double tap, bezel swipe and multi-touch, bring rich interaction into tablet devices and smartphones. However, in terms of note-taking applications, these menu invocation methods have pros and cons. Some of them may not meet our criteria, which we listed in Chapter 2.

FiftyThree's Paper [83] is an existing handwriting application and it adopts a typical menu invocation method. It is a simple menu invocation method whereby users tap a relatively small bar or button on the screen. In this case, because the bar or button for menu invocation is on the screen, the writing area on the screen becomes slightly narrow. However, though this depends on the size of the bar or button, they can occupy the writing space, and this is inconsistent with the criterion "6: menus should not occupy the writing space."

As an invocation method to pop up a context menu on a stylus nib, the long press is one of the popular methods. BambooPaper [1] adopts the long press to invoke a context menu for changing pen color. Invoking the menu near the stylus nib is an advantage. Whereas a disadvantage is that it takes time to press and hold for menu invocation.

The bezel swipe, in which users swipe from the edge of the screen, has two advantages for menu invocation. One is that it is not necessary to install bars or buttons for menu invocation and thus, the writing area of the screen is not occupied. The other advantage is that users of the application can roughly swipe without gazing at the edge of the screen.

Hinckley and Yatani [86] used the bezel swipe to invoke radial menus at the screen edge. WriteOn INKredible [47] is an application that incorporates the bezel swipe. For INKredible, users can invoke the menu by swiping bezels at both sides of the screen.

”Marking menu” by Zhao, Balakrishnan [87] and ”Bezel menus” by Jain and Balakrishnan [88] propose to select items without invoking menus. An expert mode of the marking menu enables users to select items by means of marked strokes of a pointing device without invoking menus. The bezel menu switches to menu manipulation mode with a bezel swipe. Swiping the screen from the edge in a specific direction performs like a gesture interaction for item selection without menu invocation. When we incorporate these gestures into a note-taking application, users will be required to remember these gestures and corresponding commands. Moreover, it is necessary to prepare a novice mode to manipulate with watching the menus.

The “multitouch marking menu” by Lepinski et al. [2] arranges the multi-touch interaction to invoke menus. Users can tap multiple fingers at the same time and invoke a menu according to the number and order of fingers. Harrison et al. [30] proposed unique interaction methods using hand shapes when operating various tools in the real world. Their methods can be applied to menu invocation. However, while taking notes in classes or meetings with high concentration, their methods require tapping with multiple fingers. We suspect that such manipulations will disturb the note-taking activity.

5.2.2 Menu invocation position

In most applications, menus are located at the screen edge. By contrast, the context menu is invoked near a pointing device, and, in case of tablet devices, it is invoked near the manipulating stylus or finger, for example, the aforementioned BambooPaper [1].

There is a research of changing the position of menus. If the size of the menu items is the same, because of Fitts’ law [59], selecting items on the menu displayed at the screen edge will require more time to select items on the menu displayed on nearer to the manipulating stylus. Fitzmaurice et al. [6] highlighted that the action that iteratively moves the cursor from the point of manipulation to the detached menu repeats the action of returning to the same place. They called this action “tool palette round trips.” They proposed the “tracking menu” to solve it. The tracking menu can move to the vicinity of the operating cursor. However, the problem of the tracking menu is that users need to move it manually.

5.2.3 Palm rejection

In note-taking applications equipped with palm rejection, users can put a hand onto the screen and write letters or figures. Palm rejection enables non-purposed palm touches to avoid accidental inking, which is mostly produced by unintended touches [42]. Then users can write in the same or a similar manner as writing on paper with a pen.

If a note-taking application does not embed palm rejection and a hand is placed on the screen, then the location of the hand could be determined as a pen nib. This causes a problem whereby strokes are unintentionally drawn under the hand [42]. To avoid unintentional drawing, it is necessary to float the writing hand and only allow the stylus nib to touch the screen. If such a writing approach is used for a long time, then fatigue will accumulate [19].

To overcome such an unnatural writing posture, multiple studies have proposed methods of distinguishing touches by a stylus from touches by a hand on the screen. Therefore, a user of the application can use the handwriting application by placing a hand on the screen using palm rejection.

Schwarz et al. [45] proposed a novel palm rejection algorithm called “spatiotemporal touch features.” In the algorithm, a decision tree from machine learning is used a number of times to vote which tap is the stylus, and discriminate between the stylus and taps by a writing hand with a high accuracy value of 98%.

MetaMoJi Note [43] has a specific space within which a user can put a hand. In the space, no strokes are drawn. However, each time the user moves the position of the hand, the user also needs to move the space. This is not suitable for note-taking applications that require quick movement.

Another approach to distinguishing a stylus nib from a hand is to use a specialized stylus, such as the Apple Pencil [89] of the iPad Pro. It has the advantage that the determination by hardware is more accurate than the determination by software based on a palm rejection algorithm. However, such a specialized stylus generally tends to be more expensive than general styluses. Additionally, specialized styluses are only available for specific devices, and thus, the versatility will be poor when considering use with other devices.

5.3 Palm lift manipulation

The aforementioned context menus are invoked near a manipulating stylus and hence, users can select items faster than they would for other positions, such as the edge of the screen. However, the difficulty is the menu invocation method. Most context menus are invoked using a long press, and this requires a certain time to invoke menus. The long press is not a smooth interaction, and is inconsistent with criteria “2: the more efficient menu invocation is, the better it is,” “3: the faster menu selection is, the better it is,” and “5: note-taking activity should not be disturbed by menus and their manipulation.”

To remove these inconsistencies, we use unintended touches to invoke menus and set the menu position. Palm rejection distinguishes between intended touches and unintended touches. Generally, unintended touches are considered as unnecessary; however, we recognize all touches as variables. Most unintended touches are generated by a writing hand, and these touches indicate the location of the writing hand. We consider making use of this indication for menu manipulation.

To design our menu manipulation, we develop a prototype handwriting application. The application is a web application and developed using JavaScript and HTML Canvas. When we use the web application on general multi-touch tablet devices, such as an iPad, the application detects multiple touch information. For the purpose of distinguishing between intended touches and unintended touches, we equip MLOAP, which was mentioned in Chapter 4. It can distinguish touches with almost same accuracy as Schwarz et al.'s [45] “spatiotemporal touch features.”

For our menu invocation method, there is no specific manipulation to invoke context menus, except lifting the dominant writing hand from the touch screen. When the dominant hand is lifted up, a context menu pops up at the recorded point. We call the entire manipulation “palm lift manipulation (PLM).”

5.3.1 Menu invocation using PLM

We specifically define the menu invocation method using PLM as palm lift invocation (PLI).

For applications that incorporate PLI, we first determine whether multiple touches on the screen are intended touches by a stylus, or unintended touches by manipulating hands. When there are multiple unintended touches, the mean value of the unintended touch coordinates is recorded as the center of the menu invocation position.

Then after all the touches disappear from the screen, a menu pops up under the lifted hand. Only touching of the stylus does not invoke menus, even if the stylus touch is removed from the screen.

In the scenario of note-taking, first, users put a hand on the device and take notes, second, lift the hand once they have finished writing a certain amount, and finally, move the hand to the location to write the next note.

In PLI, an interaction for menu invocation is assigned to the action of lifting the hand. It has the advantage that no special action for menu invocation is required.

However, there are disadvantages of PLI as follows:

1. The menu is not invoked unless the writing hand touches the screen .
2. PLI cannot be used at the screen edge where the writing hand cannot be placed .
3. The menu is invoked even when it is unnecessary .

When a user takes notes without placing a hand on a screen, drawback 1 occurs. In this case, to invoke the menu, it is necessary to put a hand on the screen once or tap the screen with multiple fingers.

Drawback 2 is similar to drawback 1. The user cannot invoke the menu if the writing hand is out of the screen. For example, when a right-handed user takes notes near the lower right edge of the screen, the user cannot put the hand on the screen correctly. In this case, it is necessary to put the hand on the screen or tap by manipulating fingers.

Even if the user does not intend to invoke menus, when the user lifts up the writing hand, the menu is automatically invoked. This unnecessary invocation is drawback 3, and it has no specific solutions.

Figure 5.1 shows the flow of the manipulation.

Most applications and studies have equipped long press or multi-touch interaction to invoke context menus, which requires a certain time. By contrast, PLM does not have to perform any specific manipulation to invoke the menu, and allows natural note-taking.

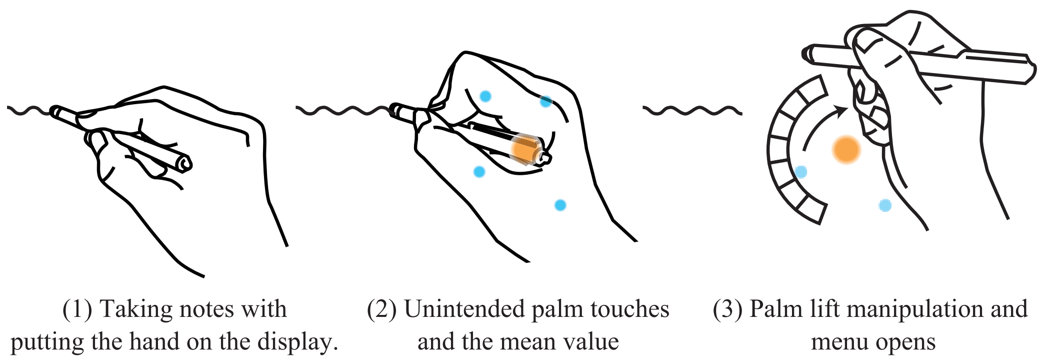

(1) Taking notes with putting the hand on the display.
(2) Unintended palm touches and the mean value
(3) Palm lift manipulation and menu opens

Figure. 5.1: Flow of palm lift manipulation

5.3.2 Position of menu invocation using PLM

In the case of displaying the menu at the screen edge, depending on the writing location, it may take time to select an item from the menu.

