

Understanding Mode and Modality Transfer in Unistroke Gesture Input

by

Jay Henderson

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2021

© Jay Henderson 2021

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Carl Gutwin
Professor, Department of Computer Science,
University of Saskatchewan

Supervisor: Edward Lank
Professor, School of Computer Science,
University of Waterloo

Internal Members: Daniel Vogel
Associate Professor, School of Computer Science,
University of Waterloo

Sylvain Malacria
Adjunct Assistant Professor, School of Computer Science,
University of Waterloo

Keiko Katsuragawa
Adjunct Professor, School of Computer Science,
University of Waterloo

Internal-External Member: Oliver Schneider
Assistant Professor, Department of Management Sciences,
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

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

Statement of Contributions

This dissertation includes first-authored peer-reviewed material that has appeared in conference and journal proceedings published by the Association for Computing Machinery (ACM). The ACM’s policy on reuse of published materials in a dissertation is as follows¹:

“Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included.”

The following list serves as a declaration of the works included in this dissertation. This material is expanded and revised from the original publication.

Portions of Chapter 3:

Jay Henderson, Sylvain Malacria, Mathieu Nancel, and Edward Lank. 2020. Investigating the Necessity of Delay in Marking Menu Invocation. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 1–13.

<https://doi.org/10.1145/3313831.3376296>

Portions of Chapter 4:

Jay Henderson, Sachi Mizobuchi, Wei Li, and Edward Lank. 2019. Exploring Cross-Modal Training via Touch to Learn a Mid-Air Marking Menu Gesture Set. In Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI ’19). Association for Computing Machinery, New York, NY, USA, Article 8, 1–9.

<https://doi.org/10.1145/3338286.3340119>

Portions of Chapter 5:

Jay Henderson, Jessy Ceha, and Edward Lank. 2020. STAT: Subtle Typing Around the Thigh for Head-Mounted Displays. In 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI ’20). Association for Computing Machinery, New York, NY, USA, Article 27, 1–11.

<https://doi.org/10.1145/3379503.3403549>

¹<https://authors.acm.org/author-resources/author-rights>. Accessed on July 26th, 2021.

Abstract

Unistroke gestures are an attractive input method with an extensive research history, but one challenge with their usage is that the gestures are not always self-revealing. To obtain expertise with these gestures, interaction designers often deploy a guided novice mode – where users can rely on recognizing visual UI elements to perform a gestural command. Once a user knows the gesture and associated command, they can perform it without guidance; thus, relying on recall. The primary aim of my thesis is to obtain a comprehensive understanding of why, when, and how users transfer from guided modes or modalities to potentially more efficient, or novel, methods of interaction – through symbolic-abstract unistroke gestures.

The goal of my work is to not only study user behaviour from novice to more efficient interaction mechanisms, but also to expand upon the concept of intermodal transfer to different contexts. We garner this understanding by empirically evaluating three different use cases of mode and/or modality transitions. Leveraging *marking menus*, the first piece investigates whether or not designers should force expertise transfer by penalizing use of the guided mode, in an effort to encourage use of the recall mode. Second, we investigate how well users can transfer skills between modalities, particularly when it is impractical to present guidance in the target or recall modality. Lastly, we assess how well users’ pre-existing spatial knowledge of an input method (the QWERTY keyboard layout), transfers to performance in a new modality.

Applying lessons from these three assessments, we segment intermodal transfer into three possible characterizations – beyond the traditional novice to expert contextualization. This is followed by a series of implications and potential areas of future exploration spawning from our work.

Acknowledgements

First, I would like to thank my PhD advisor, Edward Lank, for his guidance and support throughout my academic journey thus far. Above all, the freedom in exploration you have given me during my PhD has allowed me to foster confidence in my research ability that I never thought was possible.

Secondly, thank you to my committee, Daniel Vogel, Sylvain Malacria, Keiko Katsuragawa, Oliver Schnieder, and Carl Gutwin. Your feedback has been invaluable for improving my dissertation and I look forward to further debates in years to come.

To my colleagues, collaborators, and mentors at both the University of Waterloo and other institutions that I have been fortunate enough to work with — thank you for creating friendly and encouraging research environments for me to be a part of.

Thank you to all of the friends that I have made along the way and to the original “b’ys” who continue to support my dreams. Lastly, a special thank you to my family, for the unconditional love throughout.

Table of Contents

List of Figures	xi
List of Tables	xiv
1 Introduction	1
1.1 Terminology	4
1.2 Target Modes	5
1.2.1 The Marking Menu	5
1.2.2 The Word Gesture Keyboard	7
1.3 Research Questions	8
1.4 Contributions	11
1.4.1 Investigating Delay in Marking Menus	11
1.4.2 Cross-Modal Transfer to Mid-Air in Marking Menus	11
1.4.3 QWERTY Text Entry in a New Modality for HMDs	12
1.4.4 Characterizing Mode and Modality Transfer	13
1.5 Dissertation Outline	14
2 Literature Review	15
2.1 Gesture Classification	15
2.1.1 Gesture Taxonomy	16
2.2 Mode and Modality Transitions	18

2.2.1	Motor Learning	18
2.2.2	Novice to Expert Transfer	20
2.2.3	Marking Menus	21
2.2.4	Using Delay to Force Mode and Modality Transfer	21
2.3	When is Learning Necessary?	23
2.3.1	Mid-Air Interaction	23
2.3.2	Learning Display-less Gestures	24
2.4	Knowledge Transfer: The Word Gesture Keyboard	25
2.4.1	Head-Mounted Display Input	26
2.4.2	Text Entry Techniques	27
2.4.3	Gestural Text Entry for HMDs	29
3	Understanding the Necessity of Mode Transfer in Marking Menus	30
3.1	Motivation	30
3.2	Rationale for Delay in Rehearsal-Based Interfaces	32
3.2.1	Investigating the necessity of delay in marking menus	34
3.3	Rationale for testing no-delay Marking Menus	34
3.4	Study 1: Evaluating the Impact of Delay	37
3.4.1	Experimental Procedure	37
3.4.2	Overall Time and Accuracy	39
3.5	Study 2: investigating expert performance	44
3.5.1	Experimental Procedure	45
3.5.2	Results	46
3.6	Experiment 3: Assessing Visual Disruption	47
3.6.1	Experimental Procedure	48
3.6.2	Results	50
3.7	Discussion and conclusion	52

4	Presenting a Use Case of When Mode Transfer is Beneficial	55
4.1	Motivation	55
4.2	Assessing Touch-based Teaching of Mid-Air Gestures	57
4.2.1	Participants	58
4.2.2	Apparatus	58
4.2.3	Mid-Air Pointing	58
4.2.4	Task and Stimulus	59
4.2.5	Design and Analysis	60
4.3	Results	60
4.3.1	Error Rate	62
4.3.2	Time	62
4.3.3	NASA TLX	63
4.4	Discussion	64
4.4.1	Future Work	68
4.5	Limitations	68
4.6	Conclusion	69
5	Typing On The Thigh for HMDs	70
5.1	Introduction	70
5.2	Related Work	72
5.2.1	Around Thigh and In-Pocket Interaction	72
5.2.2	On Thigh Gestural Text Entry for HMDs	74
5.3	STAT Design	74
5.3.1	STAT Controller Design and Input	75
5.4	Experimental Protocol	78
5.4.1	Participants	79
5.4.2	Procedure	79
5.4.3	Measures	81

5.5	Results	81
5.5.1	Gesture vs. Tapping - Out of Pocket	81
5.5.2	In-Pocket vs. Out-of-Pocket	84
5.5.3	Subjective Preferences	86
5.6	Discussion	87
5.6.1	Comparison of Performance with Prior Work	87
5.6.2	Design Implications	88
5.6.3	Limitations	90
5.7	Conclusion	92
6	Lessons in Mode and Modality Transfer	93
6.1	Should we force users to transfer modalities?	94
6.2	When do we need to transfer modes/modalities?	99
6.3	Transferring to less efficient modalities	101
6.4	Transfer distance between modes/modalities	104
6.5	Conclusion	105
7	Conclusion	106
7.1	Should interaction designers force users to transition to a secondary mode?	106
7.2	Under what circumstances do users need to transition to a secondary mode?	107
7.3	Can we leverage existing interface expertise to assist in transition to a new mode in a new modality?	107
7.4	Recommendations	108
7.5	Limitations	108
7.6	Future Work	109
7.6.1	Penalty to other rehearsal-based interfaces	109
7.6.2	How far can expertise be transferred?	109
7.6.3	Quantifying the difference in modes/modalities	110
7.7	Concluding Remarks	110
	References	111

List of Figures

1.1	Invoking the <i>Copy</i> command in two separate modes.	2
1.2	Characterization of <i>intermodal expertise</i> , adapted from Scarr et. al's <i>Dips and Ceilings</i> [195].	3
1.3	The two modes with marking menus. Menu mode (right) and mark mode (left). From Gordon Kurtenbach's Doctoral dissertation: <i>The Design and Evaluation of Marking Menus</i> [127].	6
1.4	A visual depiction of the word gesture keyboard forming the word <i>quick</i> . The tracing of the word anchored on each individual letters, showing the symbolic nature of the gesture (left). The abstract visual form produced without the QWERTY keyboard context (right).	7
1.5	Illustration of our research path: questions, methodology, and main findings.	8
2.1	Unistroke gestures ranging in complexity, from simple (a) to complex (c) .	17
2.2	A reproduction from Wobbrock et al. [240], indicating percentage of gestures in each taxonomy category. From top to bottom, the categories are listed in the same order as they appear in Table 2. The form dimension is separated by hands for all 2-hand gestures.	19
2.3	Reproduced from Vatavu et al.'s <i>Nomadic Displays</i> [214]	25
2.4	Reproduced from Gustafson et al.'s <i>Imaginary Phone</i> as the process of creating a mental model for spatial transfer learning [87]	26
2.5	Reproduced from Zhu et al.'s <i>I'sFree</i> [257].	29
3.1	Top: Reproduction of figure 4.15 in [127]. Bottom: Our hypothesis on the time costs of a no-delay Marking Menu.	35

3.2	Average <i>Selection, Execution, and Preparation Times</i> per BLOCK and MARKING MENU condition. Error bars are 95 % CI.	41
3.3	Top: Distribution of modes by BLOCK in the DELAY condition. Bottom: Average <i>Error Rates</i> per BLOCK and MARKING MENU condition. Error bars are 95 % CI.	42
3.4	Effects of MENU CONDITION on <i>Execution time vs Gesture length</i> for each item. Mind the non-zeroed Y-axis on top.	46
3.5	An example of a user interacting with the drag and drop application. On top is the figure to recreate. On the bottom is the participant's current figure with an ongoing menu selection. Icons in this figure were designed by Freepik and Smashicons from www.flaticon.com	49
3.6	Likert scale questions (error bars are 95% CI).	52
4.1	Visualization of transferring gestures from touch to mid-air.	56
4.2	Interface interfaces for the TOUCH condition, showing the dragged motion path in white.	61
4.3	Interface for the MID-AIR condition on a monitor (external display), showing the movement motion path (i.e. pointer path) from the mobile device.	62
4.4	Experiment phase interface, always in MID-AIR on a monitor (external display). No visual guidance is provided.	63
4.5	Rate of correct selections across conditions (error bars indicate SD).	64
4.6	Time from prompt appearing to selection across conditions.	65
4.7	Time from beginning a gesture to selection across conditions.	65
4.8	Time from prompt appearing to selection across blocks by condition.	66
4.9	Training phase.	67
4.10	Experimental phase.	67
4.11	Overall NASA TLX scores across phases and conditions (error bars indicate SD). Training phase conditions are performing via touch or mid-air. Experimental phase is always performed in mid-air, conditions are mode which training phase was completed.	67
5.1	Sample interactions using STAT techniques – <i>STATSwype</i> on-thigh (left) and <i>STATTap</i> in-pocket (right).	71

5.2	Position of the arm or hand at resting state.	74
5.3	(a) A participant using STAT for text entry on the HMD; (b) A closeup of the STAT controller on a user's thigh, with the index finger being used on the top section trackpad as a cursor, and the thumb on the bottom section to press the button for an action; (c) The simulated pocket used for in-pocket interaction; (d) State diagram for user input using STAT; (e) The experimental interface used for performing text entry.	78
5.4	Distribution of participants' self-reported usage of word-gesture typing. . .	79
5.5	Each dependent measure across blocks and conditions (out of pocket). Error bars indicate a 95% confidence interval.	82
5.6	Categorical NASA TLX scores across conditions out of pocket. (O = Overall, PD = Physical Demand, TD = Temporal Demand, P = Performance, E = Effort, F = Frustration)	84
5.7	Each dependent measure for block 4 of out-of-pocket (SS out, ST out) and in-pocket conditions (SS in, ST in). Error bars indicate a 95% confidence interval.	85
6.1	Contextual menu zoom in.	94
6.2	Menu bar zoom in.	94
6.3	Hot key zoom in.	94
6.4	Trackpad pinch to zoom in.	94
6.5	Mode/modality transfer characterized by removing penalty for relying on recognition in rehearsal based interfaces.	96
6.6	Characterization of transferring between recognition and recall modes across modalities.	100
6.7	Characterization of transferring expertise from a higher ceiling modality to a lower ceiling modality.	103

List of Tables

2.1	Wobbrock et al.'s Taxonomy of Surface Gestures [240].	18
5.1	Summary of means by block and condition. (SS = <i>STATSwype</i> ; ST = <i>STATTap</i> ; in = in-pocket; out = out-of-pocket); HH = hand-held (T = tap, S = gesture). Note: Block 2 for regular on phone gestures is reported to control for those who were word-gesture typing novices.	83
5.2	Results of Bonferroni Post-hoc comparisons of CER for in-pocket vs. out-of-pocket. Mean differences (standard error) shown. ** indicates significance at the 0.01 level, and *** at 0.001. (Naming conventions follow Table 1).	86
5.3	Significant correlations between conditions for WPM. Naming conventions follow Tables 5.1 and 5.2. * indicates significance at the 0.05 level, ** at 0.01, and *** at 0.001.	86
5.4	Text entry speed of related techniques (WPM) in comparable (novice) learning stages. We note a challenge in this direct comparison with differing # of phrases. (WGK = Word Gesture Keyboards).	88
6.1	Examples of relative skill transfer distances from low to high.	104

Chapter 1

Introduction

In human-to-human interaction, gestures are used as expressive, nonverbal tools to convey an idea or intention. This can come in the form of simple gestures, such as a person placing their hand in front of them with fingers extended to communicate “stop”, to more extensive gesture sets such as American Sign Language (ASL) [206], that possess enough complexity to encompass an entire language. With the expressivity that gestures allow and the ubiquity of their appearance, gestural interaction has become a natural extension for humans to communicate with computing systems. It is for this reason that gestural interaction persists as a popular area of research in human-computer interaction (HCI).

Within HCI, gestural interaction can be broadly defined as a type of non-verbal communication with a system, where the use of bodily motion is used to invoke a command to a system. This could include the hands, head, feet, a finger etc. Alongside the natural extension of using gestures to mimic human-human interaction, these techniques allow for performance improvement in interaction [15, 16, 144, 127, 180, 186, 254, 255], and often garner the ability to provide display or eye’s free input [165, 143, 257]. Taking this into consideration, particularly when the gesture is symbolic or abstract, the most common use case is to serve as a *shortcut* method or *expert mode* to perform a command on a computing system.

When referring to *shortcuts*, we often describe the separation of command invocation into two separate modes. In other words, two interaction mechanisms to perform the same task. The first mode is geared toward the novice or casual user. These provide an easily discovered method of invoking a command, usually through some form of recognition within an interface. The second mode, or *shortcut* is targeted at more practiced, experienced users, who rely on recognition to perform an action associated with a command. As

an example, perhaps the most commonly used instance of two separate modes invoking the same command is the copy command, depicted in Figure 1.1a. In a typical desktop environment, the novice user will right-click to open a menu, navigate to the copy command, then left-click to select it. However, a user who is more familiar with the system can execute the hot-key *Ctrl C* or *Command C* (1.1b), as a shortcut — hence, performing the same command, just in a separate mode, geared to the user who has a higher level of expertise with the command.

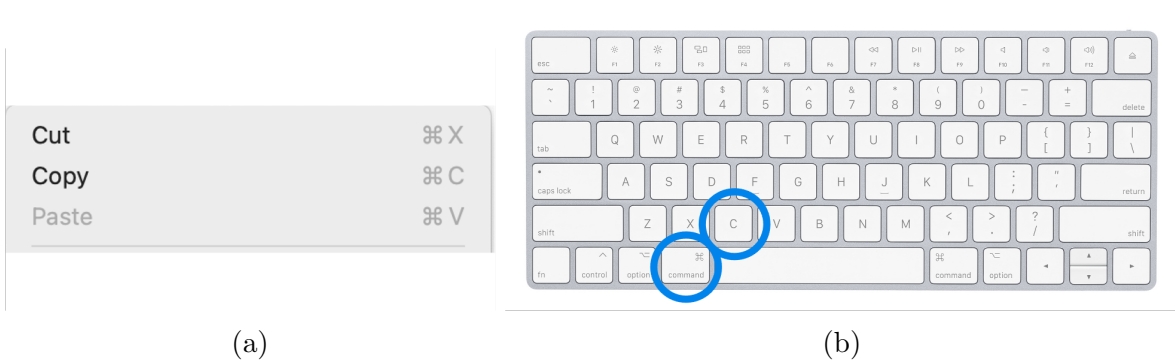


Figure 1.1: Invoking the *Copy* command in two separate modes.

When gestures are utilized as a *shortcut* or *expert mode*, similarly to the copy example, they may also be separated into two modes. As an illustration, let's say a *shortcut* could be drawing the letter 'X' on an image in a graphic design application indicated the command *Cut*. Whereas it's associated novice mode could be right-clicking and navigating a menu to select *Cut*. However, without explicitly reminding or telling the user that a second, potentially more efficient, mode exists, many will continue to use the initial mode. Which brings me to the question of, how can we support and better understand transitioning between modes? A command method of supporting the transition from a novice mode is to reveal the associated second mode while performing the novice mode. Revisiting the copy example, you can see, in Figure 1.1a, the interface displays an iconic key combination of the second mode, that is *Command C*.

Supporting and understanding this transition between modes has been of particular interest to the HCI community, with Scarr et al.'s 2011 paper introducing a framework of expertise development; and in particular *intermodal expertise*, which incorporates the transition from an initial mode to a second mode [195]. In the case of *shortcuts* or *expert modes*, their framework suggests that users will reach a ceiling or plateau in performance in an initial mode, and then to reach a higher degree of performance, they must switch to a second, expert mode. However, their framework postulates that a switch in modes will

come at a cost, which they refer to as a *performance dip* — essentially, to perform better, the user is likely to perform worse with the second mode than their ultimate performance with the first mode — but over time through practice, will surpass the performance ceiling of the initial mode [195].

