IMPROVING MULTI-TOUCH INTERACTIONS USING HANDS AS LANDMARKS

A Thesis Submitted to the College of
Graduate Studies and Research
In Partial Fulfillment of the Requirements
for the Degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon, Canada

by

Md. Sami Uddin

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

ABSTRACT

Efficient command selection is just as important for multi-touch devices as it is for traditional interfaces that follow the Windows-Icons-Menus-Pointers (WIMP) model, but rapid selection in touch interfaces can be difficult because these systems often lack the mechanisms that have been used for expert shortcuts in desktop systems (such as keyboards shortcuts). Although interaction techniques based on spatial memory can improve the situation by allowing fast revisitation from memory, the lack of landmarks often makes it hard to remember command locations in a large set. One potential landmark that could be used in touch interfaces, however, is people's hands and fingers: these provide an external reference frame that is well known and always present when interacting with a touch display. To explore the use of hands as landmarks for improving command selection, we designed hand-centric techniques called HandMark menus. We implemented HandMark menus for two platforms – one version that allows bimanual operation for digital tables and another that uses single-handed serial operation for handheld tablets; in addition, we developed variants for both platforms that support different numbers of commands. We tested the new techniques against standard selection methods including tabbed menus and popup toolbars. The results of the studies show that HandMark menus perform well (in several cases significantly faster than standard methods), and that they support the development of spatial memory. Overall, this thesis demonstrates that people's intimate knowledge of their hands can be the basis for fast interaction techniques that improve performance and usability of multi-touch systems.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Carl Gutwin for his guidance and input throughout my graduate career at the University of Saskatchewan. I also wish to thank Hasan Mahmud for introducing me to the world of HCI research, and Benjamin Lafreniere for his research assistance. A big thank you to the faculty and staff at the Department of Computer Science, and particularly to the students and staff in the HCI Lab for their help and their friendship.

This thesis is dedicated to my mother Shamim Sultana, my father Md. Nasir Uddin, and my brother Sayem Uddin.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
DH	Dominant Hand
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HM	HandMark
NDH	Non-Dominant Hand
OS	Operating System
UI	User Interface
WIMP	Windows, Icons, Menus and Pointers

CHAPTER 1

INTRODUCTION

Command selection is a ubiquitous yet important task for any graphical user interface. Many interfaces follow the traditional Windows, Icons, Menus and Pointers (WIMP) paradigm [111] for designing interactions; these interfaces let users manipulate commands or objects on the screen with the help of a pointing device such as a mouse. In WIMP interfaces there are a large number of on-screen widgets such as buttons, each typically mapped to a single system command. It is common practice to organize these commands into hierarchical tabbed toolbars or menus [103], but this hierarchical organization often reduces selection performance [11, 70] because every selection requires users to traverse through the hierarchy, even if they are very familiar with a command. Expert techniques such as keyboard shortcuts and hotkeys (e.g. [84, 91]) can improve selection performance significantly, although research shows that users often fail to adopt faster methods such as shortcuts [104].

On touch devices, fast command selection is as important as it is for traditional WIMP systems, but techniques that allow quick operation for experts are uncommon on touch interfaces. Traditional input tools such as keyboards and mice are not available in most touch surfaces; instead, most devices rely on direct touch input. As a result, command selection on multi-touch surfaces such as tablets and tabletops is often slow and dependent on menu navigation. In addition, selection techniques and widgets from desktop interfaces are often a poor match for the physical characteristics of a table – for example, menus or ribbons are typically placed at the edges of the screen, making them hard to reach on large displays, and hard to see on horizontal displays (due to the oblique angle to the user). To solve the distance problem, researchers have proposed several techniques that bring tools closer to the user's work area or allow users to invoke commands wherever they are: for example, moveable palettes and toolsheets controlled by the non-dominant hand [16, 63], gestural commands [72], finger-count menus [11], or multi-touch marking menus [79]. These techniques can work well, but are limited in the number of commands that they can show (e.g., finger-count menus are limited to 25 commands, marking menus to about 64 [70]).

One promising approach to providing a higher performance ceiling for experts involves the use of spatial location memory (e.g., [46, 47, 103]) – once users learn command locations in a spatially-stable interface, they can make selections based on memory rather than visual search. In these techniques, an important element in the transition from search-based novice operation to memory-based expert operation is the support provided for the development of spatial memory. Landmarks play a critical role in this development, because they provide an external reference frame for remembering locations. Landmarks provide anchors for people's spatial memory as they move towards full "survey knowledge" [83] of command locations – for example, people can remember items at the corners of a grid better than they remember items in the middle. Part of the difficulty in developing new high-capacity selection techniques for touch surfaces is that there are few landmarks that can help people learn the tool locations.

There is, however, a well-known landmark that is always present and visible to the user of a touch surface – their hands. People are intimately familiar with the size and shape of their hands, and proprioception allows people to easily locate features (e.g., touching your right index finger to the tip of your left thumb can be done without looking). People's intimate knowledge of their hands, however, is not exploited for command selection. This thesis carries out research to investigate how users' knowledge of hands and fingers can be used to design new interaction techniques for multi-touch surfaces that will aid in spatial memory development and improve selection performance.

1.1 PROBLEM

The problem addressed in this thesis is that it is difficult to provide both efficient command selection and a large number of commands in multi-touch interfaces.

Performance efficiency and support for large command sets could potentially be achieved in many different ways, but on touch devices such as tablets and tables, we constrain the investigation to the use of direct-touch input methods [7] and multiple-finger input [20]. However, a number of questions regarding design, usability and human performance must be answered before a hand-centric method can be deployed. For instance, menu organizations around the hands, the selection mechanism, and the number of possible commands are unknown. These and other related design questions must be answered.

Within these constraints, there are several possible directions. One possible solution uses a bimanual selection mechanism, in which widgets such as tool palettes [16, 63] are held by the non-dominant hand while command selection is performed with the dominant hand. This technique can solve the problem of tools being far away, but does not use the details of the hand as a reference frame. Another technique that does use detailed knowledge of the hands is finger-count menus [11], which select commands based on the pattern of fingers touching the surface (e.g., the fingers of the left hand select one of five menus, and the fingers of the right hand select one of five items in the menu). This allows the development of proprioceptive memory for command invocation, but does not make extensive use of people's familiarity with the size and shape of their hands. A main goal of this work is to leverage people's knowledge of their hands and investigate whether people can successfully use these features to improve command selection.

1.2 SOLUTION

The solution presented in this thesis is a new selection technique for multi-touch surfaces that uses the user's hands as landmarks to enable rapid command selection and support for a large number of commands.

The new interaction technique is called a HandMark menu. We created two different variants of the technique that accommodate different numbers of commands: first, HandMark-Finger places commands in the spaces around the spread-out fingers of the non-dominant hand; and second, HandMark-Multi accommodates larger command sets by placing a blocks of commands between the thumb and first finger, with different sets accessible through different finger combinations. We also created two versions of the techniques for different platforms – one is bimanual for large tabletops and another is single-handed for multi-touch tablets.

There were two main steps in the research:

Development of the HandMark technique

Designing an efficient multi-touch interaction technique is dependent on several other factors. The first step is understanding the ways that current interfaces work, and the ways that spatial memory can be used to make selections more efficient. The HandMark technique presented in Chapter Three was used to identify the scope of our main research; this approach was implemented on two different platforms and evaluated in controlled experiments.

Answering questions of usability and performance

In both of the implementations (for table and tablet), our target was to understand the usability and performance of HandMark menus, and their support for the development of spatial memory. We carried out four empirical studies in total with HandMark menus (Chapters Four and Five) to answer the following questions:

> How do HandMark menus perform compared to other techniques?
>
> Does the HandMark technique successfully support a large number of commands?
>
> Do HandMark menus support spatial learning?
>
> Is the HandMark technique easy to learn?
>
> How do HandMark menus perform in a bimanual selection process?
>
> How does the HandMark technique perform in a single-handed fashion?

1.3 EVALUATION

In order to provide evidence about performance and usability for the idea of hands as landmarks, we carried out four experiments with two different versions of HandMark menus. We tested two menu layouts (layout around fingers and layout in a block) and two variants with different hand requirements (bimanual for tables, and one-handed for tablets). Analyses of the results of these comparative experiments showed that HandMark menus were easy to use and easy to learn, and gave performance improvements over standard ways of command selection. In addition, the studies provided evidence that people were successfully able to use their hands as landmarks, and that this was one reason for the improved performance of the HandMark techniques.

The evaluation processes that we followed in our experiments are as follows:

- Usability and performance were evaluated using a series of target selection tasks. Both versions of the HandMark menus with their two variants were tested. Empirical results from the experiments were used to determine learning rates, selection time, error rates, and expertise development for each technique.

- In the comparative studies of the HandMark technique for tables, we compared HandMark menus against two equivalent standard tab menus. For tablet version (one-handed instead of bimanual), we compared HandMarks against a pop-up menu. In addition to these performance comparisons, we also explored the ways that hands worked as landmarks.

- In another study, the effect of memory overloading was tested. In HandMark-Multi menus, the same spatial location is used for multiple commands (from different sets); and we tested how performance changes when people learn different commands in the same spatial location.

- Subjective responses were also evaluated for each of the interfaces in the studies. Participants completed both the NASA-TLX effort questionnaire [49] and an overall preference questionnaire after using the techniques.

1.4 CONTRIBUTIONS

There are two primary contributions presented in this thesis. First, we show that hands can be used as powerful landmarks for touch interfaces, and provide empirical evidence that using hands as landmarks can improve selection performance, aid development of spatial memory, and accommodate large number of commands in multi-touch interfaces. Second, we demonstrate two new hand-centric interaction techniques – HandMark-Finger for small command sets and HandMark-Multi for larger sets – and describe their implementation for two different multi-touch device platforms.

Secondary contributions of this thesis are the set of design principles developed for HandMark techniques (that can be used to develop other spatial multi-touch interactions); empirical results about topics such as command overloading, error rates in memory-based interfaces, and reasons for participant preferences; and algorithms for detecting hand postures from a set of simple touch points.

1.5 THESIS OUTLINE

This thesis is organized into several chapters. Chapter Two presents a survey of related research and techniques for multi-touch interactions which form the foundation for the research presented here. First, the types of multi-touch interactions available are discussed. Second, we discuss a number of strategies that have been applied to improve command selection performance. Third, the use of landmarks in various interfaces for spatial memory development is discussed. Fourth, we discuss research and techniques that were intended for accommodating large number of commands. Finally, we describe and discuss currently available hand detection techniques that have been developed for vision-based surfaces.

In Chapter Three we set out the basic ideas of using hands as landmarks and introduce a technique for facilitating spatial memory development, rapid command selection, and accommodating large command set. Using this technique, we motivate our research into HandMark menus as an alternative solution to the problem of command reachability and selection in multi-touch interactions.

Chapter Four presents our work to investigate the usability and performance questions for hand-centric methods for large tables. We present two variants of the bimanual HandMark menu, and discuss design factors that can affect the performance of HandMark menus in tables. A comparative user study is carried out to answer important usability and performance questions. Both variants of HandMark menus are studied, and the results and their implications are discussed.

Chapter Five presents our work to investigate usability and performance questions for HandMark menus on touch tablets. To match the interaction paradigm of tablets, we create an alternate version of HandMark menus that can be operated with only one hand. Again, two variants of HandMark menus for tablets and their design factors are presented. Besides testing the usability and performance, we conduct an extra study to test the effect of hands as landmarks within the interface. Both variants of the single-handed HandMark menus are studied, and the results and their implications are discussed.

Chapter Six presents a discussion of the most important results from both Chapters Four and Five. Some higher-level implications of our findings and issues related to the overall work are addressed. Also, the lessons that have been learned over the course of this thesis are discussed.

Finally, Chapter Seven summarizes the research presented in this thesis. It discusses the main contributions of our work and highlights the avenues of future work revealed as a result of this thesis.

CHAPTER 2

RELATED WORK[1]

Our exploration into the use of hands as landmarks for command selection, and the design and development of the HandMarks technique, was influenced by five areas of previous related literature: multi-touch interaction, rapid command selection, landmarks, support for large command sets, and hand-detection techniques.

2.1 MULTI-TOUCH INTERACTIONS

Modern multi-touch devices such as hand-held tablets, smartphones, and digital tabletops support a wide variety of interactions. These devices primarily use direct touch-based interactions, but pen or stylus based interactions are also common. In the following sections we discuss several basic principles of multi-touch interaction.

---

[1] A portion of the material in this chapter first appeared in the following publications [120, 121]:

Uddin, M.S. and Gutwin, C. 2016. Rapid Command Selection on Multi-Touch Tablets with Single-Handed HandMark Menus. *Proceedings of the SIGCHI Conference on Interactive Surfaces and Spaces - ISS '16* (Niagara Falls, ON, Canada, 2016), in press.

Uddin, M.S., Gutwin, C. and Lafreniere, B. 2016. HandMark Menus: Rapid Command Selection and Large Command Sets on Multi-Touch Displays. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (New York, New York, USA, 2016), 5836–5848.

2.1.1 Direct-Touch Interaction

Direct-touch interaction is the most common way for users to interact with multi-touch devices, including hand-held mobile devices such as smartphones and tablets, and stationary systems such as large tabletops. Touch input is now very common, and users of these multi-touch devices are highly skilled in using multi-touch gestures to interact with them [7, 119]. Even though kinesthetic models suggest that humans are capable of using richer and more expressive forms of interaction with multiple fingers [20], however, most of the multi-touch gestures available in current devices are typically limited to a small subset of one and two-finger gestures: tap, press, double tap, drag/flick, pinch/spread, bezel swipe, and rotate gestures [7, 126]. Figure 1 shows some of the many multi-touch gestures that are available in Microsoft's Windows 8 operating system [118]. Almost all touch-enabled mobile devices also support a similar range of gestures: for example, the Apple iPad and Microsoft's Surface Pro are two examples of commercially successful products that use limited set of multi-touch gestures [7].

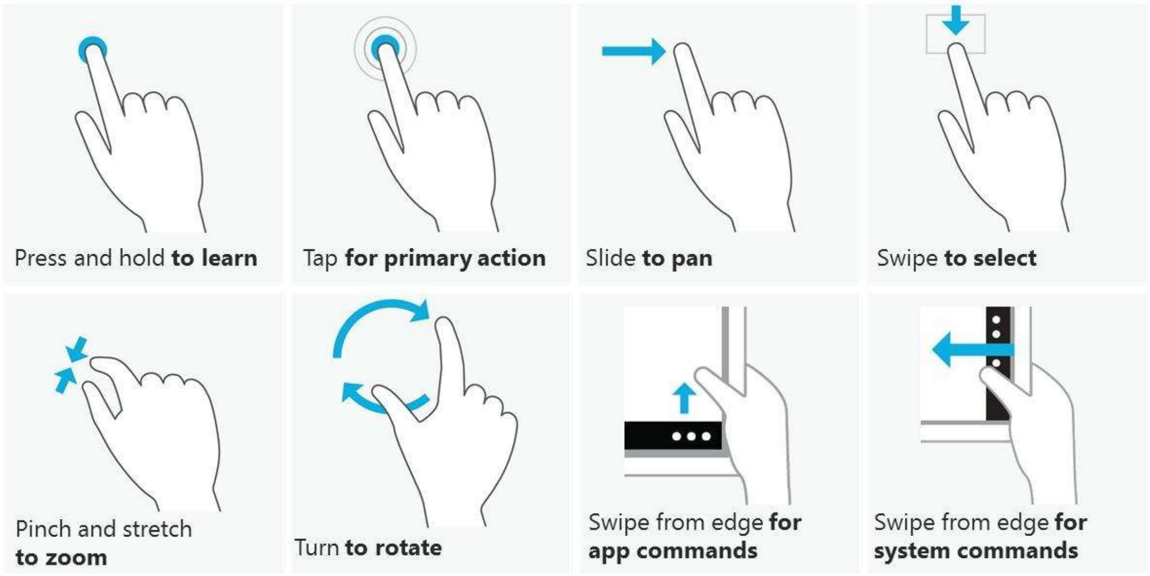


Figure 1: Some multi-touch gestures available in Windows 8 [118].

Several research projects have attempted to improve the multi-touch functionality of different touch-based systems [6, 20, 85, 92, 124]. Gutwin et al.'s FastTap menu [47] uses spatial memory to facilitate faster command selections for tablets (Figure 2), and a study showed that FastTap performed better than the widely-studied marking menu [74]. Gesture based systems, such as marking menus [70, 73], are one alternative to the hierarchical linear menus which are available in commercial products (e.g., Autodesk Sketchbook Pro). There are also variants of marking menu, including multi-touch marking menus [79] and Flower menus [10] which also use gestures to make the interaction more efficient. Ng and colleagues [89] designed techniques for low-latency direct touch inputs that provide low fidelity visual feedback immediately, which is followed by high fidelity visuals at standard latency. However, in all cases these interfaces require only one or two-finger based gestures to interact with the device.

Direct touch interactions are also the main input mechanism in large scale displays such as tables or wall displays. For example, SMARTBoards use DViT [114] technique to augment direct-touch sensing from the surface. DiamondTouch [33] is a multi-user direct multi-touch system for tabletops that provides direct touch but also identifies users within the system. Rekimoto et al.'s SmartSkin [99] is another type of direct multi-touch capacitive sensing technique for large tables. Another technique called finger count menus [11] counts the number of direct touch points from both hands to perform rapid command selection on large tables. These techniques are convenient to use in large tables, but usually support only a small number of commands.

2.1.2 Device Back or Side Interaction

Although modern touch devices support numerous simultaneous touch points [3, 60], users typically cannot use all the fingers from both hands, as the non-dominant hand is required to hold and orient the device [52]. A few projects have attempted to improve tablet interactions by introducing limited touch interaction for the holding hand – for example, in Wagner et al.'s BiTouch [124] users can interact with the thumb or fingers of the supporting hand along with a finger from the dominant hand. Other research has shown how the back of the device can be used for meaningful touch interactions [110, 127, 128].

There are other techniques which utilize device features other than the touchscreen, such as position sensors or the device bezel. For example, Baglioni et al.'s Jerktilts [9] allow quick selection of limited number of commands by tilting the mobile device quickly. Bezel Tap [109] detects taps on the edge of the device using the accelerometer sensor, allowing shortcuts to commands while the device is asleep. Hidden toolbars [108] uses swipe gestures across the bezels to facilitate fast command selection in tablets.

2.1.3 Pen or Stylus Interaction

Interaction with pen or stylus is also available in touch-based devices [54], and it is also popular among many users of hand-held portable devices such as tablets and digital tables [87]. Several projects use a combination of pen and touch for advanced interaction [52]. For example, Kitani et al.'s Palm lift manipulation leverages the unintended touches while notes are being taken to open a context menu naturally [67]. Hinckley et al.'s work investigated direct pen+touch inputs in touch sensitive surfaces [55]; they propose several mode-based pen+touch interactions for tablets.

Other research focuses on understanding and improving the usability of pen or stylus based interactions [4, 5, 48]. Grossman et al.'s Hover Widgets [42] increase the capabilities of pen-based interfaces by using the pen movements above the display surface (i.e., in the tracking state). A project by Saund et al. proposed an inferred-mode interaction protocol that avoids the mode issues seen in sketch or notetaking systems. Their technique tries to infer the user's intent from the properties of the pen trajectory and the context of the trajectory [102]. Although pen or stylus based interactions are popular, they cover only a limited range of interactions and sometimes even suffer from occlusion problems. Early research showed that while interacting with small scale touch devices (e.g., tablets) the interaction tools such as pen, hand, and forearm can occlude approximately half of the screen [122, 123], which often makes the interactions less effective.

2.2 RAPID COMMAND SELECTION

Rapid command selection in any kind of interactive menu is a primary goal in HCI. There are several ways to accelerate the command selection process, such as the use of memory-based invocation, using proprioceptive memory, and reducing invocation steps.

2.2.1 Memory-Based Command Selection

Command selection based on memory can substantially reduce the required time to select a command. Two types of memory can play crucial role in rapid command selection: spatial memory and proprioceptive memory.

Spatial memory

Spatial memory is a special kind of memory which records information about a person's surrounding environment and about the locations of objects in that environment [115]. Interaction with the surroundings and the objects helps people to develop spatial memory – for example, we can go a previously visited place without looking at the map, because our spatial memory helps us to navigate through the route.

In graphical user interfaces, spatial memory can be particularly valuable for making the interaction efficient. As users develop spatial memory through continued use of an interface, users become more efficient at locating particular commands. A novice generally relies on slow visual search to find items because of their poor spatial knowledge, but an expert user can quickly retrieve item locations from memory [27]. It is worth noting that visual search is also the first step in learning item locations – but visual search, in most cases, takes more time in finding an item's location. Experimental results by Engel [35] and Krendel et al. [68], and suggested that visual search can be either random, sequential [78] or linear [90]. People examine locations until the target is found, often revisiting already-visited items in the process which makes the visual search more time-consuming. Once stable spatial locations are learned through practice, visual search becomes unnecessary [105].

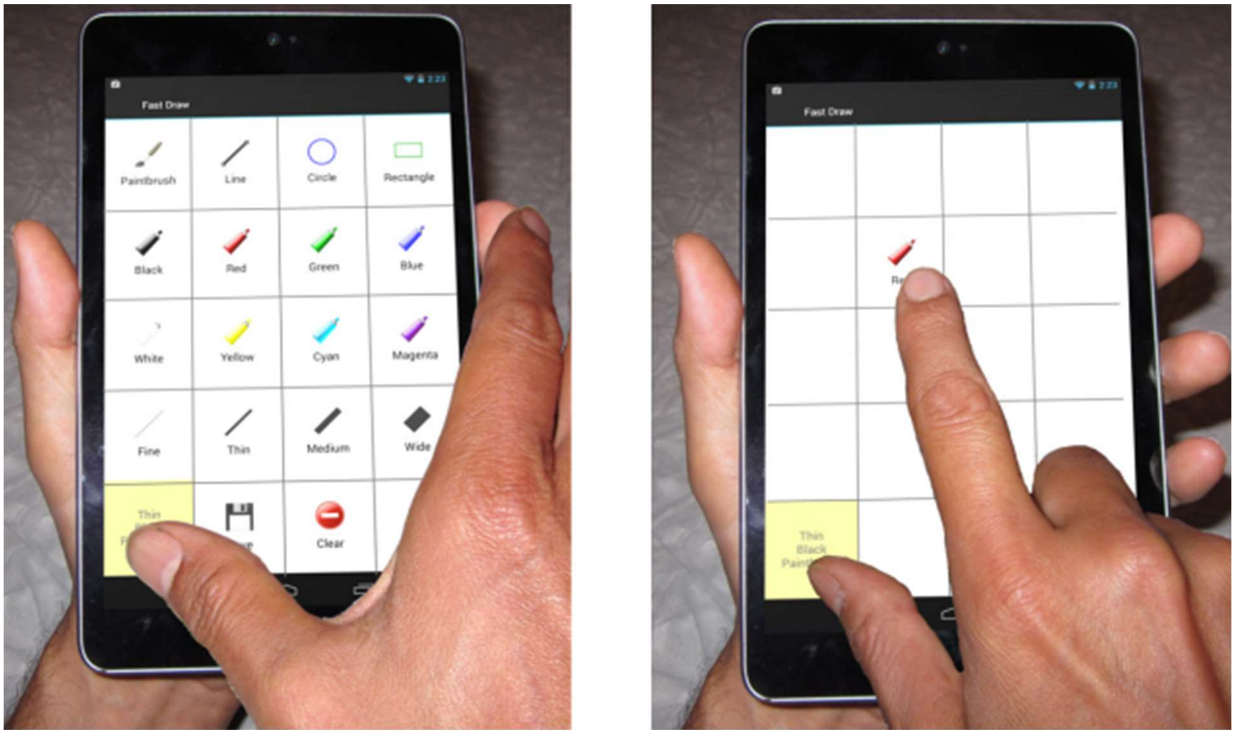


Figure 2: FastTap selection: (left) visual search by novice, (right) rapid selection from memory by experts without waiting for commands to appear [47].

Memory-based command selection is generally faster than visually-guided navigation once commands are learned [28, 45]. Many researchers have used memory-based techniques such as gestures [7, 73, 82], hotkeys [84], spatial locations [26, 47, 103], or multi-touch chords [40] in order to facilitate fast command selection. For example, studies have shown that spatial memory is built up through interactions with a stable visual representation, and as people gain experience with a particular location, they can easily retrieve that location from memory [47]. Gutwin et al. [47] showed the spatial memory is so powerful that experienced users can select a command from memory without waiting for commands to appear on the screen (see Figure 2).

The Hick-Hyman Law of choice reaction time [50, 58] can model these rapid command selections performed by leveraging the spatial memory [27]. According to the Law, remembering a mapping between locations and commands is directly related with total number of commands present within an interface. The retrieval time for any command from memory increases as a logarithmic function of the total number of commands in the mapping. When number of command increases, users have to remember lots of mappings. As a result, revisiting a specific command from memory requires extra time to take decision from the available mapping options.

Proprioceptive memory

According to Encyclopedia Britannica[2] "'proprioception' is the perception by an animal of stimuli relating to its own position, posture, equilibrium, or internal condition" [98]. From the human perspective, the understanding of the surroundings and their proximal distances with relation to adjacent body parts and the memory associated with it is called proprioceptive memory. For example, we have intimate knowledge of our own hands and fingers. Our proprioceptive memory is powerful enough that we can touch our index fingers together without looking at our hands.

Proprioceptive memory also can provide opportunities for richer interactions with multi-touch surfaces. Generally, GUIs are primarily portrayed visually, but for expert users, consistent proprioceptive memory-based actions can allow rapid command retrieval. There are several research projects which have tried to exploit the users' proprioceptive knowledge for effective interactions. For example, multi-touch marking menus [79] and finger-count menus [11] allow users to associate menu categories and items with specific combinations of fingers (e.g., as shown in Figure 3, touching with three fingers on the left hand selects the third menu, and pressing with four fingers on the right hand selects the fourth item in that menu); with practice, menu invocation and command selection make use of proprioceptive memory.

---

[2] Encyclopedia Britannica: www.britannica.com

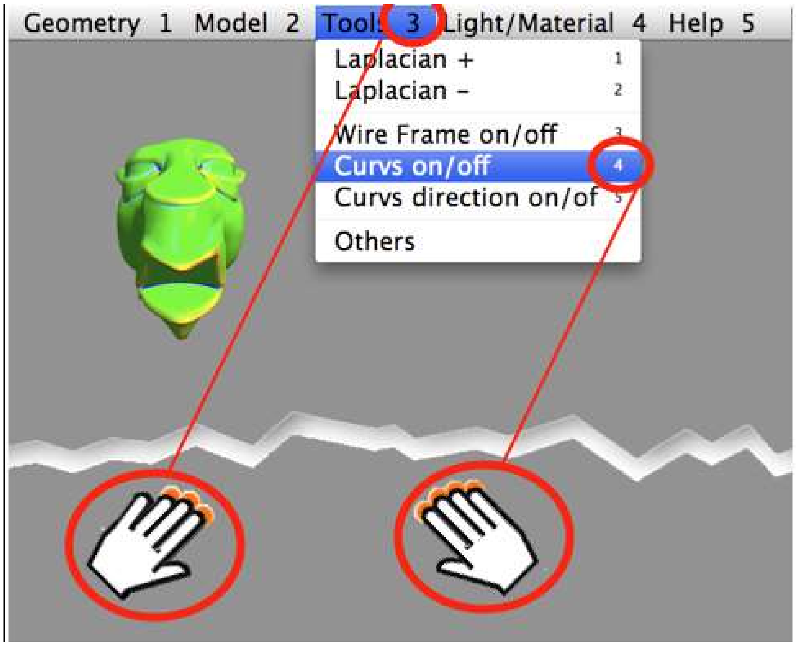


Figure 3: Finger count menus [41]. The number of fingers touched with the left hand determines the menu, and the number of fingers touched with the right hand determines the item.

However, since a more-complex control action (often requiring both hands) may take more time to recall and execute, these methods do not always improve performance [47, 66] and may require too much space for mobile tablet use.

Other work explored spatial memory in virtual spaces. For example, Li et al. [81] investigated users' proprioceptive and spatial ability to reproduce angular directions for 3D interaction where different commands were mapped to a spherical co-ordinate space placed before the user's body. Cockburn and his colleagues [28] also focused on the use of proprioception for better interaction. They investigated 'air pointing' systems in three settings: angular directions, 2D positional locations, and 3D positional locations. The items were arranged in the space around the user, and were selected by aiming the pointer at them virtually.

Other memory-based techniques

In addition to spatial and proprioceptive memory-based techniques, there are some other techniques which also use other kind of memory. For example, marking menus [70] use procedural memory, which the memory of a path. As the name suggests procedural memory is a long term memory of performing a task, which is learned implicitly [97]. Techniques such as Marking menus and their different variants [66, 70, 79] leverage the rehearsal based human motor behavior to speed up the command selection process (see Figure 4). With different variants such as multi-touch marking menu [79], two-handed marking menus [66] researchers have tried to minimize the time required for selections. These techniques place commands in radial shape pies [22] around the touch point and arrange them in different levels, each of which can accommodate eight commands and in total they can support approximately 64 commands.

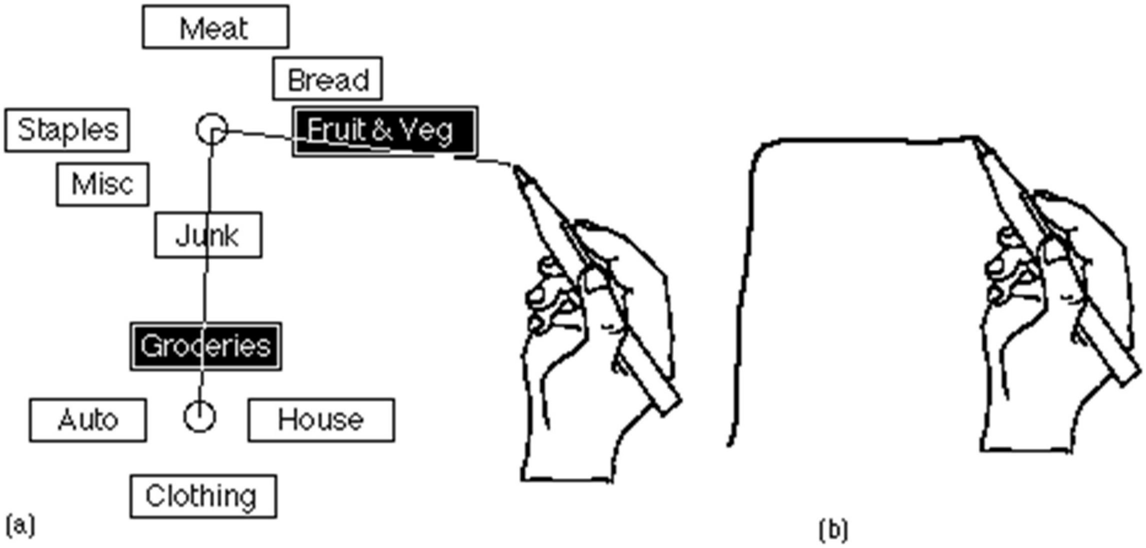
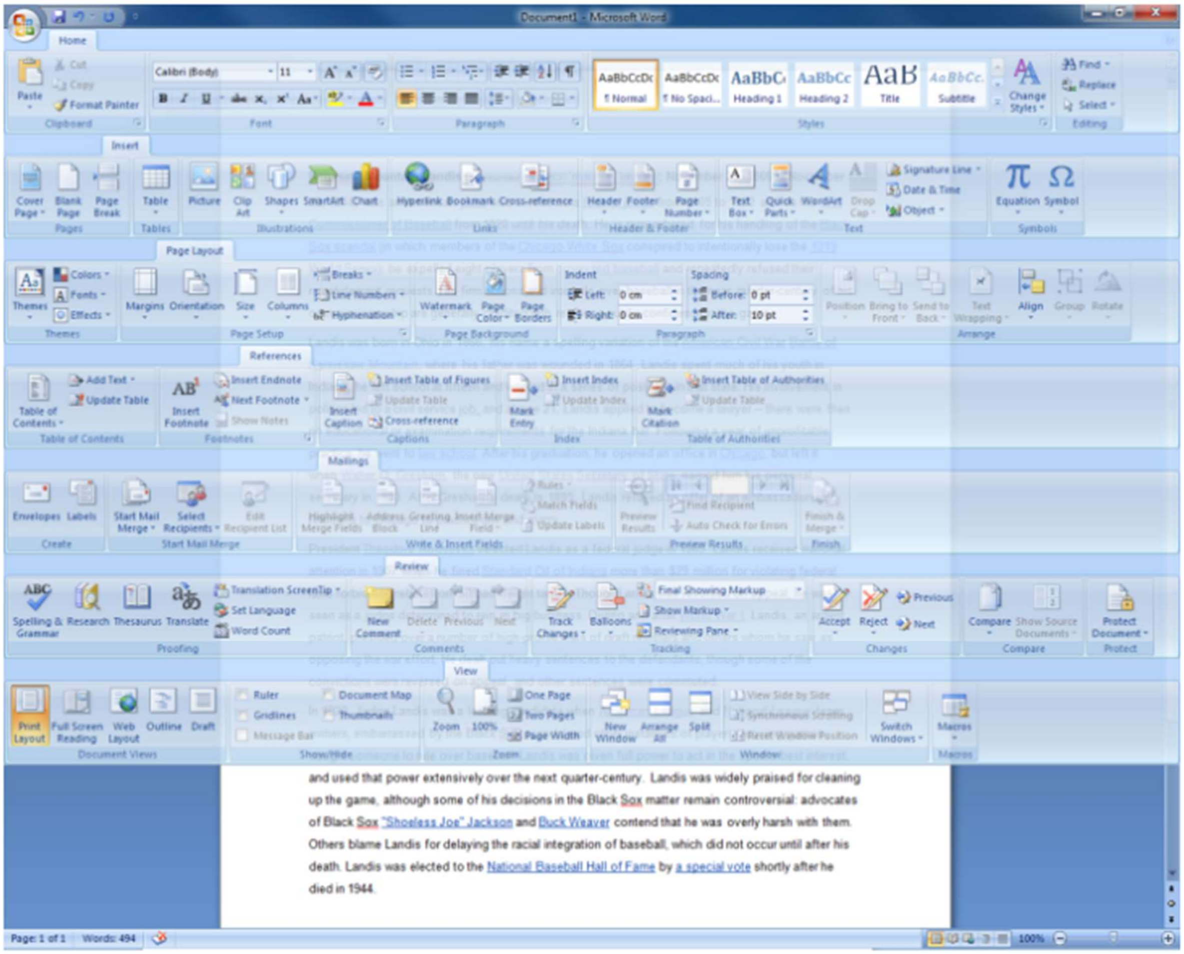


Figure 4: Selections in Marking menu: (a) novice – visual search and (b) expert gesture from memory [69].

2.2.2 Reducing the Number of Steps in a Selection

The efficiency of a command selection interface depends in part on the number of separate actions needed to find and execute a command (as well as the time required for individual actions). Reducing the total number of steps required for command invocation and execution can be an efficient way to speed up command selection. There are several techniques that attempt to minimize the number of steps – usually in contrast to hierarchical organizations of commands in toolbars, tabs, or standard menus. For example, Scarr et al.'s CommandMaps [103] used a full-screen overlay to display all commands at once, and successfully reduced the number of actions compared to hierarchical menus for desktop systems (see Figure 5). A similar approach was also used by a technique called Hotbox [71], which shows all the available menus around mouse pointer (dividing the areas into five zones) after pressing a command button. Later, specific zone's menu can be invoked simply by performing a gesture [70] toward that zone using the mouse.



and used that power extensively over the next quarter-century. Landis was widely praised for cleaning up the game, although some of his decisions in the Black Sox matter remain controversial: advocates of Black Sox "Shoeless Joe" Jackson and Buck Weaver contend that he was overly harsh with them. Others blame Landis for delaying the racial integration of baseball, which did not occur until after his death. Landis was elected to the National Baseball Hall of Fame by a special vote shortly after he died in 1944.

Figure 5: Example of CommandMaps for Microsoft Word [103] The picture shows the full-screen overlay with all seven toolbars on the screen at once. The overlay is shown when the Command key is pressed.

Gutwin et al.'s FastTap [47] uses chorded thumb and finger touches on a spatially stable grid interface to provide accelerated command selection for expert users in tablets (see Figure 2). Another project [76] implemented a FastTap-like menu system on a smartwatch, again with the intention of reducing the number of steps to make expert selections. In another work Cockburn et al. [26] overviewed all the pages of a document as thumbnails to eliminate scrolling. Although early research [57] suggest that shortcuts are often poorly adopted in real-world circumstances, Malacria et al.'s work [84] promoted the used of hotkeys throughout the interface with ExposeHK and reduced the number of steps for expert selections.

However, some of these techniques (such as full-screen overviews) are difficult to use on large touch tables because the user can be at any location and any orientation, making it difficult to accurately position a visual representation so that it is visible and reachable. At the same time, some of these advanced techniques are also unsuitable for hand-held multi-touch tablets – because of the physical characteristics of tablets, such as small screen size, or because of the need to hold the tablet with one hand, or the "fat finger" problem [112]. If many objects are placed in a small touch screen, interaction with a specific object accurately using a finger will become problematic.

2.2.3 Bimanual Techniques

Bimanual techniques (that is, techniques that use both hands) can be useful for designing more-powerful interactive systems. Numerous research projects have investigated the advantages of two-handed interactions [12, 19, 21, 23, 36, 63, 99]. While designing applications, multi-touch system designers can easily increase the input vocabulary for users and add more natural interaction techniques by leveraging input from two hands. Various input devices have been explored to enable bimanual input into computer interfaces: for example, styluses, toolglasses, and trackballs have been used along with the traditional input devices such keyboards and mice. Comparisons among these input devices [38, 51, 64, 88, 91] indicate that some perform well under certain conditions and perform poorly in others [17]. There are several research projects and studies [53, 64, 77, 93] investigating the use of techniques that use two hands, and as suggested by Brandl et al. [17] these can be divided into two categories: first, "models and frameworks", and second, "applications and input devices".

Models and frameworks

Guiard's Kinematic Chain model [43] proposes the general framework for asymmetric labor division in bimanual tasks performed by a human, and has influenced much work in bimanual interaction. The model states that when two or more motor units are assembled in series, they form a kinematic chain. This model can describe many bimanual tasks, such as working on a tablet involves two hands – one to hold the tablet, and the other to manipulate the screen (see Figure 6). The relationship between the two hands in a work is summarized by Brandl et al.: "During two-handed interaction, both hands have different roles that depend on each other with respect to three rules: the dominant hand (DH) moves within the frame of reference defined by the non-dominant hand (NDH); the sequence of motion generally sees the NDH setting the reference frame prior to actions with the DH being made within that context; and that the DH works at a higher level of precision than the NDH in both spatial and temporal terms" ([17], p. 154). As shown in Figure 6, while working with a multi-touch phone, the NDH holds the device and creates the reference frame for the DH to interact with the device.

Figure 6: Bimanual task – holding the phone with the non-dominant hand and touching the screen with the dominant hand [32].