In the case of displaying the menu near the stylus nib, the menu may be unintentionally invoked near the stylus nib while taking notes and possibly hinder note taking manipulation.

Considering these matters, for PLM, we make use of the touch coordinates, which are produced by unintended touches, and the menu position is determined by the mean value of the coordinates. Therefore, when a user lifts up the writing hand, a menu is invoked underneath the hand.

The concept of PLM originates with the similar concept of "hover widgets [90]" by Grossman et al. "Hover widgets" invoke menus using the coordinates of a specific pen-based pointing device, which can provide coordinate information even if it is floating above the screen, and can invoke menus near pointing devices. Both PLM and "hover widgets" attempt to invoke menus near the manipulating stylus and pen-based pointing device. "Hover widgets" require users to focus their attention on the tip of the pointing device. PLM uses unintended touch information and hence, users may not be conscious of the menu position.

We assume that there are several advantages of invoking the menu underneath the hand. The distance from the stylus nib to the menu does not dynamically change depending on where the user is writing on the screen. Since the menu is invoked underneath the hand, especially when taking notes from left to right with right handedness, it is less likely to hide the letters which the user has written. Even if the menu is invoked unintentionally, the unintentionally invoked menu disappears when the user taps a space other than the menu. And in the movement of taking notes naturally put the hands in the space other than the menu, the menu naturally disappears and does not interfere with taking notes. These advantages meet criteria “2: the more efficient menu invocation is, the better it is,” “3: the faster menu selection is, the better it is,” and “5: note-taking activity should not be disturbed by menus and their manipulation.” However, if MLOAP erroneously determines that the hand is a stylus nib, it is possible for the user to accidentally select the menu while taking notes.

5.3.3 Design of the PLM menu

PLM is a manipulation technique and it can adopt any type of menu design. Therefore, as the prototype handwriting application, we adopt the arc menu, which we proposed in Chapter 3.

By adopting the arc menu, the following advantages are expected. Because of the arc-shaped design, when selecting items from the menu, users can recognize all items. This will help to meet criterion “4: all items on menus should be recognized at a glance.” If the menu was a different design, such as the pie menu [74], users would need to move the hand more extensively than for the arc menu to select items. The extra movement might be inconsistent with criterion “5: note-taking activity should not be disturbed by menus and their manipulation.” Figure 5.2 shows the comparison of the movement distance.

Additionally, other context menus completely cover the space underneath the hand, and users cannot see the next writing place at all. By contrast, the middle of the arc-shaped design is empty, and hence, this helps to recognize the place that will the most probably receive writing. The slight difference may have a positive influence in achieving criterion “5: note-taking activity should not be disturbed by menus and their manipulation.” Figure 5.3 shows the comparison of the recognition of the writing place.

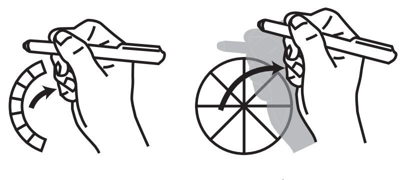


Figure. 5.2: Comparison of the movement distance

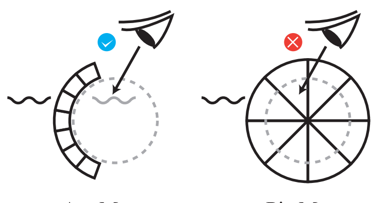


Figure. 5.3: Comparison of the recognition of the writing place

5.4 Research approach

From traditional approaches and existing applications, we found that there are several menu invocation methods and invocation positions that can be used in note-taking applications.

PLM is a unique menu manipulation method. It has not been compared with other menu manipulation methods. However, PLM is not compared and evaluated in the movement of actually writing letters. Therefore, it is crucial to clarify its usability and how its behavior affects users.

Therefore, we compare menus that can be used in note-taking applications including PLM.

First, select the terminal used in the experiment. Next, we make an experimental design. Then develop an experimental application. Finally, we classify the erroneous manipulations.

5.4.1 Device for the experiment

Currently, various tablets can be obtained. Android tablets can be obtained at a relatively low price. However, because the screen size and resolution vary, the same device may not be able to obtained to reproduce the experiment. Therefore, in this experiment, we use Apple's iPad Air 2, which is comparatively easy to obtain, and its performance is stable.

According to the scenario of note-taking, the tablet is placed on a desk and subjects sit on chairs to take notes.

The experiment is based on horizontal writing and we adopt landscape mode for the orientation of the screen. Although in Japanese, we traditionally use vertical writing, recently we accustom to use horizontal writing [91]. In many other countries, horizontal writing is more common than vertical writing.

A note-taking application can be operated by fingers; however, a stylus is used to approximate the action of taking a real note. The stylus is not an active stylus, which is dedicated to a specific device, but a general rubber-type stylus. The accuracy of the manipulation of the tablet is influenced by the size of the stylus nib [42]. Therefore, all experiments are conducted by the same stylus. For the experiment, we assume one-handed manipulation rather than both hands because we typically do not use both hands to take notes in a general scenario, and two-handed manipulation tends to be complicated.

5.4.2 Experimental design

First, we define the menu manipulation time. Second, we select a suitable combination of menu invocation methods and menu invocation positions. Finally, we describe the shape of the menu and what to write.

Definition of the menu manipulation time

We follow the same definition of the menu manipulation time as that of the arc menu experiment. The menu manipulation time is for the following series of manipulations:

1. leaving a stylus after drawing a stroke
2. invoking a menu
3. selecting an item from the menu
4. putting the stylus down to draw a new stroke

Selection of menu options for the experiment

For the purpose of evaluating PLM, which has a menu invocation method (PLI) and a menu invocation position (under the manipulating hand), we consider the options.

Because menus that can be used for note-taking applications vary, from them, we select menus that have been traditionally studied, and are actually used in existing applications.

The options for menu invocation methods are listed as follows:

1. bezel swipe
2. tap the bar at the edge of the screen
3. long press

The bezel swipe is adopted by INKredible [47], which is a well-known note-taking application, and hence. Hence the bezel swipe is appropriate as a comparison target.

In default setting, the iPad invokes the notification center by the user's swiping from the bottom of the screen, and the control center by the user's swiping from the top of the screen. Although these default interactions can be canceled by the iOS setting, by considering general use scenarios, it will be few cases to cancel those default interactions. Thus, to avoid duplication of the invocation interactions and these default interactions, in the experiment, we adopted the right or left-hand side of the edge as the orientation of the bezel swipe.

When a bar is set at the edge of the screen as an area for menu invocation, the note-taking area narrows. Tapping the bar for menu invocation was adopted by FiftyTree's Paper [83], which is also a well-known application. Thus, we adopt this menu invocation method as one of the options.

A long press is a widely used interaction, and various applications used in tablet devices adopt it. BambooPaper [1] is one of such applications. Thus, the long press is also one of the options.

A multi-touch, which simultaneously touches a number of places, is not one of the options. MLOAP, which was mentioned in Chapter 4, does not distinguish the difference between the PLI and a multi tap which intentionally taps the plural points.

Menu invocation using a stylus that can detect pressure and pen tilt angles requires dedicated hardware and cannot be used without the stylus; thus, it was not one of the options.

The invocation positions of the menu are as follows:

1. stylus nib
2. top edge of the screen

3. bottom edge of the screen
4. left edge of the screen
5. right edge of the screen

A stylus nib is widely used as the display position for a pop-up menu using a long press; thus, we set it as an objective position for comparison. General menus are often placed at the top, bottom, left, or right of the screen edge; thus, they are also objective positions for comparison.

Considering the combination of menu invocation methods and menu invocation positions, the following combinations have a similar manipulation process: tapping the bar at each side of the edge of the screen, and the bezel swipe for both the left edge and right edge of the screen.

When we compare all of them, the number of combinations increases, and the burden on the subject also increases. Therefore, we conduct a preliminary experiment to clarify which combination of menu invocation method and menu invocation position was operated quickly. Then, we reduce the options and select the fastest combination of menu invocation method and menu invocation position in that combination.

the design of the menu

In experiment, we focus on comparing several combinations of menu invocation methods and menu invocation positions. Regarding the design of the menu, we evaluated this in Chapter 3 and hence, will not compare the designs here.

It is necessary to unify the design of the menu that can be adjusted at all menu invocation positions. Instead of using the arc menu, we use the linear menu.

Each item equipped into the menu was a 60 px square, which is approximately 10 mm in actual size. A pen icon was attached to it, and each item was aligned vertically. There were seven items, and each of them had a different color. To compare the combination of menu invocation methods and menu invocation positions, we adopted a single-layered menu rather than a hierarchical multi-layered menu.

the contents of writing down

If the subjects are required to write documents on a board, the amount of memory that can be memorized at a time by watching the board by the subjects is different, and the operation time is affected. Therefore, the characters is displayed as a background image of the application. The image used in the experiment is displayed in Figure 5.4.