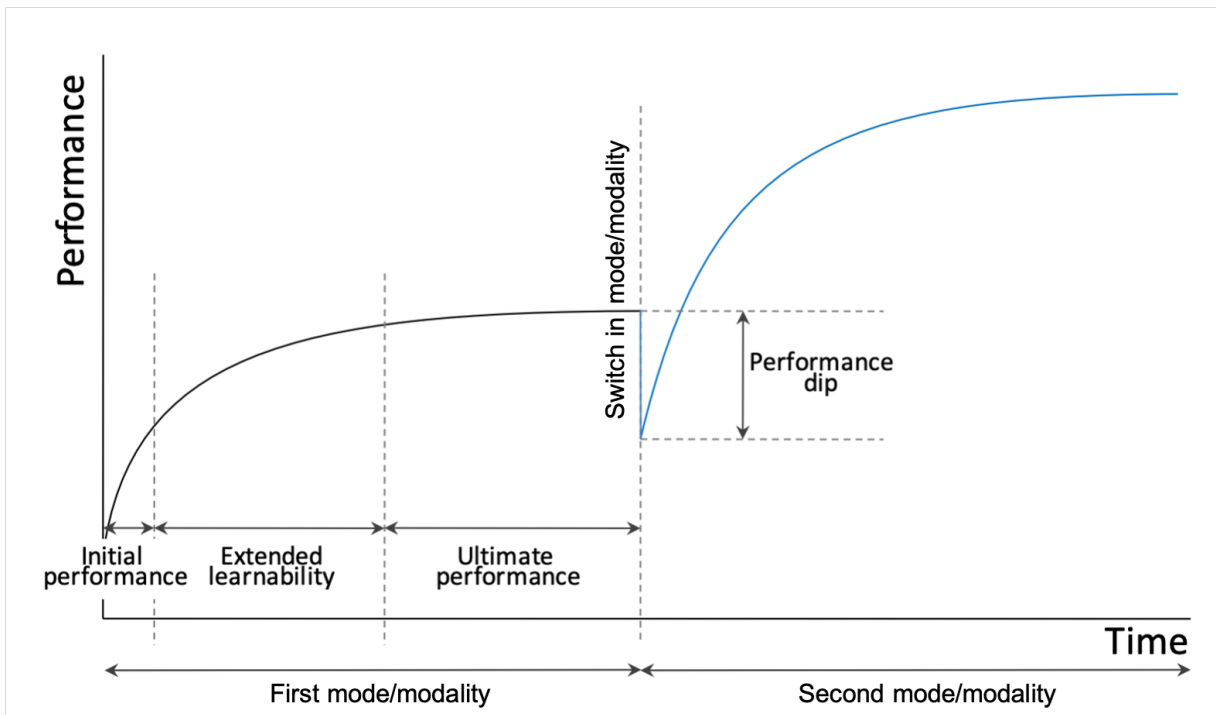


Figure 1.2: Characterization of *intermodal expertise*, adapted from Scarr et. al’s *Dips and Ceilings* [195].

The overarching goal of this thesis is to garner a better understanding of the transition between two interaction modes. In order to zero in on transition in a particular use case, I will leverage input techniques in the category of *symbolic-abstract unistroke gestures* for gaining insights into mode transfer. Symbolic-abstract gestures were chosen for their unique property of having no meaning associated with the command without prior interface experience, thus, we can obtain a greater understanding of the transition from an initial mode to the gestural input mode.

1.1 Terminology

Prior to introducing my primary research questions, I will present the following terminology to ensure sufficient grasp of the forthcoming sections.

1. **Modality** – While I acknowledge the discrepancy and controversy in the use of modality and mode in the field of human-computer interaction, for this thesis I use the definition of modality from Nigay et al. [162] as: the coupling of an interaction language L with a physical device d : $\langle d, L \rangle$. Examples of input modalities while using a mobile device include: \langle microphone, pseudo natural language \rangle , \langle inertial measurement unit, mid-air input \rangle , \langle capacitive touchscreen, surface/touch input \rangle .
2. **Mode** – Again, mode is also a complex and controversial term, with is no standardization in HCI. Thus, we extend upon Brewster et al.’s definition: *A mode is a state within a system in which a certain interpretation is placed on information* [37]. For the purpose of this dissertation, I will be focusing a specific type: two or more modes that can produce the same output via different methodologies.
3. **Intermodal Transfer** – The concept of switching between modes and/or modalities that produce the same output, as characterized by Scarr et al. [195].
4. **Unistroke Gesture** – A unistroke gesture can be defined as a single, continuous, stroke with a starting point and ending point, provided by some input mechanism (e.g. finger, hand, pen, or mouse). More technically speaking, these gestures are an ordered series of points in a 2- or 3-dimensional space.
5. **Symbolic Gesture** – We leverage Wobbrock et al.’s *Taxonomy of Surface Gestures* [240] to define a symbolic gesture as a gesture that visually depicts a symbol. Examples are tracing a caret (“^”) to perform insert, or drawing an “X” to perform a cut command.
6. **Abstract Gesture** – Utilizing Wobbrock et al.’s *Taxonomy of Surface Gestures* [240], we define an abstract gesture as a gestures that has no symbolic, physical, or metaphorical connection to their associated command. The mapping is arbitrary.
7. **Symbolic-Abstract Gesture** – A unique case of gesture input that visually depicts a symbol, but, without pre-existing knowledge or training with a particular user interface, the depiction’s meaning has no association with the command. These types of gestures are inbetween classifications of symbolic and abstract gestures (definitions 5. and 6.).

Examples include marking menu gestures (section 2.2.3) and word gesture keyboards (section 2.4).

8. **Touch Input** – We define touch input as input requiring contact of a bodily limb (usually a finger) with a physical 2 dimensional surface; e.g. a trackpad on a laptop, a touchscreen mobile phone, or touchscreen tablet. Throughout the thesis, touch input will be used interchangeably with surface input. For the purposes of the current document, this will not encompass interaction with physical buttons.
9. **Mid-Air Input** – We define mid-air input as motion interaction in free space that does not require making physical contact with a surface with their input device. This could be via the users barehand, arm, limb, etc. or via some form of physical controller that does not require contact with a separate surface (e.g. the WiiMote). This can also be referred to as “in-air” input.
10. **Head-Mounted Display (HMD)** – A head-mounted display, is display device built into glasses, a headset, or helmet, to be worn on the head for presentation of extended virtual content. The display can present exclusively virtual content (virtual reality or VR), overlaying virtual content onto the real world (augmented reality or AR), or anchoring virtual content within the real world (mixed reality or MR).

1.2 Target Modes

Throughout the document, I will focus on two main “target” modes for knowledge transfer, *Marking Menus* [127] and *Word Gesture Keyboards* [122], both of which rely on *unistroke gestures*. These gesture sets were chosen for the unique property of being classified as both symbolic and abstract gestures.

1.2.1 The Marking Menu

Similarly with other novice to expert mode transfer systems, marking menus rely on two main modes of selection: a novice mode (“menu mode”), in the form of a radial menu, and an expert mode (or “mark mode”). In the novice mode, the user presses down their input device, waits for a pre-determined delay for the menu to appear, then moves their device in the direction of a desired command; thus, in this mode, the user can rely on recognition of the menu items for selection. In expert mode, if the user knows the directional unistroke gesture corresponding to the location of the menu item, they can complete the selection

without waiting for the delay — meaning the user must solely rely on recall (or memory) of the command. A visual representation of selection is presented in Figure 1.3.

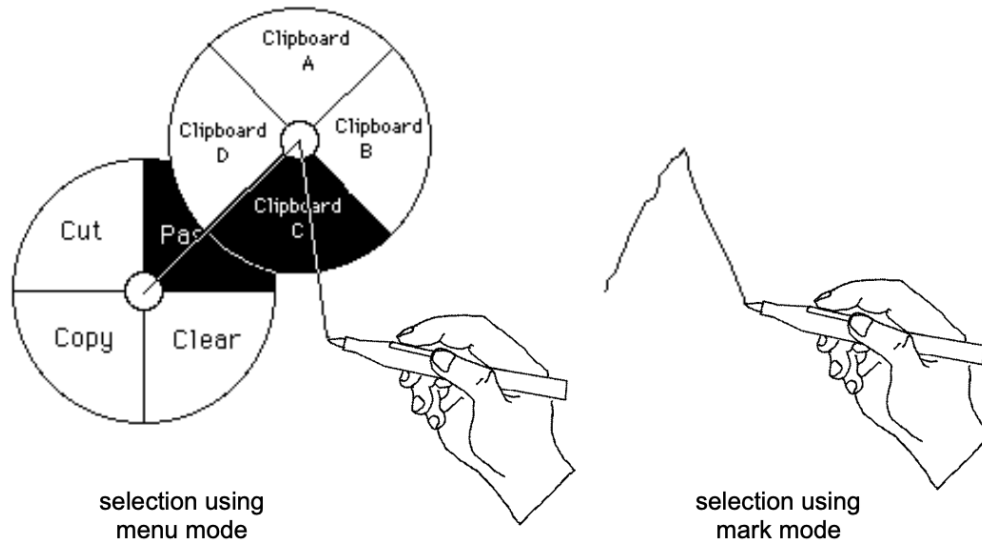


Figure 1.3: The two modes with marking menus. Menu mode (right) and mark mode (left). From Gordon Kurtenbach’s Doctoral dissertation: *The Design and Evaluation of Marking Menus* [127].

A unique property of this type of unistroke gesture, is while it is symbolic in nature (i.e. visually depicting where the menu item is located), without prior experience with a particular marking menu system, the user would have no idea what the gesture is mapped to. In other words, the gesture is arbitrarily mapped, or abstract. This contrasts other symbolic gestures, like drawing an ‘X’ to indicate a *Cut* command, as ‘X’ is a common iconic representation for some form of deletion. For this reason, we classify marking menu gestures as *symbolic-abstract* unistroke gestures. Since the mapping is arbitrary, this creates distinctive advantages over other unistroke gestures for modelling user behavior, as users can truly be taken from a true novice state (with no pre-existing knowledge of how commands may be mapped to gestures), through extended learning, to ultimate performance.

1.2.2 The Word Gesture Keyboard

Most soft-keyboards on mobile devices now employ two primary modes for text entry: the standard tap-based text entry, and word gesture text entry – both of which appear on a QWERTY style keyboard for the English language. On a word gesture keyboard, instead of tapping individual keys the user can press down their input device (usually a finger), and drag it to each individual letter on the keyboard, followed by releasing the input device once the word or phrase is complete [251]; forming a unistroke gesture representation of the word. In the same design space as marking menus, the premise, is that word-gesture keyboarding will shift from this primarily visual-guidance driven letter-to-letter tracing or typing, to recall-based gesturing [251]. We also categorize the word gesture keyboard as symbolic-abstract, as the produced trace or path forms a representation of the word on a QWERTY keyboard, without pre-existing experience with or a visual representation with the QWERTY layout, the gesture is abstract in nature (i.e. deciphering a meaning from the shape is nearly impossible, see Figure 1.4).

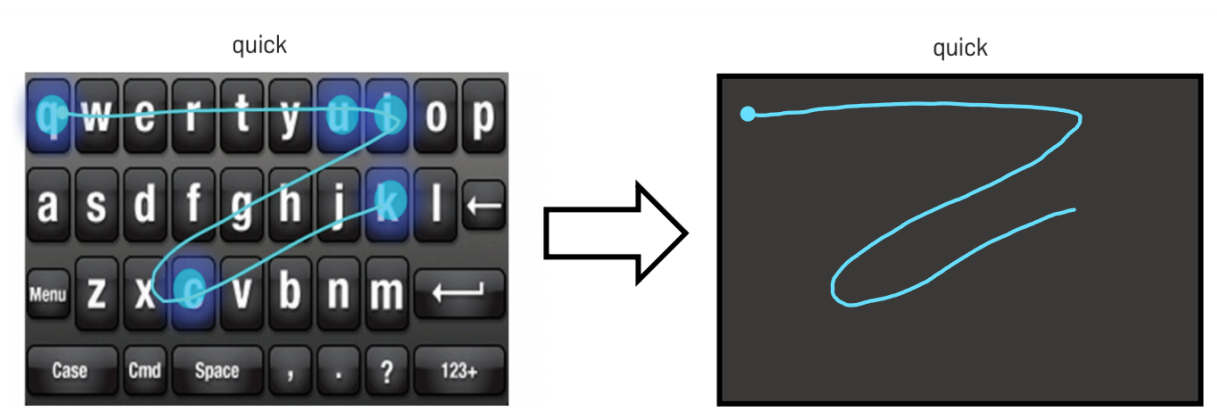


Figure 1.4: A visual depiction of the word gesture keyboard forming the word *quick*. The tracing of the word anchored on each individual letters, showing the symbolic nature of the gesture (left). The abstract visual form produced without the QWERTY keyboard context (right).

These two gesture sets provide a unique opportunity as a use case for studying mode transition in the case of unistroke gestures, as they require prior experience with an interface to allow transferring expertise to the gesture mode of interaction. They can both be classified as symbolic-abstract gestures, for reasons described prior, and are part of a larger set of interaction techniques called *Rehearsal Based Interfaces*, which attempt to reduce the performance dip when switching between modes (see Figure 1.2) by rehearsing

the interaction in a recognition-based mode prior to transferring that expertise to a recall-based mode. I will discuss the concept of rehearsal based interfaces in further depth in Section 2.2.

1.3 Research Questions

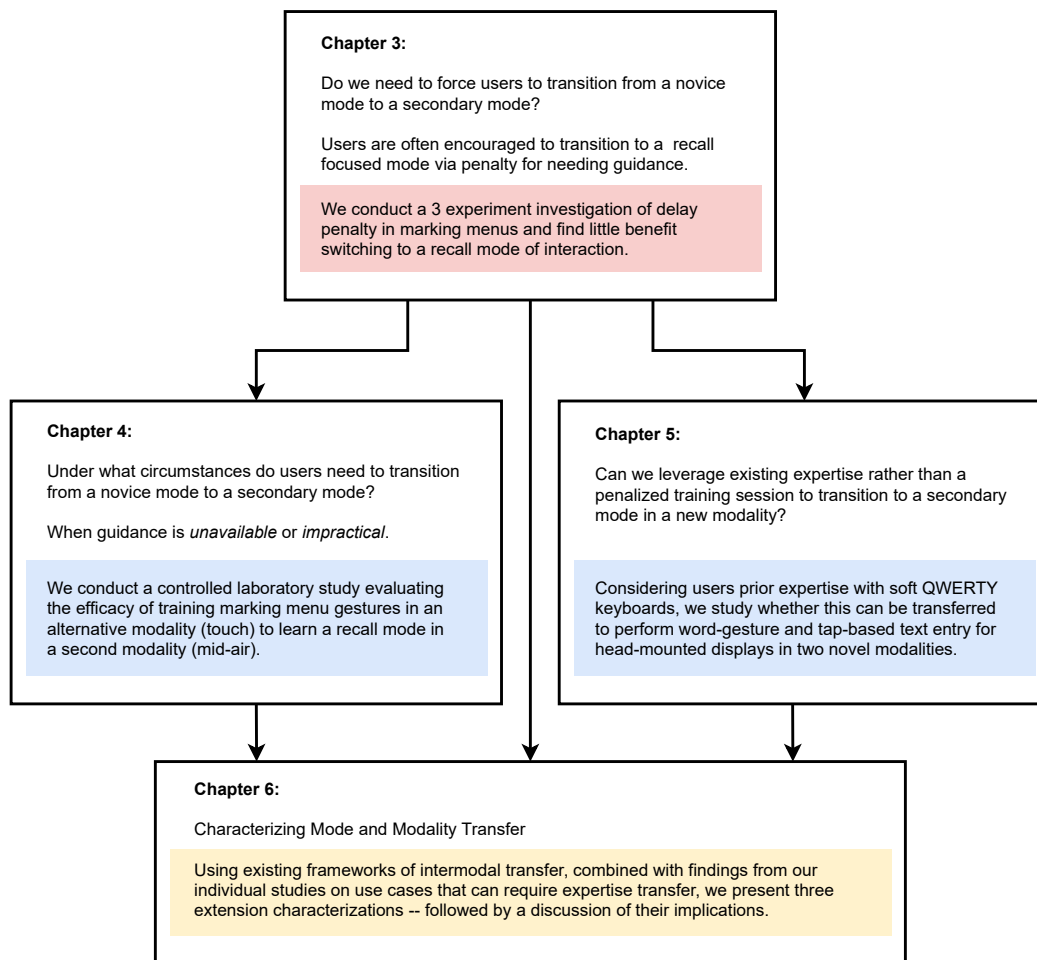


Figure 1.5: Illustration of our research path: questions, methodology, and main findings.

As stated prior, the primary aim of my thesis is to obtain a comprehensive understanding of transition between modes in the use case of symbolic-abstract unistroke gestures. In contrast to much of the prior literature, which has largely focused on the transition from an initial novice mode, to a more efficient mode (see Figure 1.2), the goal of my work is to not only study user behaviour in novice to more efficient modes, but also to expand upon intermodal transfer to different contexts. For instance, while one mode may be logical in a particular scenario – for instance mouse input to a desktop computer – this mode may not be realistic in another scenario, for example, a user interacting with their smartglasses while sitting on a train. We aim to broaden the scope of mode transfer from the traditional concept of a novice mode to a faster, more efficient mode, to observing transfer from a familiar mode to modes that may compromise speed and performance, but propose benefits in emerging ubiquitous scenarios.

Thus, throughout the document I will be answering the following research questions, further depicted in Figure 1.5.

RQ 1. Should interaction designers force users to transition from a novice mode to a secondary mode?

Much of the prior literature apply an artificial penalty to the novice user, to encourage or force the user to switch to a potentially more efficient mode — usually by hampering temporal performance with a delay [15, 16, 71, 90, 127, 132, 150].

Leveraging the use case of marking menus, we address the larger research question of *should interaction designers force users to transition from a novice mode to a new mode?* via the following:

- Do users prefer being penalized or not being penalized?
- In the general use case, do users perform better when relying on recognition rather than recall? In other words, do users perform better when remaining in a single mode or with two mode options?
- Can users reach optimal performance without switching modes?
- Do expert users perform better when the menu is not visible? How much better is performance?
- Do users prefer having having a single mode or two modes?
- Do users find the recognition or menu mode visually disrupting to their task at hand?

RQ 2. Under what circumstances do users need to transition from a novice mode to a secondary mode?

Again, within the use case of marking menus, we question *in what circumstances should users be required to transfer modes?*. In terms of marking menus, the two modes are a visual, or recognition, mode and a recall-based mode. Thus, we imagine modalities where presenting a visual mode is challenging, such as mid-air interaction, would require the transition to a memory or recall-based mode in the new modality.

- Can we leverage a modality where presenting a displayed menu mode is easy, to transfer expertise to a recall mode in an modality where presenting a display is difficult?
- Can using a familiar modality for novice mode ease transition to a new modality in a secondary mode?
- How does mode transfer compare across modalities and within a single modality? Is there a cost to learning across modalities?

RQ 3. Can we leverage existing interface expertise to assist in transition to a secondary mode in a new modality?

As a more complex symbolic-abstract gesture, we wonder whether we can take advantage of user's pre-existing expertise with the QWERTY keyboard layout and transfer it to new modalities? In particular, can we transfer this to a new, emerging, modality: interaction with a head-mounted display.

- Can users transfer expertise with the QWERTY keyboard layout to perform tap text entry in a new modality?
- Can users perform word gesture text entry in a new modality using their existing expertise with QWERTY?
- How does a more familiar, novice mode (tap text entry) compare to a secondary mode (word gesture text entry) in a new modality?
- How far can we push this transfer? After rehearsal in the new modality, can users perform these interactions in a similar, more physically constrained environment?

1.4 Contributions

In this dissertation, we make the following research contributions:

1.4.1 Investigating Delay in Marking Menus

Penalizing the novice user via delay is a core design component of marking menus, to encourage the user to switch to a secondary mode (mark mode) and, arguably, to prevent visual disruption of displaying the novice mode (menu mode). Through this particular use case, this work aims to understand RQ 1, that is: do we need to force users to transition from menu mode to mark mode? In other words, is temporal penalization actually necessary to reach optimal performance with the interaction technique? We investigate the initial assumptions from the development of marking menus, by contrasting the original marking menu design (modes separated by delay) with immediately-displayed marking menus (uni-modal, without delay) in three within-subjects experiments.

- **Experiment 1:** Using a prompt-react selection task with varying menu configurations across the two delay conditions, we found an overall performance improvement of both time and error rates for an immediately displayed, uni-modal marking menu.
- **Experiment 2:** To simulate expert performance, we used a highly constrained setting: significant training with only two menu items to learn in a two-level marking menu in a selection task. We found a slight time improvement for the delay separated modes by 260ms.
- **Experiment 3:** In visually crowded, drag-and-drop style, graphic editing interface, we found delay separated marking menus caused significantly more “loss of focus” on the task than immediately displayed marking menus. Additionally, we failed to reveal any evidence validating the initial claim – that a non-delayed marking menu would be more visually disruptive.