Applications and Input Devices

The feasibility of different bimanual tasks has been evaluated for various input methods. Rekimoto created a multi-touch interaction technique called SmartSkin [99] that can sense multiple positions of multiple hands. A prototype created for digital tables supported bimanual interaction for simple object manipulation tasks (e.g., zooming). A technique called Holowall [86] supports bimanual interactions along with single hand and whole body interactions for wall sized displays. Dietz et al. created a touch sensitive input technique (DiamondTouch [33]) that enabled bimanual interaction in table-sized displays; this technique was later successfully incorporated in other bimanual interactive projects [19, 25, 36, 38, 130].

Several research projects have been carried out by researchers in HCI to investigate the performance of different bimanual interaction techniques against standard input devices. In large tabletop interfaces, Forlines et al. [38] compared a two-handed mouse to direct touch input. Kabbash et al. [64] compared performances among different input devices (e.g., stylus, trackball and mouse) in a bimanual setting. On large tables, Brandl et al. [17] used both pen (e.g., stylus) and direct touch simultaneously for more precise bimanual interactions.

Bimanual interaction has also been studied in touch interfaces. In a comparative study, the performance of two-handed interaction was investigated with a TouchMouse [51] in one hand and a traditional input device in another hand. The Marking Menu technique [70] – a widely investigated memory dependent gesture technique – has also been implemented for two handed operation [66]. Odell et al. [91] compared marking menus, toolglasses and hotkeys in both one-handed and two-handed fashion. The results from these experiments indicate that bimanual interactions can improve overall performance.

Bailly et al.'s finger count menu [11] is also a good example of the bimanual interaction for fast command selection in touch tables. As shown in Figure 7, with this technique one of five menus can be invoked using a corresponding number of fingers from the NDH and a command from that menu can be selected by touching down a specific number of fingers from the DH.

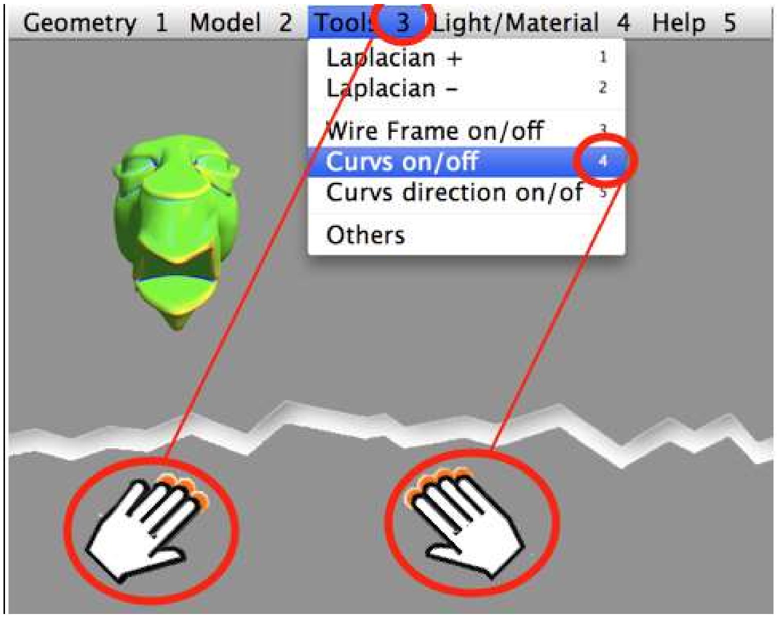


Figure 7: Finger count menu – bimanual interaction [41].

Another bimanual technique uses Palettes and Toolglasses [63], allowing users to position a menu of tools with the non-dominant hand, and make selections with a pointing device in the dominant hand. This division allows one hand to act in a supporting role to the other (following Guiard's Kinematic Chain model [43]). Although bimanual techniques such as Toolglasses can improve performance compared to traditional selection widgets [16], they only allow users to build up a coarse understanding of the locations of specific commands in relation to the hand, and only when used with an absolute input space. The intent of our HandMark menus is to go beyond the design of other multi-hand selection techniques, and use the hands as a more detailed absolute reference frame for developing memory of specific item locations.

2.3 USE OF LANDMARKS

In the real world, landmarks are known to assist people in remembering locations [83]. Landmarks are obvious visual features in an environment that do not move, and that can provide a reference frame for people to learn other locations. For example, in a city environment, a prominent building or park can act as a landmark, allowing people to remember new locations in relation to the known landmark. In the campus of University of Saskatchewan, the Thorvaldson building might act as such a landmark, allowing people to remember where other buildings are in relation to the Thorvaldson building. Landmarks help to define object locations in relation to the positions of other objects. As a result, landmarks are extremely important when it comes to encoding and retrieving location knowledge. In graphical interfaces, landmarks provide a potential method for users to mentally consolidate the data they can see into an overall spatial understanding.

In GUI-based systems, landmarks help users build up spatial memory of different command locations by providing a strong external reference frame that can anchor later retrieval. For example, Gutwin et al.'s FastTap [47] (see Figure 2) and Schramm et al.'s Hidden Toolbars [108] (see Figure 8) use the edges and corners of hand-held devices (e.g., tablets) as landmarks to organize a grid menu and toolbar. Figure 8 shows an example of the Hidden Toolbar technique, where different color markers are presented in a thin line along the four edges of a tablet; each color represents a particular icon or command. These color markings act as landmarks and help users to perform rapid selection by swiping outward from those marked locations.

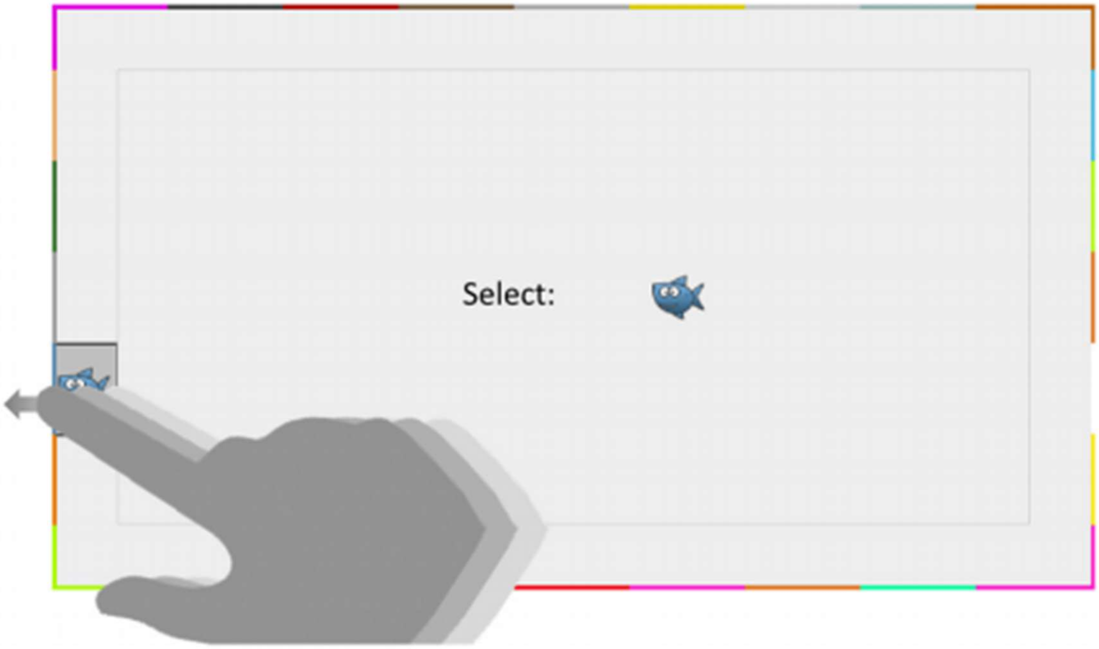


Figure 8: Hidden toolbar interface – using colored border regions as landmarks to help users remember locations [108].

These landmark based techniques have been successfully implemented for smaller scale devices, such as smartwatches. Lafreniere et al.'s WristTap and TwoTap techniques [76] use a FastTap-like grid menu, and leverage the natural landmarks of the corners and edges of the smartwatch to assist rapid command selection. However, corners and edges may not be suitable for large displays (e.g., tables) as they provide only a few landmarks compared to the size of the command set.

For large displays (e.g., desktops), a technique called CommandMap [103] accelerates command selection by presenting all the commands in flat full-screen overlay representation instead of the traditional hierarchical menu structure. The arrangement of the items in the grid provides a set of visual landmarks (e.g., a user might remember that a particular icon is next to a predominantly green item). Similarly, Cockburn et al.'s [26] space-filling thumbnails improved document navigation with an overview display of thumbnails. In this system, the grid of thumbnails also provided a set of landmarks (i.e., noticeable or visually-different pages in the grid).

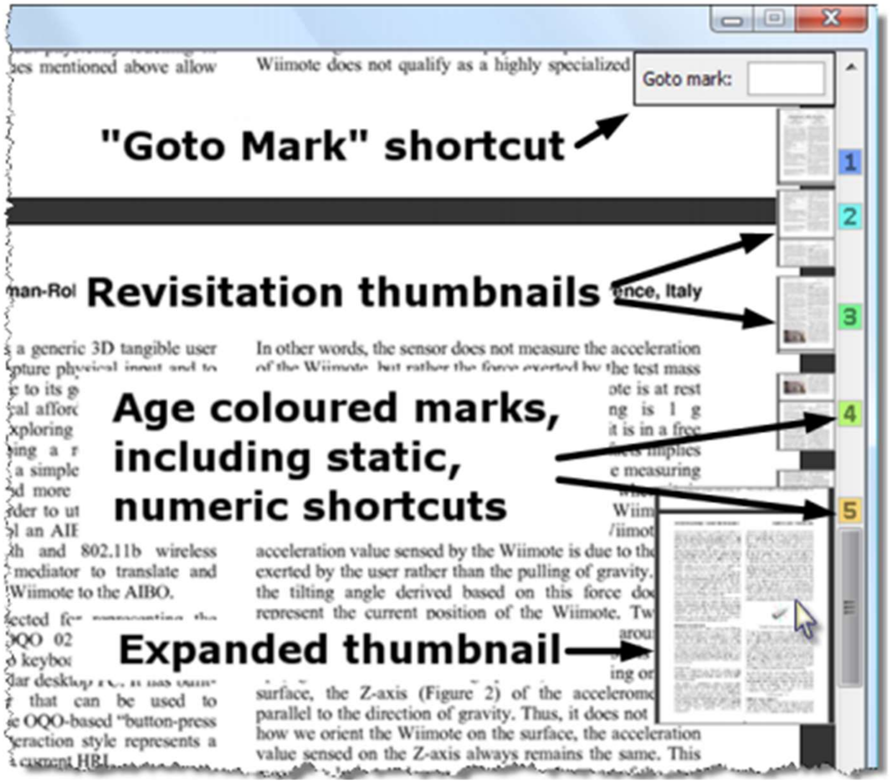


Figure 9: Footprint Scrollbar – color markers to enable revisitation [1].

In circumstances where natural landmarks are not available, dynamically generating artificial landmarks based on user behavior can be useful for supporting revisitation [105]. Alexander et al. [1] applied this idea to within-document navigation with their Footprints Scrollbar (see Figure 9); this system places colored marks in the scrollbar at recently visited document locations. The results of an evaluation showed these visual cues can act as landmarks which significantly decrease navigation time for revisited locations. Other techniques which also have successfully applied artificial landmarks include Skopik and Gutwin's "visit-wear" fisheye view [113], City Lights [132], Halo [14], Wedge [44], and the Canyon visualization [59]. However, these techniques often become less efficient with larger command sets, where many commands are not near any landmark (e.g., the middle regions of a grid).

In addition, use of landmarks within the interface is seen in many video summarization systems. In these systems the creation of visual markers such as storyboards can provide landmarks that indicate scene changes to users. For example, using captions, scripts and plot summaries as reference points for different video locations, SceneSkim [95] provides browsing and skimming facilities through video. Another method called Video Digests [96] uses navigable markers to represent sections of a long video, which enables users to efficiently browse through video content. Other tools support exploration and revisitation of different video locations by showing visual highlights on the navigable timelines that represent personal [2] or crowd [65, 131] navigation history.

2.4 SUPPORTING LARGE NUMBERS OF COMMANDS

In WIMP based GUIs, each command is typically assigned to a widget; users activate the command by clicking on the widget with a mouse. Many commercial software products provide a large number of commands: Microsoft's Office suite or Adobe's Photoshop or Maya tools have hundreds of commands in their interface, often arranged in hierarchical menu structures [103]. For example, Microsoft Office places commands in different "ribbon" toolbars which are organized into multiple tabs – but the large number of commands in different tabs and ribbons make it hard for users to remember and access the commands [103].

Use of memory-based techniques is one way to provide faster selection [47, 70, 103]. However, existing memory-dependent techniques available for touch devices support only a limited number of commands. For example, Finger Count menus [11] support only 25 commands (5 menus selected with the NDH, and five items in each that can be selected with the DH). Similarly, the FastTap menu can only accommodate 19 commands in a grid [47].

Procedural memory-based marking menu with its different variants [66, 70] which include multi-touch [79] and two-handed marking menu [66] improved selection performance, however they can accommodate approximately 64 items. Zaho et al.'s Polygon menus [133] successfully increased the per level command accommodation capacity of marking menus from 8 to 10. Other memory based gesture techniques such as Flower menus [10], Arpège [40], Augmented letters [101] and Octopocus [13] try to increase the capacity of command vocabulary by extending gesture range, or allowing chained hierarchies of gestures.

Memory-based selection techniques can allow a large command set while maintaining fast menu access. There are a few examples of high-capacity techniques: for example, Gutwin et al.'s ListMap [46] places 225 font items into a grid of buttons, and a similar spatial organization is followed in Scarr et al.'s CommandMaps [103] technique (which placed 210 items in a 2D grid). On the other hand, Kurtenbach et al.'s [71] Hotbox follows a different approach, supporting large command sets by grouping the menu items into different Marking-Menu zones. Although studies showed that only a small subset of available commands (e.g., less than 6% [8, 75]) are frequently used by most users, a problem for these large-capacity memory-depended techniques, however, is remembering command locations accurately. In these memory based techniques, when the command set size increases, the differentiability between gestures or grid locations becomes smaller, which may make it hard for the users to remember the commands. Also, most of these techniques are intended for desktop interfaces, which often makes them unfit for various touch devices.

2.5 HAND DETECTION

As discussed in earlier sections (2.1 and 2.2), hands are the primary tools for interacting with multi-touch devices; in order to design effective hand-centric interaction techniques, we need to identify hands and touches accurately. Associated issues in this task are that people have two hands, and are capable of using multiple fingers often at the same time [20, 43]; in addition, each finger differs from the others in shape and size, and while touching, fingers could be placed in different orientations. Hence, it is important to differentiate between left and right hand along with the shape and orientation of the hand once it has touched the surface.

Earlier work has explored hand detection using several methods: computer vision approaches, specialized hardware, and glove-based tracking. Several systems use computer vision to track the position of hands and to identify fingers [6, 37, 80]. In these system, researchers used cameras (which were placed under large touch screens) to detect hands, touch points, and sometimes even the arm associated with hands. Often, these techniques are robust enough that they can ignore palm contacts while touching (see Figure 10). Also, as they can detect the arm associated with touch points, they can even identify multiple persons interacting with system. However, these vision-based techniques are not suitable in mobile touch devices such as tablets.

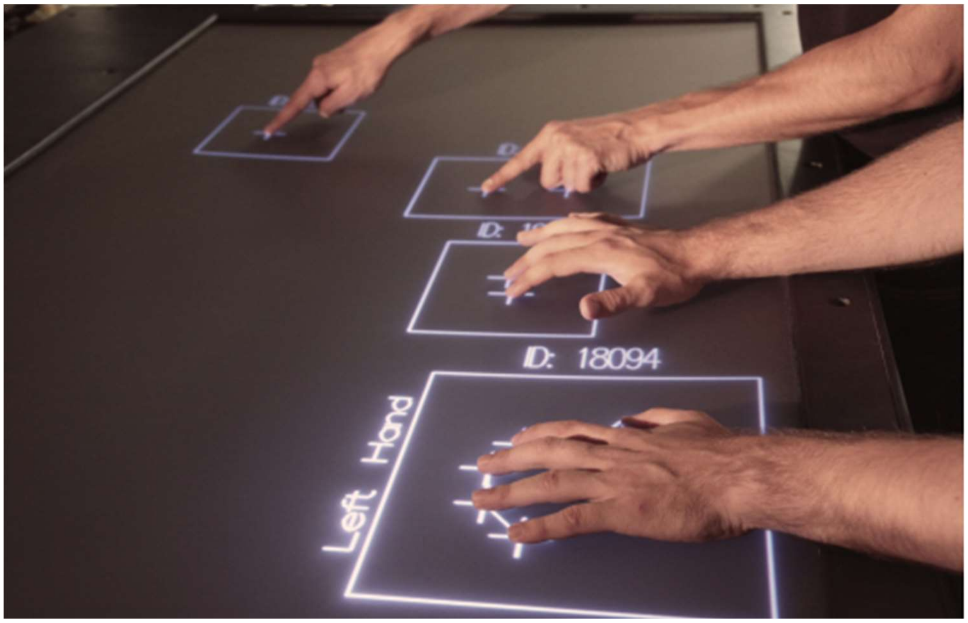


Figure 10: Vision based hand detection [37].

Another technique uses distance, orientation, and movement information of touch blobs to identify fingers and people [31, 125]. Schmidt et al.'s HandsDown [106] system allows hand detection on tabletops, and provides lenses for interaction [107]. SMARTBoards are quite common today that can also sense touches both from pen and hands – the DViT technique in SMARTBoards [114] uses computer vision-based techniques to sense touches, but it cannot identify specific hands. The reliability and accuracy of vision-based recognition, however, remains a challenge for all of these systems.

Other methods use specialized hardware to distinguish between hand parts and between users. For example, the DiamondTouch system [33] uses a capacitive coupling technique along with an overhead projector to identify different users. It can support multiple points of input from up to four users. SmartSkin [99] is another technology that can also enable touch sensing on large tabletops. Unlike the DiamondTouch technique, this technique cannot differentiate among multiple users. It offers, however, a more accurate detection of multiple input points and shapes. In addition, it describes several interaction techniques using hand shape and multi-finger input.

There are other specialized hardware-dependent approaches that can distinguish hands and body parts: for example, an EMG muscle-sensing armband that identifies a person's fingers touching a surface [15], and fingerprint recognition to provide precise touch information and user identification [56]. Other techniques distinguish a user's hands and their posture in space by using colored patches or markers on gloves [18, 129].

Most of these techniques are unsuitable for hand-held touch surfaces and also for large capacitive or resistive surface-based tables. As described below (Section 3.2.2), we developed a simple hand identification technique for HandMark menus that does not use either vision or specialized hardware, but relies only on the touch points that are reported by a multi-touch surface.

CHAPTER 3

DESIGNING EFFICIENT MULTI-TOUCH INTERACTIONS BY USING HANDS AS LANDMARKS

In this chapter[3] we introduce a new interaction technique called HandMark Menus, which provide rapid command selection on multi-touch surfaces and support a large number of commands, by using hands and fingers as spatial landmarks. This chapter focuses on the design of efficient touch interaction methods, and the approach developed here (HandMarks) is evaluated in later chapters for two different platforms (tables and tablets).

3.1 HANDMARKS: EFFICICENT HAND-CENTRIC COMMAND SELECTION

Here we present the HandMark Menu technique for efficient interactions that uses the hands and fingers as landmarks. This technique is mainly a multi-touch based interaction that leverage users' spatial knowledge of their own hands and fingers to lay out a large number of commands, and to allow quick command selection. Starting with the multi-touch interaction, we describe each part of this technique in the following sections.

3.1.1 Multi-Touch Interaction

The most common way of interacting with any touch surface is direct finger based touch. Although there are other modalities of interaction (described in Section 2.1), we chose the most common

---

[3] Portions of the material in this chapter first appeared in the following publications: [120, 121].

mechanism in order to cover maximum ground in the area of touch interaction. To organize the component parts of a touch interaction, we follow the model proposed by Cechanowicz et al., which divides augmented interaction for GUIs [24] into two parts: objects and actions. The area of multi-touch interaction can be described by the same model, but with a few modifications as described below.

Touch objects

Any distinctive item with a visual representation in a touch interface can be considered as a touch object. These can be normal text, images, launcher icons, or an icon for a specific command. The main property of touch objects is that they can be directly manipulated simply by touching. Each touch object can be assigned to perform one specific task. Touch objects can also be called commands.

Actions

The possible manipulations of touch objects or commands are known as actions. Actions in touch interfaces can be categorized by the type of data that is being manipulated, and the number of fingers used [7] (see Figure 11).

- *Single-finger discrete actions:* Simple pointing actions performed with only one finger fall in this category. These tasks can also be one of multiple states: for example, selecting an icon with a one-finger tap, and tapping twice to open an application.

- *Single-finger continuous actions:* Single finger actions can also be continuous. For example, swiping, flicking, dragging all involve continuous 2D motions of the finger. These are commonly available in most modern touch interfaces.

- *Multi-finger discrete actions:* Multiple fingers can be used to perform one state of a multi-state task – for example, a multi-finger tap.

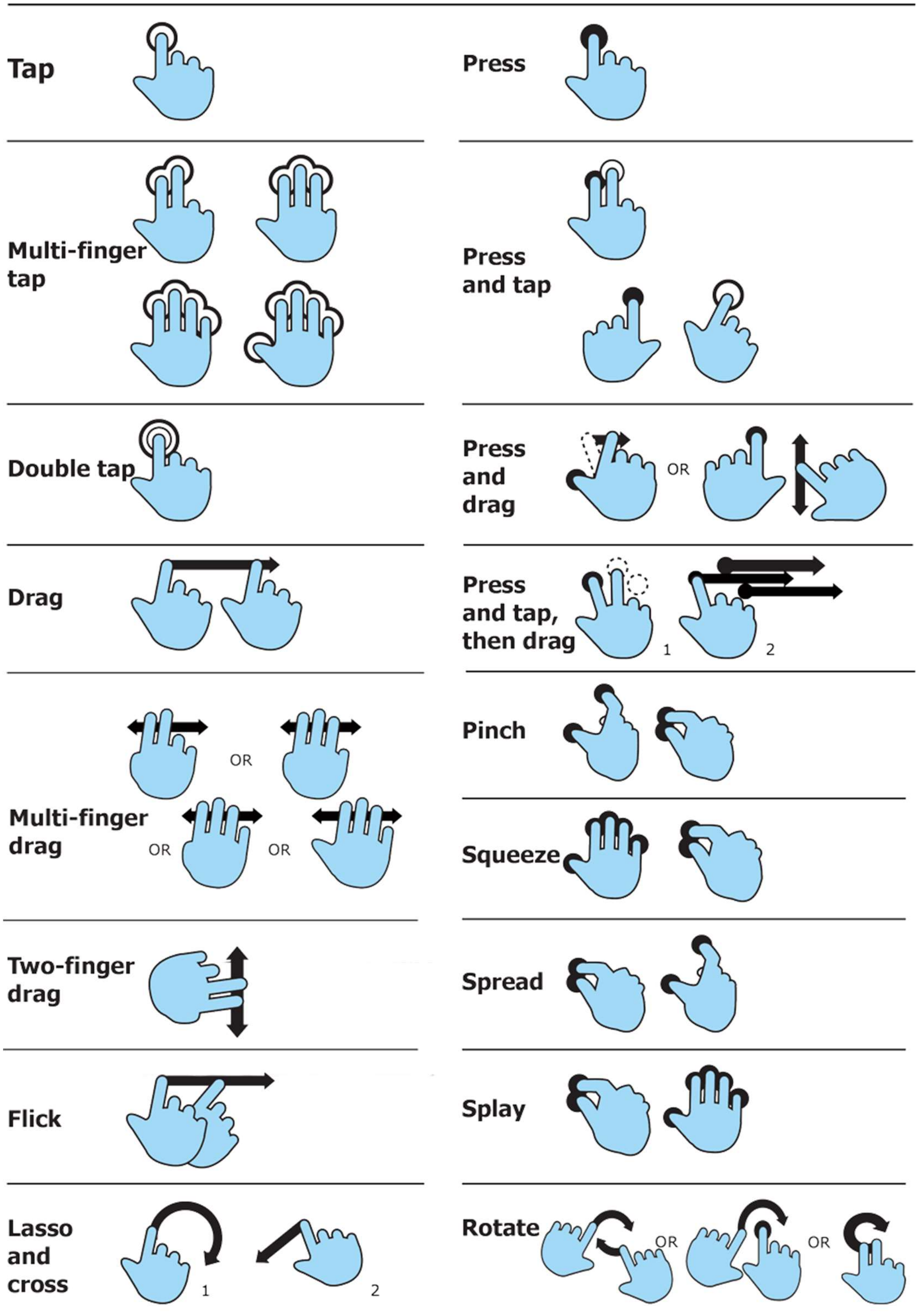


Figure 11: Examples of multi-touch actions (from [116]).

- *Multi-finger continuous actions:* a wide variety of touch actions are multi-finger continuous tasks. For example, pinching, zooming, and rotating are common facilities present in many multi-touch interfaces. However, although modern devices support many simultaneous touch points, and although people are capable of using multiple fingers at a time [20], most interfaces only provide interactions that can be performed with two fingers of one hand.

- *Complex actions:* Some actions require coordinated effort from both hands – bimanual actions. These are normally available in large tables where both hands can be used in data manipulation on the surface. In ordinary desktop systems where input is received through mice and keyboards, people can efficiently use chorded actions, often with two hands [84]. In touch interfaces, however, complex bimanual actions are rare [11, 66], although kinesthetic models suggest that humans are capable of using richer and more expressive forms of interaction with multiple fingers [20]. In the following section, we consider users' ability to perform complex tasks to make touch interactions more efficient.

3.1.2 Rapid Selection

Several studies have shown that hierarchical organization of commands (e.g., in menus or tabs) is one of the reasons behind slow selection performance [11, 70, 103]. In the previous chapter (Section 2.2), we discussed different ways of improving selection speed. In order to facilitate quick selections in touch interfaces for the design of HandMark menus, we used the ideas of hand-centric menu representation and reducing selection steps.

- *Hand-centric menu representation:* While interacting with any touch based interface, one tool that is always present is the user's hand. Based on informal observations of the characteristics of human hands, we determined that there is enough space between and around our fingers to place menu items – for example, small spaces between each pair of fingers, and a larger space between the thumb and forefinger (see Figure 12). We used a hand-centric menu presentation approach, and utilized the spaces around and between the fingers of a hand to place commands. With this menu representation, we intended to improve performance by bringing the menus close to users rather than placing menus at the edges of the screen.

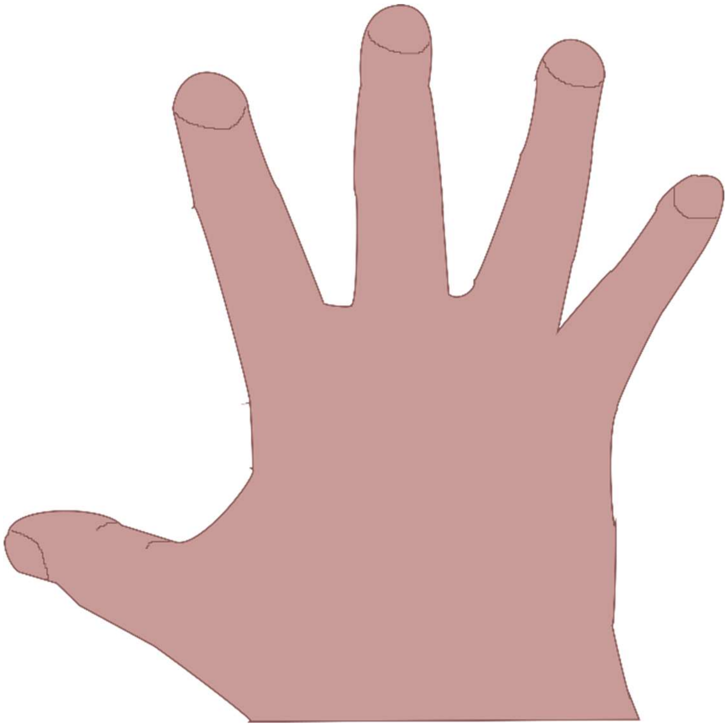


Figure 12: Example right hand, showing available space for menu items between fingers, and between thumb and forefinger.

- *Reducing steps in selection:* Reduction of the number of navigation steps is a useful way to improve selection performance, especially for expert users. Generally, there are three steps involved in command selection: menu invocation, command search by visually inspecting the menu, and selecting the desired command. An important aspect of HandMark menus is that they provide access to all commands with a minimum number of actions, and also allow experts to combine actions once locations are learned. Because people are capable of performing coordinated simple tasks simultaneously [20] even with two hands [43], our primary technique will use a bimanual selection process (sequential access using only one hand [76] is also considered later). The goal is to provide similar selection mechanisms for both novices and experts – so that with experience and the development of spatial memory, expert users will be able to drop the step of visual search to make more rapid selections [47, 103].

3.1.3 Spatial Memory Development

Spatial memory is a powerful mechanism for improving selection performance, since users can make quick decisions about command locations from the memory rather than relying on slow visual search [47, 103]. In real life, spatial learning benefits from landmarks available in the environment [83]. User interfaces typically provide few landmarks, however, particularly in large spatial interfaces when there are many items in the middle of the screen. The approach of HandMarks is to show commands around the user's spread-out fingers when the hand is touched down on the surface – and as long as the item locations are stable, the item locations can be remembered using reference to the user's own fingers and hand. For example, the 'copy' command might be located at the top of the user's index finger – with practice, the user will remember this association. The natural landmarks of the hand and fingers can act as powerful yet convenient tools to aid in spatial memory development.

3.1.4 Large Command Set

Accommodating large number of commands within user interfaces is a challenging issue. To design a technique that will support large number of commands while still allowing users to perform rapid memory-based selections, we leveraged the spaces between and around the fingers of both hands. In particular, we focused on the large space between the index finger and the thumb. We also exploit people's ability to control multiple fingers simultaneously [11, 20]. We assign different set of commands to different finger combinations of one hand. We identified four easy combinations for one hand. For example, a first command set is shown when only the index finger and thumb of the left hand are touched down; a second set is shown when the index, middle and thumb are touched (and so on). To keep the combinations simple, we decided to use index and thumb all the time (in an L-shaped posture), and created other combinations by adding other fingers (see Figure 13). When this approach is applied to both hands, we can specify eight different sets of commands (four with each hand).

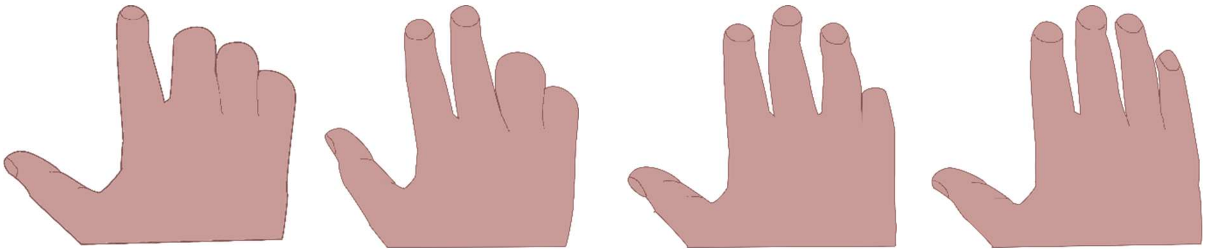


Figure 13: Finger combinations of one hand.

3.2 IMPLEMENTING THE HANDMARK TECHNIQUE

Here we discuss the implementation process and important aspects that are needed to consider while implementing new interactions using hands as landmarks.

3.2.1 Design of HandMark Menu

It is possible to design new interaction techniques for any multi-touch surface in many ways, but not all those ways would lead to efficient interactions. We have identified several issues that should be considered by interaction designers while designing HandMark interfaces.

- *Command placement:* Since HandMark menu places commands around and between fingers of a hand, care should be taken while placing the commands. As the space between index and thumb is bigger than spaces between other fingers, a larger number of commands can be placed in that region. Other places are not as convenient to reach (such as beside the pinky finger), so infrequently used commands or destructive commands (e.g., delete) can be placed in those positions. Figure 14 shows a mockup of an Android options menu with commonly used system settings and applications.

Figure 14: Command placement around and between fingers.

Figure 15: Commands in grid between index and thumb.

Figure 15 shows a mockup menu using Android’s commonly used application icons in a grid in the large space between index and thumb of right hand.

- *Size and number of commands:* While designing HandMark menus, special care is required for determining the size of the icons. Since fingers will be used to interact with those commands, size of human fingers [30] should be considered. By efficiently using the spaces around and between fingers of both hands, number of commands can be increased depending on the requirement of any system.
- *Selection mechanism:* People are capable of performing tasks with multiple finger at the same time, even with two hands. Depending on the intended platform, a suitable selection mechanism can be picked for HandMark menu. If both hands are available, then menu invocation and selection actions can be performed in parallel with two hands [11]; selection can be also performed by using two serial actions with the same hand [76], where after invoking the menu by placing the menu hand, same hand can be used to select an item from the menu (lifting hand after menu invocation and placed again). However, it is expected that there should be consistency in the selection mechanism so that users can develop familiarity and expertise with the command locations [47, 70]. Figures 16, 17 and 18, 19 show selection processes that can be performed using two hands and one hand respectively (details are provided in the following chapters).

Figure 16: Bimanual selection process (commands laid out around fingers).

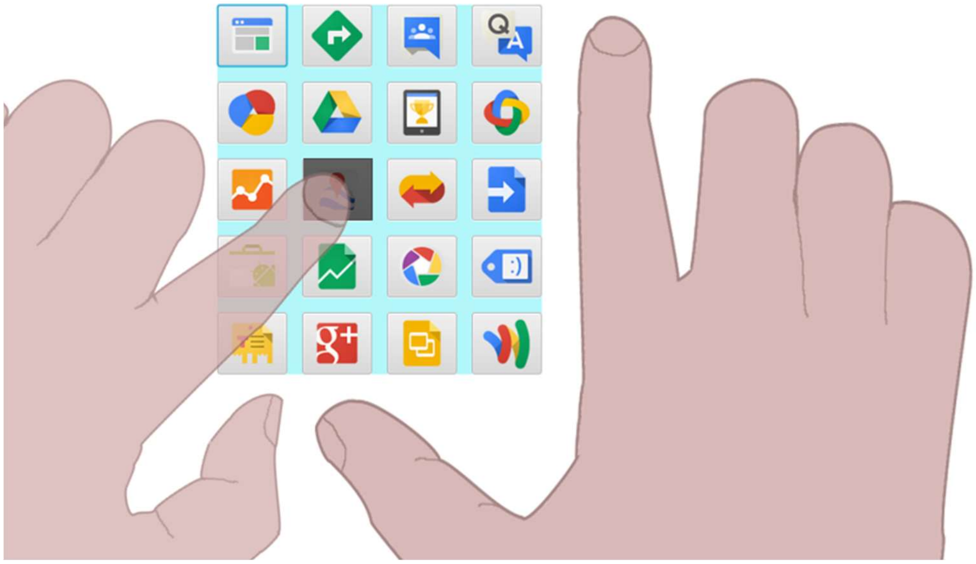


Figure 17: Bimanual selection process (commands laid out between thumb and forefinger).

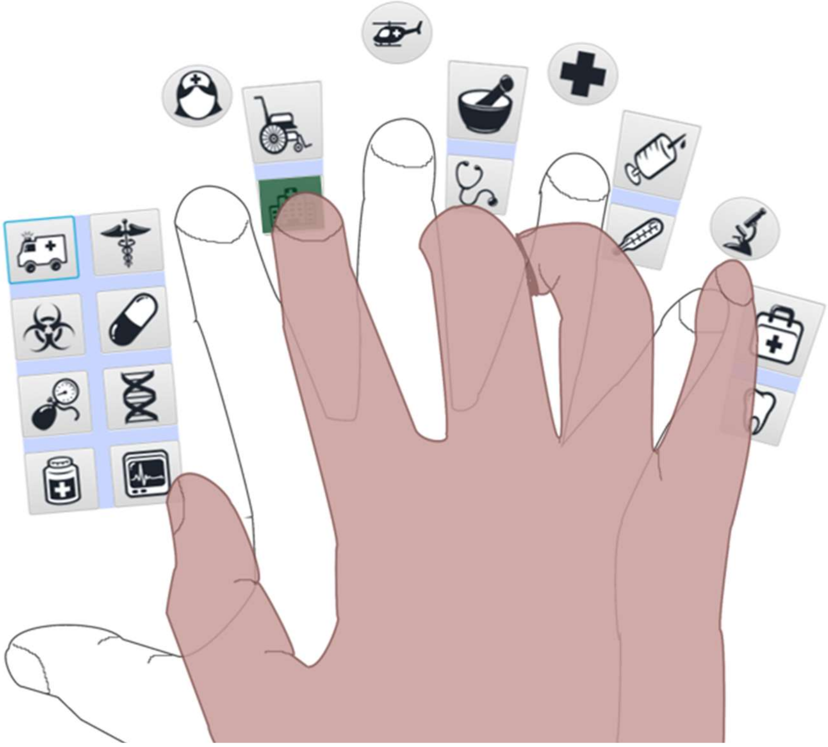


Figure 18: Single-handed selection process (the menu is shown when the hand touches the surface; then user lifts hand and selects item with forefinger).

Figure 19: Single-handed selection process (touch with hand shows menu, then user lifts hand and selects item with a second touch).

- *Feedback:* In any interface, feedback is a useful way to confirm selections. By changing the color of the command button selection, feedback can be provided to users. There are several other methods of feedback such as sound notification or vibrations which can also be used. Figures 16-19 show feedback by changing the color of the selected item.

- *Occlusion:* Some degree of occlusion is inevitable in touch based interactions. Two types of occlusions can occur in HandMark menu. First, occlusion of the background objects with the overlay menu, and second, occlusion of the menu or content by the hand itself. However, measures can be taken to minimize these effects. Semi-transparent menus can reduce occlusion of document content [47, 103], and having the menu move when the user rotate and moves their hand can reduce the effect of hand-based occlusion. We note that in memory based techniques, occlusions become less likely to affect selection as users learn the item locations, since they can begin to use the reference frame of their hands when preparing the selection action, rather than needing to carry out visual search of the actual icons on the screen.

3.2.2 Hand Detection

In order to implement HandMark menus, we need to identify the left and right hand accurately using only the fingers' touch points. We rely on the touch points since these are available in most of the capacitive and resistive touch surfaces. We make use of the distinctive geometries of people's hands in terms of the position of the thumb compared to other fingers and the individual positions of the fingers compared to the thumb. For example, the position of the thumb is always below the other fingers if the hand points upwards, and the rightmost touch is always the thumb for the left hand (and reversed for the right). Using these features, we are reliably able to differentiate the left and right hand. Other fingers (index, middle, ring, and pinky) can be found from the touch points once the hand and thumb are identified.

Simple touch-point-based hand detection algorithm

We considered only the touch points to identify a hand and fingers, in particular we used the (X, Y) coordinate values. The algorithm we use is as follows:

```
for each set of points touched down simultaneously
        identify the lowest touch point as thumb
        if thumb is the rightmost among all the points
            identify the hand as left
            identify remaining touch points sequentially in right-to-left order
        else
            identify the hand as right
            identify remaining touch points sequentially in left-to-right order
```

This simple algorithm requires that users place the fingers of one hand (all five fingers for the technique that lays out items around the fingers, and at least two for the version that uses the space between the thumb and forefinger) on the surface in an approximately upright posture, and at approximately the same time (but in any order). Other finger-identification techniques exist that are more robust (see Vogel [122]), but our simplistic approach works well for the prototypes described in this thesis.