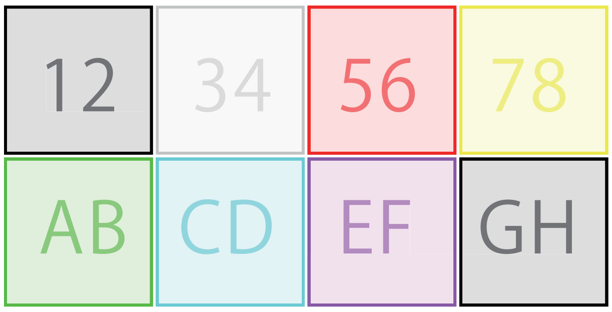


Figure. 5.4: The background image for the experiment

PLI cannot invoke menus because of disadvantages 2 at the screen edge, where hands cannot be placed. However, in any invocation method, it is necessary to set conditions that can correctly activate the menu invocation. Therefore, keep the background image at the center of the screen and secure the space where you can place your hand on the right edge as well.

For this reason, the images for tracing used in the experiments are kept close to the center of the screen, even at the right-hand side, and the space for hands to be placed was held.

The subjects are required to perform the following manipulations: first, trace the written number or letter; second, invoke the menu using the indicated methods; and finally, select the same colored items on the menu and restart tracing.

The default color is black, and hence, subjects do not have to select the menu for the upper left box, which is a black character.

We measure the manipulation time from finishing writing characters in the box, invoking the menu, selecting the color to be written next, and starting to write the next number or letter. Then we calculate the average of the total data.

5.4.3 Application development

Even if the iPad used for the experiment cannot be obtained, web applications can be used on alternative devices. Therefore, the application is developed using HTML Canvas and JavaScript as a web application. We equip MLOAP in the application.

In Safari, which is the default browser of the iPad, the URL form is displayed at the top of the screen and the drawing space of the screen becomes narrow. To avoid displaying the URL form, we create a shortcut icon on the home screen and activate the web application from this icon.

Once the application is activated, a pop-up window indicates several menu invocation methods and invocation positions: there are four types of menu invocation methods and six types of menu invocation positions.

The menu invocation methods are "PLI," "long press," "tap a menu invocation bar" and "bezel swipe." The menu invocation positions are "the screen edge (top, bottom, left and right)," "the stylus nib" and "the center of a region under the manipulating hand."

These menu invocation methods and menu invocation positions can be combined.

The way of the menu manipulation

In the case of PLI, we implement the manipulation way written in 5.3.1.

In the case of a long press, subjects put a stylus nib for 500 ms on a screen. Because a menu is invoked for approximately 500 ms for the long press with BambooPaper [1]. It was difficult to keep tapping without moving the location of the stylus nib, and hence, a width and height of 5 px were set as a permissible range, while the stylus nib keeps touching the screen for 500 ms.

In the case of a bezel swipe, when a swipe from the edge of the screen is performed, drawing strokes are not written, but the menu is invoked when the stylus nib leaves the screen.

In the case of tapping a bar, subjects tap a gray colored bar to invoke the menu. To keep a drawing area as wide as possible, we set the size of the bar as follows:

- Arranging the bar above and below, the width is 1024 px, which is same as the screen, and the height is 20 px.

- Arranging the bar on the left and right, the width is 20 px and the height is 748 px which is shorter than the length of the end of the screen.

For the iPad Air 2, 20 px is approximately 4 mm. Apple's iOS Human Interface Guidelines state that the tap area using fingers should be more than 7 mm. Whereas, the status bar of the iPad, which is 20 px in height, is assigned to the function of returning to the top after scrolling. It indicates that if the one side of an object is sufficiently long, users can tap the object with 20 px height or width. Therefore, the size of the bar is practical.

After menu invocation, the menu is hidden by the action of selecting items on the menu or by the action of touching the drawing area. Menu selection is determined when the selected stylus nib is released from the menu.

To avoid memorizing the order of the items and to diminish the possibility of changing the manipulation speed, the color of the items changes randomly for each menu invocation.

The palm rejection algorithm may erroneously determine one of the touch points by a hand as a stylus nib. When subjects display a menu by the long press, tapping the bar, and the bezel swipe, subjects can invoke a menu with putting their hand on the screen. However, if a subject operates with putting the hand on the screen even after displaying the menu, strokes will be drawn, and the menu will be hidden. For this reason, when operating those menu invocations, subjects are requested to float their hand and operate with only the stylus nib.

The menu position and the menu design

When the menu is displayed at the left and right edge, the size of the menu is 60 px in width and 420 px in height. We name this menu as the portrait linear menu. The portrait linear menu is displayed at the center of the left and right edge. When the menu is displayed at the top and bottom, the size of the menu is 420 px in width and 60 px in height. We name this menu as the landscape linear menu. The landscape linear menu is displayed at the center of the upper and lower edge.

When the invocation position is at the stylus nib or under the manipulating hand, and when the menu is invoked near the edge of the screen, a part of the menu protruded from the screen. To avoid this, the menu position is adjusted, and the entire menu is displayed inside the screen.

Preliminary experiment

To analyze the movement of menu manipulation while subjects are writing a note, the following activities are recorded and sent to a server after the experiment:

- Timing to put the stylus nib on the screen
- length of the written stroke,
- Timing to lift the stylus nib from the screen
- coordinates of the start and end points
- selected pen color
- condition of the touch points (whether the touch is to select the menu or start writing a stroke)

5.4.4 Classification of erroneous manipulation

It is difficult to distinguish erroneous manipulation using only the data acquired from the application. For example, if the menu does not appear when the subject attempts to invoke it, the menu pops up unintentionally, or the incorrect color is selected. Therefore, all the experiments are recorded on video. When we analyze them, they are checked against data recorded by the application to classify the correct manipulation and erroneous manipulation.

When menu manipulation does not operate as intended or operates with mistake, we classify the data into the following five types and record it as erroneous manipulation:

- wrong selection
- incorrect invocation
- incorrect place
- unintentional invocation

- unintentional selection

Furthermore, these types are excluded from the object of measuring the menu manipulation time depending on the content of the erroneous manipulation.

If the subject mistakenly selected a different color from the end of writing to the beginning of writing, we recorded that number as **wrong selection (WS)**. Because the case of WS is not a correct manipulation, it is excluded as menu manipulation time from the measurement targets.

If the menu is not invoked even if menu invocation manipulation is performed, the number of times it occurred is recorded as **incorrect invocation (II)**, and it is excluded as menu manipulation time from the measurement targets.

In PLI, even though the menu display position is set as the stylus nib, sometimes MLOAP determined the stylus nib incorrectly. Then, the menu is invoked near the under the hand. In this case, it is recorded as **incorrect place (IP)**, and it is excluded as menu manipulation time from the measurement targets.

In PLI, the menu is invoked when the subject removes the hand from the screen; therefore, the menu may be invoked unintentionally. Even in the case of a long press, the menu can be invoked when the subject stops moving the stylus nib in the middle of writing. For these scenarios, we determined the case in which the menu is unintentionally invoked as **unintentional invocation (UI)** and recorded the number of times it occurred.

When a UI occurs, depending on the menu display position, it is possible for the menu to be unintentionally selected by the hand during writing. In this case, we recorded the number of occurrences as **unintentional selection (US)**. When US occurred, the subject needed to reset to the original color. This menu manipulation is unnecessary if it is operated correctly. Therefore, it is excluded as menu manipulation time from the measurement targets.

5.4.5 Preliminary experiment

To execute the experiment using the same conditions, we conducted the preliminary experiment using three right-handed subjects.

The results are shown in Table 5.1.

Table. 5.1: Results of the pre-experiment

Display position	**Invocation method**	**manipulation time average (ms)**
Left edge	bezel swipe	2702
Right edge	bezel swipe	2785
Left edge	tap	2914
Right edge	Tap	2974
Bottom edge	tap	3064
Top edge	tap	3232

As a result of one way ANOVA we found that there were no significant differences between all combinations of menu display positions and menu invocation methods.

Boritz et al.[92] indicated that in the case of right handedness, moving right from the center of the pie menu is significantly earlier than moving downward. However, in the flow of writing characters and manipulating the menu at the edge of the screen from there, it was found that there was no significant difference in both of the portrait linear menu and the landscape linear menu.

Therefore, we selected the fastest combination of menu invocation method and menu display position. The menu invocation method was the bezel swipe and the menu display position was the left edge. Portrait linear menu was selected as the menu design.

Using the selected menu invocation methods and menu display positions, multiple comparisons are conducted.

To compare and evaluate the menu invocation methods, it is necessary to standardize the menu display positions. Similarly, to evaluate the menu display position, the menu invocation methods have to be standardized.

In order to compare the menu invocation method by PLI, for the bezel swipe and the long press, it is necessary to set the same positions. The left edge of the screen and the stylus nib are the positions which can be compared by the three menu invocation methods.

In order to evaluate the display position of invoking the menu under the hand proposed by the PLM, the invocation method is set to PLI. The left edge of the screen, the stylus nib, and the center of a region under the hand are compared to evaluate the menu display positions of PLI, The three types of display position is displayed in Figure 5.5.