1.4.2 Cross-Modal Transfer to Mid-Air in Marking Menus

While mid-air gestures are an attractive modality with an extensive research history, one challenge with their usage is that the gestures are not self-revealing. Scaffolding techniques to teach these gestures are difficult to implement since the input device, e.g. a hand, wand or arm, cannot present the gestures to the user. This project aims to present a potential

use case for addressing RQ2: under what circumstances do users need to transition from a novice mode to a secondary mode? Since in marking menus, the novice mode has a visual display, but the secondary (mark mode) does not, mid-air gestures appear an ideal use case that could benefit from transitioning to the secondary mode. In contrast to in-air input, when interacting via touch gestures, feedforward mechanisms (such as Marking Menus or OctoPocus) have been shown to effectively support user awareness and transition to a secondary, recall-based, interaction mode.

Through a controlled, between-subjects experiment, we explore whether touch marking menu input can be leveraged to teach users to perform mid-air marking menu gestures via a smartphone controller. We show that marking menu touch gestures transfers directly to knowledge of mid-air gestures, allowing performance of these gestures without intervention, with only a slight initial performance dip. We argue that cross-modality learning can be an effective mechanism for introducing users to mid-air gestural input.

1.4.3 QWERTY Text Entry in a New Modality for HMDs

In head-mounted display (HMD) interaction, text entry is frequently supported via some form of virtual touch, controller, or ray casting keyboard. While these options effectively support text entry, they often incur costs of additional external hardware, awkward movements, and hand encumbrance. The goal of this work is to answer RQ 3: can we leverage existing interface expertise, i.e. the QWERTY keyboard layout, to assist in transition to a secondary mode (word gesture text entry) in a new modality? This work expands upon the prior cross-modality work to another use case — head-mounted displays — where providing input is challenging. Rather than training in-lab, we question whether the user’s existing soft-typing expertise can transfer to text entry performance with a mobile device mounted on the user’s thigh.

In a within-subjects study, we compare the two interaction modes: tap text entry and word gesture text entry in the new on-thigh modality for HMDs. We found that, while words per minute did not significantly differ, tap based text entry appears to plateau (or reach a *ceiling*) in performance, where as word gesture text entry shows potential for further improvement. However, each mode has their place, as word gesture typing required significantly more corrections than tap typing. Lastly, after rehearsal atop the thigh, users were able to transfer that expertise to perform each mode with the mobile device placed in their front pocket (a more constrained environment), with a very small cost in performance.

1.4.4 Characterizing Mode and Modality Transfer

Taking into consideration lessons from answering our three primary research questions, outlined in Section 1.3, we use this section to address potential adaptations to the characterization of intermodal expertise from Scarr et al.'s Dips and Ceilings. In particular, the extension of intermodal or cross-modal expertise to modalities that may not reveal a performance increase, but possess external benefits based on context. Leveraging prior work in conjunction with our findings, we then characterize performance dip as a function of differences in mode and/or modality, for a increased understanding of transitioning between input methods.

1.5 Dissertation Outline

The remainder of the thesis is structured as follows:

- **Chapter 2: Literature Review.** We review relevant prior background literature, particularly in the space of gestural interaction styles, mode/modality transitions, and techniques that can be leveraged to introduce users to interactive techniques – particularly when they are categorized as abstract or non-self revealing.
- **Chapter 3: Understanding the Necessity of Mode Transfer in Marking Menus.** We conduct an in depth, 3 experiment, investigation of whether users should be forced into transitioning to a recall-based mode, as opposed to relying on recognition in menu mode. This is followed by a discussion of the implications of these findings for other rehearsal based techniques.
- **Chapter 4: Presenting a Use Case of When Mode Transfer is Beneficial.** Leveraging the natural ability of touch screens to provide direct manipulation feed-forward mechanisms for gesture guidance, we present the concept of utilizing surface experience to provide instruction for mid-air gesture performance. In the case of marking menus, we contrast learning mid-air gestures a cross-modal approach (touch to mid-air) versus a consistent modality approach (mid-air to mid-air).
- **Chapter 5: Leveraging Prior Expertise for Mode and Modality Transfer.** This section describes an example of using prior experience for mode/modality transfer, where user’s pre-existing knowledge and experience with soft QWERTY keyboard layouts is utilized for transfer to a new modality: on-thigh text entry for HMDs, in two separate modes: via tap typing and word gesture typing.
- **Chapter 6: Lessons in Mode and Modality Transfer.** We adapt the concept of intermodal experience development, characterized by Scarr et al., to include findings of the three former studies.
- **Chapter 7: Conclusions and Future Work.** Finally, we summarize our findings and provide recommendations for furthering understanding of mode and modality transfer in gestural input.

Chapter 2

Literature Review

As discussed in Chapter 1, this thesis will focus on understanding mode and modality transfers in *unistroke gesture interactions* — which can be defined as a single, continuous, stroke with a starting point and ending point, provided by some input mechanism. The goal of this chapter is to provide related literature of particular gesture modes and modalities, if and when it’s important to transfer to these modes and/or modalities, and how existing knowledge can be leveraged to ease this transition.

2.1 Gesture Classification

By our the most widespread ubiquitous technologies, the most commonly used gestures are performed on a touch screen surface – usually captured via some form of touch enabled motion sensing technology, such as capacitive sensing, pen, or mouse input. However, recent advances in sensing technologies have allowed capture of gestures performed in “free space”, that is, not constrained by a surface and captured via worn or environmental sensors. These free space gestures are an attractive mode of interaction as they are congruent with natural human dialogue, for instance, pointing an index finger at an object to in order to present contextual information.

For the purpose of the current project, to delve deeper into a particular topic within the broader space of gestural interaction, we have chosen to focus on *unistroke gesture interactions*, due to their wide usage and ease of recognition. *Unistroke gestures* can be defined as a single, continuous, stroke with a starting point and ending point, provided by some input mechanism (e.g. finger, hand, pen, or mouse). More technically speaking, these gestures are an ordered series of points in a 2- or 3-dimensional space.

Unistroke gestures can range in complexity. For instance, a simple unistroke gesture, and likely the most common unistroke gesture, is a single directional stroke or swipe from either left-to-right or right-to-left — commonly applied to switch between contexts or pages within an interface. This simple style of gestures (e.g. 2.1a) is the foundation of one of the most researched gesture systems within the HCI community — Kurtenbach and Buxton’s *Marking Menus* [125, 127] — where single stroke directional gestures are mapped to items within a menu selection system. At a higher degree of complexity, the larger space of unistroke gestures includes ideographic gestures (e.g. 2.1b), such as those presented in the *\$1 Gesture Recognizer* work by Wobbrock et al. [241] or alphanumeric gestures, such as *Graffiti* [149, 235], a shorthand handwriting writing recognition tool used in PDAs. At the highest degree of unistroke complexity discussed within this thesis, we look at word-gesture keyboards [122, 251], where the unistroke gesture begins at a location of a key to start a word or phrase, followed by moving to each subsequent key, and finishing at the final key of the sequence — producing an arbitrary shape that is decoded as a word or phrase.

Though the shapes and complexities in these unistroke gestures greatly differ, they retain the property of being completed in a single stroke. Retaining this property allows performance of these gestures in a number of different interaction methods — in mouse or pen based input, touch input, mid-air free space input — for a variety of ubiquitous technologies — desktop, mobile, IoT devices, as well as in augmented, mixed, or virtual realities.

2.1.1 Gesture Taxonomy

Outside of gesture complexity, in 2009, Wobbrock et al. conducted an elicitation study for surface gesture input. Leveraging the 1080 user-defined gestures, the authors manually classified each, and propose a taxonomy for categorizing surface gestures beyond their experiment [240]. The suggested taxonomy lies on four dimensions: *form*, *nature*, *binding*, and *flow*; with each containing multiple categories, summarized in 2.1 – with a taxonomy breakdown in 2.2.

The *form* dimension is applied to each hand of interaction (so if only one hand, applied to the single hand, and if two, applied to each individually). One-point path is a derivative case of static pose and path, and one-point touch is a derivative of static pose. The authors distinguished such cases due to similarities in mouse-based interaction. The remaining categories are self-explanatory.

The *nature* dimension has four categories, symbolic, physical, metaphorical, and abstract. Symbolic gestures are visual depictions, e.g. tracing a ‘?’ to indicate help. Physical

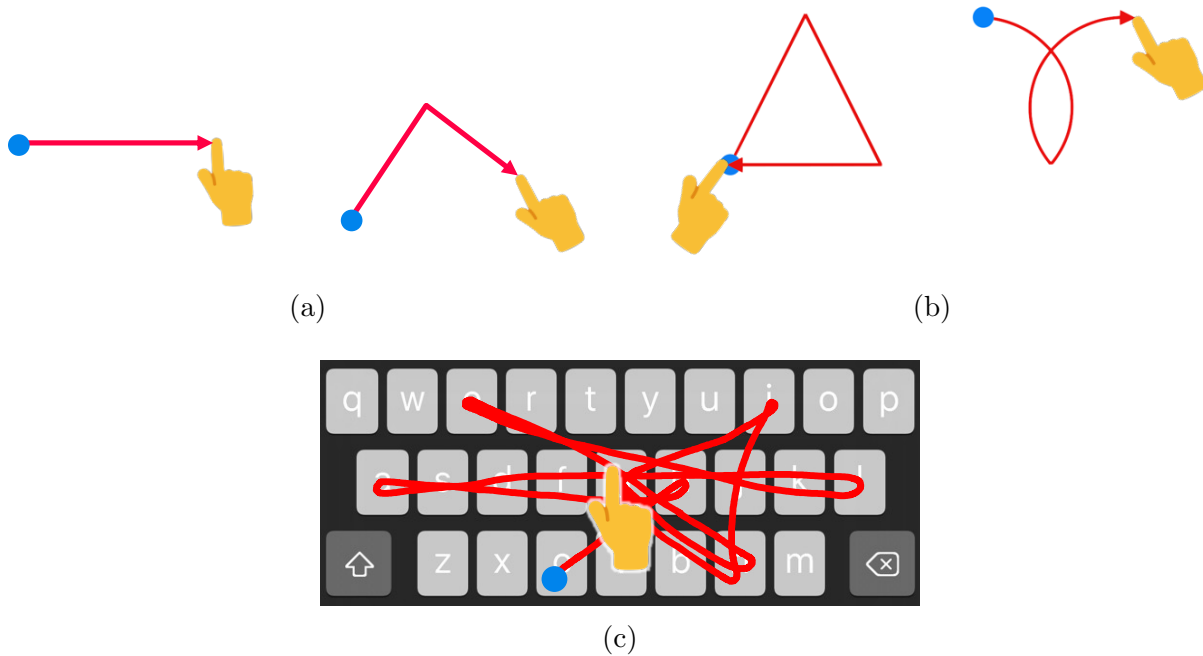


Figure 2.1: Unistroke gestures ranging in complexity, from simple (a) to complex (c)

gestures are gestures that would have the same effect when interacting with physical objects placed atop of or embedded in a surface. Metaphorical gestures indicate a metaphor of some sort – e.g. using two fingers to “walk” across the screen. Finally, abstract gestures have no symbolic, physical, or metaphorical connection to their referents – thus an arbitrary mapping.

The *binding* dimension also consists of four categories, and focuses on the location of the performed gesture. Object-centric gestures only require information about the object they effect or create, e.g. pinching to shrink an object. World-dependent gesture are defined with respect to world space, e.g. dragging a window off screen. Whereas world-independent gestures require no information about the world, for example, a non-contextual (or general) marking menu gesture. Lastly, mixed dependencies have a combination of two or more of the aforementioned styles. An example of this is two handed gestures, where one hand acts on an object and the other, elsewhere.

Finally, the *flow* dimension is either discrete or continuous. Discrete means a gesture is performed, delimited, recognized, and responded to as an event – e.g. a tick mark to indicate something is correct. Whereas continuous means ongoing recognition throughout the gesture, for instance pinch-to-zoom.

Form	<i>static pose</i> <i>dynamic pose</i> <i>static pose and path</i> <i>dynamic pose and path</i> <i>one-point touch</i> <i>one-point path</i>	Hand pose is held in one location. Hand pose changes in one location. Hand pose is held as hand moves. Hand pose changes as hand moves. Static pose with one finger. Static pose & path with one finger.
Nature	<i>symbolic</i> <i>physical</i> <i>metaphorical</i> <i>abstract</i>	Gesture visually depicts a symbol. Gesture acts physically on objects. Gesture indicates a metaphor. Gesture-referent mapping is arbitrary.
Binding	<i>object-centric</i> <i>world-dependent</i> <i>world-independent</i> <i>mixed dependencies</i>	Location defined w.r.t. object features. Location defined w.r.t. world features. Location can ignore world features. World-independent plus another.
Flow	<i>discrete</i> <i>continuous</i>	Response occurs after the user acts. Response occurs while the user acts.

Table 2.1: Wobbrock et al.’s Taxonomy of Surface Gestures [240].

2.2 Mode and Modality Transitions

Oftentimes, gestures are mapped to commands, with little meaning associated between the visual gesture and the intended command. For example, each of the gestures in Figure 2.1 requires some preexisting knowledge in order to understand what, exactly, it does. Thus, since we our focused on symbolic-abstract unistroke gestures, each will requires some form of learning mechanism to understand the mapping of the gesture in space. As mentioned in the introduction, this is often accomplished by some form of mode transition to from a novice mode, that relies on guidance to understand the form and dynamics of the gesture, to an “expert” or secondary mode, that is, being able to perform these gestures autonomically, without intervention.

2.2.1 Motor Learning

Going from a novice to expert mode of interaction, particularly in gestural input, can be viewed as a subcategory of motor learning. Fitts and Posner’s theorize that motor learning or skill acquisition follows a three stage model [67]. The theory states that learning begins with a *cognitive* stage that requires substantial attention to understand the movement,

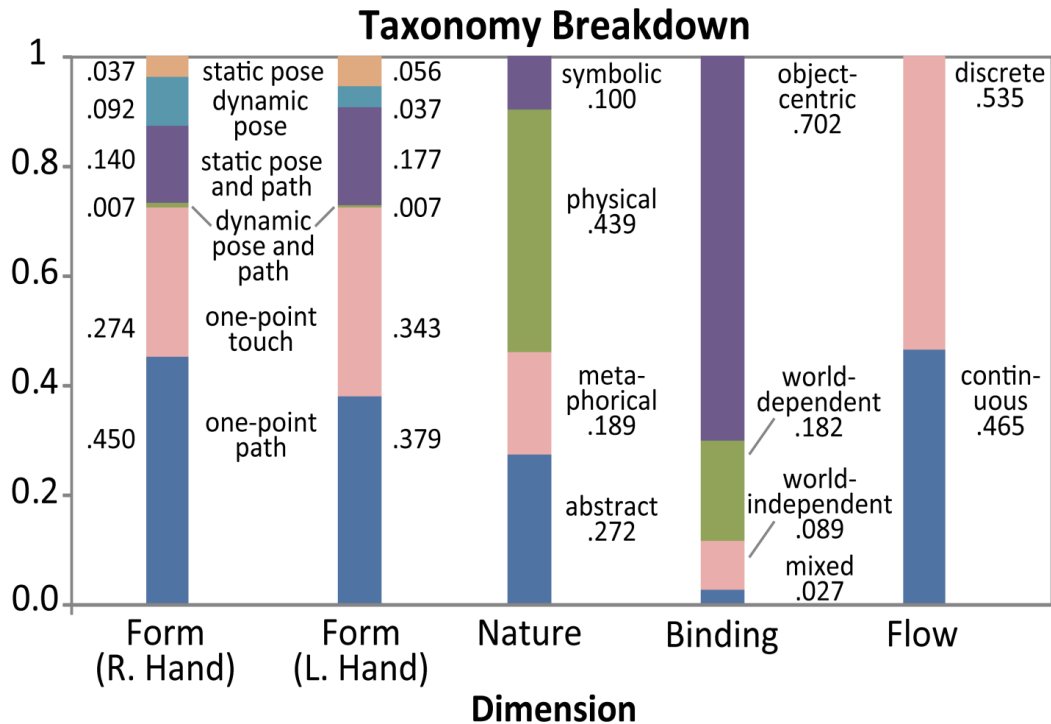


Figure 2.2: A reproduction from Wobbrock et al. [240], indicating percentage of gestures in each taxonomy category. From top to bottom, the categories are listed in the same order as they appear in Table 2. The form dimension is separated by hands for all 2-hand gestures.

often accompanied by over-corrections and a poor quality “stiff” result. The next stage is *associative*, which requires less attentional resources, where focus is on refinement of movement for increased efficiency. Lastly is the *autonomous* stage, where the movement can be completed with little attention or cognitive guidance – thus, focus is not on the skill and can be devoted to other tasks or skills [67].

Motor learning is generally measured by analyzing performance in three distinct ways: acquisition, retention, and transfer of skills [200]. Most relevant to the current work is the concept of *transfer*, which is performing a task similar in movement, yet different from the original learned or practiced task [159] — in other words, the ability to take a skill that was learned in one context or environment, and perform it in a different, but similar, way. This is of particular interest for going from novice to expert modes or modalities in user

interfaces, as user’s are often taught using a *novice* mode, and they then have to transfer that skill to the associated *expert* mode.

2.2.2 Novice to Expert Transfer

In the HCI literature, Cockburn et al. review four domains of research that help users make the transition from novice to expert modes [50]. Two of which are particularly relevant to the present research space: *intramodal improvement*, which concerns the rapidity and magnitude of performance improvement with one particular interactive method for one particular function, and *intermodal transition*, which is concerned with ways to assist users in switching to a faster method of accessing a particular action. Revisiting Figure 1.2, the *first mode/modality* indicates intramodal expertise, and the entirety of the graph depicts the overall intermodal transition.

Intramodal improvement is characterized as a power law curve, subdivided into three segments, *initial performance*, *extended learnability*, and *ultimate performance* [195, 50]. They posit this stage to be the initial stages of learning, where users rely heavily on visual search – thus, designers aim to ease comprehension via minimizing the number of controls display. However, this often results in suppression of visual guidance for more efficient methods of interaction, which inadvertently reduces their discoverability [50]; which, in turn, can introduce a trade-off of whether to cater the interface to improve initial performance, or to raise ceiling performance. The next stage is *extended learning*, focused largely on increasing recall or memorization of a particular technique – reducing the user’s need to rely on recognition factors from the *initial performance* stage [50]. Prior literature has shown that an increase in effort in this stage improves memory, but comes at a cost of frustration [51]. Lastly, is *ultimate performance*, which, in the intramodal curve, is the asymptote or the performance “ceiling” [50, 195].

Scarr et al.’s characterization of intermodal transition combines two power law curves, and involves the switch between a first mode or modality, to a more efficient, “expert” mode or modality, that would, in theory, allow for a higher performance ceiling [195]. However, their theory postulates that the initial switch to a new input method will result in a dip in performance [50, 195]. Alongside this, in order to adopt a new modality, users must first become aware that a new modality exists and second, not only optimize for immediate needs or fall fallible to ‘the paradox of the active user’, where “users are likely to stick with procedures they know, regardless of efficacy” [42]. Many interfaces that aim to solve these issues have been forceful in awareness mechanisms often leveraging some form of a delay or effort inducing mechanism to deter use of any novice mode or modality [15, 16, 81, 90, 121, 127, 124, 132].

2.2.3 Marking Menus

One of the most infamous interfaces that was developed to attempt to bridge the performance and awareness gap between two modes or modalities is *Marking Menus* [127, 124, 125], one of our chosen unistroke gesture techniques to study *Marking Menus*. Marking menus [123] are an extension of traditional pie and radial menus, with a design relying on *marks*, that are directional unistroke gestures, rather than target selections. Command selection can be performed in two main modes: the *menu mode* triggered with a predetermined delay (1/3 of a second in the original implementation), and the *mark mode* where the user simply draws the mark corresponding to the command without waiting.