3.2.3 Implementation Details

For the implementation we used a 24-inch Dell multi-touch monitor (ten simultaneous touch points with a resolution of 1920x1080) and the HandMark menu prototypes were developed using the JavaFX software environment. The software ran on an Intel(R) Core(TM) i3-2100 3.10 GHz machine (Windows 8.1 64-bit).

First, we implemented the touch-point-based simple hand detection algorithm for identifying the hand and fingers. We tracked the (X,Y) coordinates of the touch points, ran through the hand detection algorithm (Section 3.2.2). It detected the hand as left or right and assigned the touch points to specific fingers such as thumb, index, middle etc. Then we implemented the two variants of HandMark menu: HandMark-Finger and HandMark-Multi by following the parameters described below. We used 64px icons in our implemented versions and the icons were collected from open source sites and some from Android's library [34].

HandMark-Finger

In HandMark-Finger menu, we divided the location into three groups: between thumb and index, between other fingers and top of fingers. For each of them we set different configuration. Details for HandMark-Finger are provided in Table 1 considering only right hand. These parameters can be used for left hand also with a slight modification. In our implementation, all the distances are calculated from the top left corner of the button icons. We considered adult human hand size [30] while setting the value for each finger. Because of the physiological characteristics of hand and fingers we had to consider each finger separately. For the group between index and thumb, we placed eight icons square button in 2×4 grid, for the top of the finger group we used circular button, and for others we stacked two icons on top of each other which were skewed to the bottom to match the size of the space between two fingers.

Table 1: Configuration of HandMark-Finger menu.

Location	**Vertical distance (px)**	**Horizontal distance (px)**	**Angle (with y-axis)**
Between thumb and index	410 above from thumb	205 left from index	-5
Between index and middle	120 above from index	15 right from index	-5
Between middle and ring	80 above from middle	30 right from middle	5
Between ring and pinky	70 above from ring	40 right from ring	12
Side of pinky	50 above from pinky	20 right from pinky	15
Top of thumb	50 above	160 left	-1
Top of index	140 above	60 left	-5
Top of middle	140 above	40 left	0
Top of ring	140 above	10 left	10
Top of pinky	140 above	10 right	20

HandMark-Multi

In Handmark-Multi we focused on the large space between thumb and index. We placed 20 icons in a 4×5 grid of square buttons in the space between index and thumb. In particular, for the right hand, the top left corner of the grid was placed 450px above from thumb and 400px left from index finger. Since index finger are required to place on the screen in upright postures (straight with no angle), we did not use any angle. Same configuration can be used for left hand with little modifications.

3.3 RELATION WITH OTHER TECHNIQUES

There are several other techniques that attempt to provide efficient touch interactions: for example, finger count menus [11] and two-handed marking menu [66] are memory dependent multi-touch techniques that include a limited number of commands. Palettes and Toolglasses [63] are also similar in some ways to the HandMark techniques, although they were not developed for touch based devices; our HandMark Menu's two handed selection process can be seen as a descendent of these designs (i.e., controlling a menu of tools with the non-dominant hand, and making selections with the dominant hand). This division allows one hand to act in a supporting role to the other (following the Kinematic Chain model [43]).

However, although techniques such as Toolglasses can improve performance compared to traditional selection widgets [16], they only allow users to build up a coarse understanding of the locations of specific commands in relation to the hand, and only when used with an absolute input space. The intent of HandMark menus is to go beyond the design of other multi-hand selection techniques, and use the hands as a more detailed absolute reference frame for developing memory of specific item locations. This allows people to remember commands using features on their hands, and allows them to position their hands and fingers for a correct selection even before the hands have touched the surface.

3.4 SUMMARY

In this chapter, we have discussed different aspects related to the design of HandMark menus. The technique places commands in the spaces around and between the fingers so that users can more easily remember items with reference to their hands. In the next chapter (Chapter 4), we provide details of the design and evaluation of the bimanual HandMark menu for digital tables. Later (in Chapter 5) we introduce another variation that allows interaction with HandMark menus using only a single hand.

CHAPTER 4

BIMANUAL HANDMARK MENUS FOR TABLETOPS

In this chapter[4], we describe the details of HandMark menus designed for digital tables, and evaluate the technique (and the idea of using people's hands as landmarks) for command selection. We identify relevant design parameters for the table version of HandMark menus and discuss studies that we performed to answer key questions about performance and usability.

As introduced in the previous chapter, we developed and tested two hand-centric menu techniques for large multi-touch table displays. The first, HandMark-Finger, places command icons in the spaces between a user's spread-out fingers. This technique uses the hand as a clear external reference frame – once the locations of different items are learned, people can use their hand as a frame for setting up the selection action even before the fingers are placed on the touchscreen. The technique can be used with both hands to increase the number of available items.

The second technique, HandMark-Multi, provides multiple sets of commands, where the set is chosen by the number of fingers touching the surface. The technique is therefore similar to finger-count menus in the way that a category is selected, but allows many more items per category because a larger menu is displayed between the thumb and index finger (20 items in a 4x5 grid). HandMark-Multi also allows people to prepare for their selection before the hands are placed on the screen, once they have learned what menu an item is in and its location in the grid.

---

[4] Material in this chapter first appeared in the following publications: [121].

We carried out a study that compared HandMark menus to equivalently-sized tabbed toolbars presented at the top of the display. The study showed that HandMark-Finger was significantly faster than standard tabs (0.6 seconds per selection) with a similar error rate. The study also showed that although HandMark-Multi was slower than a tab UI in the early stages of use, there was no difference between the techniques once people gained experience. For both menus, it was clear that people did use their hands as a reference frame that aided memory of tool locations (e.g., people increasingly prepared their two hands for a correct selection as they gained experience). Participants also strongly preferred HandMark menus over the tab interfaces. The main contribution of this chapter is providing empirical evidence that the hands, and people's intimate knowledge of them, are an under-used resource that can improve the performance and usability of interfaces for tables and multi-touch systems.

4.1 DESIGN OF HANDMARK-FINGER MENU FOR TABLETOPS

The HandMark-Finger technique provides modal access to two different sets of commands, each belonging to one hand. Here we identify important factors in designing a HandMark-Finger menu for a relatively small number of commands. The five factors that can influence the performance of HandMark-Finger menu are: command placement, size of targets, number of commands, selection procedure, and visual feedback.

4.1.1 Command Placement

While interacting with any touch-based interface (e.g., tabletops or tablets) the hands are always present, but they are used only for selecting and manipulating commands. In the case of tabletops there are lots of spaces around the fingers of both hands which are not utilized in any interface. If we consider an adult human hand (fingers spread), we see spaces between fingers and above the fingers (see Figure 20). For example, there is a large space between forefinger and thumb. Similarly, there is space between any two adjacent fingers. In our research, HandMark-Finger menu explores the spaces around and between the fingers of both hands and places commands there. Since commands are placed in spaces between two adjacent fingers and top of every finger of both hands, those hands and fingers will work as anchoring points that will help users to develop spatial memory.

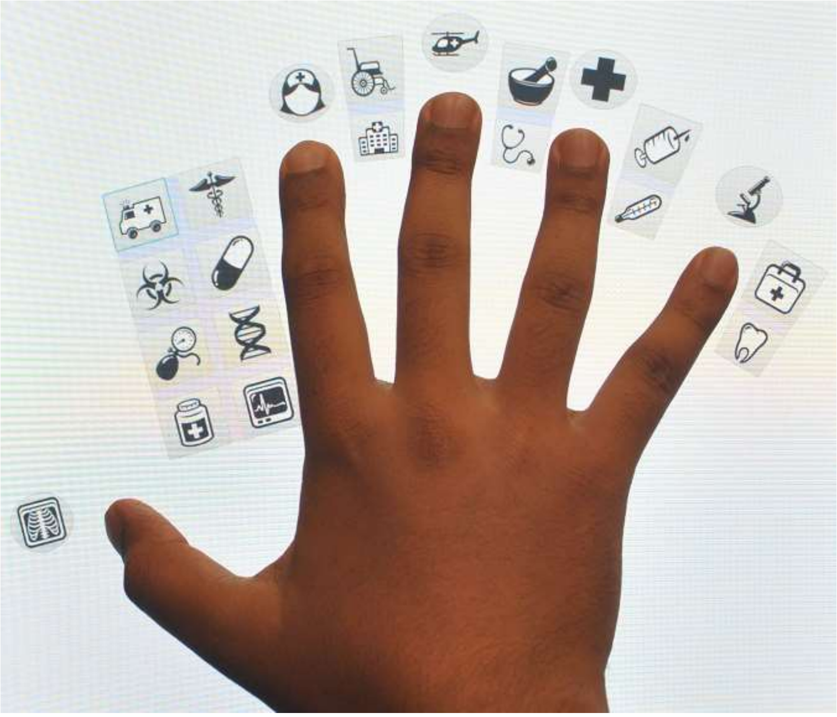


Figure 20: HandMark-Finger menu with commands (right hand).

4.1.2 Size of Targets

For a multi-touch tabletop interface, target size is a crucial factor. In general, bare hands are used to interact with the system so it will be difficult for the users to select commands accurately if the target size is not properly matched with fingers. Previous research on multi-touch interactions [47, 76] have considered these factors while designing target sizes. For HandMark-Finger, we determined the target size using the average width of an adult index finger (16-20mm [30]) as a guideline and considering Parhi et al.'s recommendation that touch targets be no smaller than 9.6mm [94].

4.1.3 Number of Commands

One of our goals was to accommodate a large command vocabulary. To achieve that goal, we designed HandMark-Finger menu by placing pairs of commands between fingers, and one at the top of each finger. As the space between the thumb and index finger is larger, we placed eight commands there in a 2×4 grid (Figure 20). In total, HandMark-Finger can support 21 commands as a set for one hand, and by combining two hands, it can accommodate 42 commands. Commands sets can be grouped by semantic factors (e.g., command type or icon color) to make it convenient for users to remember which hand contains which commands.

4.1.4 Selection Mechanism

While designing the prototype for HandMark-Finger, rapid command selection was a main priority. We created a bimanual command selection technique that ensures the use of hands and fingers as landmarks to facilitate rapid development of spatial memory and still makes selection process fast. To invoke the HandMark-Finger menu, the five fingers of the left or right hand are touched down in any order, spreading the hand to provide space between the fingers. Commands are displayed in the spaces around the hand and between the fingers (Figure 21) of that hand, and selections are made by touching an item with the other hand.

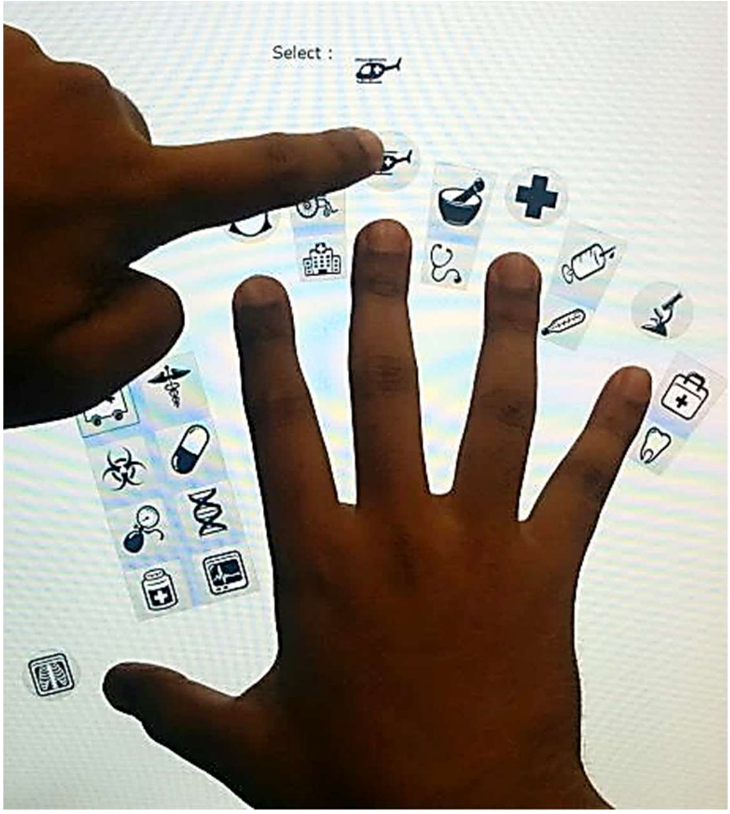


Figure 21: Command selection in HandMark-Finger menu.

The user can rotate and move the menu in any direction. Following a hand touch, the menu appears after a 300ms delay, but selections can be made immediately without waiting for the menu to appear. This enables two types of selections: *novice* and *expert*.

Novice

Novices are users who are new to the HandMark-Finger menu system and do not have any prior knowledge of different commands' positions with reference to hands and fingers. They can wait until the menu appears after placing the menu hand and use visual search to select a target with a finger from the other hand (Figure 21). Novices therefore carry out a selection process by following three sequential steps: menu invocation by placing hand on screen, visual search for an item, and command selection with a finger of the other hand.

Figure 22: HM-Finger expert selection without seeing the commands.

Expert

Expert users, who have built up spatial memory of the location of a desired item with reference to hands and fingers, can tap the location without waiting for the menu to be displayed (Figure 22).

This follows Kurtenbach's principle of rehearsal, which states that novice actions should be a rehearsal of the expert mode [74]. In expert selection, the visual search for an item step is dropped, and the other two steps (menu invocation and command selection) can be combined into a single step.

We note that the 300ms delay in showing the menu icons is not a necessary part of the design. In our prototypes we included this delay in order to more clearly see users' shift to expert operation, but in real-world versions of the menu system, the icons could be shown immediately (although the delay could be useful for avoiding occlusion of the workspace, as discussed below).

4.1.5 Feedback

The presence of feedback can often enhance the development of spatial memory. Feedback can be proprioceptive or visual. The users receive proprioceptive feedback while making physical movements to interact with computers, such as typing on a keyboard or touching a touchscreen. Prior studies have shown that direct proprioceptive interaction with items can facilitate better spatial learning [61, 100, 117], and through rehearsal, novice users can exploit spatial memory to become expert [47, 70, 74]. However, spatial memory is not always precise and users often have only approximate memory of the item-locations they have interacted before. In these circumstances, visual feedback can help users to make accurate selections. Results of previous studies [29, 62] showed that people often use their spatial memory to remember the proximal locations of the items, yet confirmation of the selections with visual feedback is suggested.

Figure 23: Feedback of selection.

In any interactive interface, visual feedback confirms users about their selection. The most common form of feedback is through a visual highlight over the active item. Rapid visual feedback for novice selections helps users to develop spatial memory quickly, and thus accelerates the transition from novice to expert. HandMark-Finger provides visual feedback to users by showing the command set as long as the respective menu hand is touched down on the screen (see Figure 23), and by highlighting the chosen item. In our experimental system where the system knew the intended target, the highlighting was green for correct selections and red for incorrect selections. In expert mode (where the full set of menu icons is not shown), the selected icon is displayed (and highlighted) for 500ms.

4.1.6 In-Place Tools and Occlusion of the Content

In-place tools appear at the user's work location (e.g., popup menus). All in-place interfaces occlude parts of the work surface [122] or the whole screen (e.g., FastTap or CommandMaps). For both of the HandMark menus we chose a hybrid overlay presentation – when used in novice mode, the menu covers part of the screen, but in expert mode, no visual presentation is needed. In addition, it is easy for the user to control the presence of the overlay (by lifting the fingers from the touch surface), allowing rapid switching between menu and content. It is also easy to move the menu hand after activating the menu, which allows the user to further manage occlusion.

4.2 DESIGN OF HANDMARK-MULTI MENU FOR TABLETOPS

We developed a second variant of HandMark menus that also uses hands and fingers as landmarks but provides access to a much larger number of commands. The technique is called HandMark-Multi. This variant also provides modal access to command sets, but this time there are eight sets of commands. Here we present the design factors of HandMark-Multi for tabletops.

4.2.1 Command Placement

While designing HandMark-Multi, we considered similar design factors as described above (see Section 4.1.1). Our goal was to accommodate a larger number of commands. We noticed that if we place a hands on a tabletop (keeping thumb and index finger in ‘L’ shape and others pointing upward), we see a large space between thumb and forefinger. We utilized that large space to place commands in a spatially-stable grid (Figure 24). This layout does not have the same degree of connection to the individual fingers and features of the hand as HandMark-Finger, but there is still a clear reference frame provided by the thumb and forefinger to help users to develop spatial memory of the grid. Compared to the extensive availability of landmarks in HandMark-Finger, HandMark-Multi provides limited landmark support with only two fingers of both hands.

Figure 24: HandMark-Multi menu showing the first set of left hand.

4.2.2 Size and Number of Commands

For selecting the size of the command set for HandMark-Multi menu, we followed the same design guidelines used for HandMark-Finger menus (discussed in Section 4.1.2) and considered previous design recommendations [30, 47, 94]. We were inspired by Gutwin et al.'s FastTap menu design for tablets [47], and placed 20 commands (a 4×5 grid) in the space between thumb and forefinger.

The other fingers of the hand can then be used to indicate different command sets. The thumb and forefinger are always used to frame the grid, so we can provide four sets of command in one hand (the first set uses only thumb and forefinger, and the others add the middle, ring, and pinky fingers).

In total, HandMark-Multi supports 160 items (20 in each set, and 4 sets in each hand). To make it convenient for the users, we grouped similar items in one set based on their type and color.

4.2.3 Invocation Mechanism and Feedback

In HM-Multi, there are eight sets of commands (four in each hand) and each set can be accessed by touching on the screen with a specific number of fingers and thumb from one hand in an L-shaped posture. Figure 25 shows the four combinations to access the sets of left hand. Similar four combinations from right hand can be used to invoke left hand's sets.

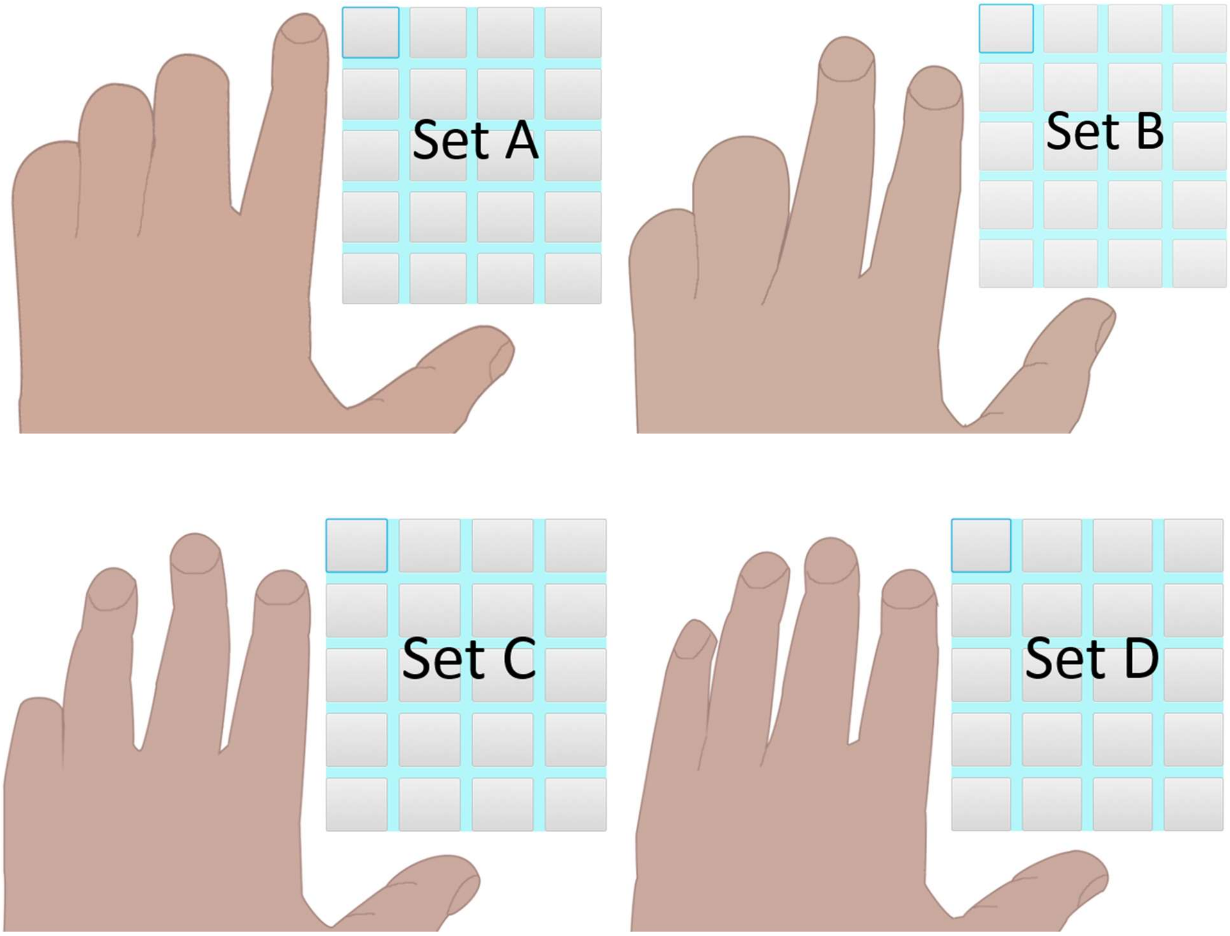


Figure 25: Finger combinations to access the sets of HM-Multi menu (left hand).

The index finger and thumb are always used, and adding other fingers accesses other sets – e.g., to access the second set on the left hand, the index and middle fingers of the left hand are touched down along with the thumb. The menu follows the user's hand as it moves or rotates on the screen. Once a desired command set is invoked by placing fingers of one hand in a specified combination, other hand's finger can be used to select a target from that set.

Handmark-Multi also supports the novice and expert selection methods (see Figure 26) similar to HandMark-Finger menu described earlier (see Section 4.1.4). Since novices are new to the system, they must rely on visual search to locate one desired item. Once found, they can select that with other hand's finger. Experts, on the other hand, cap drop the visual search step by leveraging the spatial knowledge of the item with reference to hand and fingers and can perform rapid selection by executing menu invocation and command selection with two hands in parallel.

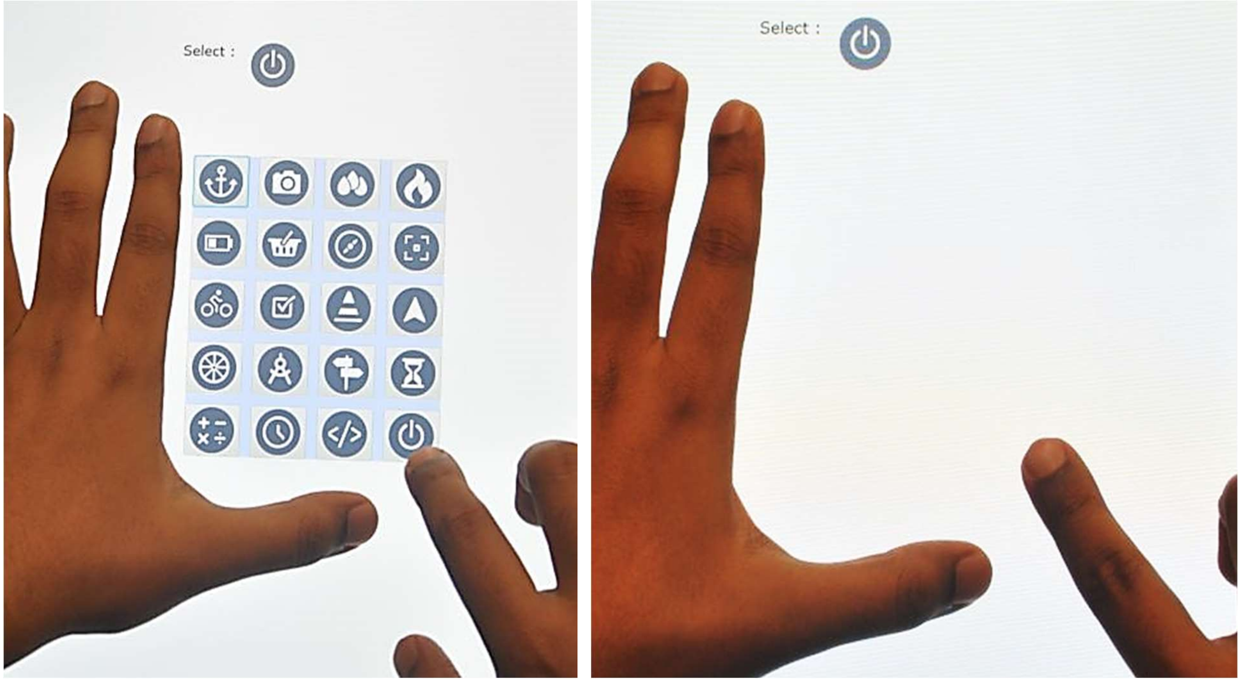


Figure 26: Selections in HM-Multi menu: (left) novice, (right) expert.

Similar to HM-Finger (see Section 4.1.5), HM-Multi menu provides feedback about item layout as long as menu hand is touched down and visual feedback (using a colored highlight) for target selection.

4.3 EXPERIMENT: HANDMARK-MENUS FOR TABLETOPS

To assess the performance of command selection using hands as landmarks, we conducted a study comparing HandMark menus to standard tab-based menus. We compared the interfaces in a controlled experiment where participants selected a series of commands over several blocks, allowing us to examine selection behaviors and learning effects in each interface.

4.3.1 Experimental Conditions

Two versions of HandMark menus, and two equivalent versions of a standard tab interface were implemented in a tabletop environment (see Figures 20 and 24). Here we describe the experimental systems used in the study.

HandMark-Finger

HandMark-Finger menu was implemented as described above (see Section 4.1). The interface used in the experiment contained 21 commands in each hand's set (grouped by color and type), for a total of 42 items. Eight items were used as study targets – four from each hand (Figure 27).

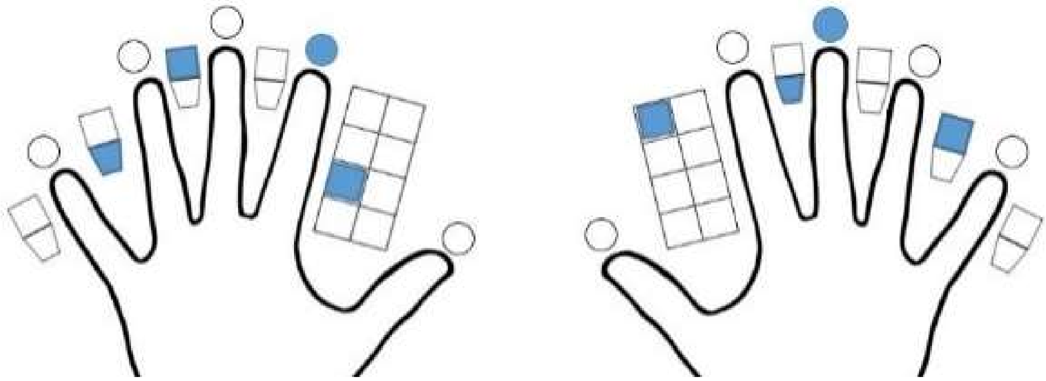


Figure 27: Target locations for HM-Finger.

HandMark-Multi

HandMark-Multi menu was also implemented as described above (see Section 4.2). There were 20 command buttons in a 4×5 grid for each set. In this menu, there were eight sets (grouped by color and type) for a total of 160 command buttons. Eight targets were used in the study, one from each set (Figure 28 shows command locations within the grid; note that each command was from a different set).

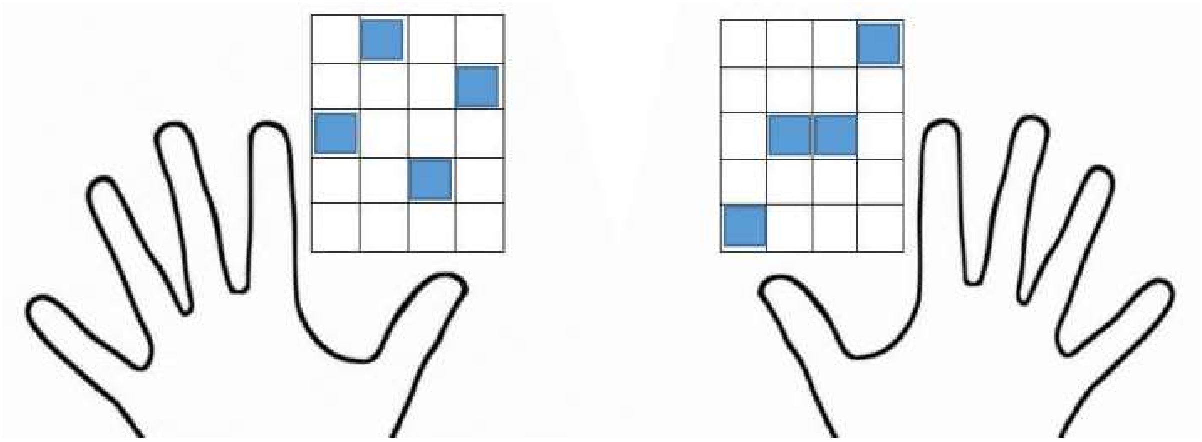


Figure 28: Target locations for HM-Multi (collapsed across different sets).

Standard Tab interfaces

We implemented two versions of standard tabbed ribbon interface (called Tabs-2 and Tabs-8) to compare with the two HandMark menus. Tabs-2 (Figure 29) had only two tabs (each consisting of 20 command buttons in a 10×2 grid) to match HandMark-Finger. For Tabs-8 (Figure 30), there were eight tabs each with 20 items in a 10×2 grid (total of 160). For both versions, items were grouped either by type or color, and the named tabs were placed side by side as a ribbon interface at the top left edge of the screen. Users could tap on a particular tab to view its commands before selection.

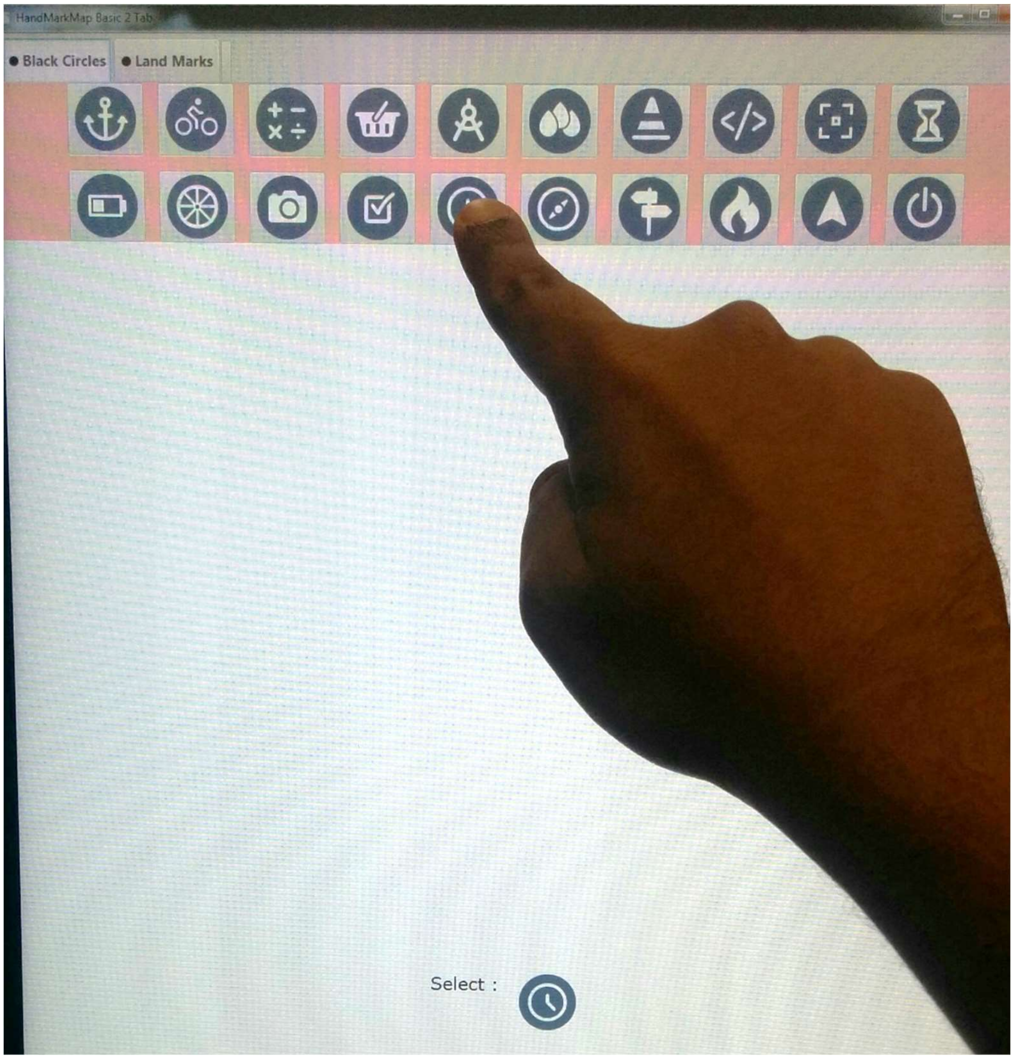


Figure 29: Standard tab menu with two tabs - Tabs-2.

We compared HandMarks to Tabs rather than other research systems for several reasons: Tabs offer equitable command range to our prototypes (which is not provided by several research techniques), and they are the de facto standard UI for many interfaces; in addition, a main goal of the evaluation was to compare the strong landmarking and proprioceptive approach of HandMarks to a traditional visually-guided approach.

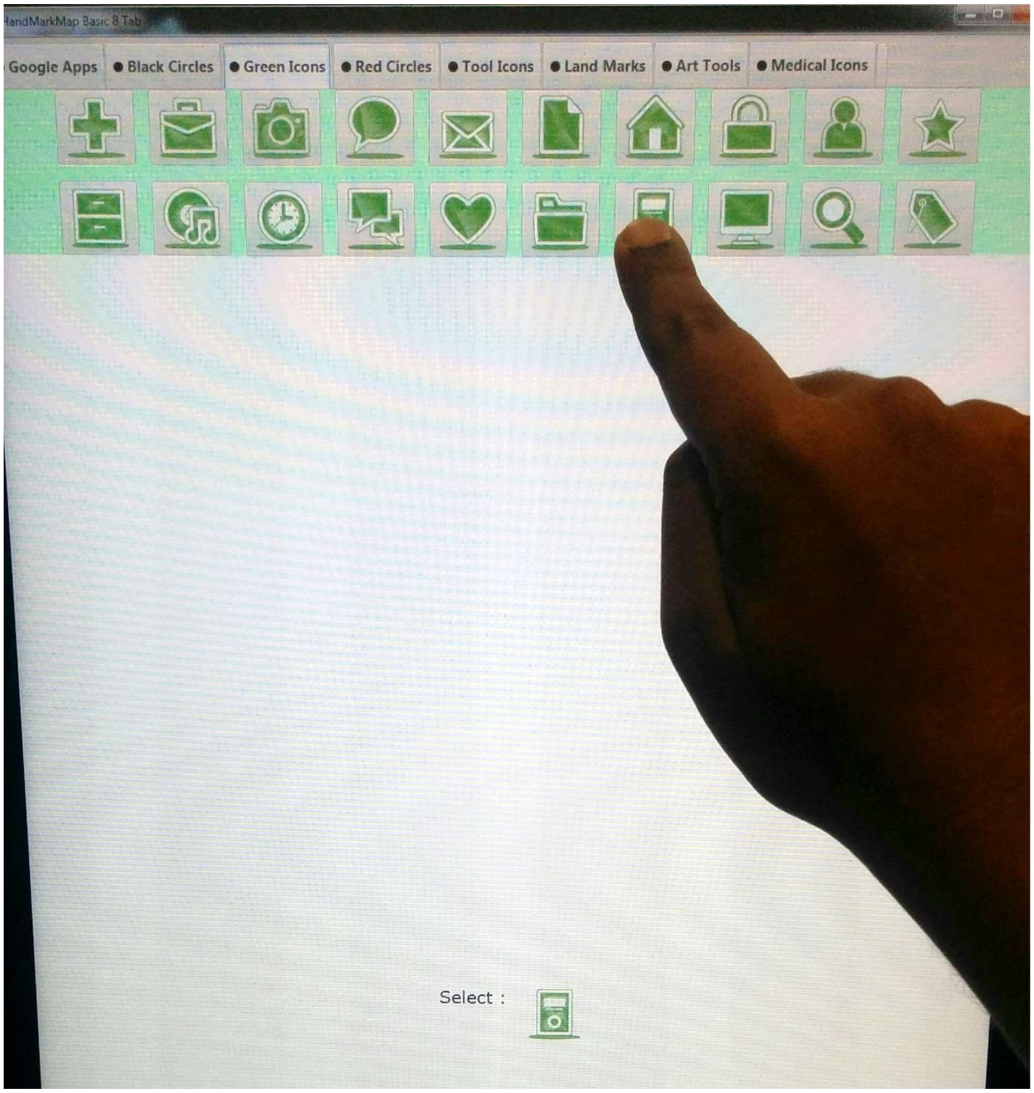


Figure 30: Standard tab menus with eight tabs - Tabs-8.

4.3.2 Hardware Configuration

Both studies were implemented on a tabletop environment and carried out using a Dell 24-inch multi-touch monitor (1920×1080 resolution) placed flat on a table in front of the participant in portrait mode. Although this is not a large-scale surface, it adequately simulated the combination of a local work area and a far edge that participants needed to reach to. Both studies were conducted using a Windows 7 PC; the interfaces were written in JavaFX.

4.3.3 Performance Measures

The study software recorded all experimental data including trial completion time, errors, and incorrect set invocations. Trial completion time is defined as the time taken by the user to select a target correctly after it was shown on the screen. It also includes the time taken to correct an incorrect selection. The software records an error when the participant selects a location which is not the displayed target. The trial ended only when the user selected the correct target, so multiple errors were possible for each trial. Along with errors, the system also recorded incorrect set selections for each trial. Number of incorrect set invocations means the number of times a wrong command set was opened by the user while searching for a target.

Trial completion time gives us a measure of the learning effect of the interfaces, while errors provides us with insights into the success of the selection task. Incorrect set invocations help us to understand the development of correct spatial memory and the amount of visual search that users are doing.

4.3.4 Methods

Participants

Fourteen participants were recruited from the University of Saskatchewan; one person's data could not be used due to technical difficulties, leaving 13 participants (6 females; mean age 24 years). All participants were students, and had no previous experience of using tabletop interfaces. However, most of them had used multi-touch systems such as tablets and smartphones before, with average weekly use between 1-10 hours.

Task and Stimuli

We carried out a simple controlled experiment where participants selected a series of commands over several blocks. We compared two versions of HandMark menus with two equivalent standard tabbed menus. For each interface, only eight commands were used as stimuli, in order to allow faster development of spatial memory. For HandMark-Finger menu, we selected targets from three areas: top of fingers, large area near thumb, and spaces between fingers (see Figures 31: top). On the other hand, for HandMark-Multi targets were selected from two areas: near fingers and far from fingers (see Figures 31: bottom).