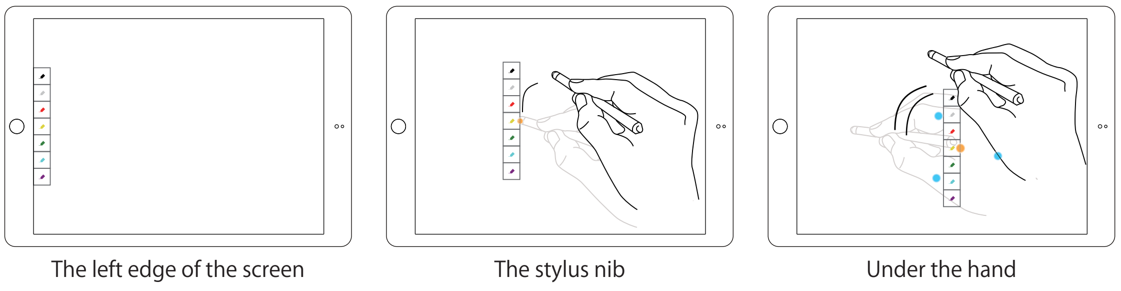


Figure. 5.5: The three types of display position

In the case of the left edge of the screen, the menu is displayed in the center of the screen at the left end same as in the preliminary experiment. In the case of a pen tip, the menu is displayed next to the coordinates of the last contact determined as the stylus nib. In the case of an under the hand, the menu is displayed next to the coordinates of the center of a plurality of points determined as the hand.

We numbered each combination, and present the comparison items for the menu display positions and menu invocation methods in Table 5.2.

Table. 5.2: Comparison list for the menus

No.	Item of menu to be compared	Invocation method	Display position
1	Invocation method	Bezel swipe	Left edge
2		Long press	Left edge
3		PLI	Left edge
4	Invocation method	Bezel swipe	Stylus nib
5		Long press	Stylus nib
6		PLI	Stylus nib
3	Display position	PLI	Left edge
6		PLI	Stylus nib
7		PLI	Under the hand

For each comparison item, the points to be compared are highlighted in gray. Seven experiments were conducted, and the menu invocation method and menu display position were compared. The null hypotheses for the items to be compared are listed as follows:

Hypothesis 1 There is no difference in the manipulation time of the menu displayed at the left of the screen depending on the menu invocation method.

Hypothesis 2 There is no difference in the manipulation time of the menu displayed at the stylus nib depending on the menu invocation method.

Hypothesis 3 There is no difference in menu manipulation time using PLI depending on the menu display position.

5.4.6 Comparison of menu invocation methods and menu display positions

Subjects were recruited from students of university classes for which the author was a part-time instructor. Eight right-handed subjects in their teens to twenties participated.

For the subjects, we first explained the experiment using the following points:

- overview of research on note-taking applications
- subjects should assume that the present application is intended to take notes
- the purpose of this experiment is the comparison of menu invocation methods and the comparison of menu display positions
- explanation of palm rejection
- explanation of menu invocation methods (PLI, bezel swipe, and long press)
- explanation of menu display positions (left edge of screen, under the hand, and stylus nib)
- subject should select from the menu the same color as that displayed and trace the numbers or letters
- the color position displayed on the menu changes every time
- if subjects select the wrong color, they should reselect the correct color and rewrite
- when the subjects cannot write a line, they can rewrite it
- when writing with a hand on the screen, the subjects should not be concerned if the line remains under the hand
- subject should operate all the menu invocation methods and menu display positions without any preference

- an explanation of the questionnaire after the experiment

Regarding the preliminary experiment, after the subject traced the numbers and letters on the screen with the same color, we measured and compared the manipulation time from finishing writing the characters in the box, invoking the menu, and selecting the color to be written next to starting to write a line with the next color.

To avoid the subjects' becoming accustomed to a specific menu invocation method or menu display position, the experiments were conducted in a random order. The subjects were required to practice twice for all patterns before the experiment.

The experiment took approximately 40 minutes, including the explanation of the experiment's purpose and two exercises. There was no break during the experiment.

5.5 Result

The experimental results are shown in Table 5.3. The detail of the results are shown in Appendix, Table 1.

The operations excluding WS, II, and IP from the total number of menu operations were the correct operations. The average of the manipulation time was the average of the correct operations. The success rate is the ratio of correct operations except WS, II, and IP to the total number of menu operations.

5.5.1 Menu invocation methods at the left edge of the screen

First, we compared the three menu invocation methods of PLI, bezel swipe, and long press with the menu display position as the left edge of the screen.

A comparison of the manipulation time of the menu invocation methods at the left edge of the screen is shown in Figure 5.6.

As a result of one-way ANOVA, when invoking the menu at the left edge of the screen, there were significant differences between all the manipulation times ($p = 1.12e - 06$). The result of one-way ANOVA is shown in Table 5.4.

Then we had multiple comparisons of menu invocation methods using Tukey's method. The result showed that the comparison between a long press and bezel swipe was ($p = 0.01$), PLI and a bezel swipe was ($P = 0.02$), and PLI and a long press was ($p| < 0.0001$) at a 95% significance level. When displaying the menu at the left edge of the screen,

Table. 5.3: Comparison of the menus

No.	Invocation method	Display position	WS [1]	II [2]	IP [3]	UI [4]	US [5]	Success Rate	Average Time (ms)
1	Bezel swipe	Left edge	1	1	0	0	0	96%	2217
2	Long press	Left edge	4	11	0	9	0	73%	2588
3	PLI	Left edge	3	4	0	32	0	88%	1893
4	Bezel Swipe	Stylus nib	1	7	0	0	0	86%	2174
5	Long press	Stylus nib	7	8	0	19	7	73%	2283
6	PLI	Stylus nib	2	15	14	40	3	55%	2117
7	PLI	Under the hand	1	6	0	25	1	88%	1834

[1] Wrong Selection
[2] Incorrect Invocation
[3] Incorrect Place
[4] Unintentional Invocation
[5] Unintentional Selection
The total number of WS, II and IP will not represent the success rate. This is because WS, II and IP can occur several times for one menu manipulation.

Table. 5.4: The result of one-way ANOVA

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Invocation	2	10772135	5386067	15.205	1.12e-06 ***
Subjects	7	9022802	1288972	3.639	0.00126 **
Residuals	134	47465742	354222		

PLI was faster than the bezel swipe and long press. A bezel swipe was faster than a long press. Therefore, the result shows that the null hypothesis 1 was rejected.

5.5.2 Menu invocation method at the stylus nib

Next, the menu display position was set as the stylus nib, and the three menu invocation methods of PLI, bezel swipe, and long press were compared.

The comparison of the manipulation time of the menu invocation method at the stylus nib is shown in Figure 5.7.

As a result of one-way ANOVA, there were no significant differences when invoking the menu at the stylus nib ($P = 0.33$). The result shows that the null hypothesis 2 could not be rejected.

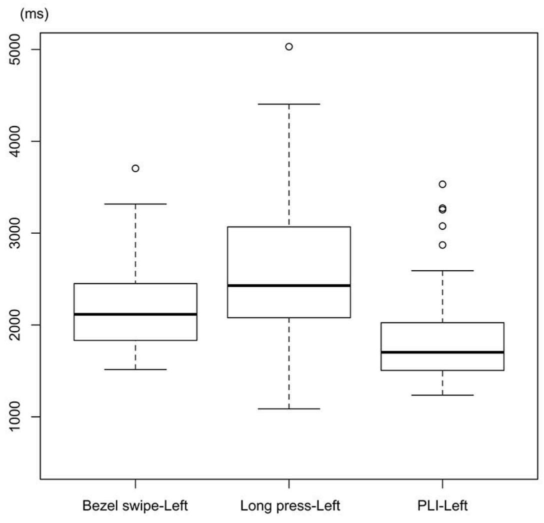


Figure. 5.6: Comparison of the menu manipulation time of the menu invocation methods at the left edge of the screen

5.5.3 Menu display position using PLI

The menu invocation method was set to PLI. The menu display positions of the three types of position, stylus nib, under the hand and left edge of the screen, were compared.

The comparison of the manipulation time of the three menu display positions using PLI is shown in Figure 5.8.

As a result of one-way ANOVA, the difference in the menu display position of the menu using PLI showed no significant difference for all manipulation times ($P = 0.09$). The result shows that the null hypothesis 3 could not be rejected.

5.6 Discussion

We discuss the results obtained by comparative experiments of menu manipulation based on the questionnaire results after the experiment. The results of the questionnaire are described in the Appendix.

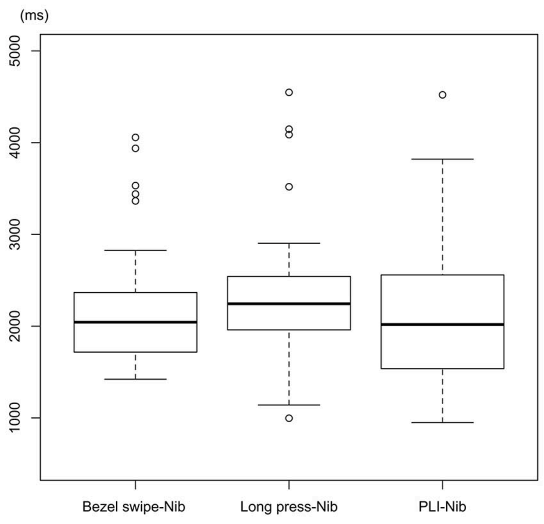


Figure. 5.7: Comparison of the manipulation time of the menu invocation methods at the stylus nib

5.6.1 Discussion of the menu invocation method at left edge of the screen

As a result of the experiment, when displaying the menu at the left edge of the screen, there were significant differences for all menu manipulation times for the three invocation methods.

A long press required the subjects to put the stylus on the screen for 500 ms to invoke the menu. Moreover, the menu display position was set to the left edge of the screen rather than the stylus nib, which took a longer time than the other menu invocation methods. For the bezel swipe, the subjects needed to swipe the left edge of the screen to invoke the menu; therefore, it took more time than PLI.