Because actions are similar in both modes, this design is intended facilitates a smooth transition from *menu* to *mark* mode by simply repeating command selections. Kurtenbach describes this as the principle of rehearsal: *Guidance should be a physical rehearsal of the way an expert would issue the command* [127]. Essentially, in rehearsal-based interactions, *physical actions made by a novice in articulating a command are a rehearsal of the actions an expert would make invoking the same command*, with the goal of leading to a more efficient transition from novice to expert interaction techniques [127].

In the case of marking menus, users begin to learn through a graphical representation of the menu, displayed as they perform the directional strokes. The marks used to activate commands are not self-revealing [127]; therefore learning a mark involves memorizing it's mapping to a command, like an accelerator key but lacking a mnemonic device. Once users memorize the spatial content of the menu, they no longer need the visible menu, and interact based on associating the directional strokes to the desired selection. Thus, the user performs an identical interaction, whether the GUI is visible or not.

Because of this, Cockburn et al. note that marking menus “lie on the cusp” between intramodal and intermodal transitions – but ultimately classify the input technique as *intramodal*, as the initial mechanisms for interaction are identical for both novice and expert modalities [50]. However, we question, if is it possible to be intramodal with two interaction modes present? If it is intramodal, then why are the two modes even separated?

2.2.4 Using Delay to Force Mode and Modality Transfer

Marking menus have inspired a large body of related research. Considering mostly expert users at first, research has explored ways to improve the breadth and depth – allowing more items – by altering marking menus’ traditional radial shape. Examples include curved lines [16] and inflection-free simple marks [254]. Each of these techniques, without question,

employed the *press-and-wait* delay technique to support novice use, i.e. the menu mode. Specifically, in order to switch from *mark mode* – the default where actions are preformed autonomically, from memory, to *menu mode* where user guidance is available, the novice users would need to pause for a timeout; thus, artificially introducing a penalty to the novice user, to foster usage of the allegedly more efficient mode. Zhao and Balakrishnan [254] noted that most of the possible advantages of their technique occurred when users made selections without waiting for the menu to be displayed. Research has also explored novice user experience through Wave menus [15] and Octopocus [24]. While both of these studies aim to improve novice use, they also still utilize a *press-and-wait* delay technique prior to triggering their respective *menu modes* of interaction.

However, *is press-and-wait necessary to foster skilled interaction?* Bailly et al.’s [15] justification for improving novice interaction provides strong motivation and rationale for an immediately-displayed marking menu. They note that *menu mode* is unavoidable – before a user can use *mark mode* they must interact with the *menu mode*, which may deter the user entirely if it is too slow or cumbersome. Further, *menu mode* never disappears: for example, 90% of commands in Microsoft Office are rarely used [140] as cited in [15], thus even expert users will still require *menu mode*. Lastly, the seamless transition between *menu* and *mark* modes [127] necessitates their co-existence.

The principle of rehearsal used in marking menus, i.e. having two accompanying modes of interaction and a delay before menu presentation, have influenced the design of alternative command selection techniques. An example of this is FastTap [73, 89, 90], a command selection technique that displays commands in a spatially-stable grid-based overlay interface. Users can display the interface by dwelling on a button located in the bottom left corner of a screen and select a command by tapping an element of the grid without releasing the first finger – thus, making selections via a chording gesture. Once the command location is known, users can also select commands with a single two-finger tap on the interface. Similarly, MarkPad [71] and InOutPad [29] are command selection techniques for trackpads that implement the principle of rehearsal, utilizing a delay prior to displaying the menu interface.

Though the above papers leverage delay, none of these papers have investigated – or even questioned – the relative costs of benefits of delay to penalize novice use. The closest work of which we are aware is recent research by Lewis [132, 133], which examines the effect different values of delay, between 200 and 2000ms, have on the use of mark mode in marking menus. Using a single level, 8-item marking menu, Lewis shows that longer delays increase mark mode use, but at the cost of a significant increase in error rate; they observe more than a six-fold error rate increase for the longest versus shortest delays initially, and more than double the error rate for the longest versus shortest delays after users reach practiced

use. Furthermore, in analysis of selection time, Lewis only observes faster selection time for delays of 2000ms (all other delays exhibit no statistical differences).

2.3 When is Learning Necessary?

If, in the case of marking menus, the two separated modes are not required, thus always having a visible menu available to guide the user’s interaction – is there a purpose to learning the gesture? The question then arises, of, what circumstances does it become vital to actually learn the gesture? The *Marking Menu* technique was originally designed for pen- and mouse-based computing, and it has since been extended to touch input [36, 68, 116, 144, 212, 255], mid-air bare-hand input [19, 131, 138], mid-air controller input [164, 165], and others inputs such as gaze [5]. In their seminal work on the Charade system, Baudel and Beaudoin-Lafon [25] note that one primary problem with mid-air gestures is that gestures are not self-revealing - the user must know the set of gestures that the system can recognize and their associated functionality; thus, a question related to gesture learning is whether mid-air interaction can benefit from the principle of rehearsal.

2.3.1 Mid-Air Interaction

Early mid-air gesture systems follow a common pattern: pointing in a direction followed by gesturing to indicate an action [31, 237, 41]. Wilson and Shafer’s XWand [237] was designed for an intelligent environment, where users would point at a location in a room (recognized using stereo-vision) and then complete a gesture, for example pointing at a television set followed by rotating the wand to indicate volume up or down. The VisionWand [41] used classical computer vision colour detection and stereo-vision to obtain orientation information with an extensive gesture set allowing rotations, pull, push, tap, tilt and selection via a pie menu. All of these follow the classic multimodal paradigm introduced by Bolt [31] of pointing followed by action invocation.

Pointing interactions have flourished in gaming systems with devices like Nintendo’s Wiimote [239], a ray-casting device for interaction mid-air. Input is captured by in-device sensors including an accelerometer, gyroscope, and IR emitter. Despite the Wiimote being a dedicated ray-casting device, Pietroszek et al. showed a smartphone had similar performance [170]. Vatavu et al. compared user-defined gestures through free-form (i.e. hand movement) and via a handheld device. Users deemed handheld gestures less difficult to execute, which the researchers hypothesize is due to their familiarity with such devices

[216]. Jakobsen et al. compared touch to mid-air techniques for large display interaction. While their analysis found touch superior, their results suggest situations where mid-air techniques would be optimal (e.g. walking to type on a keyboard) [106].

2.3.2 Learning Display-less Gestures

While touch gestures naturally possess a surface able to scaffold and guide users on interaction, in-air motion gestures do not – lacking an explicit mechanism to self-reveal interaction capabilities [25]. Most often, systems designed to teach mid-air interaction techniques require the use of additional hardware to *reveal* gestures to the user, usually in a visual [1, 9, 10, 47, 58, 59, 69, 70, 110, 197, 203, 177, 222], auditory [156, 177], or haptic form [177, 197]. Mirrored representations of the user are one common form of user training [1, 9, 10, 58, 59, 110, 184]. For example, Anderson et al. taught physical movement sequences to users through an interactive large scale augmented reality mirror – and found it improved learning and short-term retention in comparison with a standard video demonstration [10]. Geared to walk up and use displays, both Rovelo et al. [184] and Ackad et al. [1], also presented mirrored representation and dynamic guidance to introduce interactive gestures for systems.

In terms of ubiquitous display interaction, Vatavu actually suggested not requiring users to learn at all, but, rather, to use a *preferred, familiar* gesture set that is individual to each user, depicted in Figure 2.3. For example, a user who often uses a Kinect gaming system may use mid-air gestures that they have used previously, where as a user who only uses touch screen manipulations such as *pan*, *tap*, or *pinch* could use those gestures [214]. While not targeting mid-air input, but in the same realm, Scarr et al. studied the implications of consistent 2D representations in user interfaces, and found that consistency allowed users to develop spatial memory of the interface, increasing performance [194], which could be an asset when exploring in-air interfaces. Other works suggest eliciting user defined gestures [187, 241], to create more intuitive interactions as opposed to an arbitrary mapping. While not mid-air input, Gustafson et al. introduce the concept of *transfer learning* for an imaginary phone interface on a display-less surface. The idea, is that users transfer the spatial knowledge of a familiar interface that possesses a display, to a novel touch interface (such as their own palm), that does not have any visual guidance for where or how to provide input [87] – depicted in Figure 2.4

The above works give rise to rationale for utilizing a mobile device for mid-air input training. Mid-air gestural interaction is effective in interactive environments because it supports target selection and action in a single, unified input modality. While personal

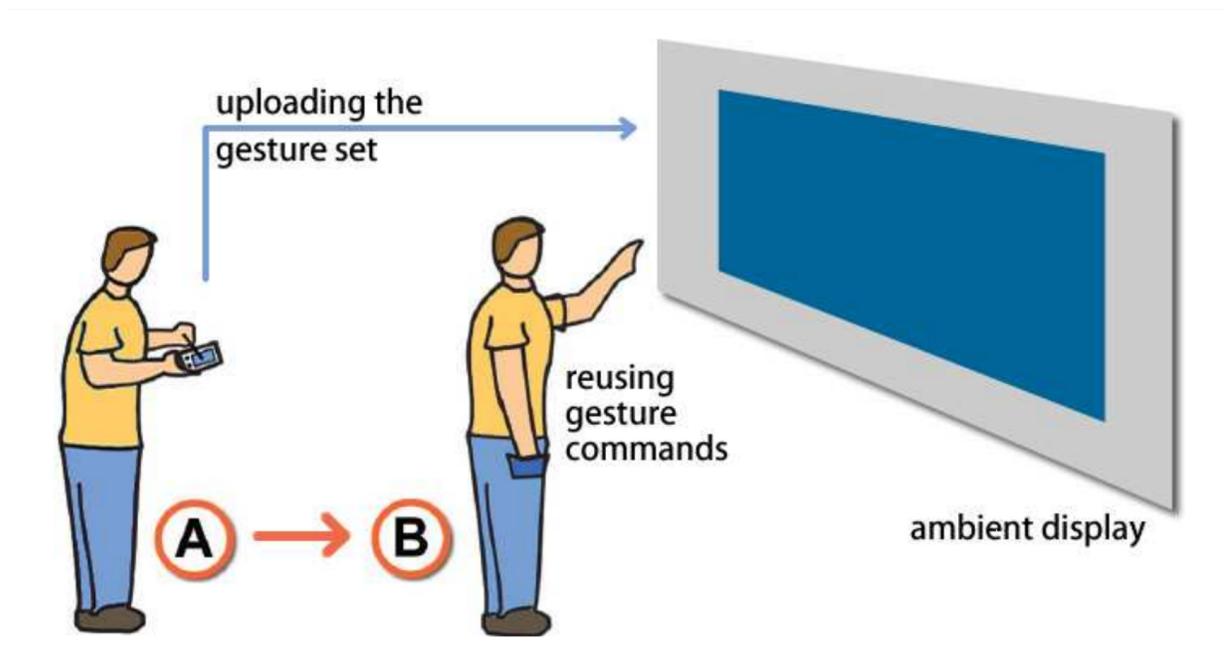


Figure 2.3: Reproduced from Vatavu et al.’s *Nomadic Displays* [214]

devices – via motion gestures [164, 188] – are an effective means of capturing and mapping gestural commands, as Oakley et al. note [165], there are challenges with guiding users mid-air inputs using a mobile device’s screen because, during performance, the screen may not be visible. For us, the attendant question is then: Can we leverage touch-based training to teach users mid-air input?

2.4 Knowledge Transfer: The Word Gesture Keyboard

While learning across the modalities of touch to in-air gestures is a novel concept, the idea of transferring information between contexts is not new – in fact, prior in this area have been shown to be effective in transferring spatial knowledge to learn complex gestural input [122, 143, 250, 251, 257]. One of the most pervasive implementations of transferring expertise between modes via rehearsal is *word gesture keyboards* — where pre-existing knowledge of a keyboard layout allows users to perform unistroke gestural input between characters (or keys) to form a word or phrase, without intervention. For example, Figure 2.1c, is a gesture spelling out “challenging”. Recent literature has shown that users are capable of performing these gestures without a visual of the keyboard to guide vertex

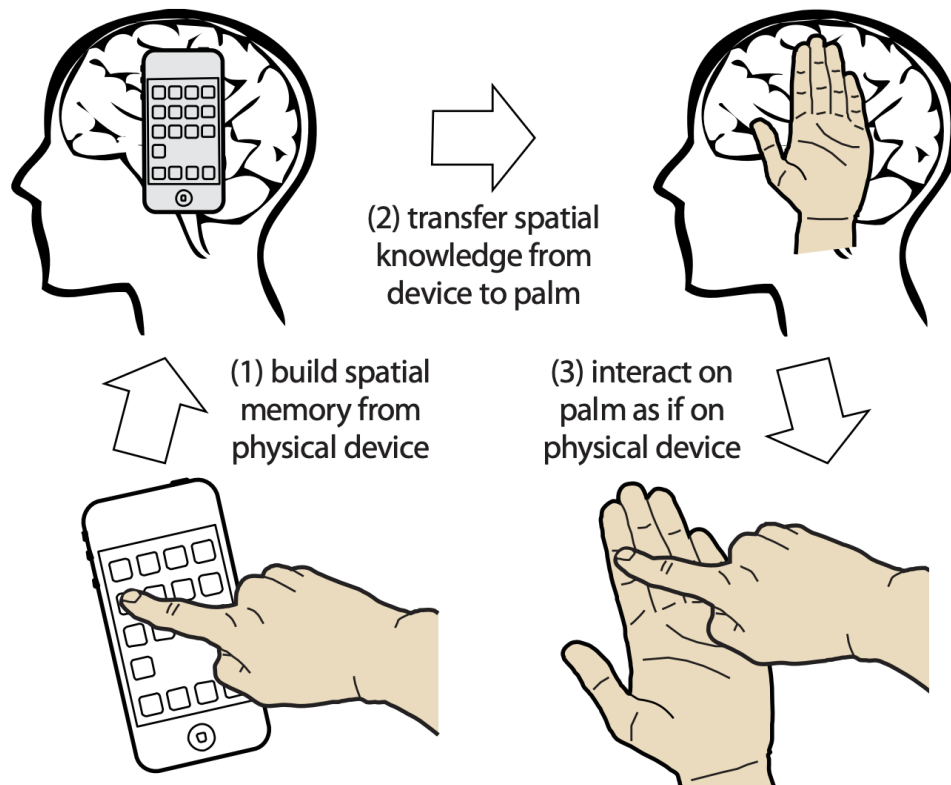


Figure 2.4: Reproduced from Gustafson et al.’s *Imaginary Phone* as the process of creating a mental model for spatial transfer learning [87]

placement [143, 257], particularly in eyes-free contexts, such as virtual or augmented reality. We then question *how far can this knowledge transfer be pushed to new contexts?*

2.4.1 Head-Mounted Display Input

Over the past decade, there has been a surge in popularity of head-mounted displays (HMDs) for presenting an augmented or virtual reality to the wearer. A challenge arising from these HMDs, which has subsequently become a roadblock in their widespread adoption (e.g. smartglasses), is the lack of an input mechanism for their control [153]. One primary input mechanism we require for HMDs is some facility for text entry. There is an extensive body of research on techniques for text entry in wearables, including HMDs (e.g. [130, 145, 232, 247]).

In general, text entry is a challenge in ubiquitous computing as input either requires a button or key associated with each character, or some form of gesture or chord to describe characters or words. This, in turn, may require specialized devices [145], additional sensors [232], or learning a new input mapping [247] to effectively input text. Gaze eliminates the need for specialized devices, but is perceived to be “complex, strenuous and slow” [7], and both speech and gaze suffer from issues of social acceptability, especially when compared with on-device interaction [181]. While it is possible to type on a virtually displayed keyboard [205], this requires tracking of finger position and, without a physical surface, it is challenging for users to localize keys—potentially reducing the speed and accuracy of such a technique. Thus, a large amount of research has been dedicated to optimizing text input when using a virtual display—resulting in increased performance via novel interaction techniques [4, 84, 205, 247, 246]. Many of these techniques, however, require specialized hardware or physical controllers that often encumber the user’s hands during interaction.

In recent work, Akkil et al. [7] note that, for users of smartglasses and other HMDs, mobile phones are considered to “complement” HMDs, particularly for functions where the HMDs are lacking, such as text entry. As users have become proficient in text entry [180] on mobile touchscreens, mobile text entry for HMDs appears an ideal area to apply knowledge transfer of a keyboard layout for seamless interaction.

2.4.2 Text Entry Techniques

Text Entry for HMDs Presenting Virtual Content

Many virtual reality (VR) scenarios leverage HMDs for their presentation. Physical keyboards have been studied as a text entry technique in VR, and have proven effective, with users performing only slightly slower than with physical keyboards outside VR [117]. However, physical keyboards may be impractical in ubiquitous HMD settings.

Where physical keyboards are impractical, specialized text entry devices can be used to support text-entry. For example, when evaluating text entry techniques in VR, Gonzales et al. [78] found mobile text entry with 12-15 physical buttons particularly effective (reaching 32.75 to 107.39 characters-per-minute, i.e. 6.5 to 20.5WPM). However, physical buttons are rarely present on modern smartphone devices. Without physical buttons, text entry rates tend to be more modest: Speicher et al. [205] conducted a study evaluating common selection based text entry techniques for VR, including head pointing [246], controller pointing (i.e. ray casting at characters), controller tapping, freehand selection, and discrete and continuous cursors. While Yu et al. [246] found that gestural head-pointing reached

WPM rates of up to 24.7 WPM after 60 minutes of training, Speicher et al. found more modest rates of text entry for head-pointing (10.2WPM). Speicher et al. also found that controller pointing achieved the highest text entry rate, 15.4 WPM. More subtle forms of text input have also been evaluated; for example, Lu et al. [143] studied various algorithms of decoding thumb-based tap text entry on a blank smartphone screen for use with HMDs and external displays. Their baseline cursor implementation achieved 7.66 WPM, while more complex statistical decoding algorithms boasted rates of up to 17-23 WPM [143], but required a user to hold their phone in their hand.

Smartphone-Based Text Entry

Modern smartphone-based text entry typically leverages a soft keyboard to capture text (i.e. an onscreen keyboard to replace the physical keyboard). On these soft keyboards, users can enter text character-by-character, or, they can take advantage of a series of intelligent typing options, including auto-correct, word-completion, and word-gesture-keyboarding (WGK) [122, 180, 250, 251]. While it seems clear that auto-correct boosts text input speed [168], word-completion and WGKs are more difficult to analyze. In an extensive study of over 37,000 smartphone users, conducted by Palin et al. [168], approximately one quarter of users reported using WGKs versus 3/4 who used tap-based typing. They found that both word-completion and WGKs actually resulted in slower text entry than typing. However, in an earlier in-the-wild study of the Google keyboard, Reyal et al. [180] found that gesture-based text entry resulted in a significantly greater text entry rate than tapping-based text entry, with average WPMs of 33.6 and 30.1, respectively. While there is a significant body of work that leverages WGKs in various forms [85, 151, 245, 246, 257], what is clear from the Palin et al. study [168], is that both character-by-character input on soft keyboard (75% of data) and WGKs (25% of data) are both common mechanisms for text entry.