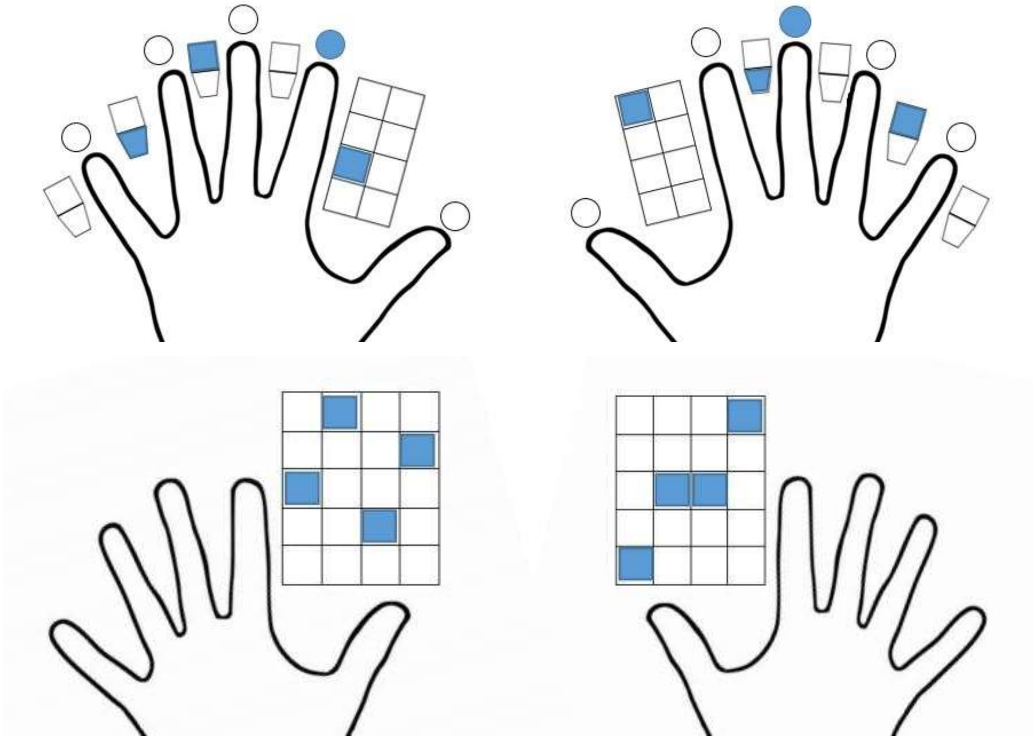


Figure 31: Item locations used in study: (top) HM-Finger, (bottom) HM-Multi (all).

During the experiment, participants were required to select targets from the proper menu sets using the specified technique. Participants were instructed to use the expert selection method if they were confident about the target locations. We provided feedback for menu layout and item selected (both expert and novice selections) for 500ms: green for correct and red for wrong selections. Figure 32 shows a wrong selection's feedback.

Figure 32: Feedback on selection.

Procedure and Design

The study was divided into two parts. Part 1 tested HandMark-Finger and Tabs-2, and part 2 tested HandMark-Multi and Tabs-8. Participants completed a demographics questionnaire, and then performed a sequence of selections in the custom study system with both interfaces. For each version, a command stimulus (one of eight icons) was displayed in the middle of the screen (Figure 33); the participants had to tap a large (easily accessible) button placed at bottom to view the command stimulus and start the trial. Trials were timed from the appearance of the stimulus until the correct target was selected. Participants were instructed to complete tasks as quickly and accurately as possible, and were told that errors could be corrected simply by selecting the correct item. In our analysis, we include error correction in completion times.

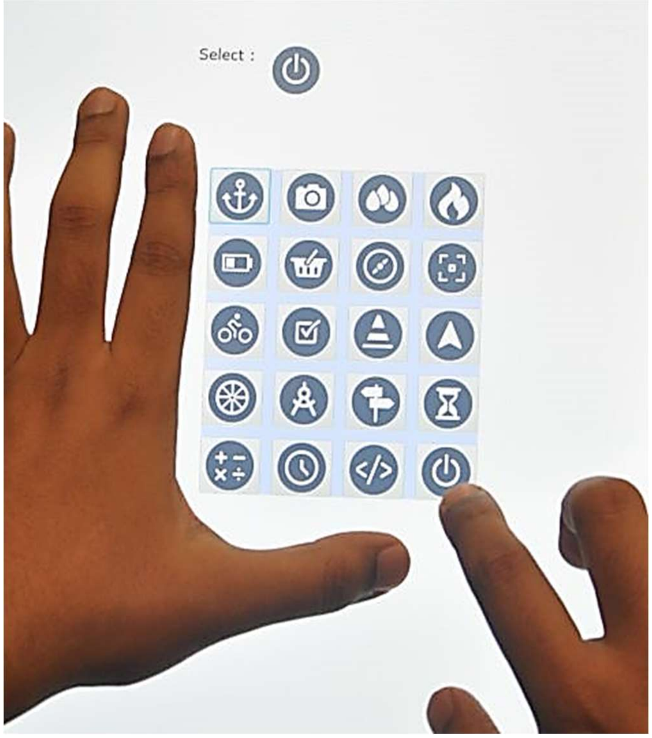


Figure 33: HandMark-Finger in experiment.

For all interfaces, targets were organized into blocks of eight trials. Participants first performed one practice session which was consisted of two commands and ten blocks (data discarded) to ensure that they could use the interfaces successfully. They then carried out 17 blocks of eight selections each. Targets were presented in random order (sampling without replacement) for each block. After each interface, participants were allowed to rest, and filled out a questionnaire based on the NASA-TLX survey [49] (see Appendix). At the end of each pair of techniques, participants gave their preferences between the systems. The order of the interfaces in each part of the study, along with the order of the parts, was counterbalanced using a Latin square design.

The study used 2×17 within-participants RM-ANOVAs; with factors *Interface* (HandMark-Finger vs. Tabs-2; and HandMark-Multi vs. Tabs-8), and *Block* (1-17). Dependent measures were trial completion time, errors per command, and incorrect tabs per command. With thirteen participants, two interfaces, seventeen blocks and eight trials, the system recorded a total of (13×4×17×8) 3536 trials for each part of the study. In total, for two parts there were 7072 trials in the experiment. The experiment took approximately 60 minutes per participant.

Hypotheses

There were two parts of the study; the first compared HandMark-Finger and Tabs-2, and the second compared HandMark-Multi and Tabs-8. For both parts the hypotheses were:

H1. Trial completion will be faster for HandMark than for Tabs.

H2. HandMark will be faster both for novices and experts.

H3. There will be no evidence of a difference in error rates between HandMark and Tabs.

H4. There will be no evidence of a difference in invoking the wrong set between HandMark and Tabs.

H5. There will be no evidence of a difference in perception of effort for HandMark and Tabs.

H6. Users will prefer HandMark over Tabs.

4.4 RESULTS: HANDMARK-FINGER VS. TABS-2

Here we present the results of the first part our study with HandMark-Finger and Tabs-2, organized by dependent variable (trial completion time, number of errors, incorrect set invocation and subjective responses).

4.4.1 Trial Completion Time

We calculated mean trial completion time for each command by dividing the total trial time by the number of commands in that block. Mean trial completion times were 0.62 seconds faster per command with *HandMark-Finger* (2.32s, s.d. 0.79s) than with *Tabs-2* (2.94s, s.d. 0.95s), see Figure 34.

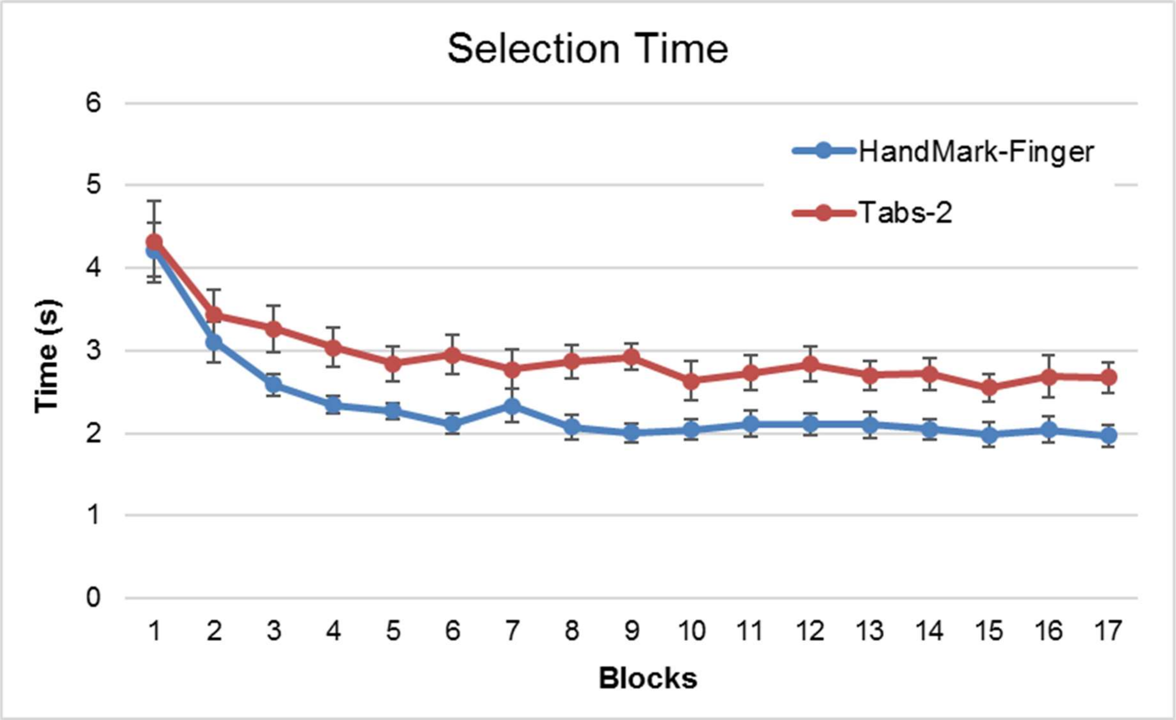


Figure 34: Mean selection time by Interface and Block.

RM-ANOVA showed a significant main effect of *Interface* ($F_{1,12}$=37.59, p<.0001). For the small menus, we therefore accept **H1** – HandMark-Finger was 21% faster than Tabs-2. As shown in Figure 34, mean trial completion times decreased across trial blocks for both interfaces; RM-ANOVA showed a highly significant effect of *Block* ($F_{16,192}$=18.04, p<.0001). There was no interaction between *Interface* and *Block* ($F_{16,192}$=1.00, p=.456) as HandMark-Finger was faster than Tabs-2 throughout the experiment. For HandMark-Finger, we therefore accept **H2**.

4.4.2 Number of Errors

We also analyzed number of errors per command (counted as any incorrect item selection) for the HandMark-Finger and Tabs-2 interfaces. RM-ANOVA showed no effect of *Interface* on errors, with HandMark-Finger at 0.04 errors/command, s.d. 0.09, and *Tabs-2* at 0.04 errors/command, s.d. 0.08 ($F_{1,12}$=0.01, p=.924). We therefore accept **H3** (errors are considered further in discussions). Our tests did not show any significant effect of *Block* ($F_{16,192}$=1.07, p=.388) on errors between two experimental interfaces. The interaction between *Interface* and *Block* was also not significant ($F_{16,192}$=1.42, p=.138).

4.4.3 Incorrect Set Invocations

During the study between HandMark-Finger and Tabs-2, we also recorded the number of times participants invoked the wrong command set (e.g., the wrong tab or the wrong hand/finger combination). RM-ANOVA did not show any significant effect of *Interface* ($F_{1,12}$=0.26, *p*=.623) on incorrect set invocations, with 0.05 incorrect sets/command, s.d. 0.09 for both *HandMark-Finger* and *Tabs-2*. There was also no significant effect of *Block* ($F_{16,192}$=0.6, *p*=.88). Therefore, we accept **H4** for HandMark-Finger.

4.4.4 Subjective Responses

We also analyzed subjective responses collected from the participants. There were two types of subjective responses: effort and preferences.

Effort

Participants' responses were positive for both interfaces, but there were no strong differences in NASA-TLX scores (see Table 2) for HandMark-Finger and Tabs-2 except regarding performance and frustration. Friedman tests showed that perceived performance of participants was significantly higher for HM-Finger with mean 8.69 (s.d. 0.95) than Tabs-2 with mean 6.92 (s.d. 1.93). For perceived frustration, HM-Finger also performed significantly better than Tabs-2. Here, low score means better performance. The score of HM-Finger was significantly lower (mean 2.00, s.d. 1.78) than Tabs-2 (mean 4.69, s.d. 3.01). However, Friedman tests did not find any significant differences on any other question, and the mean scores were similar. Therefore, we accept **H5**.

Table 2: Mean (s.d.) NASA-TLX effort scores (0-10 scale, low to high).

	HandMark-Finger	**Tabs-2**	χ_r^2	***p***
Mental	5.54(2.73)	5.46(2.37)	0.08	0.78
Physical	5.38(2.79)	6.62(2.47)	1.23	0.27
Temporal	5.00(2.89)	4.77(1.74)	0.08	0.78
Performance	8.69(0.95)	6.92(1.93)	7.69	**0.01**
Effort	5.31(2.75)	6.77(2.35)	0.31	0.58
Frustration	2.00(1.78)	4.69(3.01)	4.92	**0.03**

Preferences

After the experiment, we also asked participants about their preferred interface in terms of several qualities: speed, accuracy, memorization, and comfort, as well as their overall preference (see Table 3). Counts were easily distinguishable, and overall, 92% of participants preferred HandMark-Finger. We therefore accept **H6.**

Table 3: Counts of participant preferences.

	HandMark-Finger	**Neither**	**Tabs-2**
Speed	12	0	1
Accuracy	9	2	2
Memorization	11	2	0
Comfort	12	0	1
Overall	12	0	1

4.5 RESULTS: HANDMARK-MULTI VS. TABS-8

Here we discuss the results of the second part of the study with HandMark-Multi and Tabs-8, again organized by dependent variable (trial completion time, number of errors, incorrect set invocation and subjective responses).

4.5.1 Trial Completion Time

The analysis showed mean trial completion time was 0.62 sec/command slower with *HandMark-Multi* (3.84s, s.d. 2.1s) than with *Tabs-8* (3.22s, s.d. 1.43s), giving a main effect of *Interface* ($F_{1,12}$=4.86, *p*=.048). However, this result must be interpreted in light of the highly significant interaction between *Interface* and *Block* ($F_{16,192}$=4.96, *p*<.0001). In early blocks (see Figure 35), Tabs-8 was faster than HandMark-Multi, but by the final four blocks, the two techniques were similar (RM-ANOVA for blocks 14-17 showed no significant effect of *Interface*, $F_{1,12}$=.008, *p*=.932). Hypotheses **H1** and **H2** therefore cannot be clearly rejected – HandMark-Multi was slower overall, but there was no difference in performance once users learned item locations.

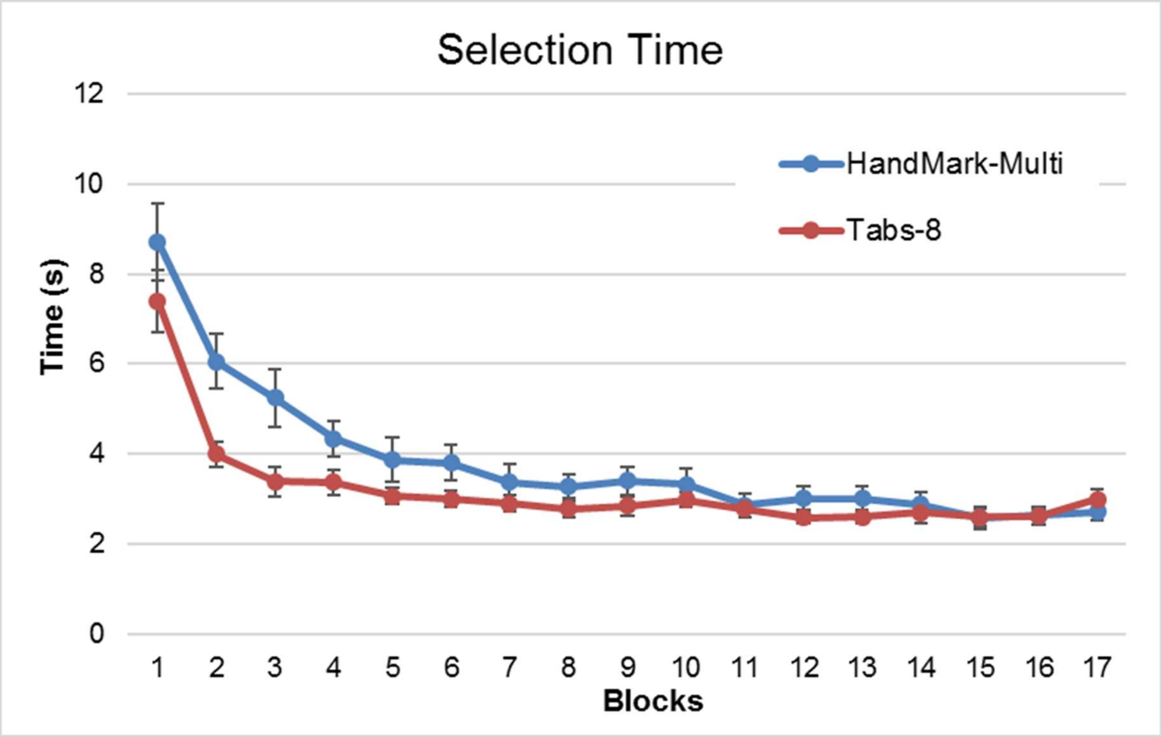


Figure 35: Mean selection time by Interface and Block.

4.5.2 Number of Errors

RM-ANOVA showed similar trends to the smaller menus: HandMark-Multi had 0.06 errors/command, s.d. 0.1, and *Tabs-8* had 0.04 errors/command, s.d. 0.09, with no main effect ($F_{1,12}$=2.47, p=.142). We therefore accept **H3** (errors are considered further in Section 4.7). There was no effect of *Block* ($F_{16,192}$=1.6, p=.07), and no interaction between *Interface* and *Block* ($F_{16,192}$=.75, p=.74).

4.5.3 Incorrect Set Selection

RM-ANOVA showed a different trend to the smaller menus: HandMark-Multi had more incorrect set selections (0.64 per command, s.d. 0.86) than *Tabs-8* (0.18 per command, s.d. 0.38), and this difference was significant ($F_{1,12}$=9.78, p<.01). Figure 36 shows that incorrect selection rate decreased significantly over *Block* ($F_{16,192}$=20.65, p<.0001). There was also a significant interaction between *Interface* and *Block* ($F_{16,192}$=6.51, p<.0001). We therefore reject **H4** for HandMark-Multi.

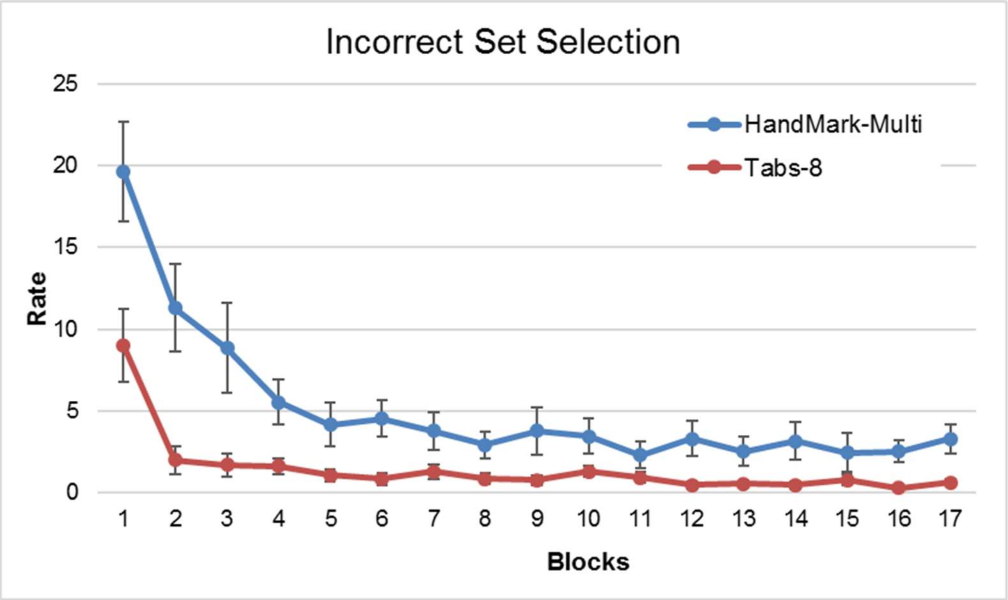


Figure 36: Incorrect set selection rate by Interface and Block.

4.5.4 Subjective Responses

Similar to HandMark-Finger vs. Tabs-2 study, we analyzed subjective responses collected from the participants for HandMark-Multi vs. Tabs-8.

Effort

Again, participants gave positive responses for both interfaces. Friedman tests did not find any significant differences (see Table 4) between HandMark-Multi and Tabs-8. Mean scores were close in all cases; therefore, we accept **H5**.

Table 4: Mean (s.d.) NASA-TLX effort scores (0-10 scale, low to high).

	HandMark-Multi	**Tabs-8**	χ_r^2	***p***
Mental	7.62(2.36)	6.23(1.83)	2.77	0.10
Physical	6.00(2.71)	7.23(2.13)	1.23	0.27
Temporal	6.46(2.76)	6.31(2.06)	0.08	0.78
Performance	7.62(1.5)	7.54(2.26)	0.69	0.41
Effort	7.00(2.45)	7.54(1.61)	0.08	0.78
Frustration	3.69(2.02)	4.08(2.5)	1.23	0.27

Preferences

Participant preference counts (Table 5) were again easily distinguishable, with a strong preference for HandMark-Multi (73% overall). We therefore accept **H6**.

Table 5: Counts of participant preferences.

	HandMark-Multi	**Neither**	**Tabs-8**
Speed	10	0	3
Accuracy	6	2	5
Memorization	12	1	0
Comfort	10	1	2
Overall	10	1	2

4.6 USE OF HANDS AS LANDMARKS

To consider whether participants made use of their hands as landmarks, we further analyzed the number of expert selections made without any visual search (meaning that people used only their hands as a reference for selection) and the performance of different locations around the hand.

4.6.1 Expert Selections

During the trials, we recorded the number of selections made without waiting 300ms for the menu icons to appear (i.e., "expert selection"). As shown in Figure 37, for both types of HandMark menu, selection without feedback started near zero in the early blocks, but increased to approximately 8% of selections in the final block. Overall, use of the expert mode was low for both types of HandMark menus; although the expert-mode rate may change if people had more practice, it was clear that people did not quickly transition to feedback-free use. One reason may be that during the 300ms delay, no visual information at all was shown (e.g., the outline of the 4x5 grid, or the outlines of the icon regions); in other studies (e.g., the FastTap research), participants always had the gridlines as a general guide to the icon locations.

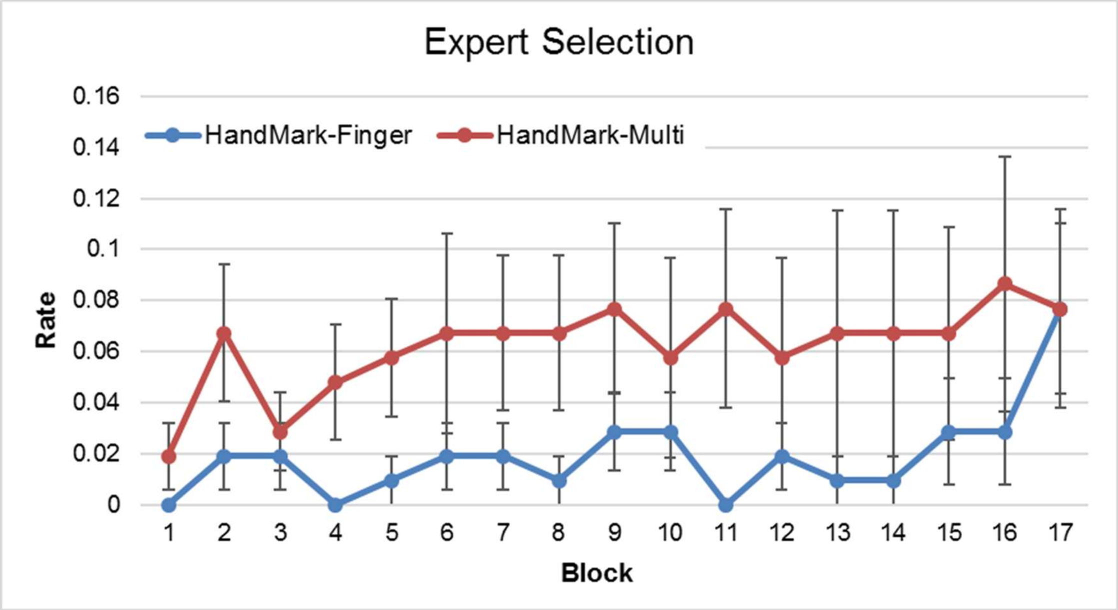


Figure 37: Expert selections in HandMark menus.

The experimenter's informal visual observations provide follow-up information that suggests that participants were making use of the hands as landmarks, even if they were not making use of the expert mode. The observations showed that all users moved their selection finger towards the correct region on the menu hand even before the menu hand was placed. That is, even when people did wait for system feedback, they were preparing for a correct selection by correctly positioning their finger before the menu was displayed. These preparatory actions suggest that people were developing proprioceptive memory and were remembering the mapping of commands to hand locations.

4.6.2 Performance by Target Location

We also analyzed selection time and expert mode use by target location. For HandMark-Finger, three areas were defined: finger-top, between-fingers and large area near-thumb (see Figure 38). RM-ANOVA showed finger-top locations were better than others with 0.03 expert-selections/command, s.d. 0.12 ($F_{2,24}$=1.41, p=.26) and faster command selection (mean 2.3s, s.d. 0.1) ($F_{2,24}$=0.74, p=.5).

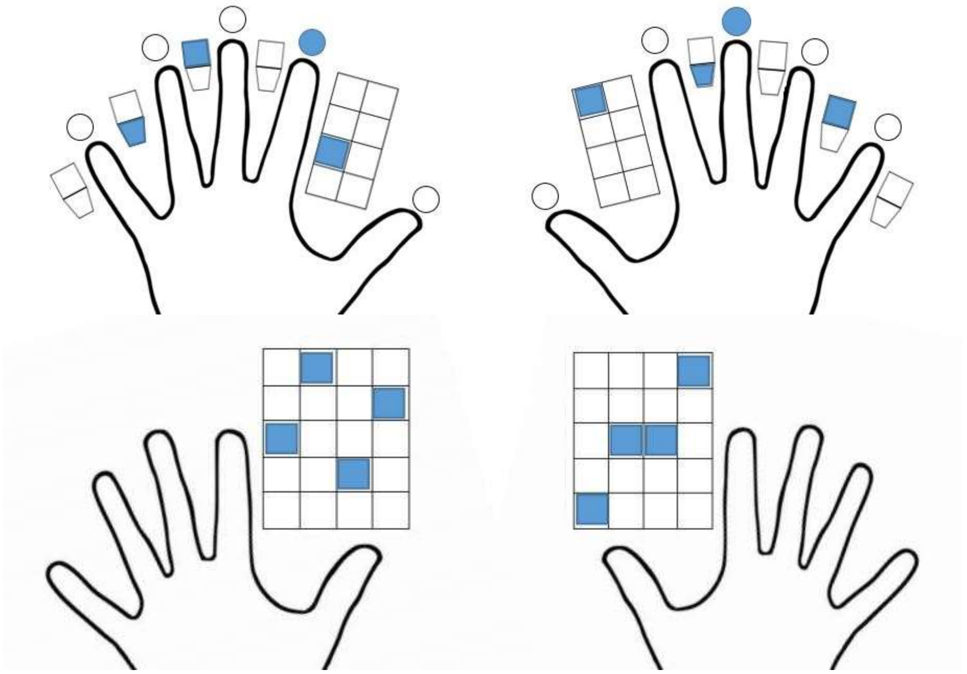


Figure 38: Target locations: (top) HandMark-Finger and (bottom) HandMark-Multi.

For HandMark-Multi, two areas were defined: close and far from index and thumb (Figure 38). Here, the targets located close to index and thumb performed best – for these better-landmarked locations, selections were faster and expert mode was used more (0.08 selections/command, s.d. 0.16) with a significant main effect ($F_{1,12}$=6.67, p<.05).

4.7 PARTICIPANT COMMENTS

Participant comments followed the pattern of preference results. Participants made several comments on how spatial stability and quick activation using both hands helped the speed of HandMark menus: one participant said *"Really neat technique that allows you to browse through different tabs based on the number of fingers."* One person, however, remarked on the difficulty of remembering the different hands' sets: *"It was difficult to remember which hand has the right kind of tool."*

Other comments suggested that the HandMark interfaces helped participants to learn command locations: one said *"It was easier to remember where icons were relative to spaces on fingers* *instead of just on the tabs."* Another said "*[HandMark was] easier to use and faster to remember,*" and another stated "*[Tabs were] more difficult to memorize."*

Some participants stated that they were initially concerned with slow memorization in HandMark-Multi, but eventually preferred it. One person stated "*Remembering was slightly slower at early stage but in a short amount of time it became quite strong and it became easier to answer."* This participant also stated that the Tab interface *"was fast at the beginning but it could not build up strong memory and hence it became tough to apply them later."*

4.8 INTERPRETATION

Our first study of HandMark menus provides four main results:

- HandMark-Finger was significantly faster than Tabs-2 (0.62s/selection), and was faster in all blocks.
- HandMark-Multi was slower overall than Tabs-8 (0.62s per selection), but only in the early blocks.
- The only difference in errors between the two approaches was that HM-Multi had a slightly higher rate than Tabs-8 (although the difference was not significant).
- There were few significant differences for perceived effort between interfaces in either pair, and where there were differences they favoured HandMarks; in addition, most participants preferred both HandMark menu types.

Here we interpret the findings of the experiment along with their explanations.

Performance analysis of HandMark menus

The study showed that both HandMark menus were faster than a visually-guided tab menu, though HandMark-Multi was slower during early use. There are a few reasons for these results, based on the command selection process for both novice and expert.

At the beginning, novice users had no idea of commands' locations in the menu, and they had to use visual search to find the commands. Novices often browsed through all the tabs available in the interface, even though menu items were grouped by type or color. This was also true for HandMark menus, but experts in this technique could remember command's location with reference to their own hand and fingers as commands were appearing around them. However, this rich landmarking facility was not available in standard tabs. Particularly with HandMark-Finger, experts seemed to be able to use their knowledge of their hands and fingers to remember commands and were able to perform quicker selection.

For HandMark-Multi, displaying a command set involves different combinations of fingers. With eight sets, there were eight different combinations of fingers. Novices spent a large amount of time determining which finger combination belonged to which set; in contrast, Tabs-8 showed names and specific positions for each tab. Even though novices also had to spend time searching through the different tabs, the visual presentation allowed people to better organize their task. This may have caused HandMark-Multi's initial slow performance. However, as people became more experienced, these differences disappeared.

In addition, the amount of time taken for reaching to the tabs (at the top edge of the display) was a substantial component of the overall performance of the Tabs technique. This is not the case for HandMark menus, since these are always invoked close to the user. Therefore, the performance difference between a tab interface and HandMarks will to some degree depend on where the tabs are located – as tabs are located further away, the performance advantage for any in-place technique will increase. We further investigate the effect of this issue by comparing HandMarks to a similar in-place menu in Chapter Five.

Error rates with HandMark

Error rates per command were high in all the techniques: 4% for both HandMark-Finger and Tabs-2; 6% and 4% for HandMark-Multi and Tabs-8. This high error rate might be an artifact of our experimental protocol, which instructed participants to select commands quickly, and noted that errors could be corrected afterwards. There are other possible explanations for the error rates, however. First, the quick execution of a selection in all the interfaces may have encouraged participants to view errors as amenable to rapid correction, thereby encouraging users towards a ‘guess and correct’ mode of operation [47].

Second, it is possible that people’s memory of a command’s spatial location was imperfect, and so participants may have experienced ‘near misses’ more often with larger sets (HandMark-Multi and Tabs-8). Third, it is very easy to touch down with fingers on touch surface but, it is somewhat more difficult to change the combination of fingers very quickly, which sometimes caused unintentional touches for HandMark-Multi. Last, for both Tabs-2 and Tabs-8, the oblique viewing angle may also have increased errors. Further work is needed to explore these sources of error, and to determine whether the high error rates for both techniques occur in real-world use.

Incorrect set invocation with HandMarks

Incorrect command-set selections were also relatively high for all the interfaces: 0.05 sets/command for both HandMark-Finger and Tabs-2; 0.64 and 0.18 tabs/command for HandMark-Multi and Tabs-8. As the number of tabs was smaller for the first pair, participants made fewer errors; in the larger menus, the rate was considerably higher. This can be explained by the larger number of items, and the increased need for visual search overall. As described above, the visual representation of the tabs in the standard interface may have allowed participants to better organize their visual search, whereas people’s search in HandMark-Multi was often poorly organized. This indicates one disadvantage of hand-centric interfaces – information such as the name of the set cannot be shown on the reference frame (i.e., the hand) in the same way that it can be for a screen-based technique like Tabs.

An additional reason for differences in set-selection errors is the physical position of fingers in HandMark-Multi. As the fingers are very close to one another, people sometimes touched the wrong finger onto the surface. More work is needed to evaluate the ergonomic and effort characteristics of different hand and finger combinations – it may be that a smaller number of menus (using only the easy-to-produce finger combinations) will improve browsing performance.

4.9 SUMMARY

In this chapter we considered design and performance questions related to bimanual HandMark menus for tables. We presented two versions of HandMark menus that use hands and fingers as landmarks to accommodate large command sets and facilitate fast command selection. First, the HandMark-Finger menu that supports 42 commands. Second, HandMark-Multi menu uses a different approach to accommodate 160 commands in two hands. Results of our study with these two menus show that both of them performed well in comparison to standard menu interfaces, and our observations suggested that the hands and fingers were being used as landmarks to support the development of spatial memory. In the next chapter, we continue our exploration of HandMarks with a different platform – handheld multi-touch tablets.

CHAPTER 5

SINGLE-HANDED HANDMARK MENUS FOR TABLETS

In this chapter[5] we continue our research to explore hands as landmarks for efficient selection performance while maintaining large number of commands by testing the technique in hand-held multi-touch tablets. The studies in the previous chapter showed that HandMark menus were as fast or faster than equal-capacity standard toolbar techniques, and that they were strongly preferred by participants. In addition, these studies provided early evidence that the benefits of the HandMark menus were associated with the value of the hand as a landmark.

However, there were limitations to the study described in Chapter 4. First, the techniques require bi-manual operation, which is feasible for tabletop systems but inappropriate for tablet use in mobile settings (where one hand is needed to hold the device). Second, the previous studies did not strongly focus on the spatial learning that takes place with HandMark menus – they compared HandMark menus only to toolbars at the edge of a table, rather than pop-up menus that are invoked at the user's work location. Therefore, it is still unclear whether HandMarks are fast because of the hand landmarks, or because of other factors such as the simple proximity of the commands.

In this chapter we address these two limitations. We first present a new version of HandMark menus that works in a single-handed fashion, and is therefore appropriate for settings where a touch tablet is held in one hand and manipulated with the other. We then report on two studies: one that establishes a baseline for spatial learning with the one-handed versions of the technique,

---

[5] Material in this chapter first appeared in the following publications: [120].

and one that compares the two HandMark techniques with a pop-up menu that does not provide hand-based landmarks.

Our results provide several new findings that argue for the use of hands as landmarks in touch interfaces. Our baseline study shows that both types of HandMark menus work well in single-hand operation, and both allow quick development of spatial memory. Second, our comparison study shows that both of the HandMark menus were significantly faster than the limited-landmark popup menu (by more than 400ms per selection), and were strongly preferred by participants. In addition, the performance improvement arose from differences in the amount of time participants took to think about the spatial location of the item, not from the basic operation of the technique.

There are two main contributions of this chapter. First, we show that adapted HandMark menus are feasible selection techniques for mobile tablets that are fast and that have high capacity. Second, we provide additional evidence that using the hand as a landmark is feasible and effective for the development of spatial-memory based selection techniques, even with one-handed use.

5.1 DESIGN OF SINGLE-HANDED HANDMARK MENUS FOR TABLETS

Here we describe the two modified HandMark menu techniques for tablets. The key point that distinguishes tablets from tabletops is that only the dominant hand is used for active interactions with items in tablets while non dominant hand holds the device. Most of the earlier design factors presented in Chapter Four still apply for the new configuration. However, as the context is different, different selection techniques must be designed to make the techniques work with a single hand.

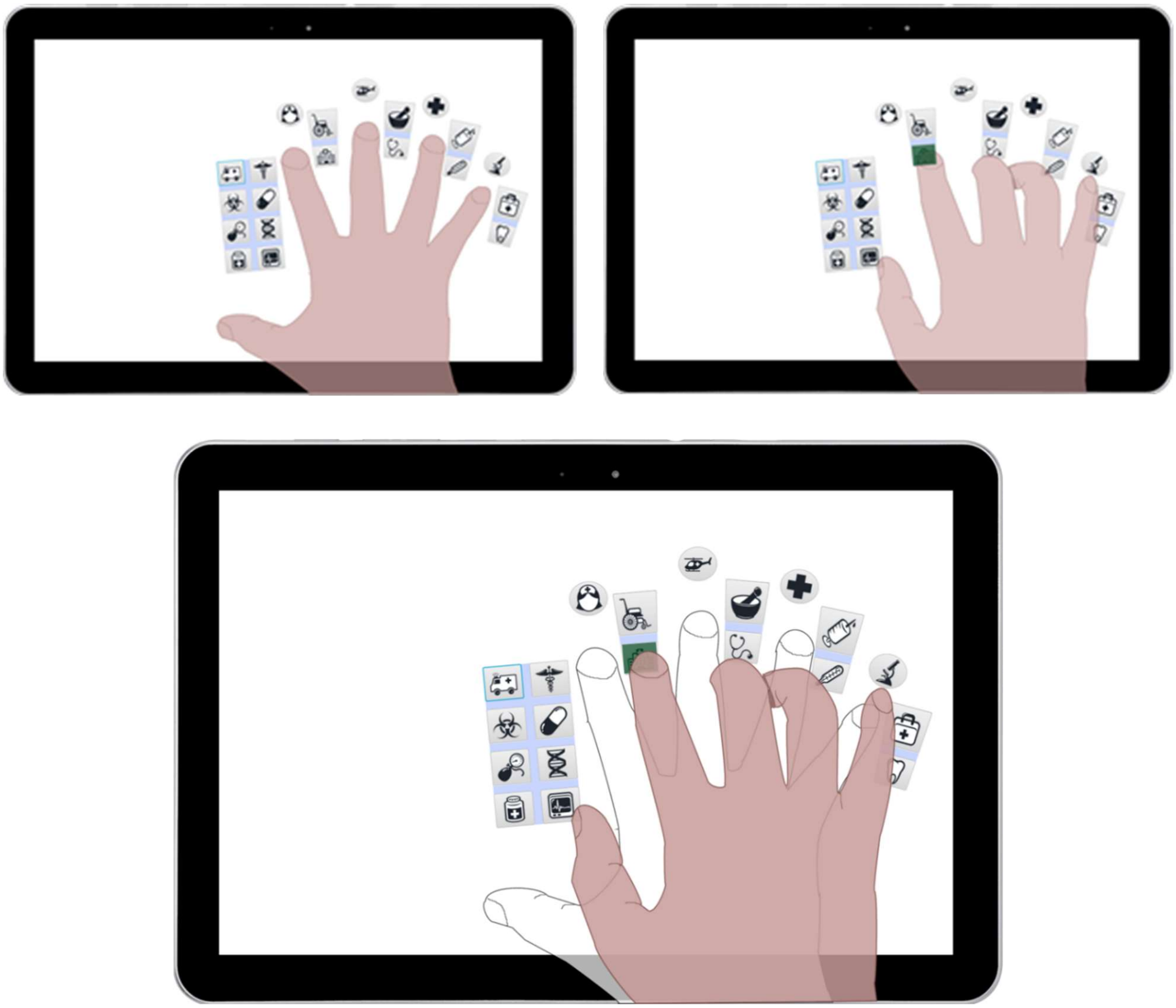


Figure 39: Single-handed HandMark-Finger menu: (left) menu invocation, (right) item selection after visual search, and (bottom) rapid selection with two sequential chunked actions.