Because PLI invoked a menu due to subjects lifting the manipulating hand from the screen, the menu invocation speed of PLI was faster than the other menu invocation methods.

For PLI, the number of occurrences of UI that were unrelated to menu manipulation was 32, which was the largest number for all menu invocation methods. However, in the case of the left-hand side of the screen, for the question "Does PLI affect writing notes when invoked at unintended timing?" seven subjects answered "no problem will occur at all." One subject answered "almost no problem will occur." Therefore, we consider

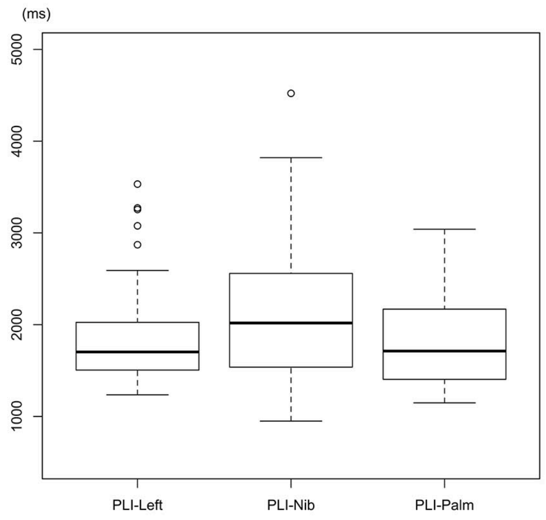


Figure. 5.8: Comparison the manipulation time of menu positions using PLI

that the influence was small.

The success rate of menu manipulation was 96% for the bezel swipe, which was the highest, and 73% for the long press, which was the lowest. The success rate of the bezel swipe was high because the subjects did not need to tap a specific position, and they only needed to swipe roughly from the edge of the screen. Four subjects answered that the bezel swipe was "frequently used and familiar." Therefore, it is possible that it was a familiar manipulation method, to some extent.

II occurred 11 times for the long press. Three subjects answered that the long press was "frequently used and familiar." However, in the recorded video, some subjects pressed the pen strongly and moved it when the menu could not be invoked. Therefore, it is possible that the unsuccessful operation caused the low success rate.

II occurred four times for PLI because MLOAP did not correctly distinguish the touch of the hand, and the menu was not invoked, even when the hand was lifted. Regarding PLI, one subject answered that it "could not display as intended at all" and four answered "could not display intended much." This will be improved by increasing the accuracy of palm rejection for invoking a menu using PLI.

5.6.2 Discussion of menu invocation method at the stylus nib

As a result of the experiment, we found that there was no significant difference in menu manipulation time using the three invocation methods. The difference in the manipulation time for each menu invocation method was smaller when the menu display position was set to the left edge of the screen.

Compared with the case in which the menu display position was at the left-hand side of the screen, there was little difference in the manipulation time of the bezel swipe.

The long press was approximately 300 ms shorter. This is considered to be a more suitable combination than displaying the menu at the left edge of the screen, because the menu is invoked at the stylus, and, menu selection can be performed immediately. However, II occurred eight times and UI occurred 19 times. We believe that the menu was invoked unintentionally because subjects had the stylus nib attached to the screen while writing the character. US occurred seven times by subjects choosing a menu unintentionally.

PLI was approximately 200 ms later than when the menu was displayed at the left edge of the screen. If the menu was correctly invoked at the stylus nib, the manipulation time of the stylus nib should have been faster than it was when the menu was at the left edge of the screen. However, the opposite result occurred. The reason of the result was the low success rate, which was 55%.

IP only occurs for the combination of PLI and the menu position at the stylus nib. When subjects lifted up the stylus first, the palm rejection algorithm was able to recognize one of the remaining touches as a stylus nib. Then, subjects lifted up their hand, and menus would be displayed under their hand rather than at the stylus nib. Two subjects made IP five times out of seven, and in total, IP occurred 14 times. Thus, the frequency of IP was one factor behind the success rate. Additionally, II occurred 15 times in total. One subject in particular made II five times, while two subjects made II three times. These results indicate that most of the subjects could not operate PLI in combination with the menu position at the stylus nib as intended.

5.6.3 Discussion of the menu display position using PLI

As a result of the experiment, we found that there was no significant difference in the three types of menu display positions of the menu for the invocation method using PLI.

When the menu was displayed under the hand, there were no subjects who answered "always displayed at the intended location," four subjects answered "generally displayed at the intended location," three subjects answered "not displayed at the intended location so much," and one answered "not displayed at the intended location at all."

For PLI, when invoking the menu under the hand, because the menu display position was calculated from the number of touching points of the hand, the menu was not always displayed under the hand at a particular position. Therefore, we considered that there was a difference between the position intended by the subject and the position actually displayed. However, the same manipulation time as the other menu display positions was still achieved. Therefore, if the accuracy improved, under the hand could be added to the options for menu display positions for the menu of the note-taking application.

5.6.4 Internal validity of the experiment

As threats to be considered, we list the following influences.

1. The condition of the long press invocation
2. The learning effect during the experiment
3. The subject's experience

Influence of the condition of the long press invocation

When we implemented the long press function, it was difficult to keep the stylus nib staying at the same point and within 1 px for 500 ms on the screen. For this reason, we designed with the intention of distinguishing the long press even if the stylus nib moves 5 px in length and, with a width of not more than 500 ms. Normally, the long press is not displayed even if you keep pressure on the pen tip after writing a line of script. However, because of this implementation, after writing a line of script, if the pen tip is kept in position without moving more than 5 px in length nor width for 500 ms, the menu is displayed. For this reason, the frequency of UI, is one of the most erroneous operations. However, the occurrences of UI are no influence on the operation time of the menu manipulation.

Influence of the learning effect during the experiment

Because all the experiments were conducted randomly, there was no possibility of a subject becoming accustomed to only the specific menu invocation method and display position.

Influence of the subject's experience

As a result of the questionnaire, because the manipulation method using the bezel swipe and long press are existing manipulation methods, we know that subjects who were accustomed to them were included. To minimize the difference in experience, we attempted all menu invocation methods twice before the experiment.

5.6.5 External validity of the experiment

For the experiments, we describe the generality of the age of the subjects, experience of using the tablet, and equipment used.

Generality of age

The subjects were eight students ranging from first year to third year at university. Similar results may not be obtained from subjects who are younger or older.

Generality of experience of using the tablet

Regarding the tablet, in the questionnaire, three subjects answered that they "use it often at home or work," four subjects answered "I have used it before; however, I am not accustomed," four subjects answered "almost never used," and there were no subjects who had "never used it before." Therefore, subjects who had never used a tablet may generate different results.

Generality of equipment used

The tablet used in this experiment was the iPad Air 2, which is widely used, and the screen size was 9.7 inches. When we compare menu display positions, experiments using devices with different screen sizes may affect the results. In particular, when experimenting with tablets with larger screen sizes, the comparison of menu display positions is expected to generate different results.

5.7 Conclusion

Comparative experiments were conducted from two viewpoints; the menu invocation method and, menu display position in the note-taking application.

In the experiment in which the menu was displayed at the left-hand side of the screen using three menu invocation methods, we found that there was a significant difference among PLI, bezel swipe, and long press; PLI was the fastest, and long press was the slowest. As a menu invocation method at the left-hand side of the screen, PLI was a practical invocation method because the manipulation time was faster than the other menu invocation methods and the success rate was higher than that of the long press.

In the experiment in which the menu was displayed at the stylus nib, there was no significant difference among the three menu invocation methods.

In the experiment in which the menu invocation method was set to PLI and the menu was displayed at the left of the screen, at the stylus nib, and under the hand, there was no significant difference among the three display positions of the menu.

Through these experiments and discussions, we clarified which menu invocation methods were manipulated rapidly.

However, in the experiments, we could not be clear which menu invocation positions were manipulated faster. We assume that the screen size affects the results. Additionally, the background image to be drawn was set to the center of the screen so that the PLI could be performed correctly. To investigate more practical usability of PLI, it is our future work to compare PLI with various screen sizes.

It is also our future work to improve the discrimination accuracy of the stylus nib and hand using palm rejection, which is the basis of the menu invocation method. How to take notes, how to hold a stylus is different for individuals. Therefore, acquiring data individually and making suitable learning data for individuals is one reasonable idea to improve the accuracy of palm rejection.

Chapter.6 Conclusion and future work

In the conclusion, we answer each research question. Then, we review each approach and discuss future work. Finally, the contribution of this research is described.

6.1 Arc menu for note-taking applications

We reviewed the development process for the arc menu. To evaluate the manipulation time of the arc menu, we compared the arc menu with two types of linear menu, which represent a general menu design for note-taking applications. The result of the experiment was that, while the users use the application, the arc menu is manipulated faster than the PopupLM. Therefore, we conclude that if it is in the same condition, the menu design we proposed makes smoother the task of taking notes compared to the general menu design.

Through this experiment, we found that besides menu design, the following factors have an influence on a smooth way of taking notes using applications. One was the difficulty of taking notes without palm rejection. Another was the menu invocation method. The two problems made us pursue the following research.

6.2 Toward the reduction of incorrectly drawn ink retrieval

After the experiment of menu design, we found that palm rejection was also important for taking notes as if users were using a real pen and notebook. If an application is not equipped with palm rejection, users are compelled to float the writing hand, and the writing posture causes fatigue.