One advantage of WGKs is that they are tolerant to a degree of imprecision in the gesture input [122], provided the word is in the dictionary and there is sufficient difference between word gestures [30]. Given this tolerance for imprecision, WGKs have been explored for eyes-free text entry. Of particular relation to the current work, Zhu et al. [257] modified the original gestural text entry algorithm to develop an eyes-free gesture typing system using a smartphone's touch screen, Figure 2.5. In their evaluation they reached an average WPM of 23.27. Similarly, Yang et al. [244] studied gesture typing on a smartphone's touch screen – motivated by first-touch imprecision for indirect touch text entry. Their technique assumes every gesture begins at 'G' and reached an average WPM of 22 in their user study. While restricted to in-dictionary words, this past research highlights the strong desire for

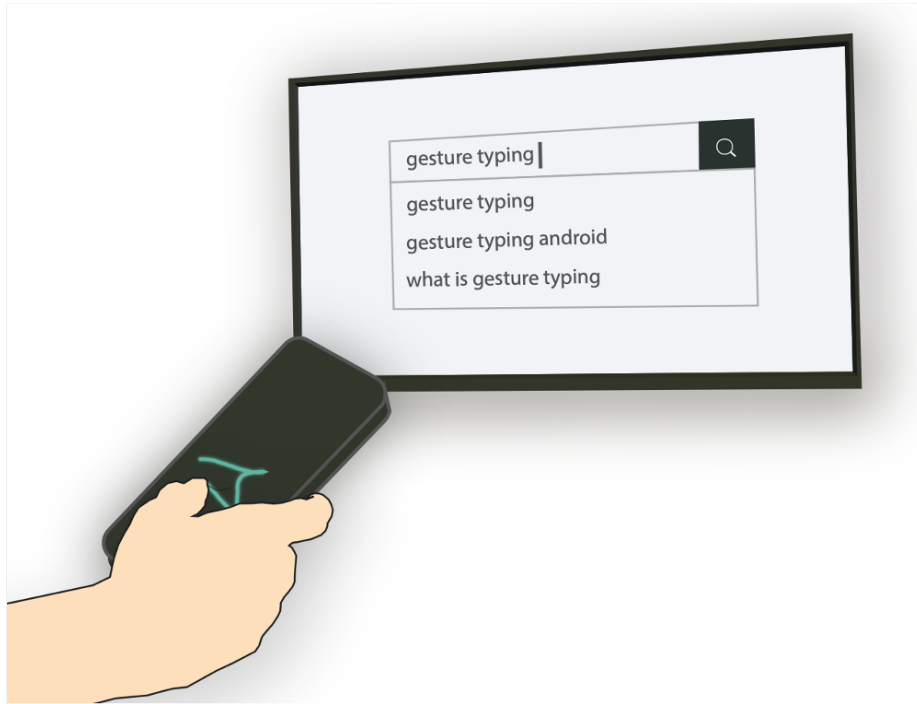


Figure 2.5: Reproduced from Zhu et al.'s *I'sFree* [257].

eyes-free text input in a variety of contexts.

2.4.3 Gestural Text Entry for HMDs

To synthesize, we revisit the question in Section 2.4 of: *how far can this knowledge transfer be pushed to new contexts?* Taking into consideration the requirement for text entry in head-mounted displays and the natural integration mobile devices can provide for wearable devices, we ask: *how can the spatial knowledge of a keyboard layout on a mobile device be leveraged to provide word-gesture text input for HMDs?*

Chapter 3

Understanding the Necessity of Mode Transfer in Marking Menus

This chapter presents the results of an in-depth study on the necessity of delay, or penalization, within marking menu invocation. Delayed display of menu items is a core design component of marking menus, arguably to prevent visual distraction and encourage the transfer to mark mode. We investigate these assumptions, by contrasting the original marking menu design with immediately-displayed marking menus. In three controlled experiments, we fail to reveal obvious and systematic performance or usability advantages to using delay and mark mode. Only in very constrained settings – after significant training and only two items to learn – did traditional marking menus show a time improvement of about 260 ms. Otherwise, we found an overall decrease in performance with delay – or when penalized, whether participants exhibited practiced or unpracticed behaviour. Our final study failed to demonstrate that an immediately-displayed menu interface is more visually disrupting than a delayed menu. These findings inform the costs and benefits of incorporating delay in marking menus, and motivate guidelines for situations in which its use is desirable.

3.1 Motivation

One reason why marking menus have been intensely studied in Human Computer Interaction [15, 16, 24, 144, 161, 186, 253, 254, 255] is because they implement the principle of *rehearsal*, whereby the selections in *menu mode* act as “rehearsal” for selecting commands in *mark mode*, easing the transition from a *novice* to an *expert* level of interaction

[123]¹. Alongside rehearsal, marking menus typically include a press-and-wait attribute which adds a cost (typically a 1/3 second delay [127]) to *menu mode* to distinguish, and encourage the use of, *mark mode*. In his doctoral thesis [127], Kurtenbach rationalizes displaying the menu after a certain delay on the basis that the menu “*can be distracting*”, “*can obliterate part of the screen*” and that “*displaying the menu takes time*”. Despite the extensive study of marking menus, surprisingly the impact of delay on users has received little attention.

In this chapter, we investigate the necessity of delay in marking menu appearance, and its possible accompanying issues. Our motivations for doing this are two-fold. First, the delay in *menu mode* creates a cost for novice interaction. While this delay might act as an incentive to use *mark mode*, it is unclear by how much it accelerates learning in real use (i.e. is penalty an effective motivation [129] in context?). Second, while users may use *mark mode* for common commands, not every command is used frequently, thus *menu mode* interaction remains necessary for many commands, even for more experienced users. To the best of our knowledge, the *menu mode* cost has never been directly measured in expert interaction. We investigate whether a no-delay marking menu really inhibits expert performance, and if so, by how much.

We report the results of three controlled experiments comparing interaction with two types of marking menus: the original DELAY marking menu [127] and a NO DELAY marking menu. The first experiment – prioritizing experimental validity – compares these two marking menus in an abstract task where participants are prompted to select commands using marking menus of different breadth and depth. The second experiment prioritizes repetition to compare these two interfaces when users have reached expert level interaction. Finally, the third experiment, more focused on marking menu use within an application, investigates whether a NO DELAY marking menu impacts subjective user experience when performing a visually demanding task requiring users to insert and move graphical objects in a 2D scene. Our results suggest that a NO DELAY marking menu offers significant benefit for novice use and comparable performance even when users are practiced with the menus, at the expense of a small cost for fully autonomic individual command selection. In particular, the NO DELAY menu exhibits fewer errors, similar learning rates, and does not significantly disturb users. These results allow us to present a more nuanced approach to delay in marking menus, to better inform their cost/benefit trade-off, and to discuss implications for other rehearsal-based interfaces.

¹Unlike common practice in the HCI literature, we will refer to “*novice*” and “*expert*” exclusively to describe user’s behavior and/or overall level of interaction, not specific modes for a given technique.

3.2 Rationale for Delay in Rehearsal-Based Interfaces

While delay is commonly used in rehearsal-based interaction techniques [16, 90, 71], the duration of the implemented delays can vary depending on the technique (typically, from 100ms [16] to 500ms [71, 255] for marking menu variants). For its part, Autodesk’s Maya 2019 incorporates a delay of approximately 230ms, as determined by counting frames from cursor change on press to marking menu appearing via screen capture running at 30Hz on an Apple Macbook Pro. Implementations of other rehearsal-based techniques also employ different values for delay (e.g. FastTap used delays of 150 [73] and 200ms [90], which suggests an explicit adjustment).

In order to better understand the rationale behind the use and selected values of these delays, we conducted non-anonymous email interviews with 4 interaction designers: G. Kurtenbach (inventor of marking menus [123, 124, 125, 126, 127]), G. Bailly (who designed marking menu variants and rehearsal based selection techniques [14, 15, 16, 17]), C. Gutwin (inventor of FastTap [73, 89, 90, 128]) and E. Lecolinet (who contributed to the design of several marking menus variants [15, 16, 176, 183] and rehearsal based selection techniques [29, 71]). We sent a single e-mail to each interviewee asking the following questions:

- In your opinion, why do Marking Menus and other rehearsal-based interfaces require delay?
- In each work, how did you choose the delay length?
- Overall, what are your views on the trade-off between penalizing novice users vs. making skill acquisition hypothetically faster?
- Are you aware of any study that tested delay’s impact on performance, learnability or visual disruption?

Three of these researchers answered these questions in a single e-mail, the fourth researcher sent an additional e-mail to complement his answers shortly after sending the first one, without us asking for additional details or clarifications

Initial motivations for delay

As expected, the goal of Kurtenbach’s initial introduction of a delay was to avoid visual disturbance. Its introduction was indeed justified “*as a means to invoke menu or mark mode*”, thus avoiding an unnecessary menu pop-up when users already know the mark of the command they wish to select. Kurtenbach also answered that once these two modes

co-exist, the delay *“may encourage learning/markings the mark”*. Avoiding visual distraction and nudging the user toward expert performance were also brought up by Gutwin as motivations for two delay-separated modes in FastTap [90]. Interestingly, Lecolinet reported that the delay in MarkPad [71] is needed, but for different reasons. MarkPad is a command selection technique that leverages gestures beginning and ending at the borders of a trackpad. However, trackpads are primarily used for cursor control. Therefore, the user could “pre-activate” a menu by mistake by starting a cursor controlling movement from a border of a trackpad that, if it were not for the delay, would display the menu. Thanks to the delay, the menu is not instantly displayed and there is no visible effect. The delay is therefore used to allow the system to distinguish between cursor control and command gesture before the gesture is completed.

Delay duration and its evolution

Kurtenbach does not remember exactly how he chose the initial duration of 333 ms, and responded that it may be based on an estimate of *“human reaction time”* even though *“if you think about it this isn’t a reaction time situation”*. In later commercial marking menu implementations, he realized that they could adjust its value as low as 100ms *“and still have users reliably control when a menu is displayed and when a mark is drawn”*. Bailly answered that he chose the delays for his implementations of marking menu variants based on values used by Kurtenbach in the literature. Gutwin reported that the delay in FastTap implementations was tuned to prevent menus from appearing during mark mode. Interestingly, Kurtenbach noted that with a *“very small delay, some users never used the marks”*, but also that *“many people have to be told about mark mode. They don’t seem to discover it”*. Finally, Bailly mentioned that he recently started to express doubt on the necessity of having two modes distinguished through a delay when it is possible to make a technique *“rely on recognition, not necessarily on memorization”*, referring to his recent research on hotkey learning as examples [74, 150].

Previous investigations of marking menus delay

The researchers were not aware of any study specifically investigating marking menus delay’s impact. However, Bailly referred to Kurtenbach’s study comparing *mark* mode to *menu* mode [127] (described in the next sub-section). Gutwin and Kurtenbach mentioned a study on hotkeys, by Grossman et al. [81], notably investigating the impact of cost applied to mouse-based selection on hotkey adoption.

3.2.1 Investigating the necessity of delay in marking menus

In addition to the above reflections, other researchers have begun to question the rationale for delay. In recent work, Zheng et al. [255] studied the progression from *menu* to *mark* mode using a 500 ms delay in a marking menu designed for mobile devices. They acknowledge the lack of literature to support delay invocation, but note cognitive psychology research supports the cost of waiting (cost-based interaction) as an incentive for active memory retrieval [56]. Exploring cost-based interaction specifically, Cockburn et al. [51] showed that hiding keys' labels improved gesture retrieval for ShapeWriter [122] input, but did not result in increased text-entry speed. Moreover, if cost-based interaction supports active memory retrieval, it is unclear whether memory retrieval is necessary for accurate, efficient marking menu interaction [15, 51, 66, 255].

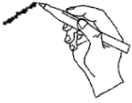
We are aware of only two studies conducted in the early 90s by Kurtenbach and Buxton that explore the value of having a delay [127, 125], and they present somewhat conflicting information. The first is a longitudinal study in which 2 participants used a 6-item marking menu for 8 and 10 hours, respectively. Results comparing command selection time in *mark* and *menu* modes (with delay subtracted) suggested that marks are faster, but the specific rationale for the improved performance (“*the user most likely waits for the menu to appear [and] must then react to the display. [...] Thus, a mark will always be faster than menu selection, even if press-and-wait was not required to trigger the menu.*” [127]) was – to the best of our knowledge – only evaluated with this limited testing based on estimates, and never contrasted with a NO DELAY condition. The second compared *menu mode* (menu always visible) with *mark mode* (menu always hidden, and only shown during the first 6s of each block), using a menu layout of numbered items labelled in clockwise order. Average execution times and error rates were significantly lower in *menu mode*. However, *mark mode* was strongly disadvantaged in this study, in spite of using a layout of ordered menu items, because participants could only consult the menu for the first 6 seconds of each block.

3.3 Rationale for testing no-delay Marking Menus


Given the above research and information elicited via email from Kurtenbach and others, we find merit in examining the use of delay for *menu mode* activation. In 1991, Kurtenbach proposed that “*even if a user did not have to pause to signal for the menu to be popped up, one would still have to wait for the menu to be displayed before making a selection. In many systems, displaying the menu can be annoyingly slow and visually disturbing.*”

However, neither he nor any of the other surveyed experts were aware of any studies that had specifically evaluated visual disturbance, and, with advances in computing over the past 27 years, the likelihood that displaying a menu would be “annoyingly slow” is low: as of 2019, the majority of systems display even the most complex user interfaces in milliseconds. Further, effective use of threading in GUI design can ensure that interfaces remain active even in the presence of costly computational tasks and users can act even before the menu appears, if displaying it is slow, as there has been anecdotal evidence that expert users avoid these issues by “*mousing ahead*” in pie menus [103], as cited in [123].

Making a mark

pen/ button down .07 secs	move to draw a mark .3 secs		pen/ button up .07 secs
------------------------------------	-----------------------------------	---	----------------------------------

Using the menu

pen/ button down .07 secs	press and wait to trigger menu .33 secs	system displays menu .15 secs	user reacts to menu display .2 secs	move to select from menu .3 secs		pen/ button up .07 secs
------------------------------------	--	--	---	--	---	----------------------------------

Using the no-delay menu towards a known item

pen/ button down .07 secs	move to select from menu .3 secs		pen/ button up .07 secs
	system displays menu		

Figure 3.1: Top: Reproduction of figure 4.15 in [127]. Bottom: Our hypothesis on the time costs of a no-delay Marking Menu.

Delving more deeply into issues of mousing ahead and temporal costs, consider Figure 3.1. Figure 3.1-top is a reproduction from figure 4.15 in [127]; it describes users’ expected behaviour when using mark mode. In the center is user’s expected behaviour in menu mode, where we can see additional costs to menu mode - including system display and user search costs. However, whether there is a cost to menu display is debatable. Consider Figure 3.1-bottom, our representation of an alternative hypothetical time costs of a no-delay marking menu. Given that the system can capture input immediately and continuously, does the user “most likely wait for the menu” (shown as *user reacts to menu display* time in Figure 3.1-center) if they already know what gesture to perform? Or might the user simply begin to move, mouse ahead [123], removing the cost of menu display (bottom)? Abundant research has shown that visual memory is quick to build and robust (see [191] for an extensive review). Cockburn et al.’s model of expert performance in linear menus argues that, in practiced use, users act without visual search delay [51]. Assuming that after

sufficient practice the user acquires knowledge of the visual layout, removing the delay (and therefore the *mark*-only mode) may reduce the temporal penalty without harming memorization. Other possible issues, such as distraction and occlusion, would remain to be investigated. The question then becomes whether or not we can determine the relative costs of these factors.

One could argue that, even if DELAY is not needed, it does no harm, so why study no delay? A reason to explore DELAY’s benefits is, alongside potential benefits in learning and preventing visual disruption, there are potential costs to delaying the visual display of the menu:

- First, delay is a *cost* when performing command selections in *menu mode*, penalizing users if they are unfamiliar with the menu or with the specific command being invoked.
- Second, selecting a command in *mark mode* requires the user to memorize the corresponding mark beforehand, i.e. it leverages recall rather than recognition.
- Finally, with delay possibly acting as an incentive to use *mark mode*, the user may try to select commands via *mark mode* even if when not entirely sure of the correct mark, which might increase error rates even for practiced use [132].

To contrast the cost of a NO DELAY marking menu with a DELAY-based marking menu, we conducted three experiments. First, in a controlled experiment we evaluate the comparative performance of DELAY and NO DELAY marking menus by testing the following hypotheses:

H1: NO DELAY *marking menus have a lower command selection time than DELAY marking menus*, because of the artificial delay imposed on the *menu mode* when selecting less practiced or less frequent commands.

H2: NO DELAY *marking menus have lower error rate than with DELAY*, because visual feedback is always available.

H3: NO DELAY *marking menus are slower than with DELAY in highly practiced use*, because trained participants will wait for the menu to appear and will suffer visual disruption.

In our first experiment, we find little benefit of delay-based activation, possibly because our participants did not reach a sufficient level of expertise – or autonomic response – for mark mode commands. We, therefore, conduct a second study that examines the limits of expert level use to identify whether – with simulated ‘perfect’ expertise – we can see and quantify a benefit from DELAY.

Finally, in a third experiment we evaluate visual distraction and occlusion issues in menu mode. Evaluating these factors is challenging: the relative costs of visual distraction

and disruption are subjective assessments of the user and are only present when using a marking menu to accomplish a specific task. We leverage a simple comic replication task, and elicit subjective assessments via Likert Scales and qualitative interview data to test the following hypothesis:

H4: *DELAY marking menus are, subjectively, less visually disruptive and occluding than NO DELAY marking menus, because the menu mode need not be used during the performance of individual practiced commands.*

3.4 Study 1: Evaluating the Impact of Delay

As a first step in evaluating the relative costs and benefits of DELAY versus NO DELAY, we conducted a controlled experiment to contrast the impact of delay on the performance of marking menus. Specifically, our experiment is designed to evaluate hypotheses 1, 2, and 3 in the previous section.

3.4.1 Experimental Procedure

Participants and Apparatus

Sixteen paid participants were recruited for the study. Average age was 23.44 (SD = 2.99). Four participants identified as female and the remaining 12 identified as male. All participants were post-secondary students from two different technically-focused universities. Each participant signed an informed consent form before starting the experiment.

The interface was displayed on a 28" ASUS PB287Q monitor with 1080p resolution. User input was achieved using a Logitech M100 mouse. We ran the application on a Macbook Pro running OS X Version 10.11.6. The application was written based on a JavaScript implementation [185] of Kurtenbach and Buxton's marking menu [123, 124, 125]. The application was modified to suit the purposes of the current research, which included, the addition of *confirmation mode*, the ability to log user behaviour and to reflect conditions outlined above.

Familiarization

Prior to completing the study, users were administered a verbal and visual demonstration of how to use marking menus within both conditions, including all modes: *menu mode*, *confirmation mode*, and *mark mode*. The demonstration was conducted by the experimenter

– thus, participants did not interact with the marking menus until the experimental task began.

Task and Stimulus

Participants were instructed to select commands with a marking menu using the right mouse button. For each trial, the participant had to select a target command (displayed on top of the window) with a MARKING MENU (DELAY or NO DELAY) of a given LAYOUT (4, 8, 4×4 , or 8×8 items).

The DELAY marking menu had a 333 ms delay to enter into *menu mode* and display the menu, and a 200 ms delay to display the sub-menu². In this condition, participants could also use *confirmation mode* which displayed the menu after a 333 ms delay. The NO DELAY marking menu had 0 ms delay to display the menu and the sub-menu was opened when participant entered the corresponding menu item.

For each menu and layout, participants performed 10 BLOCKS of command selection. For the 4-item configuration, participants performed 10 BLOCKS of 4 command selections (there were only four commands); for all other configurations, we selected 8 target commands and participants performed 10 BLOCKS of these 8 commands presented in a random order. We used two different command sets for each menu configuration, counterbalanced across conditions, to control for confounds of learning behaviour and confusion between categorical selections. Participants were permitted to take a break after each menu configuration was complete.

Given this task, the experiment was a $4 \times 2 \times 10$ within-subjects design, with the following factors and levels: LAYOUT (two 1-level configurations of 4 and 8 items, two 2-level configurations of 4×4 and 8×8 items. MENU (DELAY vs. NO DELAY), and BLOCKS (0 to 9). Presentation orders of MENUS and LAYOUTS were counter-balanced across participants using a Latin Square design. Therefore, in total, we collected $(1 \times 4 + 3 \times 8)$ commands \times 10 blocks \times 2 Menus \times 16 participants = 8960 selections in total.