5.1.1 HandMark-Finger for Tablets

Similar to the original version of HM-Finger for tables (see Section 4.1), this version of HM-Finger for tablets provides modal access to a set of commands (Figure 39). The need to hold a tablet with one hand means that we must we deviate from the bimanual technique and enable command selection using one hand only. In this single-handed HM technique, commands can be displayed by touching down all the fingers of one hand in any order, spreading the hand to provide space between the fingers. The user can rotate and move the menu in any direction by moving the fingers.

Unlike the earlier technique, however, command selection is carried out by lifting the fingers from the screen and touching the desired item with any finger. Command icons remain displayed until a selection is made (or until the menu is dismissed by touching on another part of the tablet).

As with the previous version, the spaces around the hand and between fingers are used to display commands (Figure 39). We place one command at the top of each finger (except the thumb), and pairs of commands between fingers. We utilize the large space between the thumb and index finger by placing eight commands in a 2×4 grid (20 items total).

5.1.2 HandMark-Multi for Tablets

We also modified the earlier HM-Multi technique (see Section 4.2) for use on tablets. HM-Multi uses a similar command selection mechanism as HM-Finger, but differs in the placement and number of commands. There are four sets of commands, invoked by placing a specific number of fingers (in addition to the index finger and thumb) on the screen in an L-shaped posture (Figure 40). For example, to invoke the second set of commands, the index and middle fingers of one hand along with the thumb are placed on the tablet. The menu follows the user's hand as it rotates or moves on the screen. HM-Multi uses a spatially-stable 4×5 grid to show commands in the space between thumb and index finger (Figure 40).

As shown in Figure 40 parts **a** and **d**-**f**, four different finger combinations can invoke four different item sets. In total, HM-Multi supports 80 items (4 tabs, and 20 items in each tab).

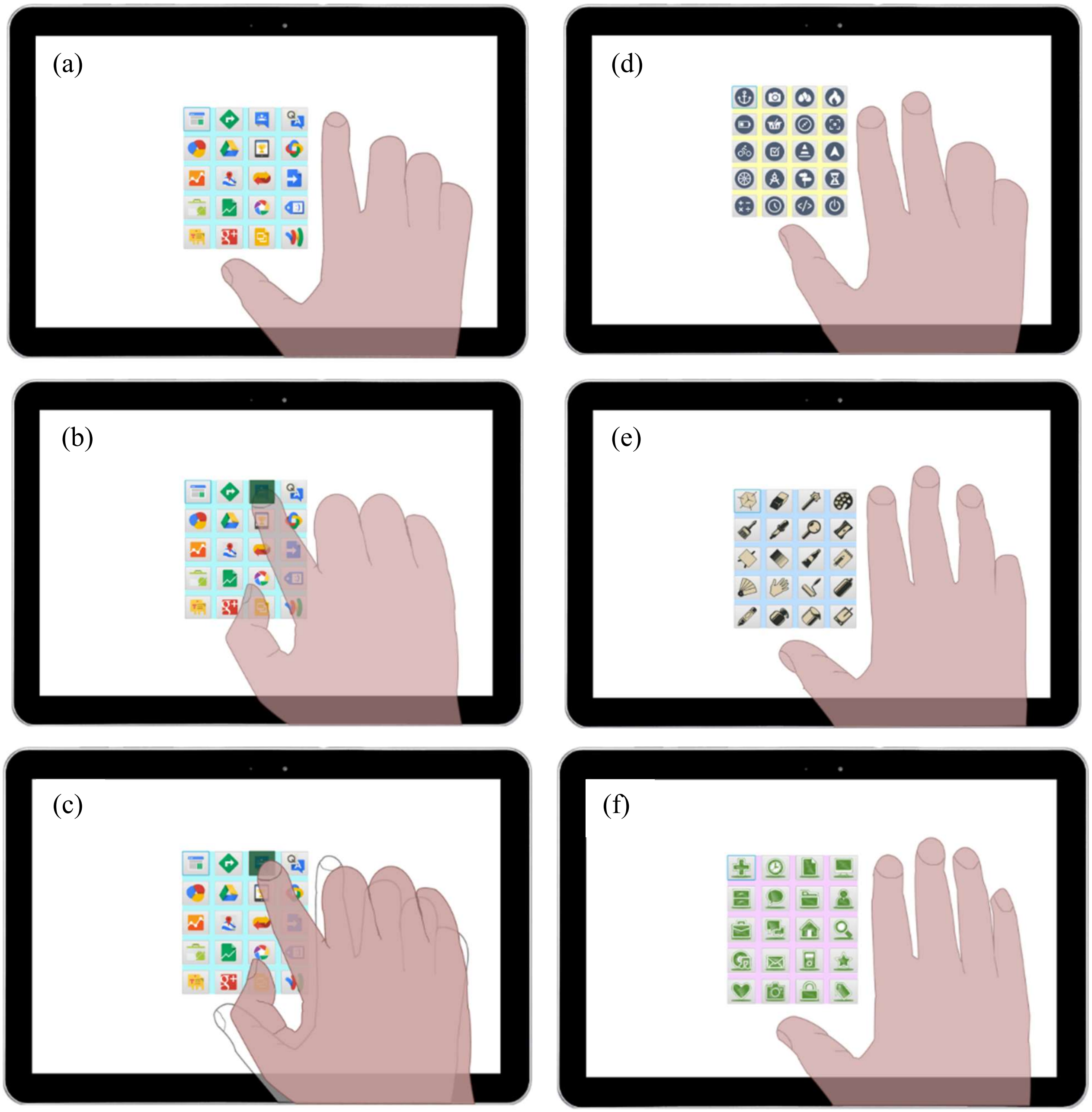


Figure 40: Single-handed HandMark-Multi menu: (a) invoking the 1st item set by placing index and thumb in ‘L’ shape posture, (b) item selection after visual search, (c) rapid selection with two sequential actions, and (d-f) invoking 2nd, 3rd, and 4th item sets with specific finger combinations respectively.

5.1.3 Enabling Rapid Execution

Spatially-stable organization of commands helps users to build up spatial memory about those commands, and allows command execution by remembering their associated locations rather than searching for a command in the interface [76]. HandMark menus (both Finger and Multi) work on the simple principle of accelerating a basic interaction process. However, instead of the bimanual process used for the original HandMark menus (invoke the menu with one hand, and select with the other), our adapted menus require the user to invoke and select with the same hand (by lifting and placing again; Figures 39 and 40). Even though our adapted menus require two sequential actions rather than a bimanual parallel action, we hypothesize that as users become familiar with commands' locations, the two separate steps (menu invocation and command selection) will be integrated into a single learned motor chunk [76]. Chunking of sequential actions is well known in operations such as double clicking, which is considered as a single action for expert users.

5.1.4 Identifying Hands and Fingers

Current multi-touch tablets typically support only ten touch points, and the operating system typically only reports these as (X,Y) locations [7]. For accurate identification of hand and fingers, therefore, we have to rely on the fingers' touch points. Chapter Three discussed how users' touch points can efficiently identify specific hand and fingers (see Section 3.2.2). For example, when the hand is pointing upwards, the lowermost position always belongs to the thumb, and the leftmost finger is the thumb of right hand (and reversed for left). We adapt this simple hand and finger detection approach here.

5.1.5 Use of Hands and Fingers as Landmarks

We designed our HandMark tablet techniques to maintain the use of hands and fingers as landmarks, as well as visual structures in the display of the menu items themselves. Both HM-Finger and HM-Multi provide visual guidance to assist command selection, including the user's hand, the display menu, the selected item, and the menu's gridlines. First, the techniques assign command sets to specific fingers. To invoke a command set, the HM-Finger technique requires all fingers to be touched down on the screen, while HM-Multi requires a specific finger combination.

Second, after invoking a command set, the commands for that menu are displayed on the screen as long as the menu fingers are touched down. Once the fingers are lifted, however, we set a timeout so that commands remain visible for 600ms. As the commands are shown in specific places around the hand, our hypothesis is that these techniques still allow users to remember locations of commands with reference to hand and fingers, even though the hand is lifted from the surface before the actual selection is made.

Third, when an item is selected after a visual search, it is displayed in its position with different color for 500ms. This visual feedback provides confirmation of selection [29, 62], which helps users to develop spatial memory.

Fourth, we place items in grids for the HM-Multi technique. Previous studies confirmed that grid marks are helpful for spatial memory development [47, 76]. We test people's ability to make selections based on landmarks (only the grid positions with reference to fingers are shown) in our studies, described below.

5.2 STUDY 5.1: LEARNING AND PERFORMANCE OF HANDMARK MENUS FOR TABLETS

Changing HandMark menus from a bimanual to a single-handed technique means that selections are not made with the hand as a permanent reference frame. Since this may compromise people's proprioceptive spatial memory, we carried out a study to determine the baseline performance for the two adapted versions of HandMark menus. We designed the study to answer three questions:

- Does spatial learning occur when HandMark operation uses two serial touch actions with the same hand?
- What is the speed and accuracy of the adapted menus?
- What is the effect of overlapping item positions from different command sets (in HandMark-Multi)?

5.2.1 Tasks and Stimulus

The study consisted of a series of trials, each involving the serial touch actions for both HandMark menus. In each trial, the participant pressed a start button, and then a stimulus icon appeared on the screen. The participant then invoked the HandMark menu and selected the item with any finger of the right hand (Figure 41). Command icons were organized into sets by color and visual style. Figures 41 and 42 show both single-handed HandMark menus in the experiment.

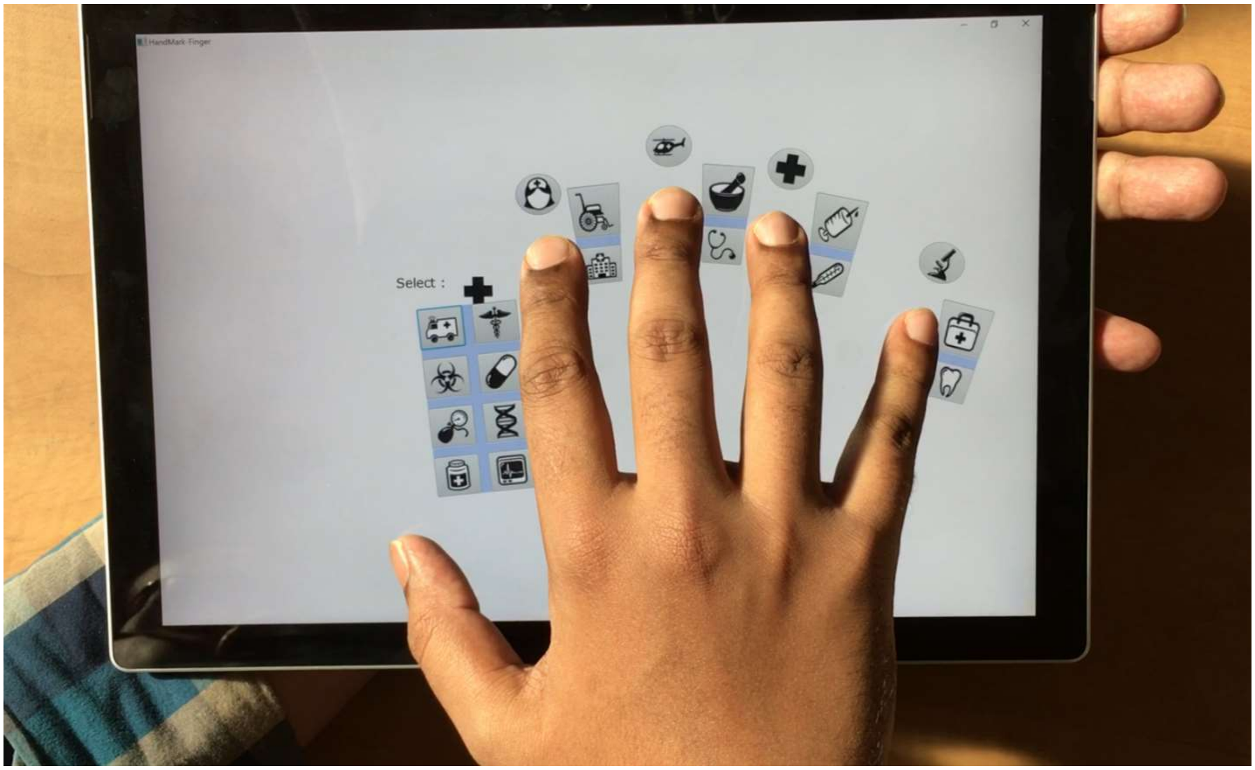


Figure 41: Single-handed HandMark-Finger in study.

HM-Finger shows 20 commands, of which six were used as study targets. HM-Multi shows four sets of 20 commands (in 4×5 grids); twelve of the 80 commands were used as targets (three from each set).

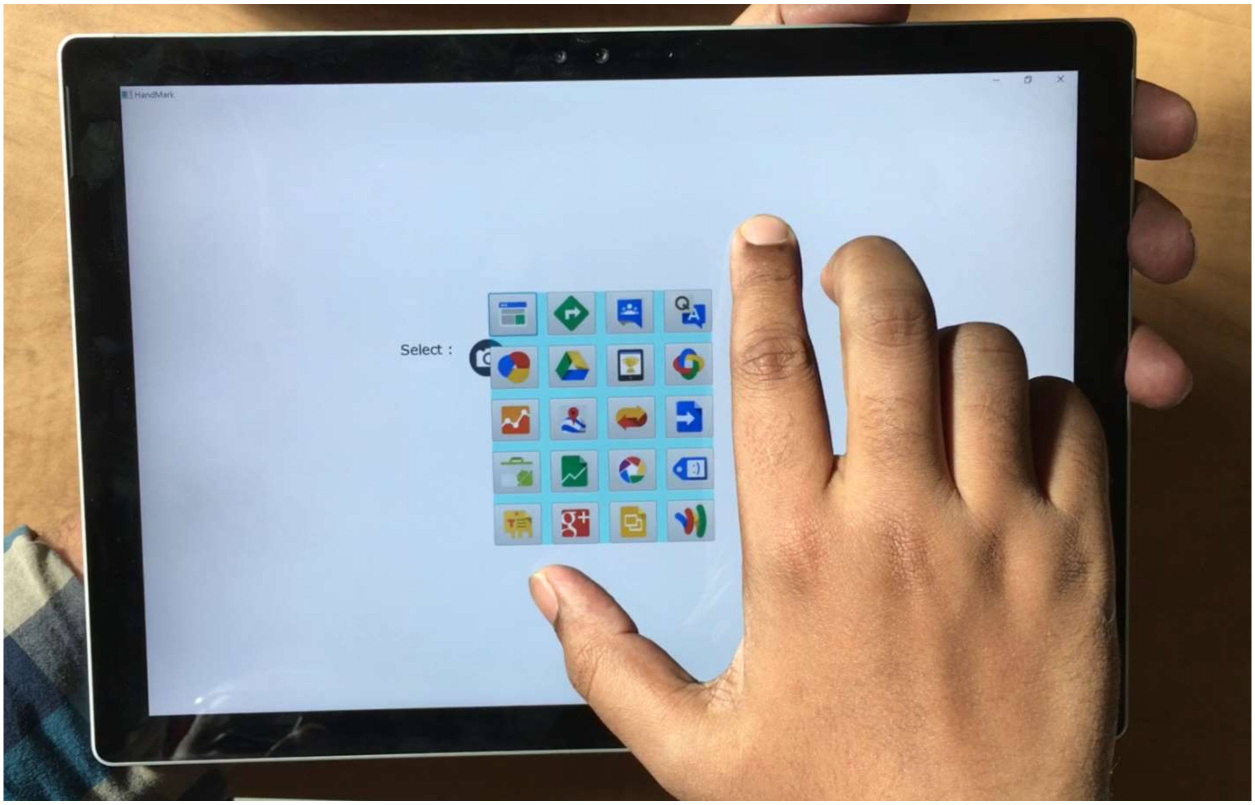


Figure 42: Single-handed HandMark-Multi in study.

5.2.2 Procedure and Study Design

The study used a within-participants design, with order of menu counterbalanced. The menus were introduced to participants and they performed 36 sample selections using HM-Finger and 54 for HM-Multi. Participants then completed several blocks of trials with the same target items (random order, sampling without replacement).

The 15 blocks were grouped into three stages. At the end of each stage, participants performed a "blind" memory test with no feedback, where only the grid lines were visible, but not the icons. The aim of these blind blocks was to test the participant's spatial memory of the items. After each stage, participants were allowed to rest, and after finishing all the stages of each interface they completed a NASA-TLX [49] questionnaire (see Appendix).

Each selection was confirmed by visual feedback – changing the button background to green for correct, and red for incorrect. Incorrect selections could be corrected by simply selecting another item (except in the blind memory tests). Participants were instructed to complete trials as quickly and accurately as possible. For each trial, we recorded task completion time, errors, the number of incorrect sets opened (for HM-Multi only), and data describing individual touches.

5.2.3 Participants

We recruited 20 people (19 right-handed, and 1 ambidextrous) from the University of Saskatchewan campus. We could not collect one person's data due to technical difficulties, leaving 19 participants (10 males, 9 females), ages 19-40 (mean 25.7). Most of the participants were students. The same participants also took part in the second and third studies discussed below. The three studies took ~60 minutes in total, and a $10 remuneration was paid to each participant. Eleven participants reported of owning and regularly using a tablet (> 10 hours per week).

5.2.4 Apparatus

The experiment was conducted on a Microsoft Surface Pro 4 tablet, with a 12.3-inch multi-touch 2736 x 1824 screen, and running the Windows 10 operating system. The interfaces were written in JavaFX. During the studies, we removed the physical keyboard, and participants held the tablet in landscape mode with their left hand and operated the system with their right hand.

5.3 RESULTS: STUDY 5.1

Here we present completion time and error rates for both versions of the adapted HandMark menus. We analyzed the 15 feedback-enabled blocks and the three 'blind' memory-test blocks separately.

5.3.1 Selection Performance

In this study, we did not compare the two versions of adapted HandMark menus, because the versions provide very different numbers of commands (20 vs. 80). So, we analyzed the data of HM-Finger and HM-Multi techniques separately.

HandMark-Finger

Average trial completion times for HM-Finger are shown in Figure 43. For the 15 feedback blocks, mean completion time was 2187ms (s.d. 1228ms); for the blind blocks, mean time was 2531ms (s.d. 1600ms). RM-ANOVA showed significant effect of *block* on completion time for both the feedback blocks ($F_{14,252}$=8.52, p<.0001) and for the blind blocks ($F_{2,36}$=5.08, p<.0001). Figure 43 clearly shows that trial completion time decreased significantly. In addition, completion time for the final memory-test block was similar to that of the feedback blocks (1848ms).

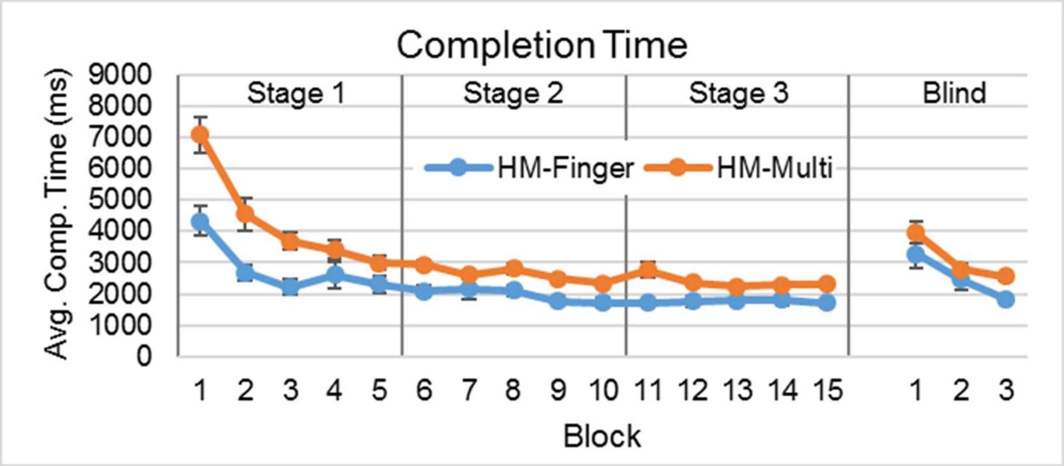


Figure 43: Average trial completion time by method and block.

HandMark-Multi

Mean completion times for HM-Multi also decreased significantly for the feedback blocks ($F_{14,252}$=29.61, p<.0001) and the blind blocks ($F_{2,36}$=19.62, p<.0001) (see Figure 43). Average completion time with feedback was 3127ms (s.d. 1648ms).

Performance with both techniques closely followed the power law of learning: fitting power-law curves to the data in Figure 43 gives R-squared values of 0.88 for HM-Finger, and of 0.94 for HM-Multi. This correspondence suggests that participants were quickly developing spatial memory.

5.3.2 Error Rates: HM-Finger and HM-Multi

We analyzed errors per trial by tracking incorrect selections for both techniques. Interestingly, participants made zero errors in the feedback blocks with HM-Finger. For blind blocks, mean errors per trial for HM-Finger were 0.28 in the first block, 0.05 in the second, and 0.03 in the third (overall 0.12 errors/trial, s.d. 0.21). RM-ANOVA showed a significant effect of *block* on errors ($F_{2,36}$=12.94, *p*<.0001). Figure 44 shows that in the blind blocks, errors rates decreased significantly across the block.

Figure 44: Error rates by method and block.

HM-Multi had a slightly higher error rate in feedback blocks (mean 0.02 errors/trial, s.d. 0.04), but with no significant effect of block ($F_{14,252}$=0.63, *p*=.84). However, blind trials showed (see Figure 44) a significant effect of *block* ($F_{2,36}$=27.64, *p*<.0001). Means were 0.43 for the first block, 0.2 for the second, and 0.11 for the third (overall 0.25 errors/trial, s.d. 0.23).

5.3.3 Impact of Overlapping Item Positions in HM-Multi

We analyzed selection errors in HM-Multi to consider overlapping target positions (i.e., targets that were in the same grid location but different sets). Four of the 12 targets overlapped. Figure 45 shows the command locations used in the HM-Multi interface including the overlapped locations.

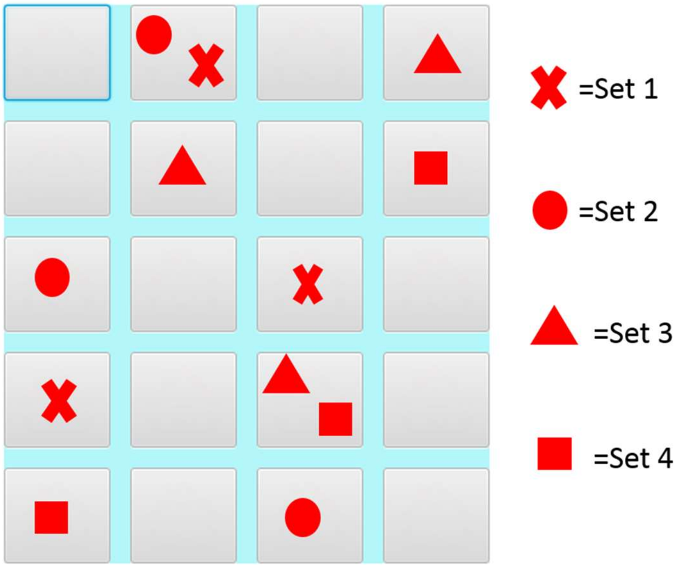


Figure 45: Command locations used in experiment (all set).

We analyzed selections from the final stage (blocks 11-15) and last blind block (Table 6). Most errors occurred by tapping the wrong location in a correct menu (2.26% of selections). There were zero errors with the correct position but in the wrong menu, suggesting that overlapping targets are not a major source of errors.

Table 6: Error analysis for HM-Multi in few final blocks.

Menu	Pos.	No Overlap	Overlap	Overall
Correct	Correct	896 (96.34%)	447 (97.17%)	1343 (96.62%)
Correct	Wrong	22 (2.3%)	8 (1.74%)	30 (2.26%)
Wrong	Correct	0	0	0
Wrong	Wrong	12 (1.29%)	5 (1.09%)	17 (1.22%)

5.3.4 Incorrect Set Selections: HM-Multi

There were four command sets in HM-Multi, each assigned to specific combinations of fingers. We recorded the number of times participants invoked an incorrect command set. For the 15 feedback, RM-ANOVA showed a significant effect of *block* on incorrect set selection ($F_{14,252}$=28.27, p<.0001) with 0.19 sets/trial (s.d. 0.34). Figure 46 shows that incorrect set selections reduced substantially by block three. The effect of *block* was not significant for blind trials ($F_{2,26}$=0.27, p=.77).

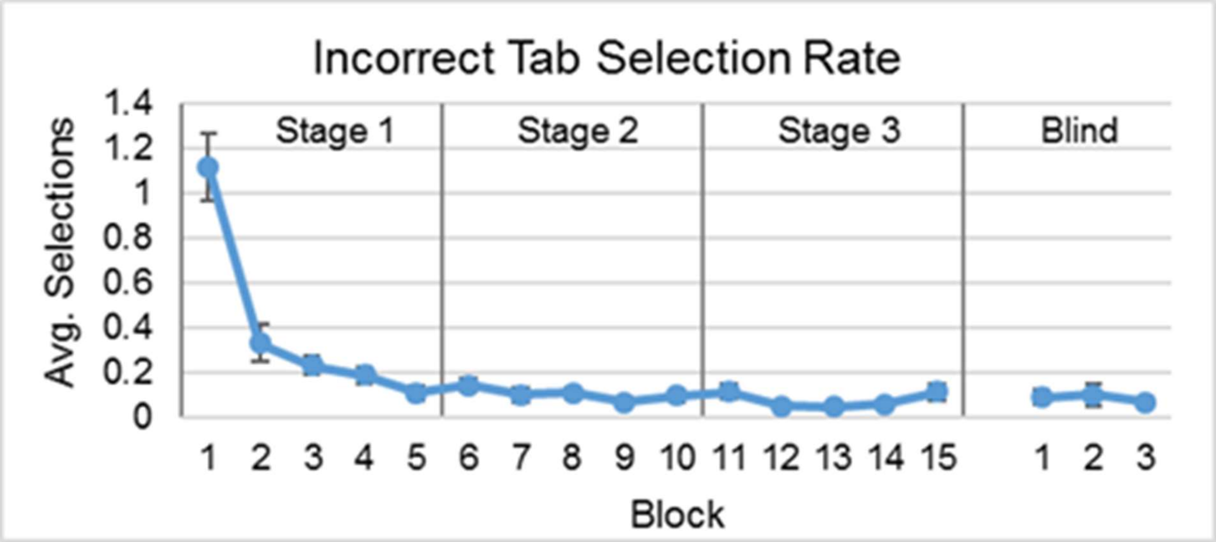


Figure 46: Incorrect tab selection rate by block.

5.3.5 Subjective responses

Participants' gave positive responses for both HandMark interfaces in NASA-TLX scores (see Table 7). We did not perform a statistical comparison on these data (because of the large difference in the number of items in the two interface), but it is clear that participants saw the HM-Multi technique as requiring additional effort – likely because of the increased capacity.

Table 7: Mean (s.d.) effort scores (0-10 scale, low to high).

HM-Finger	Questions	HM-Multi
3.94(2.37)	**Mental**	5.52(2.44)
3.68(2.58)	**Physical**	4.58(2.69)
3.47(2.14)	**Temporal**	4.68(2.14)
8.21(1.13)	**Performance**	7.00(1.6)
4.89(2.31)	**Effort**	6.16(2.27)
2.11(2.02)	**Frustration**	3.00(2.4)

5.3.6 Summary

In summary, our study showed that people were able to quickly and correctly learn both types of HandMark menus, even with single-handed use and serial (rather than bimanual) operation. Selection errors were very low, and overlapping targets did not cause additional errors.

5.4 STUDY 5.2: IMPACT OF LANDMARKS ON PERFORMANCE

Previous studies of HandMarks provide only limited evidence about the value of the hands as landmarks for anchoring spatial memory. In this study, we compared HM-Finger and HM-Multi against a version of the technique that does not strongly orient the grid menu to the position of the hand (the menu is posted underneath the user's hand). We were interested in which version of HandMark menus would perform best, and whether the two versions that are oriented towards the location of the hand performed better.

5.4.1 Study Method

This study tested three menu systems – HM-Finger, HM-Multi-1Tab (with only one tab of 20 items), and a limited-landmark variant called HM-Under. We restricted the HM-Multi technique to one tab so as to equal the number of commands available in HM-Finger. In this variant, all five fingers of the hand were required to display the menu (again, to equalize the invocation to that of HM-Finger). HM-Under was similar, with one set of commands in a 4×5 grid, but the menu was posted beneath the user's hand so that there was no clear association between the visual image of the hand and the location of items in the grid (Figure 47). We note, however, that this technique had a shorter average distance to commands than the other techniques.

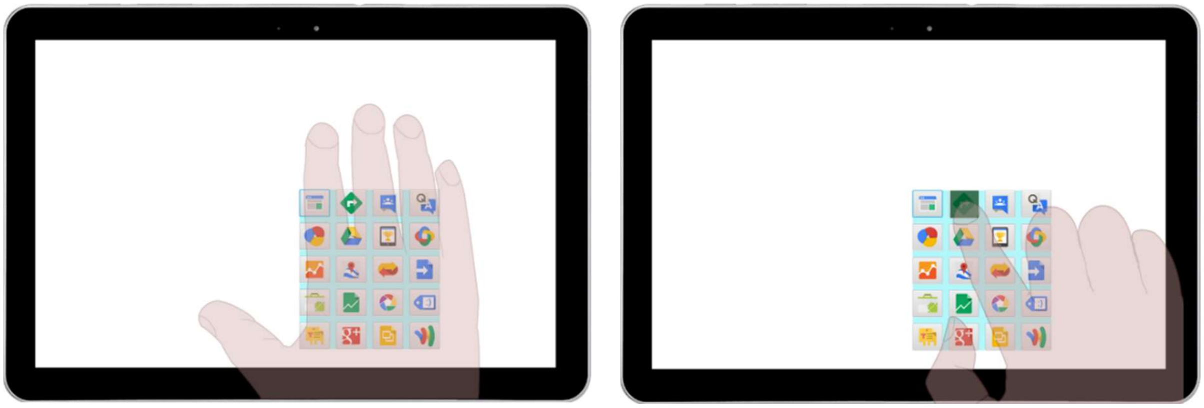


Figure 47: HandMark-Under menu: menu invocation (left), and item selection (right).

5.4.2 Tasks and Stimulus

Similar to Study 5.1 discussed above (Section 5.3), each participant performed a series of selection trials with each of the three menu systems. In each trial, a stimulus item was displayed on the screen, and participants selected the corresponding item to complete the trial. Correct and incorrect selections were confirmed with color feedback. For incorrect selection, the trial continued until the correct item was selected. To avoid learning effects, we used new command icons for this study.

5.4.3 Procedure and Study Design

The study used a within-participants design. Each menu consisted of 18 blocks of trials (using 6 items as targets), divided into three stages (5 blocks of regular trials, followed by one blind block), as described above. Participants went through a practice session before starting each menu.

The order of the menu system was counterbalanced, and items of each block appeared in randomized fashion. At the end of each menu, participants took a break and filled out a NASA-TLX [49] questionnaire; after finishing all three menus, they gave their overall preferences.

5.4.4 Participants and Apparatus

All 19 participants from our Study 5.1 (see Section 5.3) participated in this study as well, and the study was conducted with the same multi-touch tablet.

5.5 RESULTS: STUDY 5.2

Here we present the results of the comparative analysis of the second study with three command selection techniques.

5.5.1 Selection Performance

We analyzed average trial completion time for the 15 feedback blocks and 3 blind blocks separately. For feedback blocks, HM-Multi-1Tab performed best (mean 1776ms, s.d. 1048ms) compared to HM-Finger (1960ms, s.d. 1182ms) and HM-Under (2365ms, s.d. 2257ms) (see Figure 48).

RM-ANOVA showed a main effect of *interface* ($F_{2,36}$=3.26, p=.05). As shown in Figure 48, completion times decreased across trial blocks for all the interfaces; RM-ANOVA showed a significant effect of *block* ($F_{14,252}$=19.16, p<.0001), and significant interaction effect between *interface* and *block* ($F_{28,504}$=2.03, p=.002). Performance with HM-Under was not as consistent as the other two methods, and was slower overall. We suggest that because the menu was partially occluded by the fingers and hand, users could not use the hand as a landmark to aid development of spatial memory.

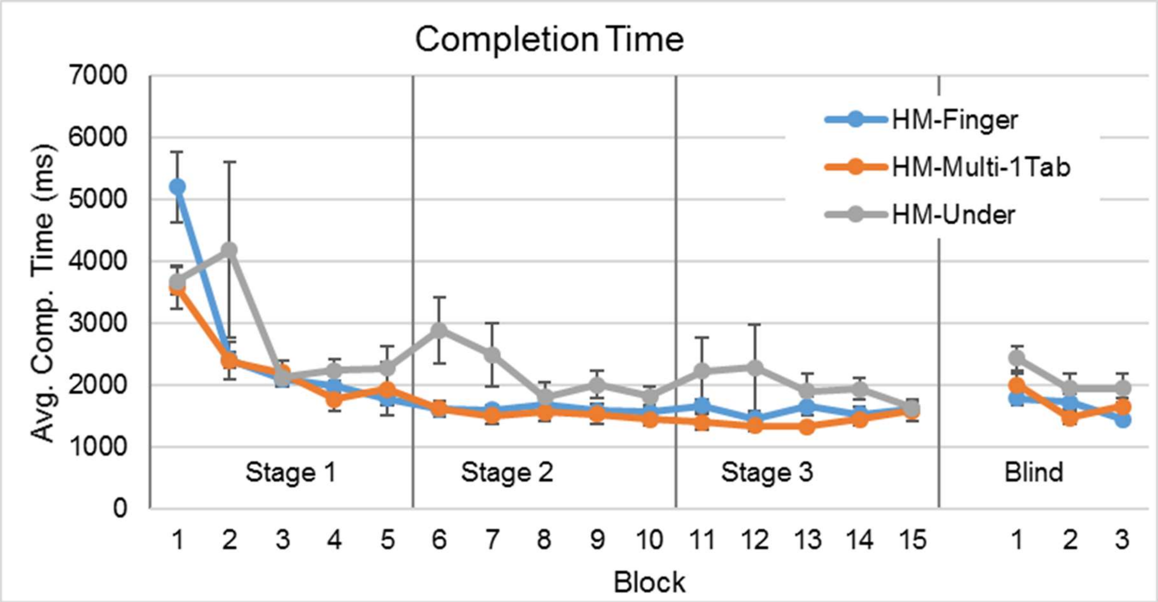


Figure 48: Average trial completion time by method and block.

We explored the issue of spatial-memory development for the three different techniques by again fitting the data to power-law curves corresponding to the classical power law of learning. The R-squared values were 0.90 for HM-Multi, 0.78 for HM-Finger, but only 0.64 for HM-Under – suggesting that participants were not able to learn item locations as well in the limited-landmarked menu.

For blind blocks, HM-Finger performed best (1653ms (s.d. 464ms), and the effect of *interface* was significant ($F_{2,36}$=5.4, p=.009). RM-ANOVA also showed significant effect of *block* ($F_{2,36}$=10.17, p<.0001), but no interaction between *block* and *interface*.

We further analyzed performance by calculating the time from invocation of the menu to the selection (i.e., the time from stimulus appearance to menu invocation was removed). RM-ANOVA showed a significant effect of *interface* ($F_{2,36}$=8.19, p=.001); as shown in Figure 49, HM-Multi-1Tab was the fastest technique (mean 735ms, s.d. 390). HM-Finger was slow at the beginning (when users need to visually search for items) but by the second block performance with this technique was similar to the others.

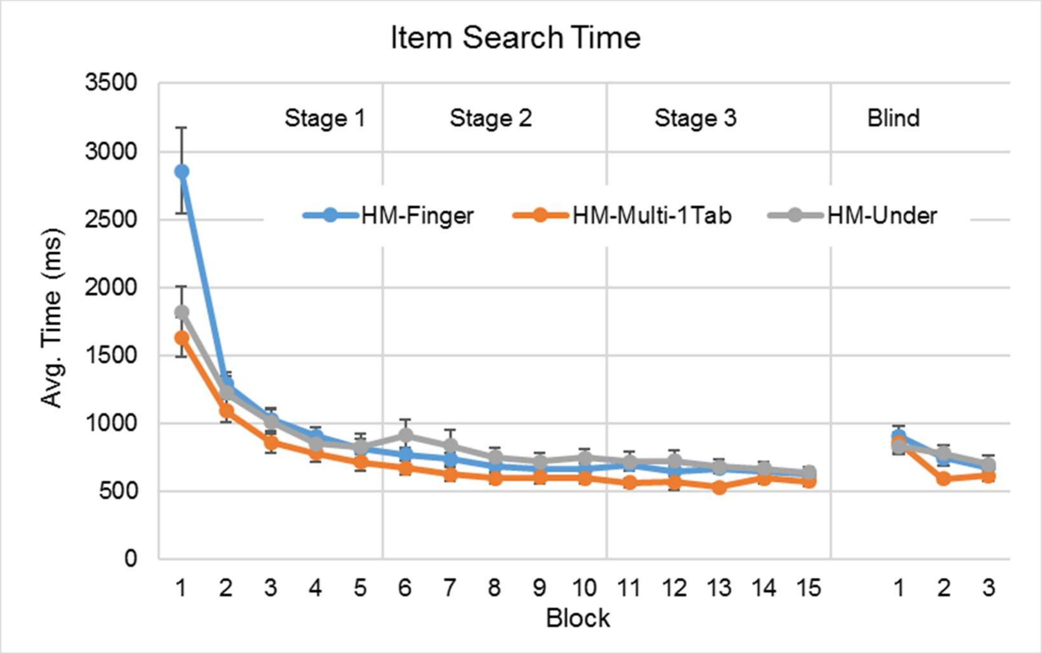


Figure 49: Average item search time by method and block.

This analysis shows that the additional time needed for the HM-Under technique (shown in Figure 48) arises primarily between stimulus appearance and menu invocation – and again we suggest that this indicates that users were having more difficulty remembering where the item was before starting the process of selection.