Even if the application is equipped with palm rejection, IDIR forces users to rewrite the retrieved strokes. To reduce the frequencies of IDIR, we developed MLOAP. While comparing an application with MLOAP and other applications are equipped with palm rejection, we confirmed that IDIR occurred less for the application with MLOAP than other applications. Therefore, the MLOAP will be an algorithm option of palm rejection in order to reduce the frequency of IDIR.

We set a fixed-sized dynamic occlusion area under the palm, although the size of the area should be able to adjust to the user. Additionally, if learning data for the SVM for each user can be collected, the data will fit each user and thus, correctness will increase. These improvements are future work.

6.3 Evaluation of menu manipulation using palm lift manipulation

PLI invokes context menus without any interaction except lifting the manipulating hand. To evaluate the menu invocation methods and menu invocation positions, we conducted experiments. The following menu invocation methods were compared; PLI, the bezel swipe and long press. The following menu positions were compared; PLI, the left edge, stylus nib and under the palm. Through the experiments, when the menu position was set at the left edge, we found that PLI was significantly faster than other menu invocation methods. However, the menu invocation positions for PLI had no significant differences between under the palm, stylus nib and left edge of the screen.

Currently, the screen size of tablet devices is becoming larger, and hence, when larger tablet devices become popular, there will be a different result for the comparison of the menu invocation position. However, we have not done experiments with multiple screen sizes. It is a future work to conduct an experiment to compare the combination of the invocation methods of the menu and the invocation positions using tablets of multiple screen sizes.

6.4 Contribution of this research

Through the investigations and discussions of menus, we developed qualitative criteria for menus of note-taking applications. This contributes to understanding menus more deeply, and to the development of menus for note-taking applications. The arc menu is manipulated faster than the PopupLM. In situations where users need to manipulate menus quickly, such as taking notes, the arc menu will contribute to expanding that choice of menus. The MLOAP reduced the frequency of the occurrence of IDIR. Therefore, the MLOAP is a possible option as an algorithm for palm rejection. PLI lets users to be able to invoke menus simply by lifting their hand. PLI can be embedded if it is a tablet with multi-touch interaction, moreover, it can be used even for tablets that can not use an active stylus. Therefore, in order to smoothly invoke a menu with a note-taking application on a general tablet device, PLI is a valid invocation method.

This series of researches is what should be considered when note-taking applications become more popular. We believe that this research will contribute to expand the choices of menus that can be used with note-taking applications and MLOAP will be practical palm rejection algorithm.

Acknowledgement

I am deeply grateful to families in my neighborhood who took part in the experiment of menu design, my colleagues at Accenture who joined the experiment of palm rejection and students at Musashino Art University who took part in the experiment of PLI. I would like to offer my special thanks to my classmates who gave me constructive comments and warm encouragement, and Mr. Jack Waldron who has assisted in the proofreading of this paper.

I offer my appreciation to Prof. Nakatani, who provided carefully considered feedback and valuable comments. I am also indebted to Prof. Tsuda, Associate Prof. Ohki and Associate Prof. Ryoke whose opinions and information helped me very much throughout the production of this study.

Appendix

Detail of table 5.3

Table. 1: Detail of table 5.3

Invocation method	Display position	Subject	WS	II	IP	UI	US	Success / Total	Success Rate	Average Time (ms)
BezelSwipe	Left Edge	Total	1	1				54/56	96%	2217
		A						7/7	100%	2255
		B	1					6/7	86%	2474
		C						7/7	100%	1990
		D						7/7	100%	1985
		E						7/7	100%	1828
		F						7/7	100%	2157
		G						7/7	100%	2193
		H		1				6/7	86%	2999
Long press	Left Edge	Total	4	11		9		41/56	73%	2588
		A		1				6/7	86%	2552
		B	1					6/7	86%	1962
		C		3		3		4/7	57%	3676
		D	1	2				4/7	57%	2299
		E	1			1		6/7	86%	2435
		F		2				5/7	71%	1703
		G	1	1		4		5/7	71%	2808
		H		2		1		5/7	71%	3522
PLI	Left Edge	Total	3	4		32		49/56	88%	1893
		A						7/7	100%	1620
		B		1		2		6/7	86%	2393
		C		1		7		6/7	86%	2152
		D		1		11		6/7	86%	1941
		E	2			5		5/7	71%	2078
		F		1				6/7	86%	1511
		G				6		7/7	100%	1837
		H	1			1		6/7	86%	1698
BezelSwipe	Stylus Nib	Total	1	7				48/56	86%	2174

Invocation method	Display position	Subject	WS	II	IP	UI	US	Success / Total	Success Rate	Average Time (ms)
		A		1				6/7	86%	2141
		B						7/7	100%	1873
		C						7/7	100%	2239
		D		2				5/7	71%	1706
		E		2				5/7	71%	1934
		F						7/7	100%	1918
		G	1					6/7	86%	2311
		H		2				5/7	71%	3444
Long press	Stylus Nib	Total	7	8		19	7	41/56	73%	2283
		A		1		3	2	6/7	86%	2259
		B	2	1		2	2	4/7	57%	2377
		C	1	2		3		5/7	71%	2491
		D						7/7	100%	2114
		E		2		1		5/7	71%	2033
		F	1	1				6/7	86%	1838
		G	2	1		10	3	4/7	57%	2800
		H	1					6/7	86%	2592
PLI	Stylus Nib	Total	2	15	14	40	3	31/56	55%	2117
		A		2		3		5/7	71%	2111
		B				1		7/7	100%	2410
		C	1	5		7	1	1/7	14%	1550
		D	1	3	5	11	1	2/7	29%	3139
		E		2	5	8		2/7	29%	2627
		F				1		7/7	100%	1410
		G		3	3	7	1	1/7	14%	2540
		H			1	2		6/7	86%	2119
PLI	Under The Hand	Total	1	6		25	1	49/56	88%	1834
		A		1		2	1	6/7	86%	2059
		B	1	1		7		5/7	71%	2526
		C				1		7/7	100%	1703
		D		1		1		6/7	86%	1810
		E				2		7/7	100%	1751
		F		2		2		5/7	71%	1640
		G		1		5		6/7	86%	1351
		H				5		7/7	100%	2021

Questionnaire

Question 1

Does the bezel swipe invoke menus as expected?

- Always invoked as expected 6
- Mostly invoked as expected 2
- Not invoked as expected many times 0
- Not invoked as expected 0

Question 2

Does the long press invoke menus as expected?

- Always invoked as expected 0
- Mostly invoked as expected 4
- Not invoked as expected many times 3
- Not invoked as expected 1

Question 3

Does PLI invoke menus as expected?

- Always invoked as expected 0
- Mostly invoked as expected 4
- Not invoked as expected many times 2
- Not invoked as expected 2

Question 4

Which menu invocation method is suitable for note-taking applications? Provide impressions for before the experiment and after the experiment.

Before the experiment

- Bezel swipe 5
- long press 2
- PLI 1

After the experiment

- Bezel swipe 8
- long press 0
- PLI 0

Question 5

Was the menu displayed in the expected position when the menu display position was under the hand and the menu invocation method was PLI ?

- Always displayed as expected 0
- Mostly displayed as expected 4
- Not displayed as expected many times 3
- Not displayed as expected 1

Question 6

Was the menu displayed in the expected position when the menu display position was the stylus nib and the menu invocation method was PLI

- Always displayed as expected 1
- Mostly displayed as expected 4
- Not displayed as expected many times 1
- Not displayed as expected 2

Question 7

When menus are invoked unintentionally by PLI, how does it affect each menu display position?

Screen edge

- Does not affect 7
- Rarely affects 1
- Slightly affects 0
- Considerably affects 0

Under the hand

- Does not affect 1
- Rarely affects 1
- Slightly affects 4
- Considerably affects 2

Stylus nib

- Does not affect 1
- Rarely affects 0
- Slightly affects 4
- Considerably affects 3

Question 8

Which menu display position is suitable for note-taking applications?

- Screen edge 7
- Under the hand 0
- Stylus nib 1

Question 9

Which menu invocation method is suitable for note-taking applications?

- Bezel swipe 7
- long press 0
- PLI 1

Question 10

Have you used a tablet device?

- Yes, I have used it very well 3
- Yes, I have used it several times, but not frequently 4
- No, I have rarely used it 1
- No, I have never used it 0

Question 11

Have you used the bezel swipe?

- Yes, I have used it very well 4
- Yes, I have used it several times, but not frequently 4
- No, I have rarely used it 0
- No, I have never used it 0

Question 12

Have you used the long press?

- Yes, I have used it very well 3
- Yes, I have used it several times, but not frequently 4
- No, I have rarely used it 1
- No, I have never used it 0

REFERENCE

[1] BambooPaper, "Wacom." http://www.wacom.com/ja-jp/jp/everyday/bamboo-paper.

[2] G. J. Lepinski, T. Grossman, and G. Fitzmaurice, "The design and evaluation of multitouch marking menus," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, (New York, NY, USA), pp. 2233–2242, ACM, 2010.

[3] M. S. Uddin, C. Gutwin, and B. Lafreniere, "Handmark menus: Rapid command selection and large command sets on multi-touch displays," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, (New York, NY, USA), pp. 5836–5848, ACM, 2016.

[4] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman, "An empirical comparison of pie vs. linear menus," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, (New York, NY, USA), pp. 95–100, ACM, 1988.

[5] P. Brandl, J. Leitner, T. Seifried, M. Haller, B. Doray, and P. To, "Occlusion-aware menu design for digital tabletops," in *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, (New York, NY, USA), pp. 3223–3228, ACM, 2009.