Dependent Measures

The primary dependent measures were *Selection Time* (time from stimulus to correct command selection), *Execution Time* (time from the last mouse press to correct command

²The duration of this delay is not specified in Kurtenbach’s thesis [127]. We empirically set it to 200 ms to minimize accidental activation without penalizing this condition’s performance.

selection) and *Error Rate*. Additional dependent measures were *Preparation Time* (time between the display of the target item and the first mouse-down event) and the proportion of *mark mode* usage in the DELAY condition.

3.4.2 Overall Time and Accuracy

In the following section, we used multi-way analyses of variance (ANOVA) for the independent variables MENU, LAYOUT, BLOCK, and their interactions. Participant is always included as a random factor using the REML procedure of the SAS JMP package. Post-hoc tests are Tukey tests when there are more than two levels.

We systematically removed the first trial of each participant (once in each DELAY/NO DELAY condition), which took distinctively more time than every other trial as participants were discovering the techniques. In what follows, TRIALS are ordered from 1 to 40 for 4-menu-item LAYOUT and from 1 to 80 for all other LAYOUTS (see subsection *Task and Stimulus*), and represent the ordered trial indexes for a given participant in a given MENU and LAYOUT condition. BLOCKS are ordered groups of trials containing exactly one selection of each target of a MENU \times LAYOUT condition: 4 trials for LAYOUT 4 \times 4, 8 for the other LAYOUTS. We present our analysis for each of our three hypotheses in turn in the remainder of this section.

H1: Menu Selection Time

Our first hypothesis posits that selection time is shorter, overall, for NO DELAY marking menus because of the cost imposed on less practiced use. To assess this, the different time measures were aggregated as medians instead of means to discard outliers and account for asymmetric distributions; residuals were found to follow a normal distribution. Time results are shown in Fig. 3.2.

To evaluate H1, we analyze *Selection Time* against MENU, LAYOUT, and BLOCK. Regarding median *Selection Time*, we found a significant effect of MENU ($F_{1,1185} = 76.29$, $p = .0001$), LAYOUT ($F_{3,1185} = 623.43$, $p = .0001$), and BLOCK ($F_{9,1185} = 63.01$, $p = .0001$), as well as significant interaction effects: MARKING MENU \times LAYOUT ($F_{3,1185} = 3.76$, $p = .05$), MARKING MENU \times BLOCK ($F_{9,1185} = 2.58$, $p = .01$), and LAYOUT \times BLOCK ($F_{27,1185} = 7.42$, $p = .0001$).

Post-hoc tests revealed that the DELAY condition (mean 2294 ms) was significantly slower than the NO DELAY condition (2042 ms). This allows us to reject H1's null hypothesis and claim that our data support improved performance overall for NO DELAY

marking menus. Our results also not only revealed that *Selection Time* was longer for 1-level marking menus than for 2-level ones, but also that all LAYOUT levels are statistically different from each other: 4 (1393 ms) \ll 8 (1777 ms) \ll 4×4 (2512 ms) \ll 8×8 (2990 ms). Overall, as expected, our results also revealed that *Selection Time* decreased with BLOCKS at a decelerating pace.

H2 Error Rate

Our second hypothesis posits that NO DELAY marking menus have lower error rate than DELAY marking menus. The corresponding null hypothesis is that *Error Rate* differences between NO DELAY and DELAY marking menus are not significant. Figure 3.3 shows comparative error rates for NO DELAY in blue and DELAY in red. To test H2, we analyze *Error Rate* as a function of MENU, LAYOUT, and BLOCK. We found a significant effect of MARKING MENU ($F_{1,1185} = 60.88, p = .0001$), BLOCK ($F_{9,1185} = 4.11, p = .0001$), and MARKING MENU×BLOCK ($F_{9,1185} = 4.01, p = .0001$) on average *Error Rate*. There was no significant effect involving LAYOUT.

Post-hoc tests highlight that participants made significantly more errors with DELAY (mean 4.2 %) than with NO DELAY (mean 1.3 %) condition. This result allows us to reject the null hypothesis and claim that *H2 is supported*.

Effects of BLOCK and of the MARKING MENU×BLOCK interaction were also significant. *Error Rate* significantly *increased* with BLOCKS overall (Fig. 3.3-bottom), but the study of the interaction effect revealed that BLOCK levels are not significantly different from each other in the NO DELAY condition. In the DELAY condition, however, BLOCKS 9 and 8 (resp. 6.9 and 8.5 %) contain significantly more errors than BLOCKS levels 0, 1 (< 1.3 %) and than every NO DELAY BLOCK (< 2.6 %). This result might be explained by the fact that with more practice in this condition, participants made more selections using *mark mode* (detail analysis of *mark mode* usage below), which is more likely to result in errors because the user must know the mark corresponding to the target, compared to the *menu mode* and the DELAY condition where the menu is displayed. It also suggests that users can switch to *mark mode* before being entirely familiar with the item.

One additional question we can pose is whether the increase in error was due to decreased use of *menu mode* in the DELAY condition. Figures 3.2 and 3.3 shows the evolution of Selection time and errors, respectively. Consider, particularly, Figure 3.3, where we can see that error rates are identical for DELAY and NO DELAY during BLOCK 1, but then diverge. Visually, it appears that, in DELAY, the use of *mark mode* is detrimental to accuracy. Analyzing this, we find that participants made significantly more selection errors

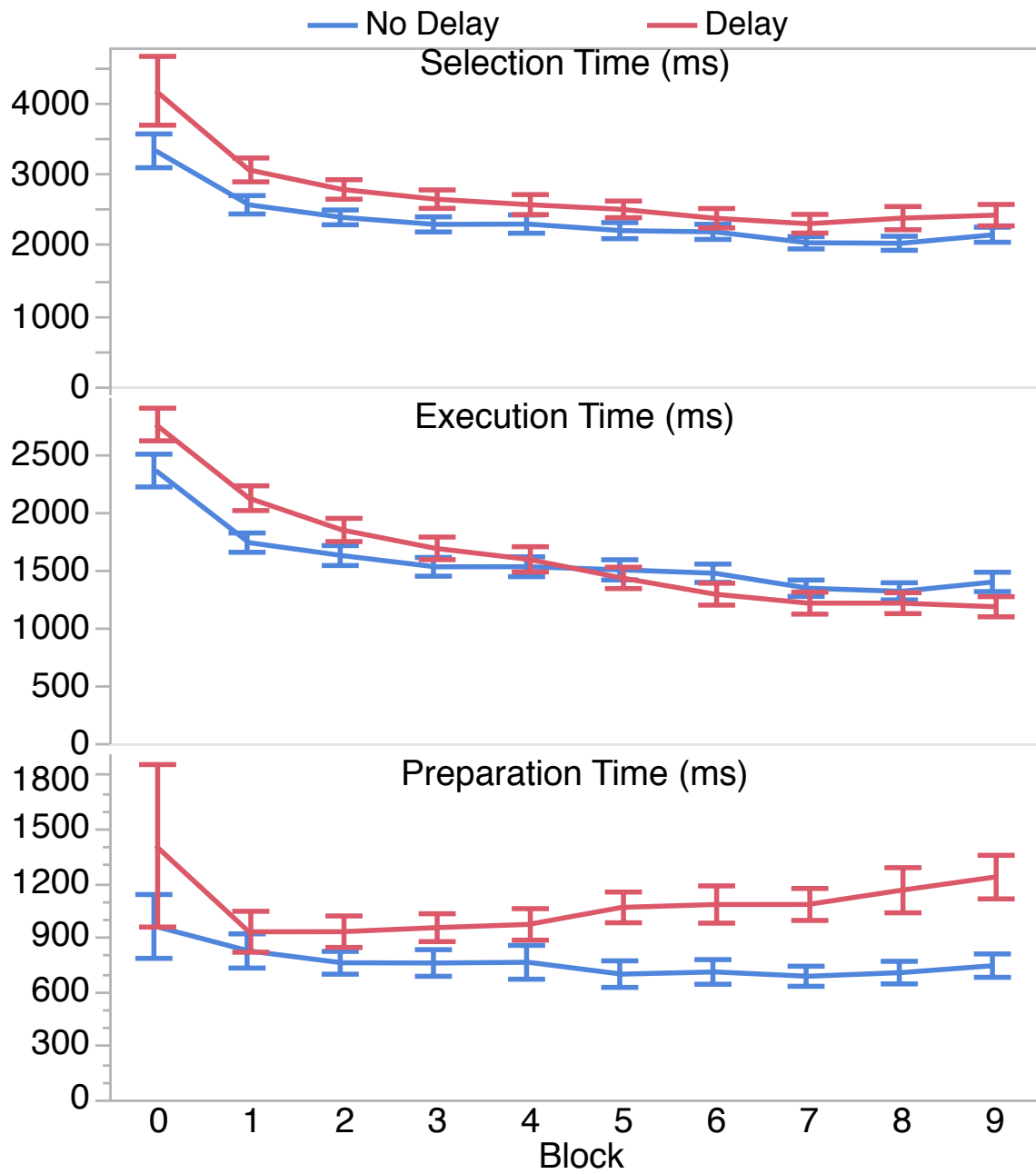


Figure 3.2: Average *Selection*, *Execution*, and *Preparation Times* per BLOCK and MARKING MENU condition. Error bars are 95 % CI.

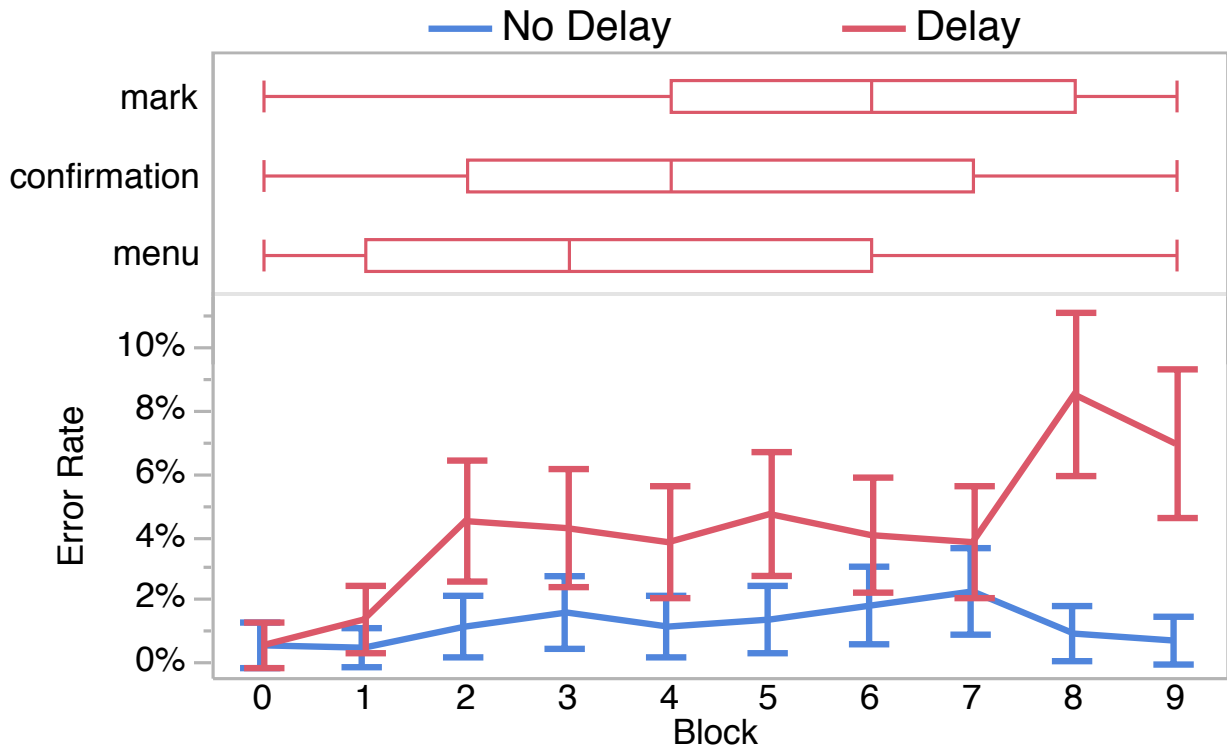


Figure 3.3: Top: Distribution of modes by BLOCK in the DELAY condition. Bottom: Average *Error Rates* per BLOCK and MARKING MENU condition. Error bars are 95 % CI.

in *mark mode* (mean 8.3 %) than in *menu* and *confirmation modes* (resp. 2 and 1.2 %).

H3 Practiced Performance

Our third hypothesis proposes that DELAY marking menus should outperform NO DELAY marking menus because mark mode should be faster to execute than menu mode, even in a NO DELAY condition because the user does not need to perceive the menu before acting. To examine H3, we need, first, an understanding of how common *mark mode* is during the DELAY. Next, we need some measure of *Practiced Use* for the NO DELAY condition, because NO DELAY has no easily distinguishable *mark mode*. Finally, we can use this understanding to contrast performance with DELAY and NO DELAY during practiced use.

To, first, explore *how common the use of mark mode is* during interaction, recall that the DELAY condition exhibits three modes, a *menu mode* (for novice use), a *mark mode* (for practiced users), and a *mark confirmation mode* to facilitate the transition between

the latter two. In DELAY, we would expect use of *mark mode* to increase over time. We would also expect more complex menus to result in reduced use of *mark mode*. Graphically, Figure 3.3, top, shows the evolution of *mark mode* use later in the study, beginning around BLOCK 4 or 5. This observation corresponds to statistical analysis: we found a significant effect of BLOCK ($F_{9,585} = 38.5, p = .0001$) and LAYOUT ($F_{3,585} = 76.78, p = .0001$) on the use of *mark mode*. It increased overall with BLOCKS (see Fig. 3.3-top), ranging from 8.1 % in BLOCK 0 to 62.3 % in BLOCK 9. It also significantly decreased with menu complexity: LAYOUT 8×8 (mean 28.9 %) \ll 4×4 (36.3 %) and 8 (37.2 %) \ll 4 (67.4 %). There was no significant interaction effect.

Analyzing the next two questions, *some measure of Practiced Use for the NO DELAY condition*, and how to *contrast performance with DELAY and NO DELAY during practiced use* is complicated to disentangle. First, considering Figure 3.2, we see that the curve of selection time (ms) versus BLOCK for DELAY and NO DELAY mirror each other. We, therefore, first examine BLOCK effects, then look at *Execution Times* and *Preparation Times* more deeply.

Contrasting *Selection Time* across BLOCKS (Fig. 3.2 left) post-hoc tests showed that BLOCKS 7-8 with NO DELAY are significantly shorter (mean < 1777 ms) than BLOCKS 0-5 with DELAY (> 2175), but only than BLOCKS 0-2 with NO DELAY (> 2139). Conversely, BLOCKS 6-9 with DELAY (< 1973) were *not* significantly different from each others, nor from any NO DELAY block other than 0 (2800 ms). Given that *Selection Time* appears stable in DELAY BLOCKS 6-9, and that there is no significant difference between them and any NO DELAY blocks, we cannot, *prima facie*, reject H3's null hypothesis.

One factor that may prevent us from rejecting H3's null hypothesis is that it is hard to match a *mark mode* (DELAY) with an equivalent binary criterion in the NO DELAY condition, because the menu always appears. Perhaps participants in NO DELAY perform more “practiced” command selections.

To test this, we can assume that the “practiced” selections in any given condition form a subset of trials that span all blocks, and with distinctively good time performance. In order to properly compare practiced performance between conditions, we identify the BLOCKS in which *Selection Time* stabilized towards its minimum using Hsu's HSB contrast, i.e. blocks 7 to 9. For *Selection Time* on these “stable” BLOCKS, we found a significant effect of LAYOUT ($F_{3,105} = 60.51, p = .0001$), but no longer of MARKING MENU nor any interaction between the two. Discarding trials with selection errors, we found significant effects of DELAY MODE ($F_{2,414} = 331.57, p = .0001$) and BLOCK ($F_{9,413.1} = 2.31, p = .05$) on median *Selection Time*. For the DELAY condition, participants were significantly faster in *mark mode* (mean 1445 ms) than in *menu* and *confirmation modes* (resp. 2982 and

3102 ms). However, DELAY does not outperform NO DELAY in terms of overall selection time; it remains the case that there is no significant difference between DELAY and NO DELAY during practiced use.

A final analysis we can perform involved breaking down *Selection Time* into *Execution Time* and *Preparation Time* to try to identify, with finer granularity, contrasting effects of DELAY and NO DELAY. We found a significant effect of MARKING MENU ($F_{1,105} = 7.12$, $p = .0088$) and LAYOUT ($F_{3,105} = 41.2$, $p = .0001$) on *Execution Time*, with no interaction. Post-hoc tests revealed that the DELAY condition has a smaller *Execution Time* (mean 948 vs. 1130 ms), and a similar LAYOUT effect as before. Finally, we found a significant effect of MARKING MENU ($F_{1,105} = 50.61$, $p = .0001$) and LAYOUT ($F_{3,105} = 6.32$, $p = .0007$) on *Preparation Time*, again with no interaction. Post-hoc tests revealed that the DELAY condition has a larger PREPARATION TIME (mean 830 vs. 569 ms), and a similar LAYOUT effect as before.

In summary, for practiced use, we find no significant difference between DELAY and NO DELAY in terms of *Selection Time*, and that while *Execution Time* is lower with DELAY (182 ms difference), it is compensated by a lower *Preparation Time* (261 ms difference) for NO DELAY, regardless of the LAYOUT.

3.5 Study 2: investigating expert performance

Study 1 supports both H1 and H2 – NO DELAY has significant advantages over DELAY even as users become practiced. H3, that DELAY outperforms NO DELAY in highly practiced use, was not supported; DELAY suffered from higher error rates and reaction times, even when selecting commands consistently in mark mode. However, it may be that, with sufficient practice, users could reach a theoretical level of perfect expertise – autonomic response – that would result in an overall performance benefit. Thus, study two was designed to balance the need for the “best case” of an autonomic response, versus the confound of anticipation, or the cost of deciding what selection to perform. With this design, we are able to examine the limits of expert level use to identify whether – with fully autonomic reaction – we can quantify a benefit from DELAY.

3.5.1 Experimental Procedure

Participants

Eight participants (average age 26.88, $SD = 2.56$) were recruited for the study, two who identified as female, six as male. All were either graduate students or post-doctoral researchers from technical disciplines in a technically focused university.

Task and Stimulus

The study application was a modified version of Study 1, run on the same computer and with the same mouse. Prior to completing the study, users were familiarized using the same process as study 1. The experiment followed the same procedure as study 1 except that:

- Users had to select only two items from each of two different menus repeatedly in DELAY and NO DELAY conditions.
- The two items per menu were carefully balanced through pilot studies to ensure equivalent speed and precision.

We chose the items to select in each menu based on geometric characteristics: one acute angle and one right angle per menu. Prompted items for menu 1 were ↖ [north][south-west] and ↘ [south][east], and for menu 2 ↗ [west][north-east] and ↙ [east][south].

For each trial, the participant had to select one of the two target commands, whose label was displayed on top of the window, within a MARKING MENU of 8×8 (64) items. For each menu participants performed 8 BLOCKS of 10 selections per command. Ordering of the commands was randomized within each menu. Similarly to study 1, we used different command sets in each of the two 8×8 menus.

The result was a fully counterbalanced, within subjects, 2X2 design (MENU – DELAY/NO DELAY and ITEMS). In total, we collected 2 commands per marking menu \times 10 prompts per command \times 8 blocks \times 2 MENUS \times 8 participants = 2560 selections (640 per command) in total.

At the end of each MENU condition, we instructed participants to select the items of the corresponding pair 4 times each using arrows as instructions, to obtain a temporal floor for gesture performance. Dependent measures and analysis method followed those of study 1.