5.5.2 Error Rates

Participants made zero errors in the feedback blocks. For the blind blocks, RM-ANOVA showed a similar pattern to Study 5.1: HM-Finger had 0.053 errors/trial, s.d. 0.1, HM-Multi-1Tab had 0.099 errors/trial, s.d. 0.19, and HM-Under had 0.11 errors/trial, s.d. 0.16, with no main effect of *interface* ($F_{2,36}$=2.44, p=.101). There was a significant effect of *block* ($F_{2,36}$=17.73, p<.0001), and a significant interaction between *interface* and *block* ($F_{4,72}$=2.94, p=.026). Figure 50 also shows that error rates decreased significantly over the blind blocks. These results also suggest that the presence of the hand as a landmark made the HM-Finger technique in particular less error-prone.

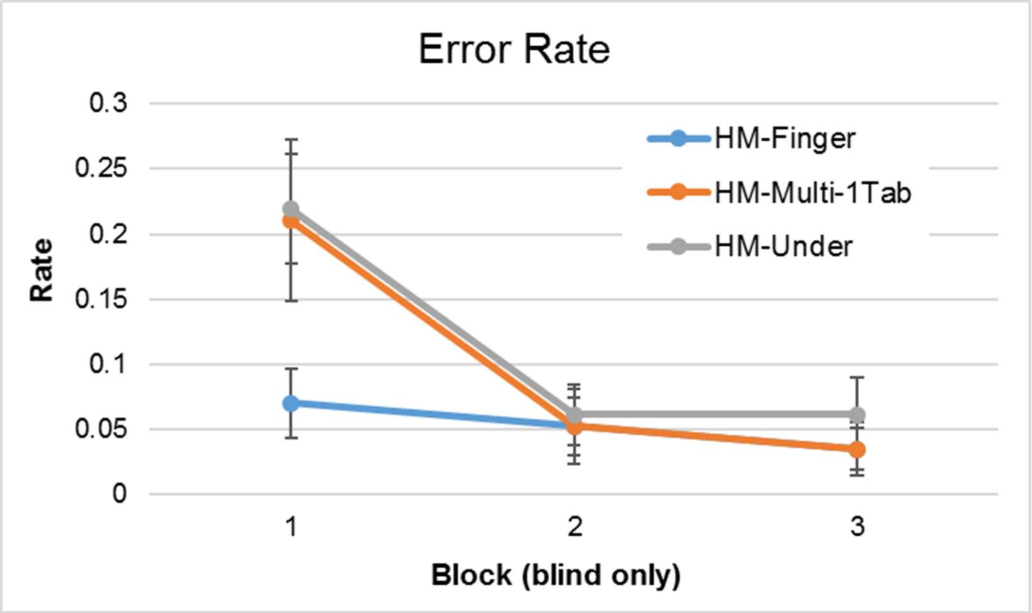


Figure 50: Error rates by method and block (blind only).

5.5.3 Subjective Responses

Friedman tests on NASA-TLX responses showed several significant differences (see Table 8) in physical demand required, perceived performance, effort required, and level of frustration. In all cases, HM-Finger and HM-Multi-1Tab scored better than HM-Under.

Table 8: Mean (s.d.) effort scores (0-10 scale, low to high).

	HM-Finger	**HM-Multi-1Tab**	**HM-Under**	χ_r^2	***p***
Mental	3.47(2.04)	3.21(2.07)	4.0(2.65)	3.18	.2
Physical	3.79(2.88)	3.63(2.77)	5.21(2.74)	8.71	**.01**
Temporal	3.53(2.06)	3.37(1.92)	4.16(2.22)	5.08	.08
Performance	8.32(0.95)	8.53(1.17)	7.32(1.67)	9.03	**.01**
Effort	4.32(2.83)	4.68(3.13)	5.47(2.82)	8.84	**.01**
Frustration	1.58(1.8)	1.58(2.01)	3.21(2.66)	10.45	**.01**

Preference counts (see Table 9) showed that nearly all participants preferred either the HM-Finger or HM-Multi techniques on five different criteria. Only one participant of the 19 preferred HM-Under on any measure.

Table 9: Count of participant preferences.

	HM-Finger	**HM-Multi-1Tab**	**HM-Under**	**None**
Speed	8	10	0	1
Accuracy	8	8	0	3
Memorization	9	6	0	4
Comfort	8	10	1	0
Overall	11	8	0	0

5.5.4 Participant Comments

Participants preferred the HM-Finger and HM-Multi-1Tab almost equally, and these preferences were also reflected in the comments provided by them after completing the study. HM-Finger method had more landmarks, allowing more opportunity to use proprioception: as one participant said, *"I can see every icon easily in the gaps between my fingers, and it requires an easy hand gesture."* Another person commented how their fingers facilitated memorization: *"I could use their location as reference."*

Others preferred HM-Multi as it required less visual search. One participant said *"everything was packed into one spot so I had to look only one spot."* Others commented about the occlusion problem with other methods: one stated that in HM-Finger *"I had to stretch my fingers apart to see all the icons."* Another participant commented that with HM-Under, *"I had less confidence in the location of icons."* Finally, all participants indicated that they used the hand and fingers as landmarks for all the three methods, even if the icons were partially occluded (HM-Finger and HM-Under).

5.6 STUDY 5.3: OVERLOADING SPATIAL MEMORY

When memory-based techniques are used in real life, there is a possibility that one application's command associations could interfere with another's commands. For single-handed HandMark menus, because participants learned one set of icons for Study 5.1 (see Section 5.2), and a different set for Study 5.2 (see Section 5.4), we saw an opportunity to test this real-world issue.

Figure 51 shows the command set we used in Study 5.1, and Figure 52 shows the command set we used in Study 5.2.

Figure 51: Command set of single-handed HM-Finger Study 5.1.

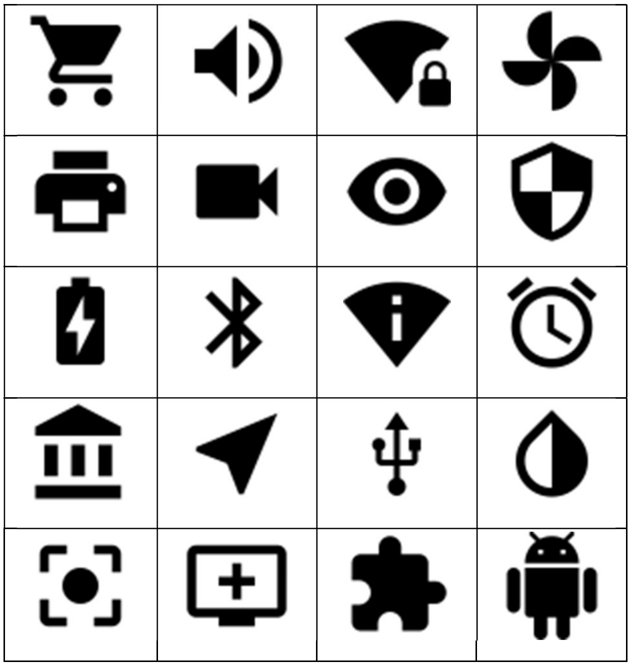


Figure 52: Command set of single-handed HM-Finger Study 5.2.

5.6.1 Study Method

As a final test at the end of the experiment, we asked our 19 participants to perform one final 'blind' memory test using all of the targets from both Study 5.1 and Study 5.2 (HM-Finger version). The 12 targets were used as stimuli in random order for the memory test. In this study, the menus items were not visible to participants (see Figure 53); they solely had to rely on their spatial memories that were developed while using our study systems in Study 5.1 and Study 5.2 with the single-handed HandMark-Finger menu. There were two locations which overlapped between the two sets. Figure 53 shows the memory test interface with those two overlapped locations.

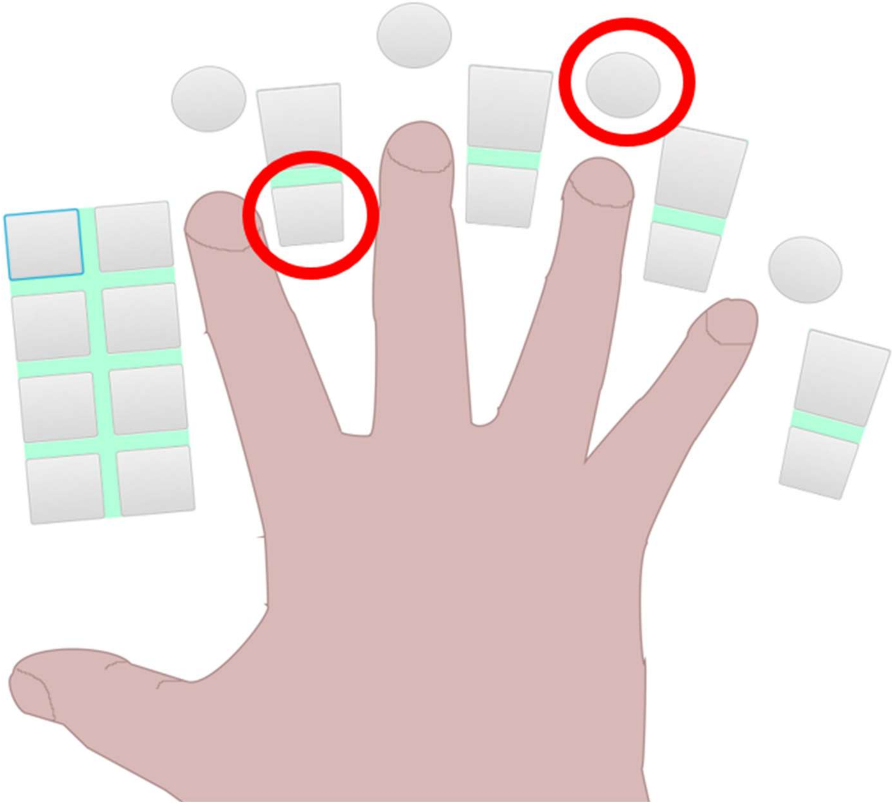


Figure 53: Memory test interface with overlapped locations.

During the study, participants were allowed to execute command selection only once per stimuli regardless of the accuracy. In single-handed HandMark-Finger menu of Study 5.1, participants learnt command locations with reference to different fingers. However, same locations were used to show different set of commands in single-handed HM-Finger menu of Study 5.2. In this study, we were interested in whether participants' memory of the Study 5.1 command locations decayed after learning the locations in Study 5.2.

5.7 RESULTS: STUDY 5.3

As shown in Table 10, participants were able to recall a large majority of both command sets. However, items from Study 5.1 were recalled less accurately (72% overall) than items from Study 5.2 (91% overall). These results do not indicate whether the reduced accuracy is because of interference or simply the passage of time (participants had not used the Study 5.1 icons for about 20 minutes, whereas they had just used the Study 5.2 icons). However, this finding suggests that further research is needed on the issue of decay in memory-based techniques.

Table 10: Error analysis for Study 5.3.

	No Overlapping		**Overlapping**		**Overall**	
Selection	**Study 5.1**	**Study 5.2**	**Study 5.1**	**Study 5.2**	**Study 5.1**	**Study 5.2**
Correct	60(79%)	69(91%)	22(58%)	35(92%)	82(72%)	104(91%)
Incorrect	16(21%)	7(9%)	16(42%)	3(8%)	32(28%)	10(9%)

5.8 INTERPRETATION

Previous experiments presented in Chapter Four indicated that proprioceptive knowledge of hands and fingers can be used as landmarks for better bimanual multi-touch interaction on tabletops (see Sections 4.4, 4.5 and 4.6). Our results from these experiments suggest that the same approach can be used effectively for single-handed interactions on tablets:

- Study 5.1 showed that both HM-Finger and HM-Multi allowed rapid selections (~2 seconds).

- Study 5.2 showed that using hands as landmarks improved performance: HM-Finger and HM-Multi outperformed HM-Under and were strongly preferred.
- Studies 5.2 and 5.3 showed that overlapping targets are not a main source of errors, but that location memory can degrade because of time or because of interference.

Here we discuss potential reasons for single-handed HandMark menus' performance in the three studies.

Selection performance of single-handed HandMark menus

Study 5.1 and 5.2 showed that both HM-Finger and HM-Multi facilitate rapid command selection once participants are familiar with item locations. The main reason for this performance is in the menu arrangement and item selection mechanism. For both techniques, menu items are placed in spatially-stable positions. When people are unfamiliar with the locations, they must visually search for a desired item (and explore multiple tabs with HM-Multi). After practice, users can use fingers as landmarks to remember item positions, and perform a "chunked" invoke-and-select sequence that is similar to double-clicking.

Overall, the adaptation of the HandMark menus to one-handed use was highly successful. The final performance of our serial version of the technique was very similar to results from previous studies of the bimanual version (both versions of HM-Finger take approximately 2.0 sec/selection, and both versions of HM-Multi take approximately 2.5 sec/selection).

Error rates in selections

In memory-based selection techniques, some errors are inevitable [47]. However, our studies showed that the adapted HandMark menus had extremely low error rates. We believe that the low error rate arises from the switch from parallel bimanual to serial operation – the latter enforces a short period between invocation and selection where the visuals of the menu can be used to check the selection.

5.9 SUMMARY

In this chapter we examined design and performance questions for single-handed HandMark menus for tablets. Similar to the earlier version of HandMark menus for tables, new version for tablets also used hands as landmarks for fast command selection. However, since one hand is occupied in holding the tablet, menus were assigned to one hand only. HandMark-Finger and HandMark-Multi had 20 and 80 items respectively. Results of our studies provide clear evidence that HandMark menus are a feasible and useful technique for tablets, and that they support development of spatial memory and fast performance even when used with a single hand.

CHAPTER 6

DISCUSSION

Here we discuss the findings of our studies with hand-centric menu techniques. We begin by discussing the implications of our findings[6] for both versions of HandMark menus – bimanual and single-handed, and then consider the lessons that were learned through the research.

6.1 SUMMARY OF FINDINGS

Our four quantitative user studies have resulted a number of findings. Here we summarize our main results.

6.1.1 Hands as Landmarks Aid in Spatial Memory Development

The HandMark technique uses the hands as a reference frame for layout of menu items. In contrast to traditional toolbar-based menu designs, we placed command items around and between the spaces of fingers. HandMark-Finger uses the fingers extensively; HandMark-Multi uses only the space between thumb and index finger (and also multiplexes the same space by assigning different sets to different finger combinations).

Our analyses of the command selection and expert selections results from both bimanual and single-handed versions of HandMark menus show that both Finger and Multi variants performed

---

[6] Material in this chapter first appeared in the following publications: [120, 121].

well compared to ordinary menus. Although bimanual HandMark-Multi was slow at the beginning, it performed similarly to standard menu system when users became familiar with the items. These results, particularly in Study 5.2 with the single-handed version (Section 5.5), indicate that participants were able to remember item locations using their hands and fingers as anchoring points. As a result, they could efficiently revisit those item locations [1, 46, 47, 103, 113]. Our findings suggest that use of hands as landmarks can be beneficial for spatial memory development.

Figure 54: Selection preparation with both hands.

Also, in the experiment with bimanual HandMark menus (Section 4.3) we observed that participants were carrying out preparatory actions with their menu fingers and selection finger even before placing their hands on the screen (see Figure 54). In these preparatory actions, the menu hand was used as a reference frame for staging the action. This indicates that hands and fingers helped in developing spatial memory of the commands.

6.1.2 HandMark Menus Can Improve Command Selection Performance

We carried several studies with the bimanual and single-handed versions of HandMark menus. Although the main working principle behind two versions was same – using hands as landmarks – we had to adjust the command selection process and capacity to match the physical characteristics of tables and tablets. Our analyses from these experiments showed that both techniques were easy to learn and followed similar learning patterns. Mean trial completion times for both versions were also similar for the two platforms: for Handmark-Finger, the bimanual version was 2320ms and the single-handed version was 2187ms; for HandMark-Multi, the bimanual version was 3840ms, and the single-handed version 3127ms. The larger number of commands in the bimanual version may explain the overall longer time compared to single-handed versions. Also, despite the significant difference in command capacity, mean trial completion times of HandMark menus, especially for the HandMark-Finger of both versions follow the trends of other memory-based techniques' findings [11, 47, 70].

The studies showed that HandMark techniques performed faster than some standard techniques (although bimanual HandMark-Multi was slower overall due to its performance during early blocks, as discussed in Section 4.8). In the situations where it was faster, there are a few reasons for HandMark's improved performance, based on the command selection steps for both novice and expert. For novices, there are three steps needed: invoking the correct command set, searching for the target command, and executing a selection action. Invoking the menu was different for both interfaces. For HandMark-Finger, invocation involves pressing with all five fingers anywhere on the touch surface, which was easy and fast. For HandMark-Multi, displaying a command set involves different combinations of fingers. With multiple sets, there were different combinations of fingers. Novices spent a large amount of time determining which finger combination belonged to which set.

Searching for a specific command within a set required a similar strategy for all interfaces. The final step – executing the selection action – was similar for all interfaces. For experts, selection in HandMark menus requires only two steps: retrieval of the command's set and location from memory [47, 103], and execution of the selection action at any position on the touch surface. The lack of a strong spatial reference frame for Tabs and hidden pop-up menu, however, means that users may still need to perform visual search, even when they are familiar with the location. The performance advantage for expert use of HandMark menus arises in the speed of execution, which can be achieved by selecting an item either in parallel (for bimanual versions) or in sequential chunked actions (for the single-handed version).

The results from Gutwin et al.'s FastTap [47], Lafreniere et al.'s research on smartwatch [76], Bailly et al.'s finger-count menu [11] and Kurtenbach et al.'s Marking menu [70] also agree with the performance of our HandMark menus, suggesting that HandMark techniques can improve command selection performance. Though our results follow similar trends, they are slightly higher than other memory-based techniques, which can be explained by the large difference in available commands within the interfaces. Also, selection process that requires multiple finger combinations from both hands (e.g., HandMark-Multi) can be a factor for these differences. Single-handed HandMark menus' better trail completion time may answer the that question. The similarities of results between HandMark menus and other memory-based techniques, however, indicate that using hand as landmarks can be another way to improve command selection performance in touch interfaces.

6.1.3 Errors in HandMark Menus

Errors are common in all memory-based interfaces [47, 76], because as users begin to use their memory rather than visual search, they can make mistakes. Our analyses showed substantially higher error rates in the bimanual version than the single-handed version. A main reason for this difference is that the single-handed techniques always present the visual information of the menu; the speedup for experts occurs because of a chunking of two serial actions, rather than a reliance on memory alone. In contrast, we believe that higher errors for the bimanual versions occurred because people were starting to use the memory-based expert selection of the technique with weak spatial knowledge. When people were using the novice method at earlier stages of use, there were no more errors than with standard techniques.

Error rates for the bimanual technique may also relate to the larger number of commands in these versions. In the short span of the experiments it may have been difficult for users to become familiar with the command set; a longer experience might produce a different result.

6.1.4 HandMark Menus Are Easy to Use

Ease of use is another important aspect of any new technique. We analyzed subjective responses using the NASA-TLX effort scales (mental, physical, temporal, performance, effort and frustration) [49]. The absolute scores from these measures suggest that participants were positive overall about the effort requirements of the new techniques. Our analyses did not show any significant differences in any factors for the bimanual use of HandMark menus, except for the performance of HandMark-Finger (where participants rated their own performance significantly higher, and rated their frustration significantly lower, than Tabs-2).

6.1.5 Users Prefer HandMark Menus

The analyses of subjective preference feedback revealed that participants preferred both HandMark-Finger and HandMark-Multi menus over the standard menus, in all measures. We did not compare between the two HandMark menus in the bimanual version, but we saw an opportunity of comparing them in single-handed use of HandMark menus. Analysis showed that most participants preferred HandMark-Finger overall. The richer landmarking features of this version helped users to easily memorize the locations, which was reflected in the preference count. However, the simplistic design and representation of HandMark-Multi-1Tab was preferred for rapid selection and comfortability measures.

6.2 RELATIVE MERITS OF HANDMARK MENUS

In general, both bimanual and single-handed techniques have strengths and weaknesses depending on the usage situation. Since bimanual techniques require two hands, larger surfaces are required for using bimanual HandMark menus. Though we developed the bimanual techniques for large tabletops, they can also be used in touch screen desktops. Generally, the bimanual selection process is faster than a single-handed serial two-step selection, as invocation and selection can be made in parallel. Single-handed HandMark menus are originally intended for multi-touch tablet interaction, but they can be used equally in touch laptops, desktops, and even large tabletops. Although there are two sequential steps for selection in single-handed HandMark menus, it can be faster than the visual guided hierarchical menus because of the chunked motor actions. One important factor is the command capacity, and bimanual techniques are capable of supporting exactly double compared to single-handed version. However, in situations where one hand is not needed to hold the device (e.g., on a table), the single-handed versions of HandMarks could also be used with either hand.

Of the two variants of the technique, HandMark-Finger has the potential of higher performance in situations where fewer commands are used, as it can support relatively small number of commands (only 20 items in single-handed version and 42 in bimanual). However, sometimes items were partially occluded by parts of the hand (as reported by a few participants); this was not a major problem, as the menu's placement can be adjusted by moving the hand. One main advantage of HandMark-Finger technique is that each finger is very close to some of the locations – in future, we plan to test menus where frequently-used items are placed in easier-to-reach locations (such as around the index finger).

The HandMark-Multi technique is more suitable for interfaces where larger command sets are required. Our designs support 80 commands (20 items in each set) for the single-handed version, and 160 for the bimanual version. Multiple tabs can be difficult to learn at first, because there is no clear indication of which finger combination is required (thus leading to the incorrect tab selection errors discussed in Sections 4.5.3 and 5.3.4). However, our participants quickly overcame this limitation (and we note that moving between tabs is fast enough that even when users make tab errors it does not greatly slow the technique).

Another advantage of HandMark-Multi is that all the items are placed in one general location, making initial visual search easier. Additionally, HandMark-Multi does not suffer from the occlusion problem, and involves a simpler hand posture when invoking the menu. Even though HandMark-Multi does not have as rich a set of landmarks compared to HandMark-Finger, the modified single-tab version performed as well or better than HandMark-Finger in Study 5.2 (Section 5.5). Therefore, HandMark-Multi can also be used as a single menu for a small number of items.

Finally, more research is required with both bimanual and single-handed HandMark techniques to explore integration with different sizes of multi-touch surfaces.

6.3 LESSONS LEARNED

Here we present some important lessons learned from our experiments.

6.3.1 Rich and Expressive Interactions with Multiple Fingers

In all the versions of our implemented HandMark technique, users were required to use multiple fingers at the same time. Most of the time they had to perform coordinated gestures using multiple fingers (often all the fingers) of one hand, and sometimes they were required to perform parallel actions using both hands. The results from our studies described in Chapters Four and Five show that participants performed well in carrying out those tasks with multiple fingers and hands. This provides additional evidence that people are capable of performing rich and expressive multi-finger and multi-hand interactions [20, 43]. Though people frequently use these capabilities in real life – e.g., typing in keyboard, playing piano, or driving a car – these are less explored in multi-touch GUIs.

6.3.2 Hand Postures and Ergonomics

A few participants reported difficulties with the finger combinations required to choose different command sets with HandMark-Multi, particularly in the bimanual version. Participants noted that changing quickly between sets was initially difficult as it required good finger dexterity (even though all of our hand postures are "relaxed" [40]). These initial problems were quickly overcome, and it was seen as helpful that the menus move and adapt as the hands are moved, allowing users to choose comfortable hand positions. Finally, two participants had longer fingernails in our bimanual study, but they did not have difficulty using HandMark menus.

6.3.3 A Few Locations Are Hard to Access

We chose an unconventional menu representation approach in our HandMark techniques, where we placed items in the spaces around and between the fingers (see Sections 4.1, 4.2 and 5.1). Although kinesthetic models [20] and Guiard's Kinematic Chain model [43] suggest that humans are capable performing rich and expressive tasks with multiple fingers and hands, some areas are inconvenient to access because of the physiological characteristics of human hands. For example, accessing the space beside pinky finger of right hand with left hand's finger or vice versa was more difficult than other regions. This information can be leveraged to design an efficient and ergonomic menu, such as placing infrequently used commands in those locations.

6.3.4 Indicating Hand Menu Contents

One important factor identified during the studies was that HandMark-Multi does not provide any visual indication of the mapping between finger combinations and command sets in order to assist users who are in the novice stages of learning. A visual "map legend" could be shown on the display as a reminder, but it would also be possible to use augmented-reality techniques to show menu contents (e.g., project symbols on the display near the hands above the table, or on the hands themselves).

6.3.5 Fingers Occluding Menu Contents

Our HandMark-Finger technique shows items between fingers, so it is possible for the hand itself to occlude the menu, particularly if the hand is not directly in front of the user. In our studies, some participants with smaller fingers occasionally experienced this problem. Occlusion should be considered when implementing hand-centric techniques. The problem of occlusion primarily affects the learning stages, however, when users are still using the visual guidance of the displayed menu; once item locations are known, users can position their selection finger using the hand rather than the display.

6.3.6 Real-World Use of HandMarks

The HandMark prototypes represent a tradeoff between landmarks and command capacity – HandMark-Finger makes more extensive use of the hands and is faster overall, but is limited in size; HandMark-Multi can accommodate more commands, but overloads one region (between thumb and index finger).

Although further studies with the techniques are needed, we speculate that people will be more successful at learning locations with HandMark-Finger, due to its richer landmarking, and therefore more likely to use the expert selection mode in real-world use. HandMark-Multi, on the other hand, does not have the same rich landmarks, but the consistent presentation of command sets between thumb and index finger is still likely to be valuable in real-world use. HandMark-

Multi can be considered as a version of earlier Palette techniques [63], but with the tool items always presented using a consistent spatial reference frame.

CHAPTER 7

CONCLUSION

Command selection on multi-touch surfaces can be difficult, because techniques are not well suited to the display setting, and also because the lack of landmarks makes it harder for users to build up familiarity with spatial locations. People's hands are always present in the workspace, however, and can be used as a reference frame for designing efficient touch-based selection techniques. A few techniques take advantage of hands, but often these methods are limited in the number of items they can accommodate. We designed two hand-centric techniques – HandMark-Finger and HandMark-Multi for multi-touch displays, implemented them in two different touch platforms – large tabletops allowing bimanual selection operation and hand-held tablets allowing single-handed operation, and tested them in empirical comparisons against equivalent-capacity tab menus and pop-up menus. Our results provide clear evidence that HandMark menus are a feasible and useful technique for touch interfaces (except for the early stage use of bimanual HandMark-Multi), and that they support development of spatial memory and fast performance in both bimanual and single-handed use. In addition, we showed that having the hand as a reference frame for the menu contents can provide significant performance and preference advantages[7].

---

[7] Material in this chapter first appeared in the following publications: [120, 121].

7.1 CONTRIBUTIONS

Primary contributions

There are two primary contributions presented in this thesis. First, we show that hands can be used as powerful landmarks for touch interfaces, and provide empirical evidence (Chapters Four and Five) that using hands as landmarks can improve selection performance, aid development of spatial memory, and accommodate large number of commands in multi-touch interfaces. Second, we demonstrate two new hand-centric interaction techniques – HandMark-Finger for small command sets and HandMark-Multi for larger sets – and describe their implementation for two different multi-touch device platforms (Chapters Four and Five).

Secondary contributions

Secondary contributions of this thesis are the set of design principles developed for HandMark techniques (that can be used to develop other spatial multi-touch interactions) (Chapter Three); empirical results about topics such as command overloading (Chapter Five), error rates in memory-based interfaces, and reasons for participant preferences; and algorithms for detecting hand postures from a set of simple touch points (Chapter Three).

7.2 FUTURE WORK

The research conducted in this thesis has laid the foundation for future hand-centric multi-touch interaction research, and opened a number of paths for future research for using real life tools – hands and fingers as landmarks within interactive touch surfaces.

7.2.1 Increasing the Number of Commands

Our bimanual HandMark prototypes explored two command-set sizes (42 for HandMark-Finger, and 160 for HandMark-Multi), and single-handed variant with 40 and 80 for HandMark-Finger and HandMark-Multi respectively. It is possible to increase these numbers considering the human finger and accessible minimum target size [30, 94]. For example, a larger grid can be placed between the space of thumb and index finger. Another approach is to stack additional layers above the fingers in HandMark-Finger (currently there is only one item), or by using the different positions in HandMark-Finger as triggers for second-level sets. However, further work is needed to determine whether larger command sets are beneficial – for example, our study showed that initial learning was more difficult for the multiple sets in HandMark-Multi, and it may be advantageous to restrict the number of commands to improve learnability.

7.2.2 Occlusion of the Menu

Since commands are displayed near and between fingers, there is a good possibility of menu item occlusion by users' hand. A few of our participants with smaller fingers reported experiencing this problem in the study. However, some alternate measures can overcome this problem, such as better determination of the actual shape of the hand, automatically scaling the icons for different hand sizes, or moving the icons upwards (while still maintaining relative spatial positioning) if there is not enough space between fingers. Previous work by Vogel and colleagues [122] has shown that unrestricted models of hand occlusion can be inferred from touch points, so we believe that our technique can be extended to the general usage case. However, further work is needed in this specific area to tackle the menu occlusion problem by hand.

7.2.3 Mapping the Commands to Menus and Locations

In all of our studies, we arbitrarily assigned commands to locations, and command sets to different hands – in real use, however, performance and learning could likely be substantially improved with a more thoughtful mapping. For example, with bimanual HandMark-Multi, both the left and right hands held four different command sets, and some participants initially had difficulty remembering which hand contained their desired set. The study with this technique suggested that choosing the wrong hand was a costly error, as participants tended to check each of the sets on that hand before trying the other hand. Minimizing the number of sets could be one way to tackle this issue, as single-handed variant produced less wrong set selections. But doing so will limit the overall command accommodation capacity. Therefore, further work is needed to determine how menu contents are best mapped to different hands.

In addition, our analysis of the performance based on target locations in Section 4.6.2 showed that some locations around the hand are faster and easier to learn. In order to make the technique more ergonomic and natural, frequent or important commands could be assigned to these locations. For example, we can place frequently used commands on top of each finger in HandMark-Finger menus. Also the large space between index and thumb in this technique can be used for placing commonly used items. Similarly, in case of HandMark-Multi technique, the grid positions closer to index and thumb can be used to place the frequently used commands from each set.

7.2.4 Hand Detection

In our prototype implementation, we used a simple hand detection algorithm (see Section 3.2.2) that we created to identify hands and fingers. We only considered touch (X, Y) coordinates, and based on the relative associations we identified hands and fingers. Although this technique is not robust enough to detect hand and fingers from any orientations (e.g., on tables where people can stand on any side), it worked well for the prototypes and the experiments described here. More research is required to create a precise hand identification technique for real life implementation of this technique.

7.2.5 Multiple Users and Different Orientations

Our prototype systems of bimanual HandMark menus for large tabletops supported only one person at a fixed location. In real life, however, multiple users interacting with a table from different edges is a common scenario. In contrast, the single-handed versions of HandMark menus do not require multiple hand identification since a tablet is primarily intended for single-person use. But it is not likely that a person will always place their hand in an upright orientation. So further work is needed to determine how the hand detection technique will perform with multiple hands and with hand(s) at any orientation. We believe that our hand-posture algorithms can handle these additional demands with additional sensing of the environment, such as finger-contact areas and shapes for tablets, or a depth camera that can track each person's approximate location around the table [39].

7.2.6 Device Orientation and Size

In our experiment with bimanual HandMark techniques we used a relatively small tabletop touch surface (24-inch diagonal). We believe that this setup reasonably approximates the actions that will be needed on a larger table, but future studies with larger surfaces are required to confirm this aspect. As we have seen that the reachability of menu items in standard tab menus was one of the likely reasons for their slow performance, it will be interesting the see how standard tabs and HandMarks actually perform in larger tables.

We also experimented with single-handed HandMark techniques on a 12-inch surface tablet. In reality, however, different sizes of tablets are available, some of which might be smaller in size (e.g., 7-inch tablet). Since our techniques require multiple fingers to interact with system, often all the fingers of a hand are required (e.g., HandMark-Finger), it will be interesting to explore the whole hand interaction with HandMark techniques in smaller touch devices.

7.2.7 Advanced Interactions

Most interfaces include widgets that are more advanced than buttons – for example, sliders or color pickers can be used to provide a finer degree of control over application parameters. We will explore how these kinds of widgets can be converted to work with HandMark menus – for example, HandMark could be adapted to use a Toolglass-style interaction [16] in which users click through the command and start their manipulation at the same time. It may also be possible to combine HandMarks with other gesture-based techniques such as marking menus [70]. For example, people could activate different command modes with one hand and perform gestures with the other in a bimanual interaction setting, or with the same hand in single-handed use.

In addition, since there is a large number of commands available in HandMark menus as icons without any text description, discoverability of the commands could be an issue. In real life applications, most of the cases, each command's name is provided along with the icon. This could be handled by adding long press feature to the icons. For instance, people could see the short description or name of the command with a simple long press over any command. Though it is not required for the expert users, novices could be benefitted with this extra feature in learning stage.

7.2.8 Handedness Effect

We designed and tested the single-handed HandMark prototype for right-handed people only. However, with simple changes the techniques can be used by left-handed users. Handedness could be an issue for bimanual versions of the HandMark techniques also. Though we did not perform any specific study of the handedness effect on performance, it is possible to get different performance results depending on the handedness of the users. Further research is required in this area to understand the effect of handedness on performance.

7.2.9 Overloading Memory Space

In real life we use different applications for different purposes, and each of them involves interaction with several widgets and commands. When using memory-based interaction techniques such as HandMark menus, some command locations will overlap. In addition, learning one application's command locations also might cause interference on the learning of another application. Although we tested a limited version of this question with the single-handed variant of HandMark menus, the results did not provide a comprehensive look at the issue. Further work is required to explore this real life implementation issue of HandMark techniques.

REFERENCES

[1] Alexander, J., Cockburn, A., Fitchett, S., Gutwin, C. and Greenberg, S. 2009. Revisiting read wear: analysis, design, and evaluation of a footprints scrollbar. *Proceedings of the 27th international conference on Human factors in computing systems - CHI '09* (New York, New York, USA, 2009), 1665–1674.

[2] Al-Hajri, A., Miller, G., Fong, M. and Fels, S.S. 2014. Visualization of personal history for video navigation. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (New York, New York, USA, 2014), 1187–1196.

[3] Android Developer Library: *https://developer.android.com/guide/index.html*. Accessed: 2016-10-08.

[4] Annett, M., Anderson, F., Bischof, W.F. and Gupta, A. 2014. The Pen is Mightier: Understanding Stylus Behaviour While Inking on Tablets. *Proceedings of Graphics Interface 2014* (Toronto, Ont., Canada, 2014), 193–200.

[5] Annett, M., Gupta, A. and Bischof, W.F. 2014. Exploring and Understanding Unintended Touch during Direct Pen Interaction. *ACM Transactions on Computer-Human Interaction*. 21, 5 (Nov. 2014), 1–39.

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

[7] Avery, J. and Lank, E. 2016. Surveying Expert-Level Gesture Use and Adoption on Multi-Touch Tablets. *Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS '16* (New York, New York, USA, 2016), 577–581.

[8] Baecker, R., Booth, K., Jovicic, S., McGrenere, J. and Moore, G. 2000. Reducing the gap between what users know and what they need to know. *Proceedings on the 2000 conference on Universal Usability - CUU '00* (New York, New York, USA, 2000), 17–23.

[9] Baglioni, M., Lecolinet, E. and Guiard, Y. 2011. JerkTilts: using accelerometers for eight-choice selection on mobile devices. *Proceedings of the 13th international conference on multimodal interfaces - ICMI '11* (New York, New York, USA, 2011), 121–128.

[10] Bailly, G., Lecolinet, E. and Nigay, L. 2008. Flower menus: a new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. *Proceedings of the working conference on Advanced visual interfaces - AVI '08* (New York, New York, USA, 2008), 15–22.

[11] Bailly, G., Müller, J. and Lecolinet, E. 2012. Design and evaluation of finger-count interaction: Combining multitouch gestures and menus. *International Journal of Human-Computer Studies*. 70, 10 (Oct. 2012), 673–689.

[12] Balakrishnan, R. and Patel, P. 1998. The PadMouse: facilitating selection and spatial positioning for the non-dominant hand. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '98* (New York, New York, USA, 1998), 9–16.

[13] Bau, O. and Mackay, W.E. 2008. OctoPocus: a dynamic guide for learning gesture-based command sets. *Proceedings of the 21st annual ACM symposium on User interface software and technology - UIST '08* (New York, New York, USA, 2008), 37–46.

[14] Baudisch, P. and Rosenholtz, R. 2003. Halo: a technique for visualizing off-screen objects. *Proceedings of the conference on Human factors in computing systems - CHI '03* (New York, New York, USA, 2003), 481–488.

[15] Benko, H., Saponas, T.S., Morris, D. and Tan, D. 2009. Enhancing input on and above the interactive surface with muscle sensing. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09* (New York, New York, USA, 2009), 93–100.

[16] Bier, E.A., Stone, M.C., Pier, K., Buxton, W. and DeRose, T.D. 1993. Toolglass and magic lenses: the see-through interface. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques - SIGGRAPH '93* (New York, New York, USA, 1993), 73–80.

[17] Brandl, P., Forlines, C., Wigdor, D., Haller, M. and Shen, C. 2008. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. *Proceedings of the working conference on Advanced visual interfaces - AVI '08* (New York, New York, USA, 2008), 154–161.

[18] Buchmann, V., Violich, S., Billinghurst, M. and Cockburn, A. 2004. FingARtips: gesture based direct manipulation in Augmented Reality. *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Austalasia and Southe East Asia - GRAPHITE '04* (New York, New York, USA, 2004), 212–221.

[19] Butler, C.G. and St. Amant, R. 2004. HabilisDraw DT: a bimanual tool-based direct manipulation drawing environment. *Extended abstracts of the 2004 conference on Human factors and computing systems - CHI '04* (New York, New York, USA, 2004), 1301–1304.

[20] Buxton, W., Hill, R. and Rowley, P. 1985. Issues and techniques in touch-sensitive tablet input. *ACM SIGGRAPH Computer Graphics*. 19, 3 (Jul. 1985), 215–224.

[21] Buxton, W. and Myers, B. 1986. A study in two-handed input. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '86* (New York, New York, USA, 1986), 321–326.

[22] Callahan, J.R., Hopkins, D., Weiser, M.D. and Shneiderman, B. 1988. An empirical comparison of pie vs. linear menus. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '88* (New York, New York, USA, 1988), 95–100.

[23] Casalta, D., Guiard, Y. and Lafon, M.B. 1999. Evaluating two-handed input techniques: rectangle editing and navigation. *CHI '99 extended abstracts on Human factors in computing systems - CHI '99* (New York, New York, USA, 1999), 236–237.

[24] Cechanowicz, J.E. 2008. *Improving expressivity in desktop interactions with a pressure-augmented mouse*.

[25] Chen, C.-H. and Nian, C.-X. 2009. VTouch: A Vision-Base Dual Finger Touched Inputs for Large Displays. *Proceedings of the 13th International Conference on Human-Computer Interaction. Part II: Novel Interaction Methods and Techniques*. Springer-Verlag. 426–434.

[26] Cockburn, A., Gutwin, C. and Alexander, J. 2006. Faster document navigation with space-filling thumbnails. *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06* (New York, New York, USA, 2006), 1–10.