[6] G. Fitzmaurice, A. Khan, R. Pieké, B. Buxton, and G. Kurtenbach, "Tracking menus," in *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, (New York, NY, USA), pp. 71–79, ACM, 2003.

[7] G. P. Kurtenbach, A. J. Sellen, and W. A. S. Buxton, "An empirical evaluation of some articulatory and cognitive aspects of marking menus," *Hum.-Comput. Interact.*, vol. 8, pp. 1–23, Mar. 1993.

[8] G. Bailly, E. Lecolinet, and L. Nigay, "Flower menus: A new type of marking menu with large menu breadth, within groups and efficient expert mode memorization," in *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, (New York, NY, USA), pp. 15–22, ACM, 2008.

[9] N. Banovic, F. C. Y. Li, D. Dearman, K. Yatani, and K. N. Truong, "Design of unimanual multi-finger pie menu interaction," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, (New York, NY, USA), pp. 120–129, ACM, 2011.

[10] M. Kam, J. Wang, A. Iles, E. Tse, J. Chiu, D. Glaser, O. Tarshish, and J. 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, (New York, NY, USA), pp. 531–540, ACM, 2005.

[11] K. Hinckley, S. Zhao, R. Sarin, P. Baudisch, E. Cutrell, M. Shilman, and D. Tan, "Inkseine: In situ search for active note taking," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, (New York, NY, USA), pp. 251–260, ACM, 2007.

[12] A. Bauer and K. R. Koedinger, "Note-taking, selecting, and choice: Designing interfaces that encourage smaller selections," in *Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '08, (New York, NY, USA), pp. 397–406, ACM, 2008.

[13] N. PLUS, "Writeon notes plus." https://www.writeon.cool/notes-plus/.

[14] Evernote, "Penultimate." https://evernote.com/intl/jp/penultimate/.

[15] GoodNotes, "Time base technology limited." http://www.goodnotesapp.com.

[16] Office, "Microsoft." https://www.office.com/.

[17] Illustrator, "Adobe." http://www.adobe.com/jp/products/illustrator.html.

[18] B. Shneiderman, "Designing menu selection systems," *Journal of the American Society for Information Science.*, vol. Vol. 37, pp. p57–70. 14p., Mar 1986.

[19] M. J. Camilleri, A. Malige, J. Fujimoto, and D. M. Rempel, “Touch displays: the effects of palm rejection technology on productivity, comfort, biomechanics and positioning,” *Ergonomics*, vol. 56, no. 12, pp. 1850–1862, 2013. PMID: 24134774.

[20] J. Wagner, S. Huot, and W. Mackay, “Bitouch and bipad: Designing bimanual interaction for hand-held tablets,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’12, (New York, NY, USA), pp. 2317–2326, ACM, 2012.

[21] K. Kim, S. A. Turner, and M. A. Pérez-Quiñones, “Requirements for electronic note taking systems: A field study of note taking in university classrooms,” *Education and Information Technologies*, vol. 14, pp. 255–283, Sept. 2009.

[22] A. Piolat, T. Olive, and R. T. Kellogg, “Cognitive effort during note taking,” *Applied Cognitive Psychology*, vol. 19, no. 3, pp. 291–312, 2005.

[23] A. Baddeley, “Exploring the central executive,” *THE QUARTERLY JOURNAL OF EXPERIMENTAL PSYCHOLOGY*, pp. 5–28, 1996.

[24] T. NISHIMURA, K. DOI, and H. FUJIMOTO, “Evaluation of size and spacing of buttons in consideration of contact angle of forefinger in tablet terminal with touch-sensitive screen,” *Transactions of Japan Society of Kansei Engineering*, vol. 14, no. 3, pp. 343–350, 2015.

[25] C. P. Thacker, E. M. McCreight, and B. W. Lampson, “Alto: A personal computer,” *CSL-79-11*, 1979.

[26] B. Shneiderman, “Direct manipulation: A step beyond programming languages (abstract only),” in *Proceedings of the Joint Conference on Easier and More Productive Use of Computer Systems. (Part - II): Human Interface and the User Interface - Volume 1981*, CHI ’81, (New York, NY, USA), pp. 143–, ACM, 1981.

[27] Apple, “ios human interface guidelines.” https://developer.apple.com/library/prerelease/content/documentation/UserExperience/Conceptual/OSXHIGuidelines/.

[28] Google, “Google material design.” https://material.google.com/patterns/gestures.html.

[29] Microsoft, "Universal windows platform touch interactions." https://msdn.microsoft.com/en-us/windows/uwp/input-and-devices/touch-interactions.

[30] C. Harrison, R. Xiao, J. Schwarz, and S. E. Hudson, "Touchtools: Leveraging familiarity and skill with physical tools to augment touch interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, (New York, NY, USA), pp. 2913–2916, ACM, 2014.

[31] F. Matulic and M. C. Norrie, "Supporting active reading on pen and touch-operated tabletops," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, (New York, NY, USA), pp. 612–619, ACM, 2012.

[32] V. Roth and T. Turner, "Bezel swipe: Conflict-free scrolling and multiple selection on mobile touch screen devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, (New York, NY, USA), pp. 1523–1526, ACM, 2009.

[33] P. Ewerling, A. Kulik, and B. Froehlich, "Finger and hand detection for multi-touch interfaces based on maximally stable extremal regions," in *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, (New York, NY, USA), pp. 173–182, ACM, 2012.

[34] N. Marquardt, J. Kiemer, D. Ledo, S. Boring, and S. Greenberg, "Designing user-, hand-, and handpart-aware tabletop interactions with the touchid toolkit," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, (New York, NY, USA), pp. 21–30, ACM, 2011.

[35] D. Schmidt, M. K. Chong, and H. Gellersen, "Handsdown: Hand-contour-based user identification for interactive surfaces," in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, (New York, NY, USA), pp. 432–441, ACM, 2010.

[36] D. Vogel and G. Casiez, "Hand occlusion on a multi-touch tabletop," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 2307–2316, ACM, 2012.

[37] O. K.-C. Au and C.-L. Tai, "Multitouch finger registration and its applications," in *Proceedings of the 22Nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction*, OZCHI '10, (New York, NY, USA), pp. 41–48, ACM, 2010.

[38] K. Hinckley, X. A. Chen, and H. Benko, "Motion and context sensing techniques for pen computing," in *Proceedings of Graphics Interface 2013*, GI '13, (Toronto, Ont., Canada, Canada), pp. 71–78, Canadian Information Processing Society, 2013.

[39] K. Hinckley, M. Pahud, H. Benko, P. Irani, F. Guimbretière, M. Gavriliu, X. A. Chen, F. Matulic, W. Buxton, and A. Wilson, "Sensing techniques for tablet+stylus interaction," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, (New York, NY, USA), pp. 605–614, ACM, 2014.

[40] K. Hinckley, K. Yatani, M. Pahud, N. Coddington, J. Rodenhouse, A. Wilson, H. Benko, and B. Buxton, "Pen + touch=new tools," in *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, (New York, NY, USA), pp. 27–36, ACM, 2010.

[41] Y. Li, K. Hinckley, Z. Guan, and J. A. Landay, "Experimental analysis of mode switching techniques in pen-based user interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, (New York, NY, USA), pp. 461–470, ACM, 2005.

[42] *Exploring and Understanding Unintended Touch During Direct Pen Interaction*, vol. 21, (New York, NY, USA), ACM, Nov. 2014.

[43] NoteAnytime, "Metamoji." http://product.meta moji.com/ja/anytime/.

[44] D. Vogel, M. Cudmore, G. Casiez, R. Balakrishnan, and L. Keliher, "Hand occlusion with tablet-sized direct pen input," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, (New York, NY, USA), pp. 557–566, ACM, 2009.

[45] J. Schwarz, R. Xiao, J. Mankoff, S. E. Hudson, and C. Harrison, "Probabilistic palm rejection using spatiotemporal touch features and iterative classification," in

Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems, CHI '14, (New York, NY, USA), pp. 2009–2012, ACM, 2014.

[46] M. Annett, F. Anderson, W. F. Bischof, and A. Gupta, "The pen is mightier: Understanding stylus behaviour while inking on tablets," in *Proceedings of Graphics Interface 2014*, GI '14, (Toronto, Ont., Canada, Canada), pp. 193–200, Canadian Information Processing Society, 2014.

[47] INKredible, "Writeon inkredible." http://inkredibleapp.com/.

[48] G. Kurtenbach, G. Fitzmaurice, T. Baudel, and B. Buxton, "The design of a gui paradigm based on tablets, two-hands, and transparency," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, (New York, NY, USA), pp. 35–42, ACM, 1997.

[49] H. Song, H. Benko, F. Guimbretiere, S. Izadi, X. Cao, and K. Hinckley, "Grips and gestures on a multi-touch pen," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, (New York, NY, USA), pp. 1323–1332, ACM, 2011.

[50] M. Sun, X. Cao, H. Song, S. Izadi, H. Benko, F. Guimbretiere, X. Ren, and K. Hinckley, "Enhancing naturalness of pen-and-tablet drawing through context sensing," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, (New York, NY, USA), pp. 83–86, ACM, 2011.

[51] J. I. Kiger, "The depth/breadth trade-off in the design of menu-driven user interfaces," *International Journal of Man-Machine Studies*, vol. 20, no. 2, pp. 201 – 213, 1984.