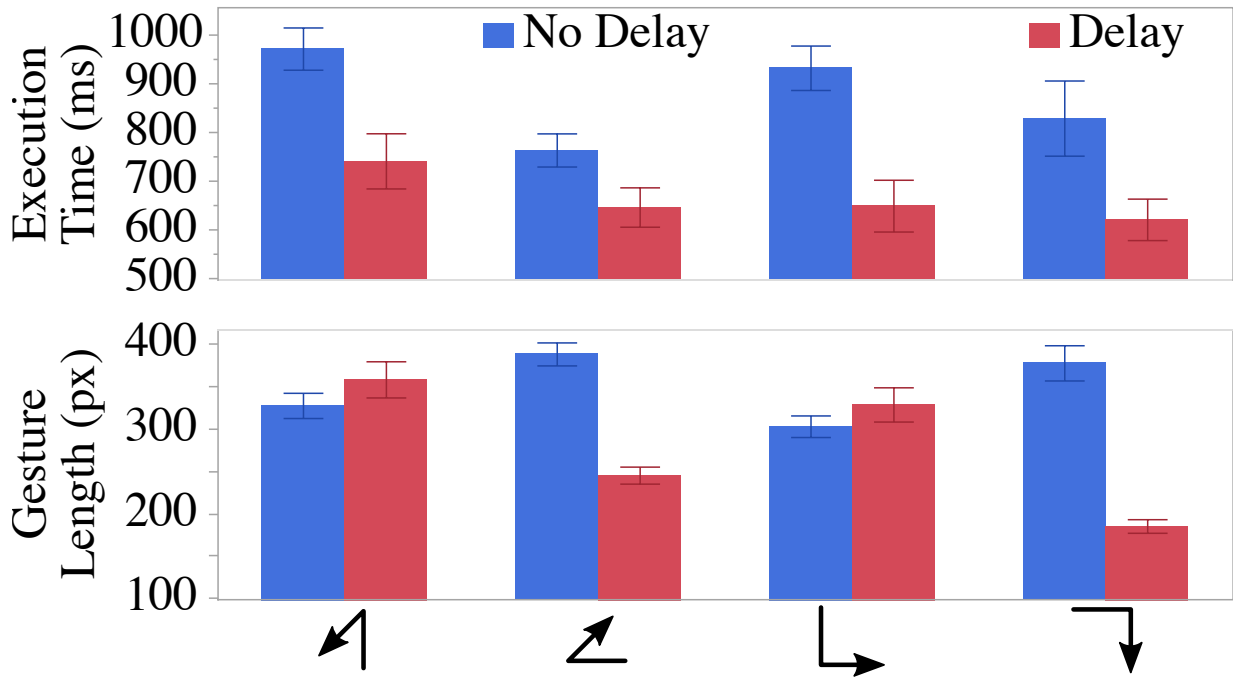


Figure 3.4: Effects of MENU CONDITION on *Execution time* vs *Gesture length* for each item. Mind the non-zeroed Y-axis on top.

3.5.2 Results

Fig 3.4 shows the overall *Preparation*, *Execution*, and *Selection* times throughout the study for each MENU condition. We found a significant effect of BLOCK on *Preparation time* ($F_{7,2545} = 14.59, p < 0.0001$), on *Execution time* ($F_{7,2545} = 49.83, p < 0.0001$), and on *Selection time* ($F_{7,2545} = 47.55, p < 0.0001$). In all three measures, BLOCK 0 took significantly longer than the rest (300 ms to 1 s), with no other significant difference between BLOCK numbers. There was no effect of BLOCK on *Error rate*. We excluded BLOCK 0 from further analyses.

We found significant effects of MARKING MENU on *Execution time* ($F_{1,2231} = 164, p < 0.0001$) and *Selection time* ($F_{1,2231} = 75.1, p < 0.0001$). Selections with DELAY were faster to *Execute* (mean 663.14 ms vs. 873.08 ms with NO DELAY), which was directly translated into *Selection time* (mean 1380.37 ms, vs. 1633.01 ms with NO DELAY), i.e. a total improvement of about 250 ms. We therefore reject H3's null hypothesis and conclude that extensive training with two targets yields faster command selection in *mark mode* with DELAY marking menus, than with NO DELAY marking menus. There was no significant

effect on *Preparation time* nor on *Error rate*.

The other factor that significantly impacted time and error rate was ITEM. ITEM had a significant effect on *Execution time* ($F_{3,2229} = 16.75, p < 0.0001$) and *Selection time* ($F_{1,2229} = 6.78, p < 0.0001$). ITEM ↖ had significantly longer *Selection time* (mean 1608 ms) than ↘ and ↗ (means $\bar{}$ 1480 ms), with ↙ in between and not significantly different than the other three. For *Execution time*, we had ↖ (mean 855 ms) \ll ↙ (790 ms) \ll ↘ and ↗ (means $\bar{}$ 725 ms). We also found a significant effect of ITEM on *Error rate* ($F_{3,2229} = 2.76, p < 0.05$): ↙ (mean 2.32%) caused significantly fewer errors than ↗ (mean 5.54%). We found no effect on *Preparation time*, nor interaction effect.

Next, we explored the common hypothesis that MARK mode is faster because it allows smaller marks. We found significant effects of MARKING MENU ($F_{1,2226} = 211.48, p < 0.0001$), ITEM ($F_{3,2226} = 27.55, p < 0.0001$), and MARKING MENU \times ITEM ($F_{3,28} = 5.12, p < 0.01$). DELAY caused shorter strokes overall (mean 279 vs. 349 px), and ↖ (343 px) \gg ↗ & ↙ (316 px) \gg ↘ (281 px). No clear pattern emerges from the Tukey test for the interaction effect, and correlations (R^2) for linear regressions between *Execution time* and *Gesture length*, for each MARKING MENU, are both 0. They all remain below 0.3 (mostly below 0.1) if regressions are performed independently for each participant. Overall, *Gesture length* seems a poor predictor of *Execution time* in the context of marking menus.

3.6 Experiment 3: Assessing Visual Disruption

Our previous studies suggest that NO DELAY marking menus yield fewer errors and lower command selection times overall for all but extremely practiced targets. Kurtenbach’s original design rationale for marking menus includes one additional benefit of *mark mode*, i.e. that no menu is displayed and that menus “*can be distracting*” and “*can obliterate part of the screen*” [127]. H4 presents this hypothesized benefit of DELAY. While studies 1 and 2 evaluated marking menu performance with and without delay *for prompted commands*, generalization to real-world use-cases with occlusion and visual distraction requires we test marking menus in an interactive program where occlusion of content might inhibit interaction.

To evaluate occlusion and disruption, we conducted a third experiment. In a 2×2 within-subjects protocol, we evaluated the effects of DELAY vs. NO DELAY marking menus in two simple yet realistic graphic arrangement applications in which participants were instructed to replicate existing figures: one involving CARTOONS and a second involving

FLOW CHARTS. Our rationale for this design is twofold: it requires many repeated menu selections, so participants can exhibit practiced behaviour and possibly reach a higher level of expertise with the menus; and it creates a crowded canvas with many elements, requiring manipulation through contextual menus, so occlusion may become an issue for users.

3.6.1 Experimental Procedure

Participants and Apparatus

Sixteen paid participants were recruited for the study. Average age was 24.38 (SD = 2.16). Three participants identified as female and the remaining 13 identified as male. While all participants came from technical backgrounds, none of them participated in the first two studies. 2 participants had heard about Making Menus before (4 unsure, 10 no), 2 had already interacted with one (5 unsure, 9 no), and 10 had already interacted with a Pie/Radial Menu (1 unsure, 5 no).

The interface used the same display monitor and marking menu implementation as in Study 1. Mouse input was obtained through a VicTsing Slim Wireless Mouse. The experimental software was an object arrangement application in Javascript. A right button press on the page triggered a main, two-level marking menu containing the various available image items, arranged in categories, which upon selection were spawned onto the page. These images could then be dragged into and within the scene, using the left mouse button, for precise positioning. A right button press on any of those images triggered a contextual marking menu containing manipulation functions such as "rotate", "send to front", "delete", etc. whose effects were applied, upon selection, to the right-pressed image.

Task and Stimulus

As in the previous experiment, users were administered a verbal and visual demonstration of how to use marking menus within both conditions, including all modes and verbal instruction of how to use the drag and drop interface. Participants did not interact with the interface until the experimental task began. The task began with a reference image appearing in the top panel of our application (see Figure 3.5). Participants were then instructed to recreate the image displayed in the above panel to the below panel by "spawning" and modifying items using marking menus. Participants were asked to complete four of these tasks, two per menu condition. This experiment employed a think aloud protocol [134] inviting participants to comment while recreating the images.

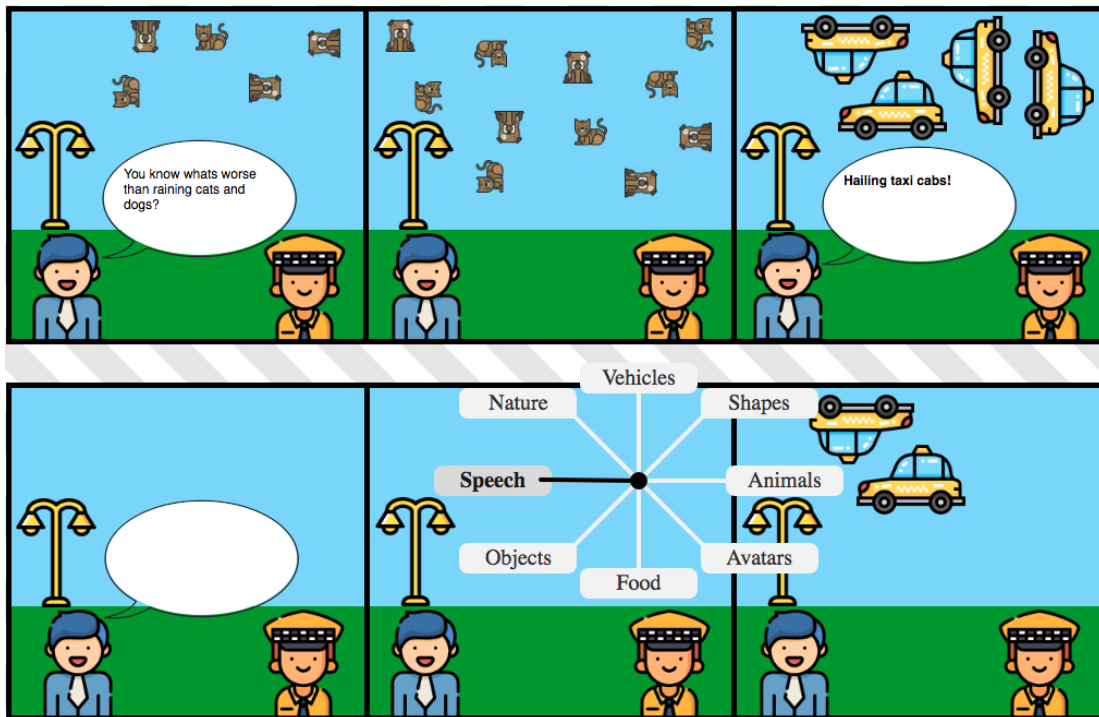


Figure 3.5: An example of a user interacting with the drag and drop application. On top is the figure to recreate. On the bottom is the participant's current figure with an ongoing menu selection. Icons in this figure were designed by Freepik and Smashicons from www.flaticon.com.

There were two command sets: one for CARTOONS and one for FLOW CHARTS. Each main menu had the same layout, but different configuration. For instance, the CARTOONS menu had a single one-level item in the left stroke direction and the FLOW CHARTS menu had a single one-level item in the right stroke direction. The layout consisted of a single one-level item, one five-item sub-menu category, three 4-item sub-menu categories and three 8-item sub-menu categories. The contextual menus remained consistent throughout the experiment, with a contextual menu for text (including font, style, alignment and delete) and images (including rotate or change colour, send to front, send to back and delete). Each task required the same minimum number of menu selections from each level and size of submenu.

Design and Analysis

The experiment used a 2×2 within-subjects design, with the following factors and levels:

- Main command sets / image theme: two different command sets, CARTOONS and FLOW CHARTS, were used for each menu condition to control for confounds of learning behaviour and confusion between categorical selections.
- MARKING MENU (DELAY or NO DELAY) which remained consistent with those explained in section 3.4.

Order of menu conditions and command sets were counter-balanced, each combination happening four times per subject.

Subjective Data Collection

Instead of determining whether occlusion occurs or if the display is “visually disruptive”, our goal is to assess whether these factors affect the perceived usability of marking menus with and without delay. To compare the subjective experience of DELAY and NO DELAY, in addition to capturing spontaneous comments expressed during the tasks, participants were administered a questionnaire after each condition, comprised of Likert-scales (1 = Strongly Disagree to 7 = Strongly Agree). Questions and responses are shown in Fig. 3.6.

We also recorded think-aloud comments and conducted an exit interview. All subjective data (Likert, think-aloud, and interviews) were leveraged for analysis: the Likert responses as ordinal statistical data, and the qualitative data via transcripts and focused coding regarding occlusion and disruption.

3.6.2 Results

Practiced performance When users can decide the order they wish to perform the overall task it, becomes a challenge to assess selection errors. To confirm that the study was long enough for users to become experienced with the menu layouts, we counted selections in the DELAY condition performed in each available mode (menu, mark, confirmation) that were not cancelled before completion. Among the 1279 successful command selections performed by all 16 participants in DELAY condition, 48% (615, mean 38.4 per participant, SD 17.5) were performed entirely in *menu* mode, 18% (225, mean 15/p, SD 9.2) involved in *confirmation* mode, where participants pause upon completion to verify whether they

successfully selected the command, and 34% (439, mean 31.4/p, SD 19.7) were performed entirely in *mark* mode. Because over 1/3 of DELAY commands were invoked in *mark mode*, we believe that this indicates that participants were, generally, able to learn the menus.

Subjective preferences Some overall trends appeared for both conditions. Questions on performance with items whose location is already remembered (a, b) received neutral or positive scores (4-7). Participants were not generally bothered by occlusion: only one scored above 5 to question (g) (NO DELAY condition). Finally, only two participants scored below neutral (4) to memorization (e) and real-world use (i) questions, both for DELAY.

An ordinal logistic fit found a single significant effect of MARKING MENU on the answers to question (h): “The [DELAY or ABSENCE OF DELAY] made me lose focus on my main task” ($\chi^2 = 3.97, p = .046$), with DELAY (mean score 4, most frequent scores 1, 3, 4, and 7 each with N=3) found to be more problematic than ABSENCE OF DELAY (mean 2.56, most frequent score 1 with N=7). For all other questions, we found no significant effect of MARKING MENU. Ratings are summarized in Fig. 3.6.

At the end of the study, 7 participants preferred DELAY, 7 preferred NO DELAY, and 2 expressed no preference.

Subjective Comments We did not observe a consensus among participants against DELAY or NO DELAY marking menu in terms of disruption or disturbance. For example, P_8 said they “*didn’t find any disturbance [in (NO DELAY)]*” and P_{12} mentioned “*I noticed no disturbance or disruption [in NO DELAY] because I could always control visibility of the marking menu by simply releasing the right mouse button*”.

A single participant (P_6) reported an issue with object occlusion in the NO DELAY condition, specifically when attempting to rotate an arrow image: “*I cannot see which direction the arrow currently is, so I don’t know how to rotate it. The arrow is hidden, the menu should come somewhere else without hiding the picture [...] The menu should not hide the existing element about to be manipulated*”. That being said, this particular issue impacted only one participant and would be present in either condition to a user unfamiliar with the menu layout. One can expect that a user familiar with the menu would not forget the orientation of the arrow while performing the command selection gesture. In fact, P_{12} noted, in the DELAY condition, “*the fact that the menu hid some of the content was only problematic because I knew that closing the menu just to see what’s behind would come at the additional cost of waiting when re-opening it later*”.

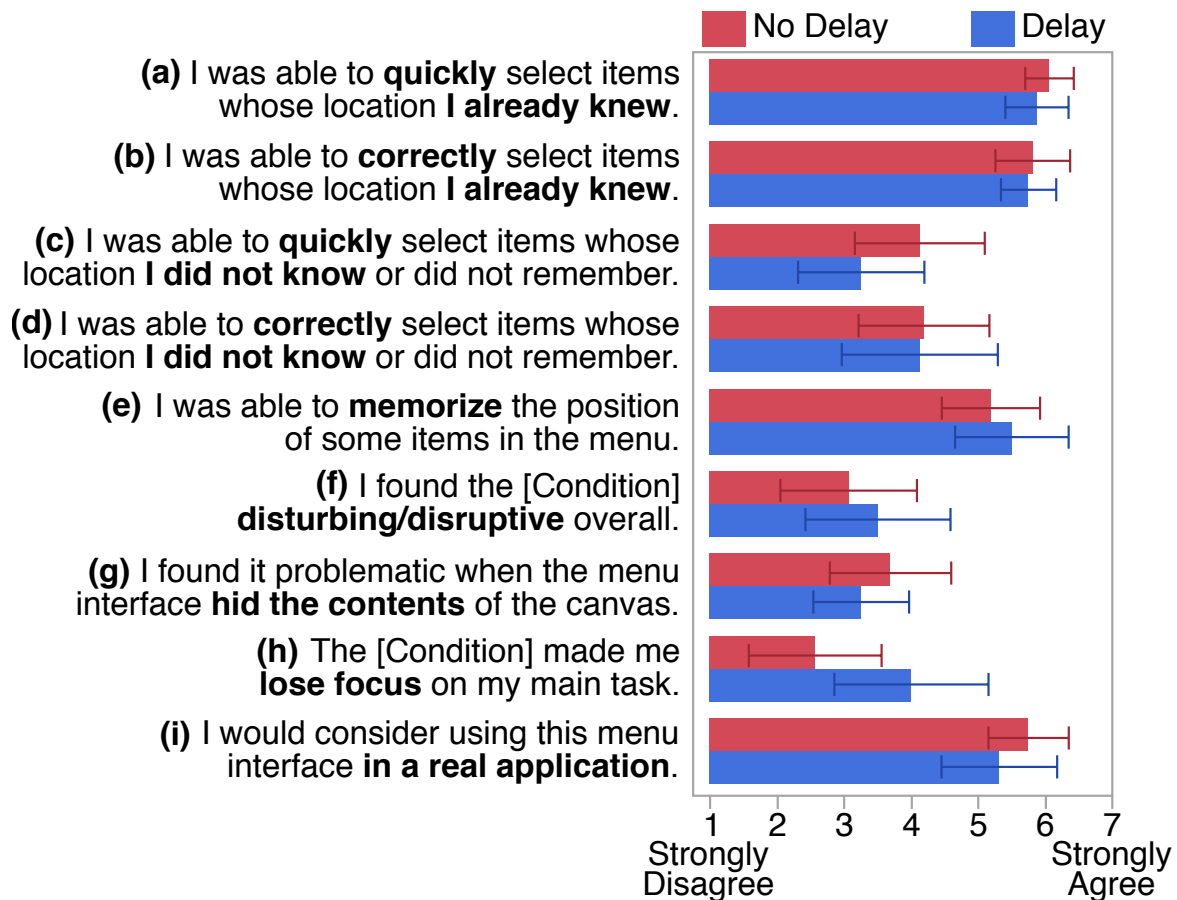


Figure 3.6: Likert scale questions (error bars are 95% CI).

In terms of speed, echoing findings in the initial experiment and Likert questions (a) and (c), P_{15} noted, “*I’ve been working with Photoshop, it’s basically like this [system] ... the user wants to do the process as fast they can, so they don’t want to wait [in the DELAY condition], it’s slow*”.

3.7 Discussion and conclusion

Delay has been an integral component of menu mode activation in marking menus since their introduction. Despite this, little has been done to explicitly evaluate the costs and benefits of marking menu delay to learning, performance, and visual disruption. There is

good reason to evaluate this trade-off, however, because there exists a theoretical tension around the use of delay. Specifically:

- Delay penalizes the novice experience, which may frustrate learners – but may also promote faster learning.
- Delay stresses recall over recognition, which may lower throughput via errors and high cognitive load – but may also speed performance for well-known, frequent commands.
- Delay eliminates visual occlusion of content and visual disruption of the interface for practiced commands – but it is unclear how costly visual occlusion and disruption are during command invocation, especially if it is expected.