[27] Cockburn, A., Gutwin, C. and Greenberg, S. 2007. A predictive model of menu performance. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07* (New York, New York, USA, 2007), 627–636.

[28] Cockburn, A., Quinn, P., Gutwin, C., Ramos, G.A. and Looser, J. 2011. Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. *International Journal of Human-Computer Studies*. 69, 6 (Jun. 2011), 401–414.

[29] Czerwinski, M. and Robertson, G. 1999. The Contribution of Thumbnail Image, Mouse-over Text and Spatial Location Memory to Web Page Retrieval in 3D. *Proceedings of Seventh International Conference on Human-Computer Interaction - INTERACT '99* (Edinburgh, Scotland, Scotland, 1999), 163–170.

[30] Dandekar, K., Raju, B.I. and Srinivasan, M.A. 2003. 3-D Finite-Element Models of Human and Monkey Fingertips to Investigate the Mechanics of Tactile Sense. *Journal of Biomechanical Engineering*. 125, 5 (2003), 682–691.

[31] Dang, C.T., Straub, M. and André, E. 2009. Hand distinction for multi-touch tabletop interaction. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09* (New York, New York, USA, 2009), 101–108.

[32] Designing for Mobile, Part 2: Interaction Design | UX Booth: *http://www.uxbooth.com/articles/designing-for-mobile-part-2-interaction-design/*. Accessed: 2016-10-08.

[33] Dietz, P. and Leigh, D. 2001. DiamondTouch: a multi-user touch technology. *Proceedings of the 14th annual ACM symposium on User interface software and technology - UIST '01*

(New York, New York, USA, 2001), 219–226.

[34] Downloads | Android Developers: *http://developer.android.com/design/downloads/index.html*. Accessed: 2016-09-19.

[35] Engel, F.L. 1977. Visual conspicuity, visual search and fixation tendencies of the eye. *Vision Research*. 17, 1 (1977), 95–108.

[36] Esenther, A. and Ryall, K. 2006. Fluid DTMouse: better mouse support for touch-based interactions. *Proceedings of the working conference on Advanced visual interfaces - AVI '06* (New York, New York, USA, 2006), 112–115.

[37] Ewerling, P., Kulik, A. and Froehlich, B. 2012. Finger and hand detection for multi-touch interfaces based on maximally stable extremal regions. *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces - ITS '12* (New York, New York, USA, 2012), 173–182.

[38] Forlines, C., Wigdor, D., Shen, C. and Balakrishnan, R. 2007. Direct-touch vs. mouse input for tabletop displays. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07* (New York, New York, USA, 2007), 647–656.

[39] Genest, A.M., Gutwin, C., Tang, A., Kalyn, M. and Ivkovic, Z. 2013. KinectArms: a toolkit for capturing and displaying arm embodiments in distributed tabletop groupware. *Proceedings of the 2013 conference on Computer supported cooperative work - CSCW '13* (New York, New York, USA, 2013), 157–166.

[40] Ghomi, E., Huot, S., Bau, O., Beaudouin-Lafon, M. and Mackay, W.E. 2013. Arpège: learning multitouch chord gestures vocabularies. *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces - ITS '13* (New York, New York, USA, 2013), 209–218.

[41] Gilles Bailly: Finger-Count: *http://www.gillesbailly.fr/fingercount.html*. Accessed: 2016-10-08.

[42] Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M. and Balakrishnan, R. 2006. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06* (New York, New York, USA, 2006), 861–870.

[43] Guiard, Y. 1987. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behavior*. 19, 4 (Dec. 1987), 486–517.

[44] Gustafson, S., Baudisch, P., Gutwin, C. and Irani, P. 2008. Wedge: clutter-free visualization of off-screen locations. *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08* (New York, New York, USA, 2008), 787–796.

[45] Gustafson, S., Bierwirth, D. and Baudisch, P. 2010. Imaginary interfaces: spatial interaction with empty hands and without visual feedback. *Proceedings of the 23nd annual ACM symposium on User interface software and technology - UIST '10* (New York, New York, USA, 2010), 3–12.

[46] Gutwin, C. and Cockburn, A. 2006. Improving list revisitation with ListMaps. *Proceedings of the working conference on Advanced visual interfaces - AVI '06* (New York, New York, USA, New York, USA, 2006), 396–403.

[47] Gutwin, C., Cockburn, A., Scarr, J., Malacria, S. and Olson, S.C. 2014. Faster command selection on tablets with FastTap. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (New York, New York, USA, 2014), 2617–2626.

[48] Hancock, M.S. and Booth, K.S. 2004. Improving menu placement strategies for pen input. *Proceedings of Graphics Interface 2004* (London, Ontario, Canada, 2004), 221–230.

[49] Hart, S.G. and Staveland, L.E. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*. 52, (1988), 139–183.

[50] Hick, W.E. 1952. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*. 4, 1 (Mar. 1952), 11–26.

[51] Hinckley, K., Baudisch, P., Ramos, G. and Guimbretiere, F. 2005. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05* (New York, New York, USA, 2005), 451–460.

[52] Hinckley, K. and Buxton, B. 2016. Inking Outside the Box: How Context Sensing Affords More Natural Pen (and Touch) Computing. Springer International Publishing. 27–47.

[53] Hinckley, K., Pausch, R., Proffitt, D., Patten, J. and Kassell, N. 1997. Cooperative bimanual action. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '97* (New York, New York, USA, 1997), 27–34.

[54] Hinckley, K., Wilson, A., Pahud, M., Benko, H., Irani, P., Guimbretière, F., Gavriliu, M., Chen, X. "Anthony," Matulic, F. and Buxton, W. 2014. Sensing techniques for tablet+stylus interaction. *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14* (New York, New York, USA, 2014), 605–614.

[55] Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H. and Buxton, B. 2010. Pen + touch = new tools. *Proceedings of the 23nd annual ACM symposium on User interface software and technology - UIST '10* (New York, New York, USA, 2010), 27–36.

[56] Holz, C. and Baudisch, P. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10* (New York, New York, USA, 2010), 581–590.

[57] Howes, A., Payne, S.J. and Woodward, A. 2000. The trouble with shortcuts. *CHI '00 extended abstracts on Human factors in computing systems - CHI '00* (New York, New York, USA, 2000), 267–268.

[58] Hyman, R. 1953. Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*. 45, 3 (1953), 188–196.

[59] Ion, A., Chang, Y.-L.B., Haller, M., Hancock, M. and Scott, S.D. 2013. Canyon: providing location awareness of multiple moving objects in a detail view on large displays. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (New York, New York, USA, 2013), 3149–3158.

[60] iOS Developer Library: *https://developer.apple.com/library/ios/navigation/*. Accessed: 2016-10-08.

[61] Jetter, H.-C., Leifert, S., Gerken, J., Schubert, S. and Reiterer, H. 2012. Does (multi-)touch aid users' spatial memory and navigation in "panning" and in "zooming & panning" UIs? *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '12* (New York, New York, USA, 2012), 83–90.

[62] Jones, W.P. and Dumais, S.T. 1986. The spatial metaphor for user interfaces: experimental tests of reference by location versus name. *ACM Transactions on Information Systems*. 4, 1 (Jan. 1986), 42–63.

[63] Kabbash, P., Buxton, W. and Sellen, A. 1994. Two-handed input in a compound task. *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94* (New York, New York, USA, 1994), 417–423.

[64] Kabbash, P., MacKenzie, I.S. and Buxton, W. 1993. Human Performance Using Computer Input Devices in the Preferred and Non-preferred Hands. *Proceedings of the INTERCHI '93 conference on Human factors in computing systems* (Amsterdam, The Netherlands, 1993), 474–481.

[65] Kim, J., Guo, P.J., Cai, C.J., Li, S.-W. (Daniel), Gajos, K.Z. and Miller, R.C. 2014. Data-driven interaction techniques for improving navigation of educational videos. *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14* (New York, New York, USA, 2014), 563–572.

[66] Kin, K., Hartmann, B. and Agrawala, M. 2011. Two-handed marking menus for multitouch devices. *ACM Transactions on Computer-Human Interaction*. 18, 3 (Jul. 2011), 1–23.

[67] Kitani, A. and Nakatani, T. 2016. A Suitable Design for Natural Menu Opening Manipulations When Note-Taking on Tablet Devices. Springer International Publishing. 358–363.

[68] Krendel, E.S. and Wodinsky, J. 1960. Search in an unstructured visual field. *Journal of the Optical Society of America*. 50, (Jun. 1960), 562–568.

[69] Kurtenbach, G. and Buxton, W. 1993. The limits of expert performance using hierarchic marking menus. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '93* (New York, New York, USA, 1993), 482–487.

[70] Kurtenbach, G. and Buxton, W. 1994. User learning and performance with marking menus. *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94* (New York, New York, USA, 1994), 258–264.

[71] Kurtenbach, G., Fitzmaurice, G.W., Owen, R.N. and Baudel, T. 1999. The Hotbox: efficient access to a large number of menu-items. *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99* (New York, New York, USA, 1999), 231–237.

[72] Kurtenbach, G., Moran, T.P. and Buxton, W. 1994. Contextual Animation of Gestural Commands. *Computer Graphics Forum*. 13, 5 (Dec. 1994), 305–314.

[73] Kurtenbach, G., Sellen, A. and Buxton, W. 1993. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Human-Computer Interaction*. 8, 1 (Mar. 1993), 1–23.

[74] Kurtenbach, G.P. 1993. *The design and evaluation of marking menus*. University of Toronto.

[75] Lafreniere, B., Bunt, A., Whissell, J.S., Clarke, C.L.A. and Terry, M. 2010. Characterizing Large-Scale Use of a Direct Manipulation Application in the Wild. *Proceedings of Graphics Interface 2010 (GI '10)* (Toronto, Ont., Canada, Canada, 2010), 11–18.

[76] Lafreniere, B., Gutwin, C., Cockburn, A. and Grossman, T. 2016. Faster Command Selection on Touchscreen Watches. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (New York, New York, USA, 2016), 4663–4674.

[77] Latulipe, C., Mann, S., Kaplan, C.S. and Clarke, C.L.A. 2006. SymSpline: symmetric two-handed spline manipulation. *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06* (New York, New York, USA, 2006), 349–358.

[78] Lee, E. and Macgregor, J. 1985. Minimizing User Search Time in Menu Retrieval Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*. 27, 2 (1985), 157–162.

[79] Lepinski, G.J., Grossman, T. and Fitzmaurice, G. 2010. The design and evaluation of multitouch marking menus. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10* (New York, New York, USA, 2010), 2233–2242.

[80] Letessier, J. and Bérard, F. 2004. Visual tracking of bare fingers for interactive surfaces. *Proceedings of the 17th annual ACM symposium on User interface software and technology - UIST '04* (New York, New York, USA, 2004), 119–122.

[81] Li, F.C.Y., Dearman, D. and Truong, K.N. 2009. Virtual shelves: interactions with orientation aware devices. *Proceedings of the 22nd annual ACM symposium on User interface software and technology - UIST '09* (New York, New York, USA, 2009), 125–128.

[82] Lü, H. and Li, Y. 2011. Gesture avatar: a technique for operating mobile user interfaces using gestures. *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11* (New York, New York, USA, 2011), 207–216.

[83] Lynch, K. 1960. *The image of the city*. MIT Press.

[84] Malacria, S., Bailly, G., Harrison, J., Cockburn, A. and Gutwin, C. 2013. Promoting Hotkey use through rehearsal with ExposeHK. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (New York, New York, USA, 2013), 573–582.

[85] Malacria, S., Lecolinet, E. and Guiard, Y. 2010. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10* (New York, New York, USA, 2010), 2615–2624.

[86] Matsushita, N. and Rekimoto, J. 1997. HoloWall: designing a finger, hand, body, and object sensitive wall. *Proceedings of the 10th annual ACM symposium on User interface software and technology - UIST '97* (New York, New York, USA, 1997), 209–210.

[87] Meyer, A. 1995. Pen computing: a technology overview and a vision. *ACM SIGCHI Bulletin*. 27, 3 (Jul. 1995), 46–90.

[88] Myers, B.A., Lie, K.P. and Yang, B.-C. 2000. Two-handed input using a PDA and a mouse. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '00* (New York, New York, USA, 2000), 41–48.

[89] Ng, A., Lepinski, J., Wigdor, D., Sanders, S. and Dietz, P. 2012. Designing for low-latency direct-touch input. *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12* (New York, New York, USA, 2012), 453–464.

[90] Nilsen, E.L. 1992. *Perceptual-motor control in human-computer interaction*. University of Michigan.

[91] Odell, D.L., Davis, R.C., Smith, A. and Wright, P.K. 2004. Toolglasses, marking menus, and hotkeys: a comparison of one and two-handed command selection techniques. *Proceedings of Graphics Interface 2004 - GI '04* (Waterloo, Ontario, Canada, 2004), 17–24.

[92] Olwal, A., Feiner, S. and Heyman, S. 2008. Rubbing and tapping for precise and rapid selection on touch-screen displays. *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08* (New York, New York, USA, 2008), 295–304.

[93] Owen, R., Kurtenbach, G., Fitzmaurice, G., Baudel, T. and Buxton, B. 2005. When it gets more difficult, use both hands: exploring bimanual curve manipulation. *Proceedings of Graphics Interface 2005* (Waterloo, Ontario, Canada, 2005), 17–24.

[94] Parhi, P., Karlson, A.K. and Bederson, B.B. 2006. Target size study for one-handed thumb use on small touchscreen devices. *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services - MobileHCI '06* (New York, New York, USA, 2006), 203–210.

[95] Pavel, A., Goldman, D.B., Hartmann, B. and Agrawala, M. 2015. SceneSkim: Searching and Browsing Movies Using Synchronized Captions, Scripts and Plot Summaries. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15* (New York, New York, USA, 2015), 181–190.

[96] Pavel, A., Reed, C., Hartmann, B. and Agrawala, M. 2014. Video Digests: a browsable, skimmable format for informational lecture videos. *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14* (New York, New York, USA, 2014), 573–582.

[97] Procedural Memory: Definition and Examples: *http://www.livescience.com/43595-procedural-memory.html*. Accessed: 2016-10-08.

[98] Proprioception | Biology | Britannica.com: *https://www.britannica.com/science/proprioception*. Accessed: 2010-10-09.

[99] Rekimoto, J. 2002. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02* (New York, New York, USA, 2002), 113–120.

[100] Robertson, G., Czerwinski, M., Larson, K., Robbins, D.C., Thiel, D. and van Dantzich, M. 1998. Data mountain: using spatial memory for document management. *Proceedings of the 11th annual ACM symposium on User interface software and technology - UIST '98* (New York, New York, USA, 1998), 153–162.

[101] Roy, Q., Malacria, S., Guiard, Y., Lecolinet, E. and Eagan, J. 2013. Augmented letters:

mnemonic gesture-based shortcuts. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (New York, New York, USA, 2013), 2325–2328.

[102] Saund, E. and Lank, E. 2003. Stylus input and editing without prior selection of mode. *Proceedings of the 16th annual ACM symposium on User interface software and technology - UIST '03* (New York, New York, USA, 2003), 213–216.

[103] Scarr, J., Cockburn, A., Gutwin, C. and Bunt, A. 2012. Improving command selection with CommandMaps. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12* (New York, New York, USA, 2012), 257–266.

[104] Scarr, J., Cockburn, A., Gutwin, C. and Quinn, P. 2011. Dips and ceilings: understanding and supporting transitions to expertise in user interfaces. *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11* (New York, New York, USA, 2011), 2741–2750.

[105] Scarr, J.L. 2014. *Understanding and Exploiting Spatial Memory in the Design of Efficient Command Selection Interfaces*. University of Canterbury. Computer Science and Software Engineering.

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

[107] Schmidt, D., Chong, M.K. and Gellersen, H. 2010. IdLenses: dynamic personal areas on shared surfaces. *ACM International Conference on Interactive Tabletops and Surfaces - ITS '10* (New York, New York, USA, 2010), 131–134.

[108] Schramm, K., Gutwin, C. and Cockburn, A. 2016. Supporting Transitions to Expertise in Hidden Toolbars. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (New York, New York, USA, 2016), 4687–4698.

[109] Serrano, M., Lecolinet, E. and Guiard, Y. 2013. Bezel-Tap gestures: quick activation of commands from sleep mode on tablets. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (New York, New York, USA, 2013), 3027–3036.

[110] Shen, E.E., Tsai, S.D., Chu, H., Hsu, Y.J. and Chen, C.E. 2009. Double-side multi-touch input for mobile devices. *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems - CHI EA '09* (New York, New York, USA, 2009), 4339–4344.

[111] Shneiderman, B. 1987. Direct manipulation: A step beyond programming languages. *Human-computer interaction: A Multidisciplinary Approach*. M. Kaufmann. 461–467.

[112] Siek, K.A., Rogers, Y. and Connelly, K.H. 2005. Fat Finger Worries: How Older and

Younger Users Physically Interact with PDAs. *Proceedings of the 2005 IFIP TC13 international conference on Human-Computer Interaction* (2005), 267–280.

[113] Skopik, A. and Gutwin, C. 2005. Improving revisitation in fisheye views with visit wear. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05* (New York, New York, USA, 2005), 771–780.

[114] SMART Technologies Inc. invents new technology for touch-sensitive displays - bobsguide.com: *http://www.bobsguide.com/guide/news/2003/May/23/smart-technologies-inc-invents-new-technology-for-touch-sensitive-displays/*. Accessed: 2016-10-08.

[115] Spatial memory: *http://en.wikipedia.org/wiki/Spatial_memory*. Accessed: 2016-10-08.

[116] Swipe, pinch, zoom & tab - All you want to know about Touchscreen Technology - Advantech: *http://www.advantech.eu/pressroom/corporate_news/45b9a76d-d8f3-4a81-ae7f-66ff92510275*. Accessed: 2016-10-08.

[117] Tan, D.S., Pausch, R., Stefanucci, J.K. and Proffitt, D.R. 2002. Kinesthetic cues aid spatial memory. *CHI '02 extended abstracts on Human factors in computing systems - CHI '02* (New York, New York, USA, 2002), 806–807.

[118] Touch hardware and Windows 8 – Building Windows 8: *https://blogs.msdn.microsoft.com/b8/2012/03/28/touch-hardware-and-windows-8/*. Accessed: 2016-10-08.

[119] Tuncer, Z., Selim, M., Liwicki, M. and Kormann, G. 2015. Evaluation of Multi-Touch Gestures. *GIL Jahrestagung*. 189–192.

[120] Uddin, M.S. and Gutwin, C. 2016. Rapid Command Selection on Multi-Touch Tablets with Single-Handed HandMark Menus. *Proceedings of the SIGCHI Conference on Interactive Surfaces and Spaces - ISS '16* (Niagara Falls, ON, Canada, 2016).

[121] Uddin, M.S., Gutwin, C. and Lafreniere, B. 2016. HandMark Menus: Rapid Command Selection and Large Command Sets on Multi-Touch Displays. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (New York, New York, USA, 2016), 5836–5848.

[122] Vogel, D. 2010. *Direct Pen Input and Hand Occlusion*.

[123] Vogel, D., Cudmore, M., Casiez, G., Balakrishnan, R. and Keliher, L. 2009. Hand occlusion with tablet-sized direct pen input. *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09* (New York, New York, USA, 2009), 557–566.

[124] Wagner, J., Lecolinet, E. and Selker, T. 2014. Multi-finger chords for hand-held tablets. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*

- *CHI ’14* (New York, New York, USA, 2014), 2883–2892.

[125] Wang, R.Y. and Popović, J. 2009. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*. 28, 3 (Jul. 2009).

[126] Wigdor, D. and Wixon, D. 2011. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann Publishers Inc.

[127] Wobbrock, J.O., Myers, B.A. and Aung, H.H. 2008. The performance of hand postures in front- and back-of-device interaction for mobile computing. *International Journal of Human-Computer Studies*. 66, 12 (Dec. 2008), 857–875.

[128] Wolf, K. 2015. Interaction Areas on the Front and on the Back of Tablets. Springer International Publishing. 49–66.

[129] Wu, M. and Balakrishnan, R. 2003. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *Proceedings of the 16th annual ACM symposium on User interface software and technology - UIST ’03* (New York, New York, USA, 2003), 193–202.

[130] Wu, M., Chia Shen, Ryall, K., Forlines, C. and Balakrishnan, R. Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces. *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP ’06)* 185–192.

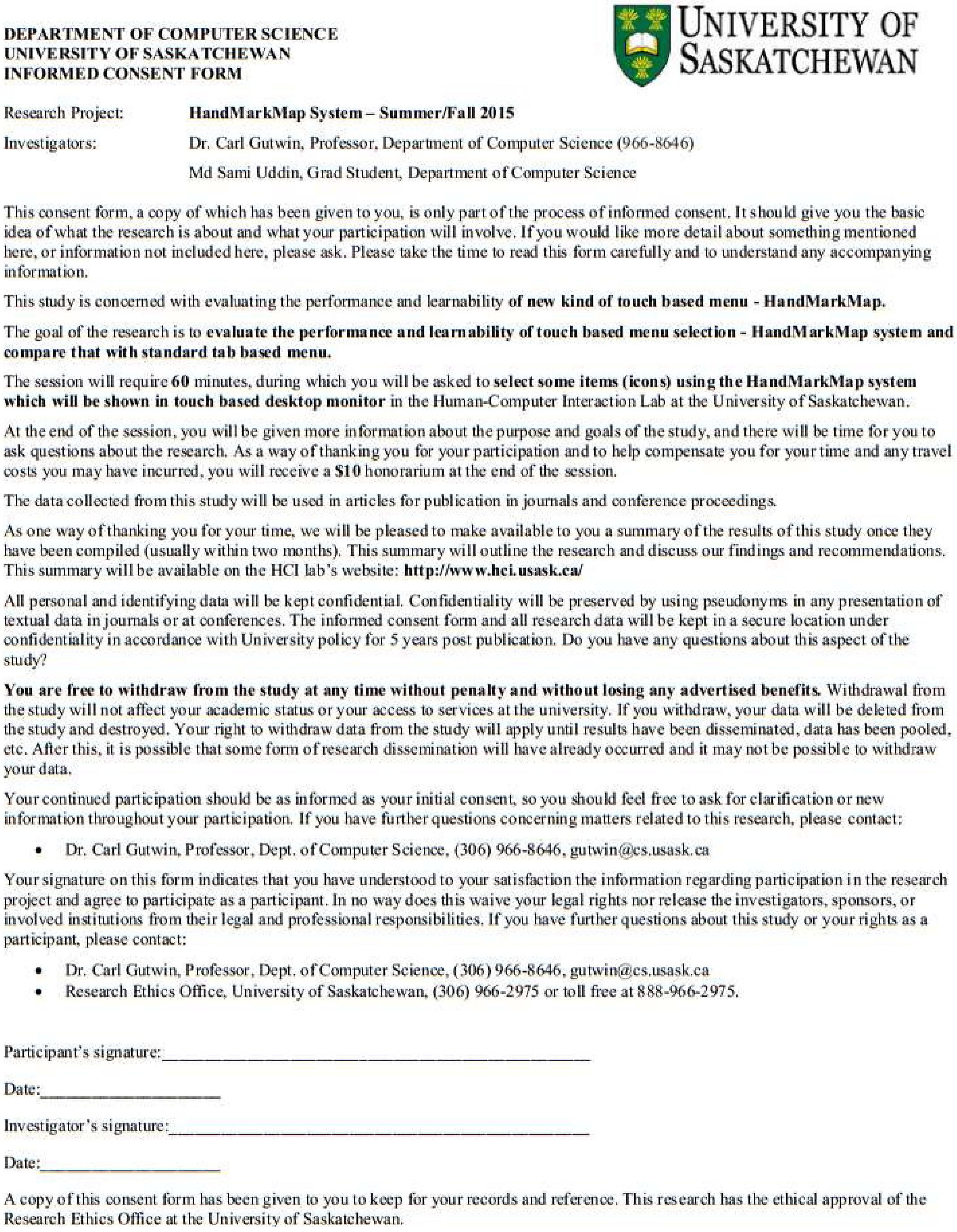
[131] Yu, B., Ma, W.-Y., Nahrstedt, K. and Zhang, H.-J. 2003. Video summarization based on user log enhanced link analysis. *Proceedings of the eleventh ACM international conference on Multimedia - MULTIMEDIA ’03* (New York, New York, USA, 2003), 382–391.

[132] Zellweger, P.T., Mackinlay, J.D., Good, L., Stefik, M. and Baudisch, P. 2003. City lights: contextual views in minimal space. *CHI ’03 extended abstracts on Human factors in computing systems - CHI ’03* (New York, New York, USA, 2003), 838–839.

[133] Zhao, S., Agrawala, M. and Hinckley, K. 2006. Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI ’06* (New York, New York, USA, 2006), 1077–1086.

APPENDIX

STUDY CONSENT FORMS



DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM

UNIVERSITY OF SASKATCHEWAN

Research Project:	**HandMarkMap System – Summer/Fall 2015**
Investigators:	Dr. Carl Gutwin, Professor, Department of Computer Science (966-8646)
	Md Sami Uddin, Grad Student, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with evaluating the performance and learnability **of new kind of touch based menu - HandMarkMap.**

The goal of the research is to **evaluate the performance and learnability of touch based menu selection - HandMarkMap system and compare that with standard tab based menu.**

The session will require **60** minutes, during which you will be asked to **select some items (icons) using the HandMarkMap system which will be shown in touch based desktop monitor** in the Human-Computer Interaction Lab at the University of Saskatchewan.

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research. As a way of thanking you for your participation and to help compensate you for your time and any travel costs you may have incurred, you will receive a **$10** honorarium at the end of the session.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. This summary will be available on the HCI lab's website: **http://www.hci.usask.ca/**

All personal and identifying data will be kept confidential. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca
- Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll free at 888-966-2975.

Participant's signature:____

Date:__________________

Investigator's signature:____

Date:__________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM

Research Project: **HandMark Menu for Tablet – Summer 2016**

Investigators: Dr. Carl Gutwin, Professor, Department of Computer Science (966-8646)
Md. Sami Uddin, Grad Student, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with evaluating the performance and learnability **of new kind of landmark-based menu for Tablet – HandMark Menu.**

The goal of the research is to **evaluate the performance and learnability of command selection in Tablet with HandMark Menu system and compare among three techniques.**

The session will require about **60** minutes, during which you will be asked to **select some targets (icons) using the HandMark Menu system which will be shown in a Tablet** in the Human-Computer Interaction Lab at the University of Saskatchewan.

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research. As a way of thanking you for your participation and to help compensate you for your time and any travel costs you may have incurred, you will receive a **$10** honorarium at the end of the session.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. This summary will be available on the HCI lab's website: **http://www.hci.usask.ca/**

All personal and identifying data will be kept confidential. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca
- Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll free at 888-966-2975.

Participant's signature:____

Date:____________________

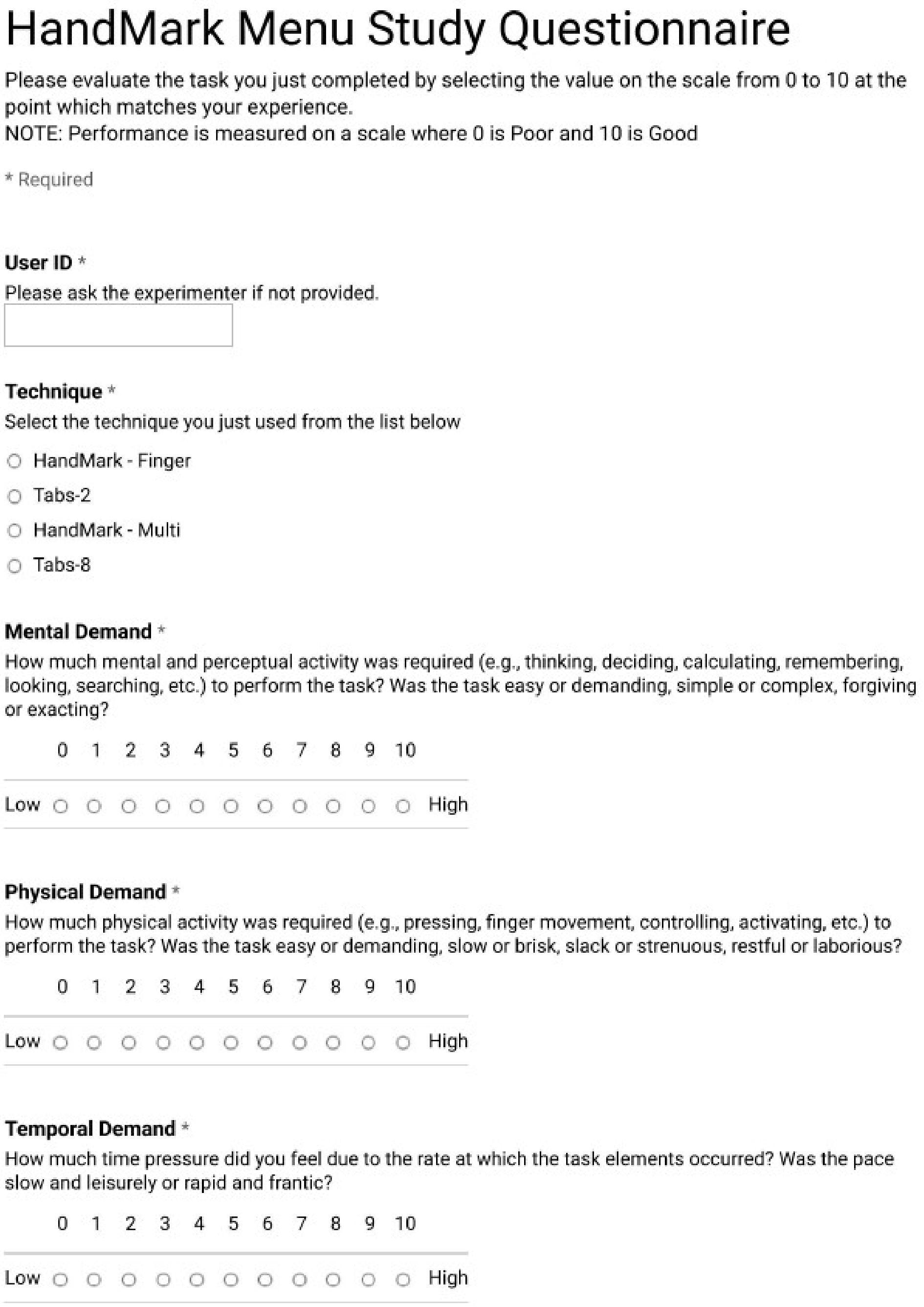
Investigator's signature:____

Date:____________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

APPENDIX

QUESTIONNAIRES



HandMark Menu Study Questionnaire

Please evaluate the task you just completed by selecting the value on the scale from 0 to 10 at the point which matches your experience.
NOTE: Performance is measured on a scale where 0 is Poor and 10 is Good

* Required

User ID *
Please ask the experimenter if not provided.

Technique *
Select the technique you just used from the list below

○ HandMark - Finger
○ Tabs-2
○ HandMark - Multi
○ Tabs-8

Mental Demand *
How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.) to perform the task? Was the task easy or demanding, simple or complex, forgiving or exacting?

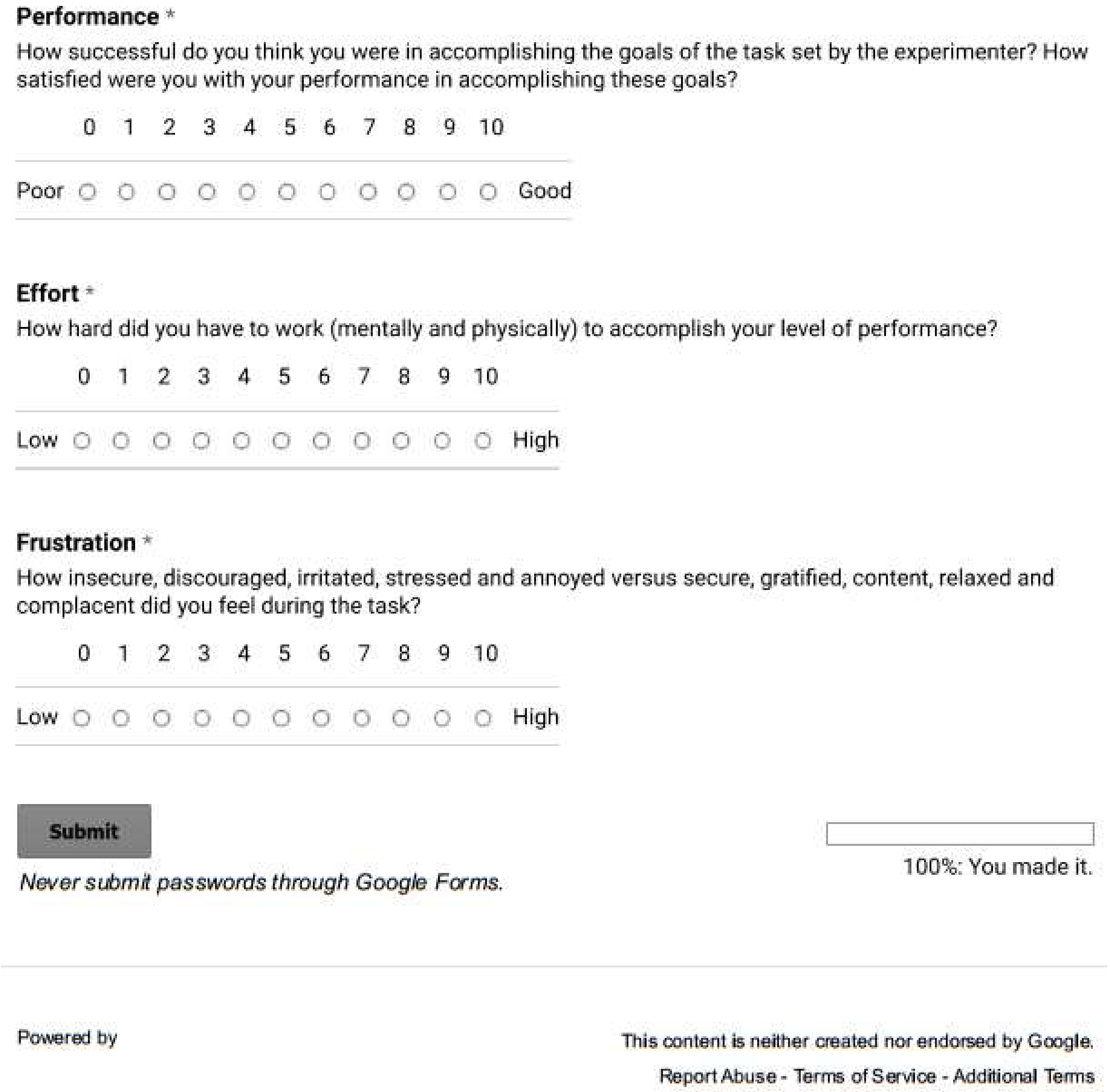
	0	1	2	3	4	5	6	7	8	9	10	
Low	○	○	○	○	○	○	○	○	○	○	○	High

Physical Demand *
How much physical activity was required (e.g., pressing, finger movement, controlling, activating, etc.) to perform the task? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

	0	1	2	3	4	5	6	7	8	9	10	
Low	○	○	○	○	○	○	○	○	○	○	○	High

Temporal Demand *
How much time pressure did you feel due to the rate at which the task elements occurred? Was the pace slow and leisurely or rapid and frantic?

	0	1	2	3	4	5	6	7	8	9	10	
Low	○	○	○	○	○	○	○	○	○	○	○	High



Performance *

How successful do you think you were in accomplishing the goals of the task set by the experimenter? How satisfied were you with your performance in accomplishing these goals?

	0	1	2	3	4	5	6	7	8	9	10	
Poor	○	○	○	○	○	○	○	○	○	○	○	Good

Effort *

How hard did you have to work (mentally and physically) to accomplish your level of performance?

	0	1	2	3	4	5	6	7	8	9	10	
Low	○	○	○	○	○	○	○	○	○	○	○	High

Frustration *

How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

	0	1	2	3	4	5	6	7	8	9	10	
Low	○	○	○	○	○	○	○	○	○	○	○	High

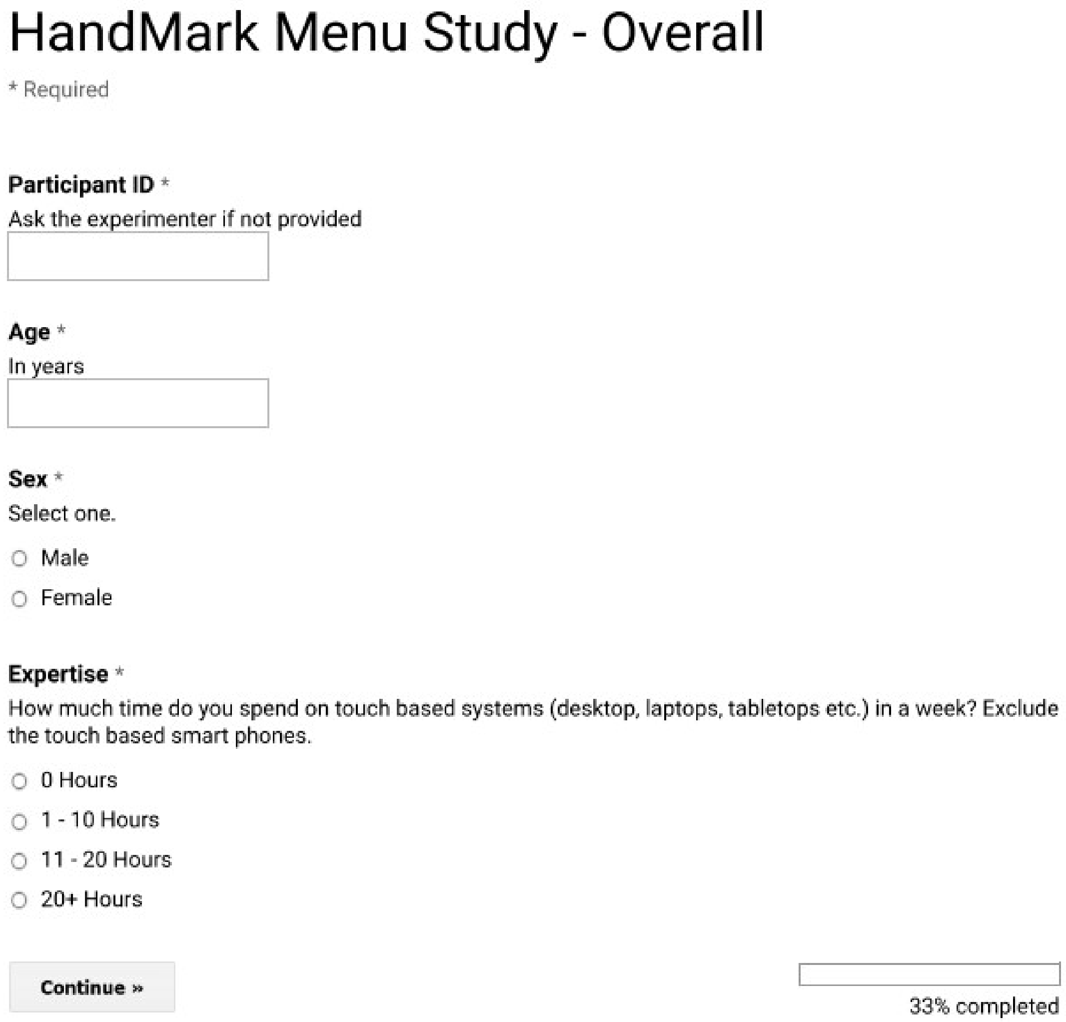
Never submit passwords through Google Forms.