[52] T. Tsandilas and m. c. schraefel, "Bubbling menus: A selective mechanism for accessing hierarchical drop-down menus," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, (New York, NY, USA), pp. 1195–1204, ACM, 2007.

[53] A. Tomita, K. Kambara, and I. Siio, "Slant menu: Novel gui widget with ergonomic design," in *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, (New York, NY, USA), pp. 2051–2056, ACM, 2012.

[54] S. R. Parkinson, M. D. Hill, N. Sisson, and C. Viera, “Effects of breadth, depth and number of responses on computer menu search performance,” *International Journal of Man-Machine Studies*, vol. 28, no. 6, pp. 683 – 692, 1988.

[55] T. Hesselmann, S. Flöring, and M. Schmitt, “Stacked half-pie menus: Navigating nested menus on interactive tabletops,” in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS ’09, (New York, NY, USA), pp. 173–180, ACM, 2009.

[56] P. Isokoski and M. Käki, “Comparison of two touchpad-based methods for numeric entry,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’02, (New York, NY, USA), pp. 25–32, ACM, 2002.

[57] Y. Liu, X. Chen, L. Wang, H. Zhang, and S. LI, “Pinyinpie: A pie menu augmented soft keyboard for chinese pinyin input methods,” in *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI ’12, (New York, NY, USA), pp. 271–280, ACM, 2012.

[58] K. Samp and S. Decker, “Supporting menu design with radial layouts,” in *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI ’10, (New York, NY, USA), pp. 155–162, ACM, 2010.

[59] P. M. Fitts, “The information capacity of the human motor system in controlling the amplitude of movement.,” *Journal of Experimental Psychology*, vol. 47(6), pp. 381–391, 1954.

[60] G. A. Miller, “The magical number seven, plus or minus two: Some limits on our capacity for processing information.,” *Psycholog- ical Review*, pp. 81–97, 1956.

[61] G. Kurtenbach and W. Buxton, “The limits of expert performance using hierarchic marking menus,” in *Proceedings of the INTERACT ’93 and CHI ’93 Conference on Human Factors in Computing Systems*, CHI ’93, (New York, NY, USA), pp. 482–487, ACM, 1993.

[62] M. A. Tapia and G. Kurtenbach, “Some design refinements and principles on the appearance and behavior of marking menus,” in *Proceedings of the 8th Annual ACM*

Symposium on User Interface and Software Technology, UIST '95, (New York, NY, USA), pp. 189–195, ACM, 1995.

[63] G. Bailly, E. Lecolinet, and L. Nigay, "Wave menus: Improving the novice mode of hierarchical marking menus," in *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction*, INTERACT'07, (Berlin, Heidelberg), pp. 475–488, Springer-Verlag, 2007.

[64] J. Francone, G. Bailly, L. Nigay, and E. Lecolinet, "Wavelet menus: A stacking metaphor for adapting marking menus to mobile devices," in *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, (New York, NY, USA), pp. 49:1–49:4, ACM, 2009.

[65] F. Guimbretiére and T. Winograd, "Flowmenu: Combining command, text, and data entry," in *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, (New York, NY, USA), pp. 213–216, ACM, 2000.

[66] S. Zhao, M. Agrawala, and K. Hinckley, "Zone and polygon menus: Using relative position to increase the breadth of multi-stroke marking menus," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, (New York, NY, USA), pp. 1077–1086, ACM, 2006.

[67] J. Zheng, X. Bi, K. Li, Y. Li, and S. Zhai, "M3 gesture menu: Design and experimental analyses of marking menus for touchscreen mobile interaction," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, (New York, NY, USA), pp. 249:1–249:14, ACM, 2018.

[68] J. Francone, G. Bailly, E. Lecolinet, N. Mandran, and L. Nigay, "Wavelet menus on handheld devices: Stacking metaphor for novice mode and eyes-free selection for expert mode," in *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, (New York, NY, USA), pp. 173–180, ACM, 2010.

[69] G. Bailly, A. Demeure, E. Lecolinet, and L. Nigay, "Multitouch menu (mtm)," in *Proceedings of the 20th International Conference of the Association Francophone D'Interaction Homme-Machine*, IHM '08, (New York, NY, USA), pp. 165–168, ACM, 2008.

[70] D. Kammer, F. Lamack, and R. Groh, "Enhancing the expressiveness of fingers: Multi-touch ring menus for everyday applications," in *Proceedings of the First International Joint Conference on Ambient Intelligence*, AmI'10, (Berlin, Heidelberg), pp. 259–264, Springer-Verlag, 2010.

[71] S. Heo, J. Jung, and G. Lee, "Melodictap: Fingering hotkey for touch tablets," in *Proceedings of the 28th Australian Conference on Computer-Human Interaction*, OzCHI '16, (New York, NY, USA), pp. 396–400, ACM, 2016.

[72] Y. Luo and D. Vogel, "Pin-and-cross: A unimanual multitouch technique combining static touches with crossing selection," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software &#38; Technology*, UIST '15, (New York, NY, USA), pp. 323–332, ACM, 2015.

[73] A. Roudaut, E. Lecolinet, and Y. Guiard, "Microrolls: Expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, (New York, NY, USA), pp. 927–936, ACM, 2009.

[74] X. Ren, J. Yin, T. Oya, and Y. Liu, "Enhancing pie-menu selection with pen pressure," in *2008 3rd International Conference on Innovative Computing Information and Control*, pp. 364–364, June 2008.

[75] F. Tian, L. Xu, H. Wang, X. Zhang, Y. Liu, V. Setlur, and G. Dai, "Tilt menu: Using the 3d orientation information of pen devices to extend the selection capability of pen-based user interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, (New York, NY, USA), pp. 1371–1380, ACM, 2008.

[76] I. Aslan, I. Buchwald, P. Koytek, and E. André, "Pen + mid-air: An exploration of mid-air gestures to complement pen input on tablets," in *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, NordiCHI '16, (New York, NY, USA), pp. 1:1–1:10, ACM, 2016.

[77] S. Chu and J. Tanaka, "Design of a motion-based gestural menu-selection interface for a self-portrait camera," *Personal Ubiquitous Comput.*, vol. 19, pp. 415–424, Feb. 2015.

[78] Y. Suzuki, K. Misue, and J. Tanaka, “Stylus enhancement to enrich interaction with computers,” in *Proceedings of the 12th International Conference on Human-computer Interaction: Interaction Platforms and Techniques*, HCI’07, (Berlin, Heidelberg), pp. 133–142, Springer-Verlag, 2007.

[79] SAMSUNG, “S pen.” http://www.samsung.com/us/.

[80] Digitizer, “Wacom.” http://www.wacom.com/en-us.

[81] Microsoft, “Pro pen.” http://www.microsoft.com/surface/en-us.

[82] Paper, “Fiftythree.” https://www.fiftythree.com/paper.

[83] Pencil, “Fiftythree, inc..” http://www.fiftythree.com/pencil.

[84] D. Yoon, N. Chen, and F. Guimbretière, “Texttearing: Opening white space for digital ink annotation,” in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST ’13, (New York, NY, USA), pp. 107–112, ACM, 2013.

[85] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “Liblinear: A library for large linear classification,” *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, June 2008.

[86] K. Hinckley and K. Yatani, “Radial menus with bezel gestures,” Aug. 25 2011. US Patent App. 12/709,301.

[87] S. Zhao and R. Balakrishnan, “Simple vs. compound mark hierarchical marking menus,” in *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST ’04, (New York, NY, USA), pp. 33–42, ACM, 2004.

[88] M. Jain and R. Balakrishnan, “User learning and performance with bezel menus,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’12, (New York, NY, USA), pp. 2221–2230, ACM, 2012.

[89] A. Pencil, “Apple.” http://www.apple.com/apple-pencil/.

[90] T. Grossman, K. Hinckley, P. Baudisch, M. Agrawala, and R. Balakrishnan, “Hover widgets: Using the tracking state to extend the capabilities of pen-operated de-

vices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, (New York, NY, USA), pp. 861–870, ACM, 2006.

[91] Y. Obana, "Vertical or horizontal? reading directions in japanese," *Bulletin of the School of Oriental and African Studies, University of London*, vol. 60, no. 1, pp. 86–94, 1997.

[92] J. BORITZ, "Fitts' law studies of directional mouse movement," *Proc. graphics interface '91*, pp. 216–223, 1991.

ACHIEVEMENT

[1] A. Kitani, M. Shiraishi, and T. Nakatani, “A proposal of Arc Palette for a hand writing application used on a tablet device,” *Proc. of the 5th International Congress of International Association of Societies of Design Research,* pp.3489-3498, 2013.

[2] A. Kitani, and T. Nakatani, “A Suitable Design for Natural Menu Opening Manipulations When Note-Taking on Tablet Devices,” *HCI International 2016 Posters' Extended Abstracts: 18th International Conference, HCI International 2016, Toronto, Canada, July 17-22, 2016, Proceedings, Part I,* pp.358-363, 2016.

[3] A. Kitani, and T. Nakatani, “Menu Designs for Note-taking Applications,” *Japan Society for Software Science and Technology,* vol.34, no.3, pp. 3_148-3_160, 2017.

[4] A. Kitani, T. Kimura, and T. Nakatani, “Toward the reduction of incorrect drawn ink retrieval,” *Human-centric Computing and Information Sciences,* vol.7, no.1, pp.18, 2017.

[5] A. Kitani, and T. Nakatani, “Evaluations of menu manipulations with Palm Lift Manipulation,” *Human Interface Society,* vol.20, no.2, pp.229-242, 2018.