100%: You made it.

Powered by

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms



HandMark Menu Study - Overall

* Required

Participant ID *

Ask the experimenter if not provided

Age *

In years

Sex *

Select one.

○ Male

○ Female

Expertise *

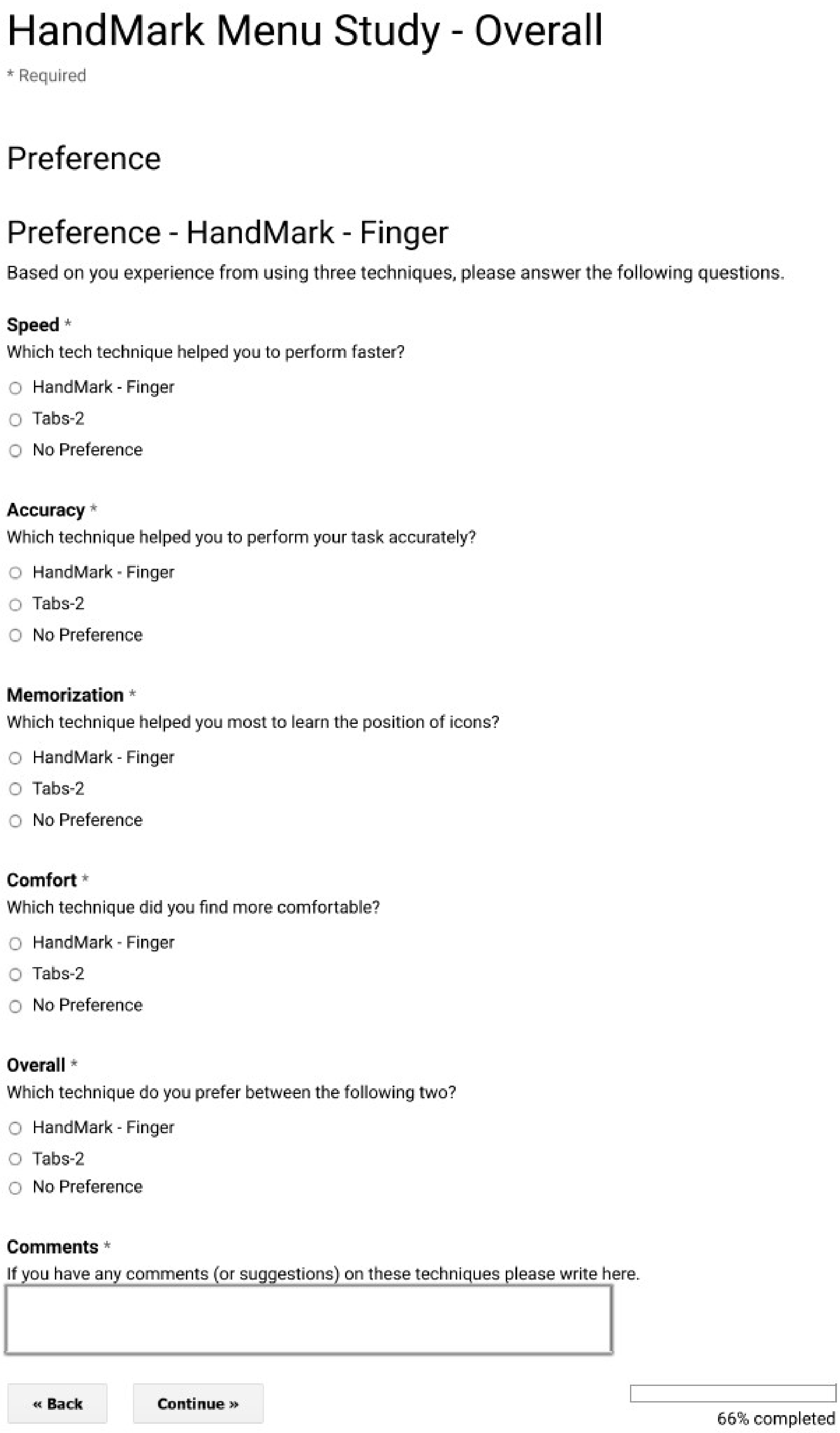
How much time do you spend on touch based systems (desktop, laptops, tabletops etc.) in a week? Exclude the touch based smart phones.

○ 0 Hours

○ 1 - 10 Hours

○ 11 - 20 Hours

○ 20+ Hours



HandMark Menu Study - Overall

* Required

Preference

Preference - HandMark - Finger

Based on you experience from using three techniques, please answer the following questions.

Speed *

Which tech technique helped you to perform faster?

○ HandMark - Finger

○ Tabs-2

○ No Preference

Accuracy *

Which technique helped you to perform your task accurately?

○ HandMark - Finger

○ Tabs-2

○ No Preference

Memorization *

Which technique helped you most to learn the position of icons?

○ HandMark - Finger

○ Tabs-2

○ No Preference

Comfort *

Which technique did you find more comfortable?

○ HandMark - Finger

○ Tabs-2

○ No Preference

Overall *

Which technique do you prefer between the following two?

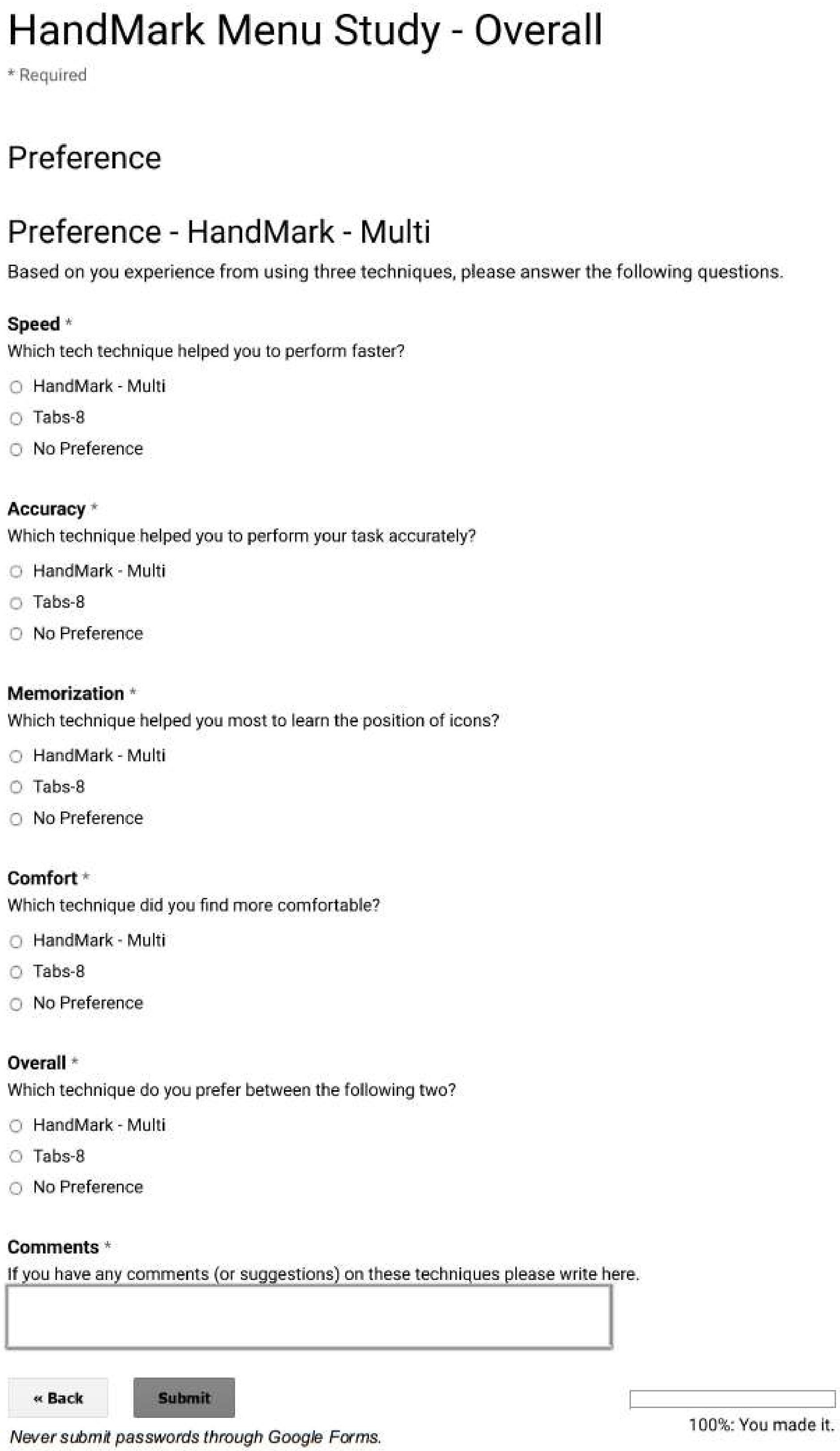
○ HandMark - Finger

○ Tabs-2

○ No Preference

Comments *

If you have any comments (or suggestions) on these techniques please write here.



HandMark Menu Study - Overall

* Required

Preference

Preference - HandMark - Multi

Based on you experience from using three techniques, please answer the following questions.

Speed *

Which tech technique helped you to perform faster?

- ○ HandMark - Multi
- ○ Tabs-8
- ○ No Preference

Accuracy *

Which technique helped you to perform your task accurately?

- ○ HandMark - Multi
- ○ Tabs-8
- ○ No Preference

Memorization *

Which technique helped you most to learn the position of icons?

- ○ HandMark - Multi
- ○ Tabs-8
- ○ No Preference

Comfort *

Which technique did you find more comfortable?

- ○ HandMark - Multi
- ○ Tabs-8
- ○ No Preference

Overall *

Which technique do you prefer between the following two?

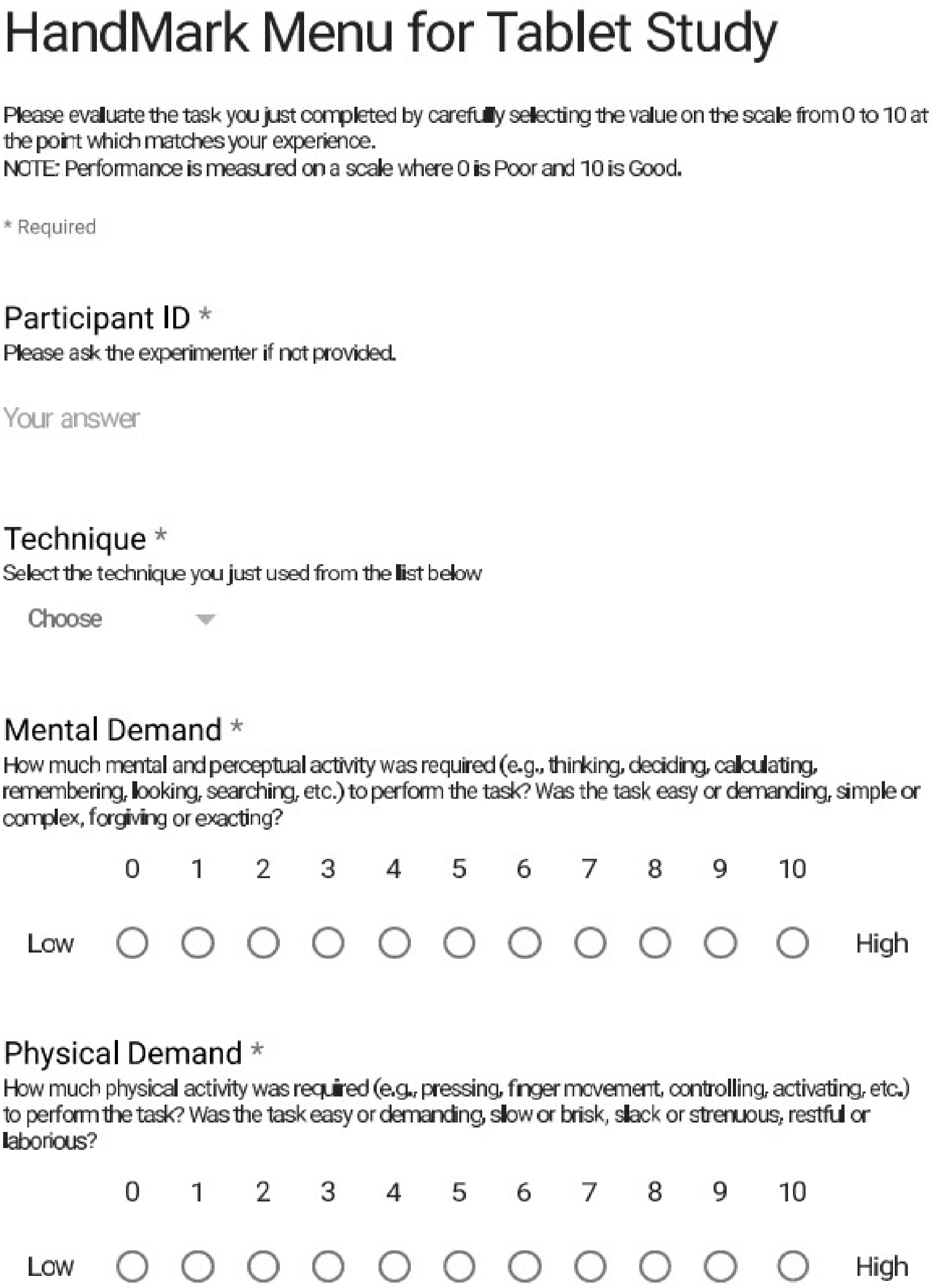
- ○ HandMark - Multi
- ○ Tabs-8
- ○ No Preference

Comments *

If you have any comments (or suggestions) on these techniques please write here.

100%: You made it.

Never submit passwords through Google Forms.



HandMark Menu for Tablet Study

Please evaluate the task you just completed by carefully selecting the value on the scale from 0 to 10 at the point which matches your experience.
NOTE: Performance is measured on a scale where 0 is Poor and 10 is Good.

* Required

Participant ID *

Please ask the experimenter if not provided.

Your answer

Technique *

Select the technique you just used from the list below

Choose ▾

Mental Demand *

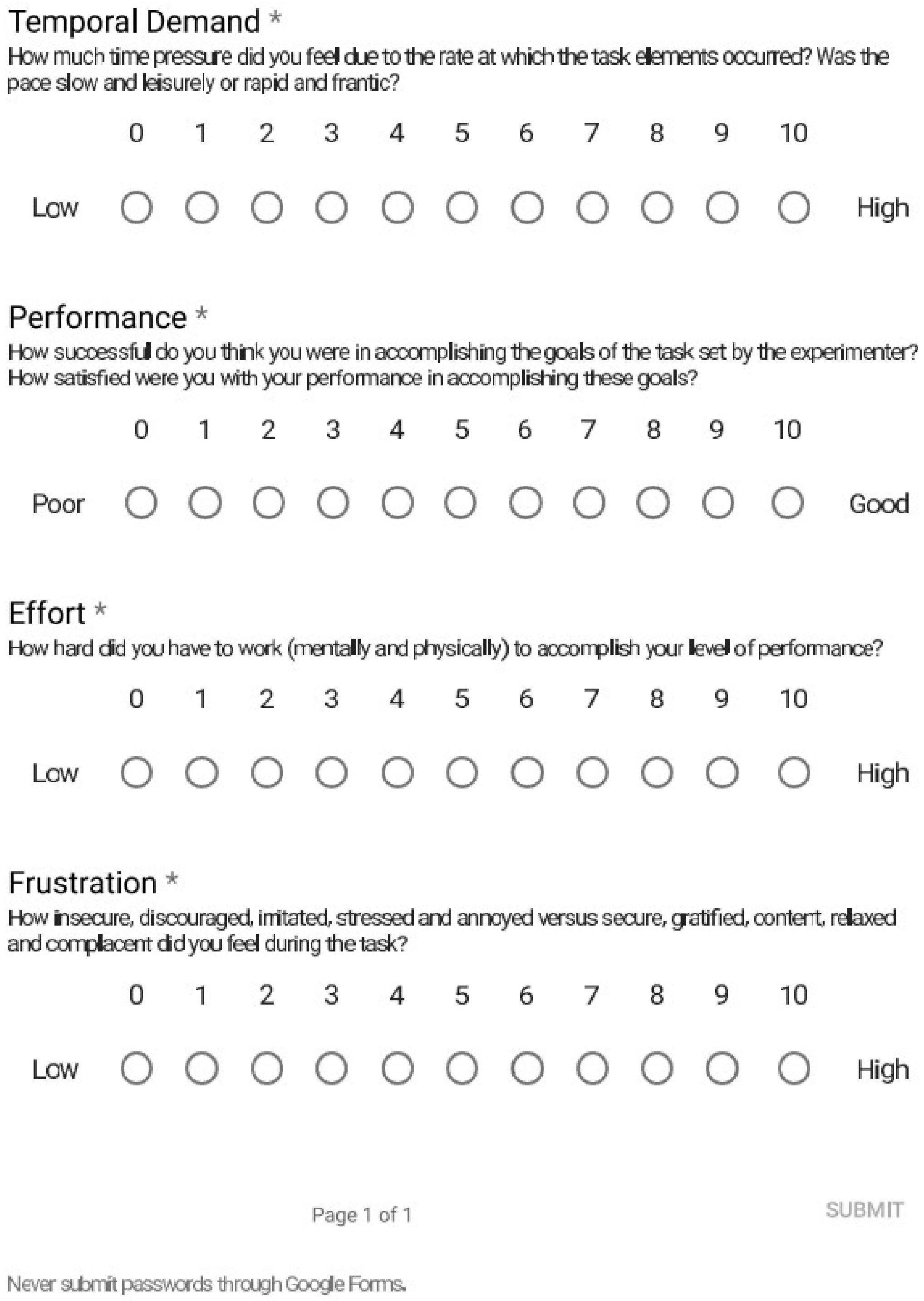
How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.) to perform the task? Was the task easy or demanding, simple or complex, forgiving or exacting?

	0	1	2	3	4	5	6	7	8	9	10	
Low	○	○	○	○	○	○	○	○	○	○	○	High

Physical Demand *

How much physical activity was required (e.g., pressing, finger movement, controlling, activating, etc.) to perform the task? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

	0	1	2	3	4	5	6	7	8	9	10	
Low	○	○	○	○	○	○	○	○	○	○	○	High



Temporal Demand *

How much time pressure did you feel due to the rate at which the task elements occurred? Was the pace slow and leisurely or rapid and frantic?

	0	1	2	3	4	5	6	7	8	9	10	
Low	○	○	○	○	○	○	○	○	○	○	○	High

Performance *

How successful do you think you were in accomplishing the goals of the task set by the experimenter? How satisfied were you with your performance in accomplishing these goals?

	0	1	2	3	4	5	6	7	8	9	10	
Poor	○	○	○	○	○	○	○	○	○	○	○	Good

Effort *

How hard did you have to work (mentally and physically) to accomplish your level of performance?

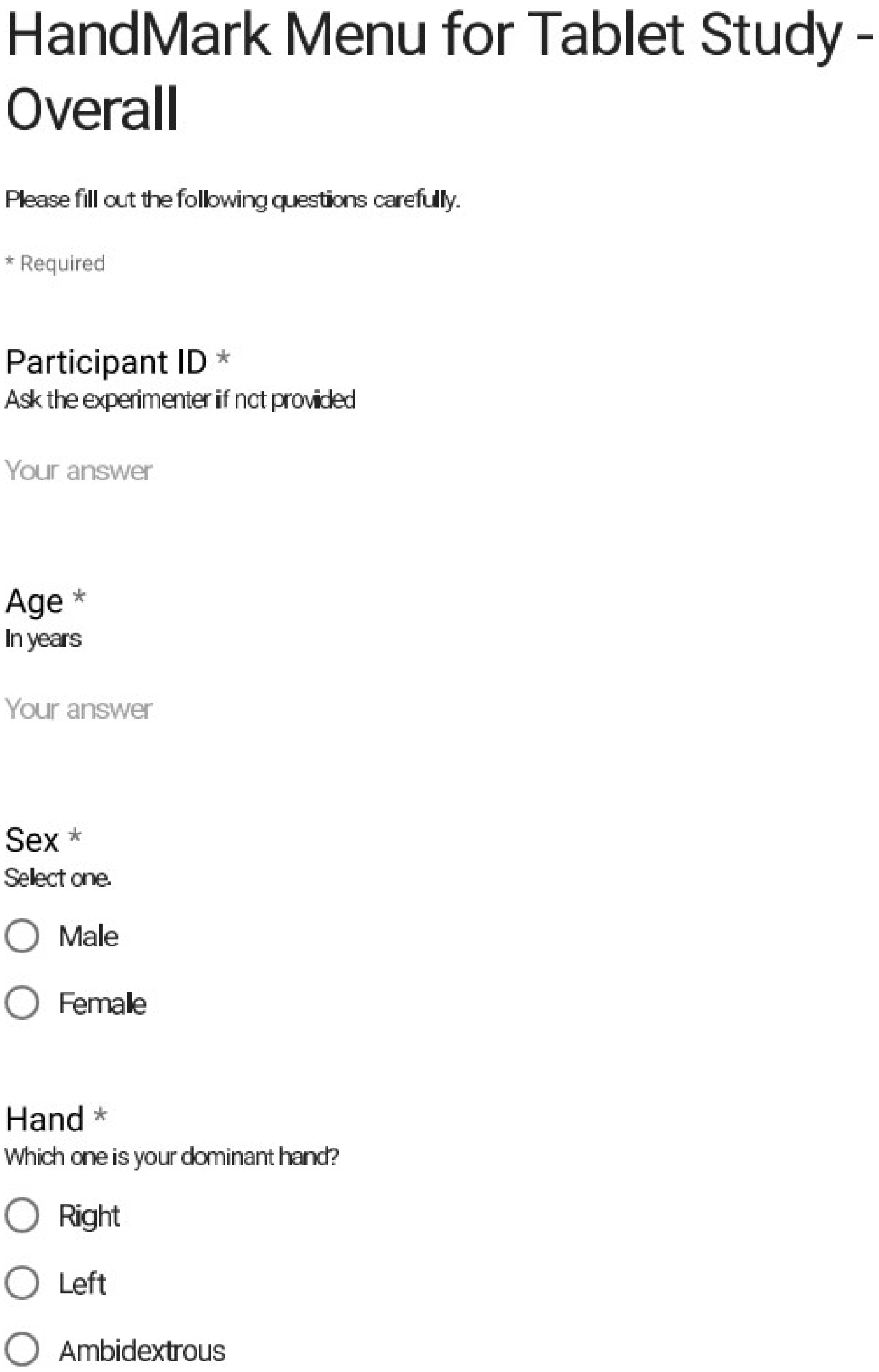
	0	1	2	3	4	5	6	7	8	9	10	
Low	○	○	○	○	○	○	○	○	○	○	○	High

Frustration *

How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

	0	1	2	3	4	5	6	7	8	9	10	
Low	○	○	○	○	○	○	○	○	○	○	○	High

Never submit passwords through Google Forms.



HandMark Menu for Tablet Study - Overall

Please fill out the following questions carefully.

* Required

Participant ID *

Ask the experimenter if not provided

Your answer

Age *

In years

Your answer

Sex *

Select one.

○ Male

○ Female

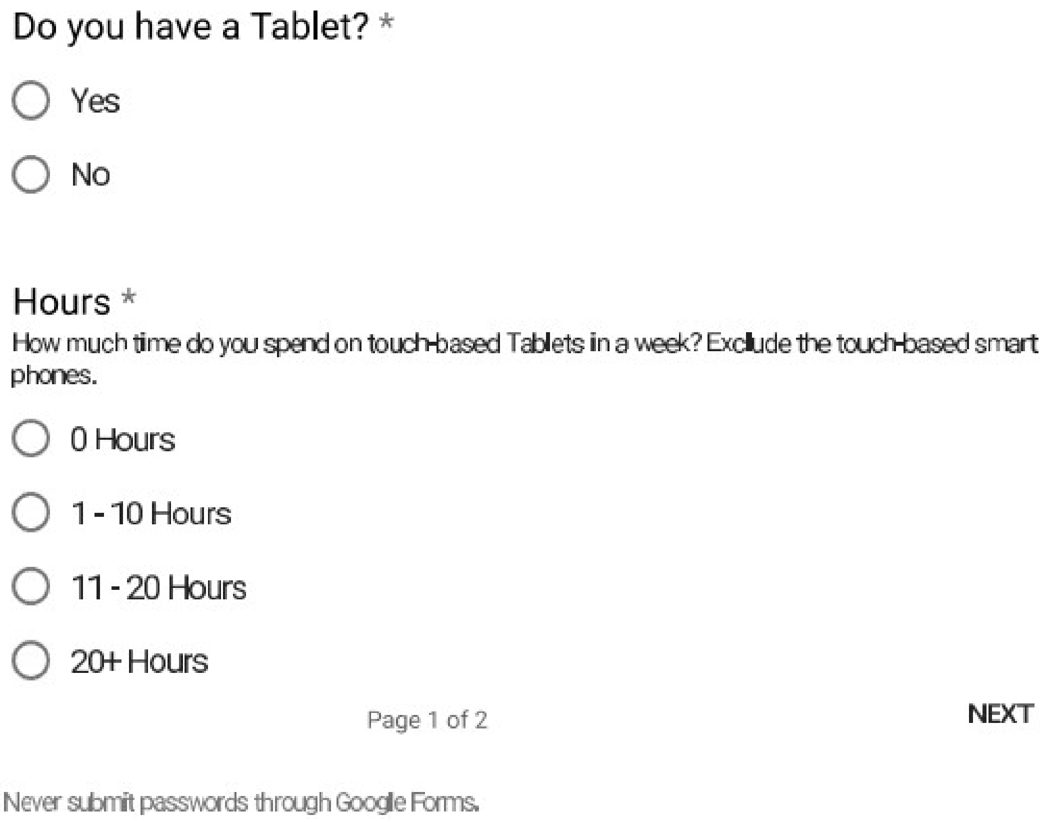
Hand *

Which one is your dominant hand?

○ Right

○ Left

○ Ambidextrous



Do you have a Tablet? *

○ Yes

○ No

Hours *

How much time do you spend on touch-based Tablets in a week? Exclude the touch-based smart phones.

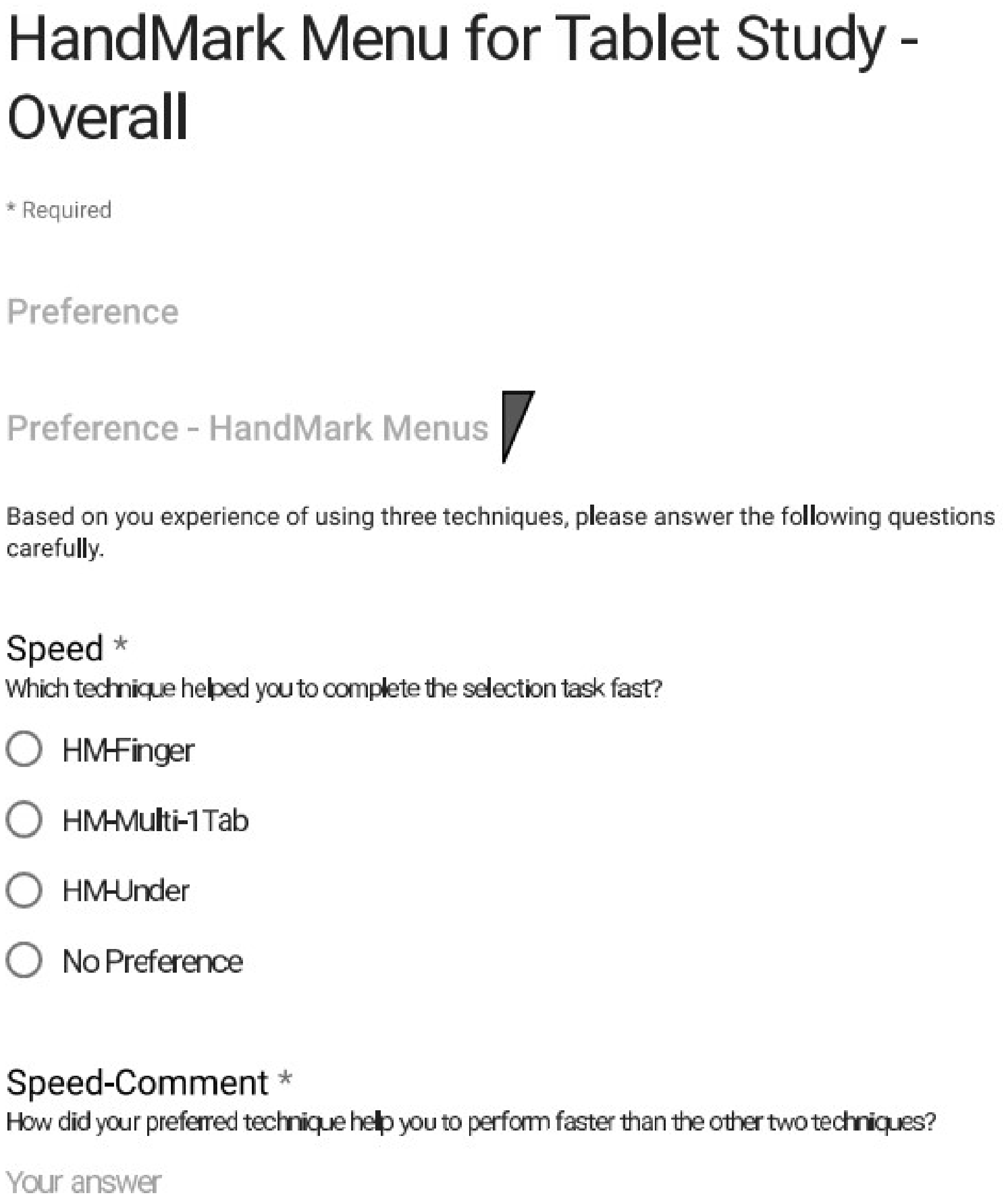
○ 0 Hours

○ 1 - 10 Hours

○ 11 - 20 Hours

○ 20+ Hours

Never submit passwords through Google Forms.



HandMark Menu for Tablet Study - Overall

* Required

Preference

Preference - HandMark Menus

Based on you experience of using three techniques, please answer the following questions carefully.

Speed *

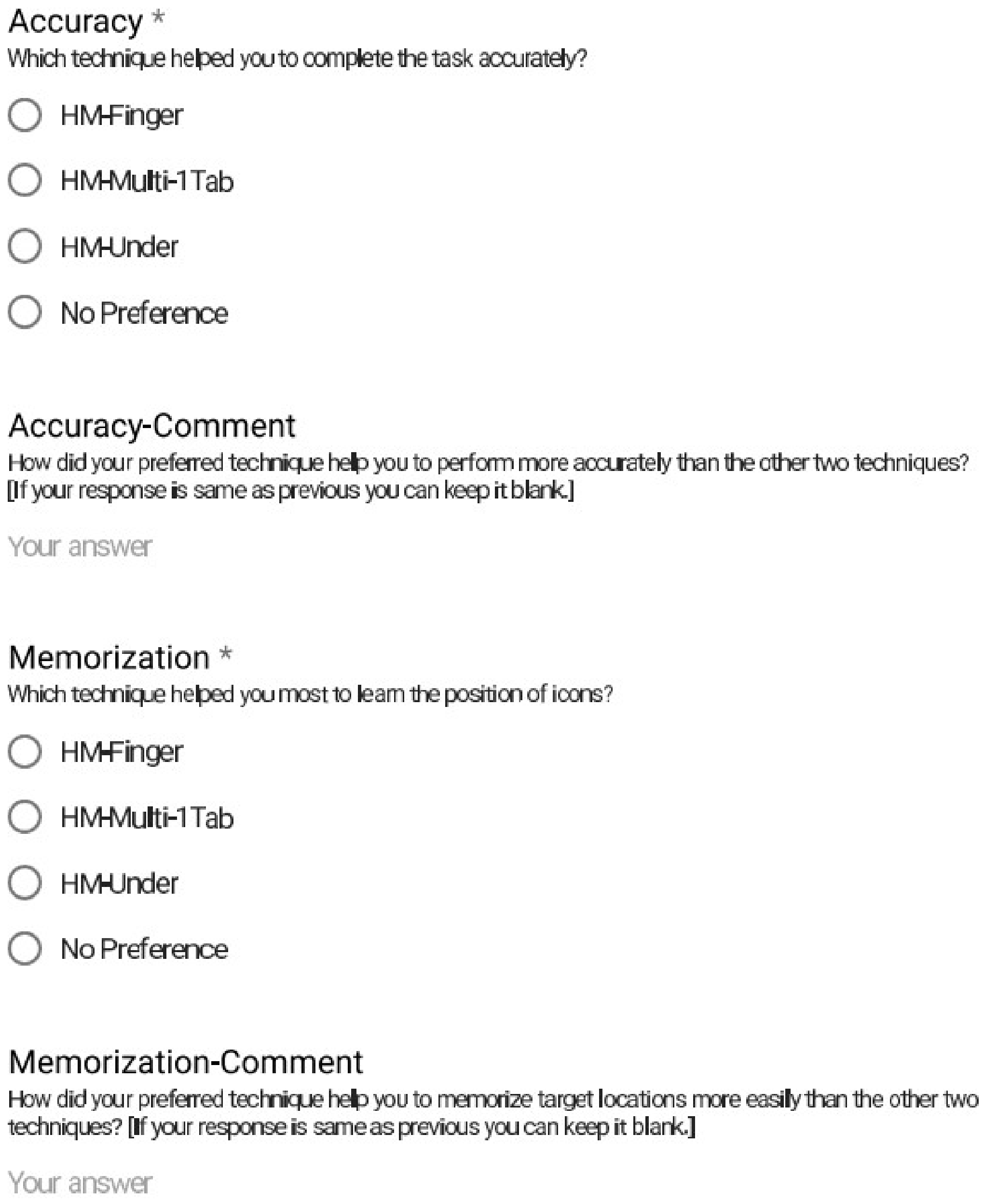
Which technique helped you to complete the selection task fast?

- ○ HM-Finger
- ○ HM-Multi-1Tab
- ○ HM-Under
- ○ No Preference

Speed-Comment *

How did your preferred technique help you to perform faster than the other two techniques?

Your answer



Accuracy *

Which technique helped you to complete the task accurately?

○ HM-Finger

○ HM-Multi-1Tab

○ HM-Under

○ No Preference

Accuracy-Comment

How did your preferred technique help you to perform more accurately than the other two techniques? [If your response is same as previous you can keep it blank.]

Your answer

Memorization *

Which technique helped you most to learn the position of icons?

○ HM-Finger

○ HM-Multi-1Tab

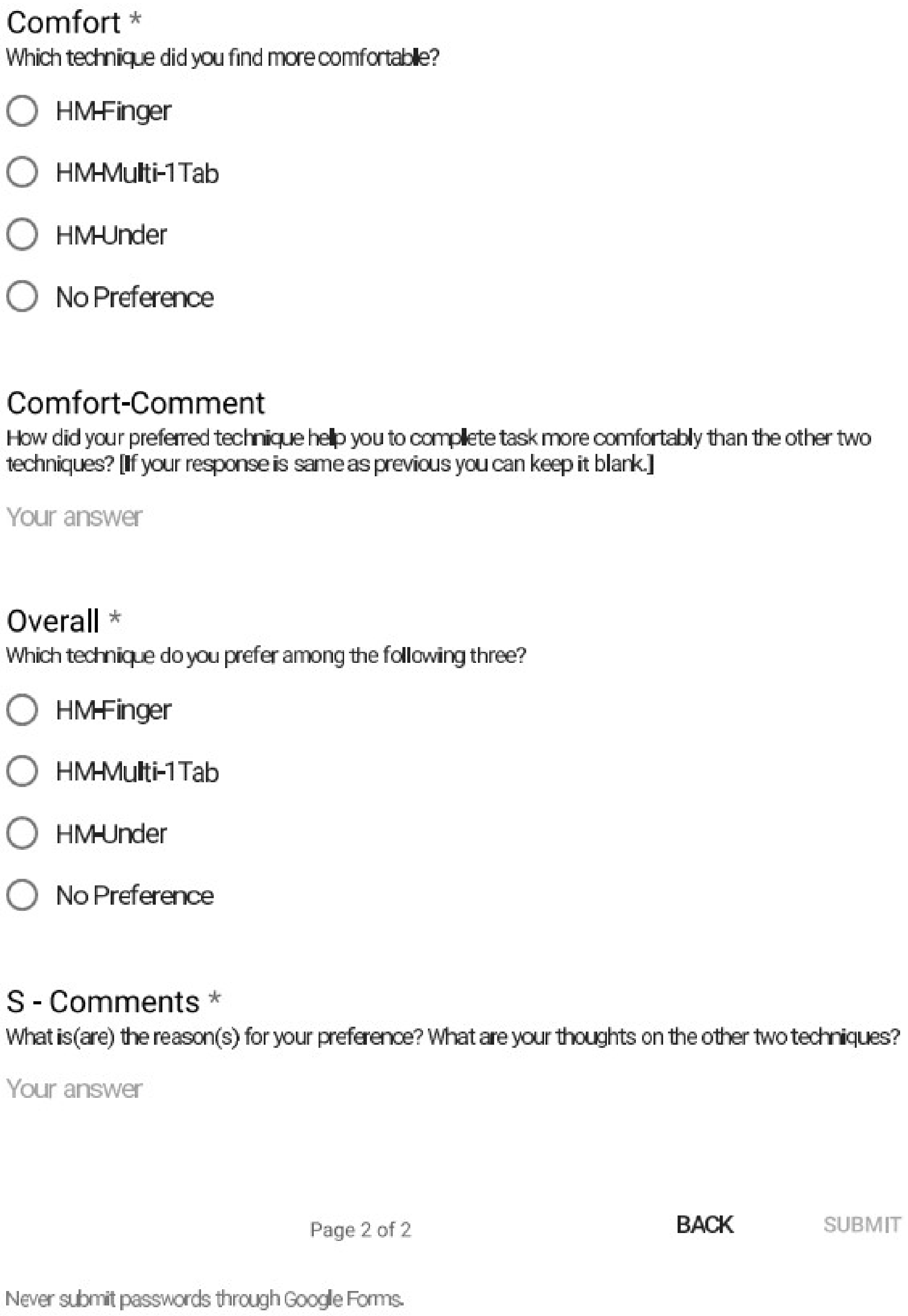
○ HM-Under

○ No Preference

Memorization-Comment

How did your preferred technique help you to memorize target locations more easily than the other two techniques? [If your response is same as previous you can keep it blank.]

Your answer



Comfort *

Which technique did you find more comfortable?

○ HM-Finger

○ HM-Multi-1Tab

○ HM-Under

○ No Preference

Comfort-Comment

How did your preferred technique help you to complete task more comfortably than the other two techniques? [If your response is same as previous you can keep it blank.]

Your answer

Overall *

Which technique do you prefer among the following three?

○ HM-Finger

○ HM-Multi-1Tab

○ HM-Under

○ No Preference

S - Comments *

What is(are) the reason(s) for your preference? What are your thoughts on the other two techniques?

Your answer

APPENDIX

ICON SETS USED IN STUDIES