Our hypotheses probe these questions, exploring overall cost during learning and into early practiced use [H1, Experiment 1], error rate [H2, Experiment 1], speed during highly practiced use [H3, Experiments 1 and 2], and perceived visual disruption [H4, Experiment 3]. We find support for H1 and H2, that DELAY does significantly impact the cost of marking menus. We also find mixed results for H3, with no benefit for DELAY observed during *mark mode* activations in experiment 1 and only a benefit during autonomic command invocation in experiment 2. Finally, in experiment 3, we note limited issues around visual occlusion despite the fact that *mark mode* was used 34% of the time during the DELAY condition; in fact, in experiment 3, we noted significantly more impact due to the cost of DELAY than issues with occlusion.

Our difficulty finding support for H3 and H4 was somewhat surprising, as the assumption with marking menu is that *mark mode* use should have performance benefits, and that *mark mode* eliminates visual disruption. Temporal benefits of *mark mode* were difficult to identify in experiment 1. Visual disruption issues were difficult to identify in experiment 3. Only in study 2 did we identify a statistically significant benefit of *mark mode*, and then only for two highly practiced commands.

The results of study 1 and 2 in concert suggest that extensive training may be required to overcome performance costs, and it remains unclear whether users would be able to reach expertise with enough commands for the delay to be beneficial overall when novice, practiced, and expert behaviours are considered altogether. Assuredly, it is virtually impossible for a user to remember a complete menu layout: even projections using a Zipfian distribution in a 8 items only marking menu suggest that users would be far from selecting all commands in *mark mode* after extensive use, selecting only 43% when delay is 200 ms delay, 55% with 333ms, and only reaching a higher rate with significantly longer delays, up to 2s, but at the cost of a doubling or tripling of error rate [132].

The most controversial result in our work is the lack of subjective impact of visual disruption in Experiment 3. However, in hindsight it is questionable whether this result should be surprising: consider past work by Cockburn et al. on menu performance [50, 51] and past work on “mousing ahead” by Hopkins [103, 123]. In Cockburn et al.’s work, novice (or unpracticed) performance is modeled via linear visual search because the user still needs to read each menu item to find their desired target, and expert (practiced) performance is modeled via the Hick-Hyman Law because the user simply mentally selects the desired action from among candidates without the need to look for menu options. Cockburn et al.’s model includes no temporal cost that results from visual disruption in their model of menu access, and their model correlates perfectly with user performance. Regardless, visual disruption may also be an over-stated issue: while one way to eliminate visual disruption caused by occlusion is to avoid displaying the menu, partial transparency of the menu may also allow users to continue to see underlying content without the need to include novice mode penalty and increased error rate in marking menus.

While our results begin to explore the costs and benefits of DELAY in marking menu interfaces, additional work remains. In our email questionnaire, Kurtenbach noted that, in industry, he has used delay values significantly shorter than those in the academic literature – around 100ms, and that many users fail to discover *mark mode*. These shorter delays may reduce error, speed novice performance, and still support autonomic performance as per experiment 2. We also note limitations in our work, including a relatively small sample size and the lack of an in-situ scenario, i.e., utilizing an application where marking menus are actually used, such as Autodesk’s Maya.

Our results open similar questions regarding other command selection techniques that rely on delay-separated modes [71, 73, 75, 89, 90] to implement the rehearsal design principle. Future work should investigate whether similar results would be found with these interfaces. For example, the FastTap technique displays a full-screen grid that may result in higher perceived visual disturbance.

In the end, our results are valuable to designers exploring the use of Marking Menus and other rehearsal-based interfaces. Understanding DELAY’s benefit for highly practiced commands allows designers to choose to incorporate delay depending on their perspective of whether the system being designed should penalize novice and in order to force transition to mark mode or other “expert” modes in rehearsal-based interaction. This work is, to our knowledge, the first quantitative and qualitative exploration of the relative costs of these factors in rehearsal-based interface design.

Chapter 4

Presenting a Use Case of When Mode Transfer is Beneficial

Taking into account the drawbacks of mode separation in marking menus exhibited in Chapter 3, we then raise the question of under what circumstances do users need to transition from a novice mode to a secondary mode? In other words, in what scenarios do users need to rely on recall as opposed to self-revelation via guidance? It would be reasonable to assume these scenarios arise when revealing gestures to users is impractical.

While mid-air gestures are an attractive modality with an extensive research history, one challenge with their usage is that the gestures are not self-revealing. Scaffolding techniques to teach these gestures are difficult to implement since the input device, e.g. a hand, wand or arm, cannot present the gestures to the user. In contrast, for touch gestures, feedforward mechanisms (such as Marking Menus or OctoPocus) have been shown to effectively support user awareness and learning. In this chapter, we explore whether touch gesture input can be leveraged to teach users to perform mid-air gestures. We show that marking menu touch gestures transfer directly to knowledge of mid-air gestures, allowing performance of these gestures without intervention. Thus, we argue that cross-modal learning has potential to be an effective mechanism for introducing users to mid-air gestural input.

4.1 Motivation

Mid-air gestures are an attractive method of interaction in the context of internet of things (IoT) and ubiquitous computing environments (ubicom). Free space gestures have the

ability to provide eyes-free input, a benefit for many IoT devices that do not have a display to guide users, e.g. a smart light-bulb or smart thermostat. In ubiquitous interaction scenarios, mid-air gestures can free the external display to hold information pertaining to its particular context, rather than an arbitrary gesture scaffolding.

Mid-air gestures are still rarely deployed in practice due to three primary challenges: reliability in tracking and recognition [112, 113], user fatigue [98, 201], user discomfort [3], and user awareness [110] (i.e. gestures are not self-revealing [25]). In this work, the primary interest is in examining ways we can address the challenge of user awareness of mid-air gestures; specifically, how can we help users learn an extensive mid-air gesture set.

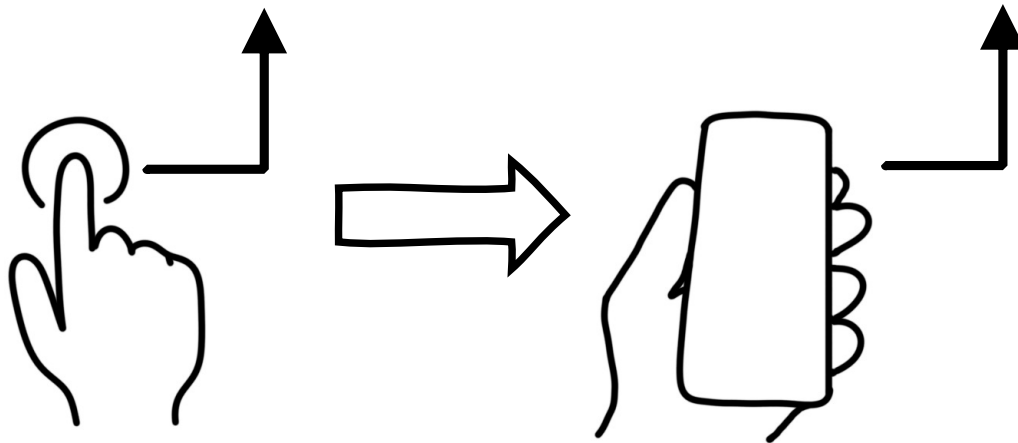


Figure 4.1: Visualization of transferring gestures from touch to mid-air.

One primary advantage of surface gestures with respect to this challenge is that surface gestures – because they are frequently performed on a display surface via direct manipulation – have a natural visual feedback mechanism via a screen that allows feed-forward techniques [24] to guide the user to a particular gesture that they wish to complete. These feed-forward, rehearsal-based techniques generally display the structure or path of a gesture, as in Bau et al.’s OctoPocus [24] and Kurtenbach et al.’s Marking Menus [125]. Mid-air motion gestures lack a natural visual display to scaffold gesture learning unless additional hardware is deployed, which, for the majority of work, is accomplished by including displays such as screens or projections [1, 10, 9, 47, 59, 70, 110, 177, 184, 197, 222, 230].

Requiring a display constrains the interaction environment – negating IoT contexts (as mentioned prior) and/or consumes all or part of that display with gesture representations – either permanently, reducing usable display space for other uses, or temporarily, requiring some means to invoke the gesture help display. Even in the presence of these training mechanisms, it is often the case that training occurs as a separate task for the user [36, 110]; in contrast, feed-forward techniques allow a user to learn the gestures while performing them in context [24, 125].

In this chapter, we explore whether we can leverage surface-based gesture representations to teach users mid-air gestures. Leveraging marking menus [125, 230], we train users in one of two-ways: 1) We show marking menus on an external display to reveal gestures to users [110]; and 2) We show marking menus performed and displayed on a touch-screen and explore whether users can map 2D actions learned on the touchscreen onto mid-air actions, i.e. *cross-modal* training. We evaluate both the error rate and speed of interaction and find no statistically significant differences between touch and mid-air training on mid-air performance of gestures. We find, somewhat counter-intuitively, that error rate is unaffected for participants trained on a mid-air gesture set using touch-based training. The only cost we observe in touch-based training of mid-air gestures is in the first block of the experimental phase of our two-phase (training and experimental phases) study, where participants’ speeds differed significantly for only that first block as they move from one modality (touch) in the training phase to a new modality (mid-air) in the experimental phase [50]. To the best of our knowledge, this work represents the first instance of evaluating how well touch can be leveraged to train users to perform mid-air gestures.

4.2 Assessing Touch-based Teaching of Mid-Air Gestures

In past work, the primary mechanism for providing feedback and teaching users to perform mid-air gestures is through a visual representation of the current and required movement path, typically via an external display [1, 10, 9, 47, 58, 59, 70, 110, 177, 222]. While this makes sense in environments where external displays exist, in other environments (e.g. environments populated with IoT artifacts), it may be the case that the environment does not have physical displays and provides output more subtly. The other tension within this space is, if we wish to train with surface gestures, then why move to motion gestures? We believe that, alongside motion gestures representing an effective means for combining target and command, motion gestures can serve as an eyes-free shortcut to command with the attendant benefit that the touchscreen is not impacted. We view motion gesture input

as a form of short-cut, analogous to a system-wide hotkey, to support dedicated gestural input.

While our earlier discussion treated mid-air gestural input as a generalized input modality, in our evaluation we focus specifically on motion gesture input, i.e. mid-air gestural input where the user moves a mobile device in order to issue commands. Our rationale for this is similar to that of Vatavu et al. [215]: users are familiar with their personal devices, almost always have them with them, and we can leverage the display as an opportunistic training platform and as a convenient, on-hand, input sensor to capture interactions. That said, we believe our results should generalize to bare-hand mid-air input, provided the environment supports hand-tracking to capture gesture input.

In this section, we describe participants, apparatus, and an experiment that explores the comparison of touch and mid-air training of mid-air gestures in detail.

4.2.1 Participants

Fourteen participants between the ages of 21 and 28 volunteered for the study. Average age was 24.47 (SD = 2.36). Six participants identified as female and the remaining eight identified as male. Participants were remunerated \$15 for their participation. 13 participants were post-secondary students and one was a teacher. Five participants had experience with marking menus (e.g. Maya, Pinterest for iPad, or other application contexts) and the remaining nine did not.

4.2.2 Apparatus

The visual interface was displayed either on an ASUS PB287Q monitor with 1080p resolution using an Nvidia Shield TV running Android 8.0 or on a smartphone. Participants interacted using a Huawei Nexus 6P running Android 8.0, with dimensions: $159.3 \times 77.8 \times 7.3$ mm, a weight of 178g and a 5.7" screen. Cross-device communication was facilitated through a dedicated WiFi network.

4.2.3 Mid-Air Pointing

Mid-air interaction was captured through sensor fusion on the mobile device, i.e. obtaining rotation of the Android smartphone as described in [167]. Our rotation technique maps changes in device orientation on the Yaw and Pitch axis, which can then be converted to

a 2D position relative to the center of the display. Participants were asked to keep the device’s roll with the screen facing up within 45 degrees in each direction, as this facilitated stability in orientation detection.

4.2.4 Task and Stimulus

The experiment was a 2-factor between-subjects experiment. Participants completed a demographic questionnaire followed by a two phase study, a training phase that leverages touch of mid-air followed by an experimental phase involving mid-air input.

In the *training phase* participants had a visual representation of a 4×4 , 16 item marking menu. For each trial in the training phase, a prompt was displayed on the top-left corner of the screen indicating what selection to make. Participants selected a command based on the prompt, which then displayed on the top-right hand side of the screen, in either green with a check-mark or red with an \times , to indicate a correct or incorrect selection, respectively. Each new prompt was one of 8 items from the marking menu and only displayed upon correct selection. Prompts were presented in random order for each block. Short breaks were given every 32 correct selections (4 blocks). The *training phase* consisted of 20 blocks of 8 selections for 160 selections in total. Participants were assigned to one of two conditions in the *training phase*, either TOUCH or MID-AIR, as follows.

- **Touch:** Participants learned the marking menu via a touch interface on mobile device. They were instructed to press their finger down on the screen, navigate through the menu (drawing a gesture on the screen), until they have reached the item they wish to select. Once they have completed the gesture, they release their finger. The experimenter gave a brief demonstration with no interface on the mobile device’s screen of how to complete a gesture. The interface was displayed on the mobile device screen.
- **Mid-Air:** Participants learned the marking menu via an external display positioned in front of them, motivated by prior work [1, 10, 9, 47, 58, 59, 70, 110, 177, 222]. They were instructed to navigate through a menu, using a mobile device as a “wand” to point where to select on the screen. Their movement path was displayed on the screen as they completed a motion. To begin a gesture, they press down on the volume down button on the device. When they have completed the gesture, they release the button. The experimenter gave a brief demonstration with no interface on the monitor of how to complete a gesture. Menu and motion path are displayed on the external display (to avoid obfuscation as in the Oakley et al. training [165]).

In both conditions, participants were asked to learn the menu as best as possible, because in the *experiment phase* they would have to complete the gestures in MID-AIR with no visual interface guiding them. In the *training phase* participants were required to select the correct target before moving to the next selection.

The *experiment phase* followed the exact procedure of the *training phase*, with the exception that no visual marking menu interface was displayed (Figure 4.4) and prompts changed upon every selection made (regardless of correctness, to keep a consistent experiment time and reduce frustration in the case participants did not learn the menu). The experiment phase was always conducted with MID-AIR gestures. Prompts and correct/incorrect indicators were displayed on the external monitor. The experimenter gave a brief demonstration with no interface on the monitor of how to complete a MID-AIR gesture, but participants were not permitted to confirm whether or not their interpretation was correct.

In each phase, participants were permitted a break after 32 (4 BLOCKS \times 8) selections. After each phase, both *training* and *experiment*, participants completed the NASA-TLX [72].

4.2.5 Design and Analysis

Our 2-factor between-subjects experiment included the following parameters: TRAINING CONDITION (TOUCH vs. MID-AIR, between-subjects), \times BLOCKS (20) \times commands (8) \times participants (14) for a total of 2240 command selections in the experiment phase.

Our analysis focuses on whether cross-modal training (i.e. learning mid-air gestures via touch input) is as effective as learning mid-air gestures via mid-air training. As a result, dependent measures are error rate and time to perform mid-air gestures in the experimental phase of the experiment. We also assess, quantitatively, whether mid-air training provides a statistical advantage versus touch training for learning mid-air gestures, and qualitatively, on how large the advantage of consistency in training (mid-air to mid-air) is compared to cross-modal (touch to mid-air) training.

4.3 Results

Our quantitative analysis consists of the error rate and timing of mid-air gestures. Error rate represents the fraction of mid-air gestures ($[0, 1]$) performed correctly in the experimental block given two training conditions – touch or mid-air training. Likewise, time

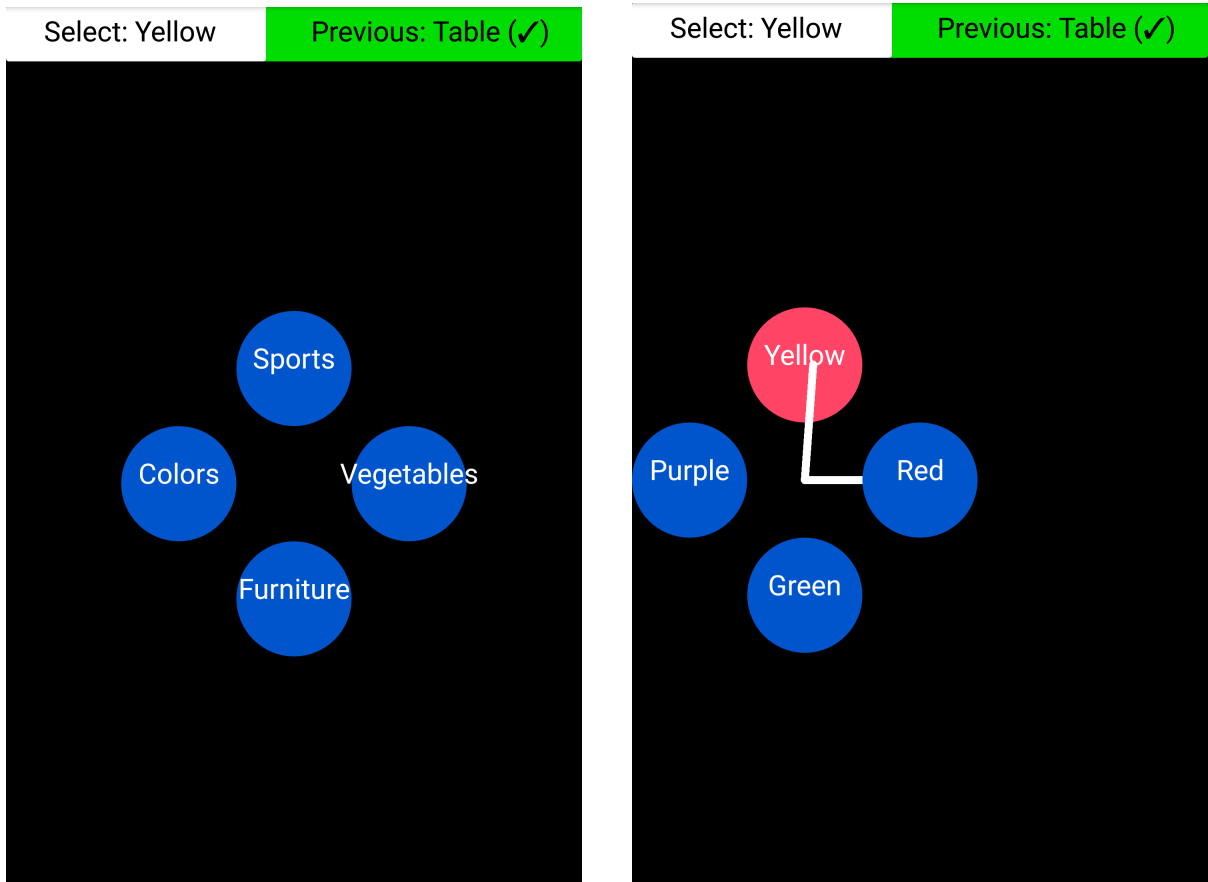


Figure 4.2: Interface interfaces for the TOUCH condition, showing the dragged motion path in white.

represents the time from prompt or from gesture initiation for mid-air gesture input in the experiment phase given each training condition in the training phase.

Before beginning our analysis, we performed a power and sample size determination. We set a threshold of 100ms for a temporal cost that would represent a significant difference in temporal performance. The standard deviation of our data set for time from prompt to selection was 1876ms and time from beginning a gesture to selection was 804ms. Because our data was not normally distributed, we applied a log-normal transform of our data, yielding a normal distribution. We found that, to support a 95% confidence interval for a 2-tailed analysis of effects, we required a sample size of at least 5 participants per condition, or 10 participants. Our data set of 14 participants exceeds this threshold.

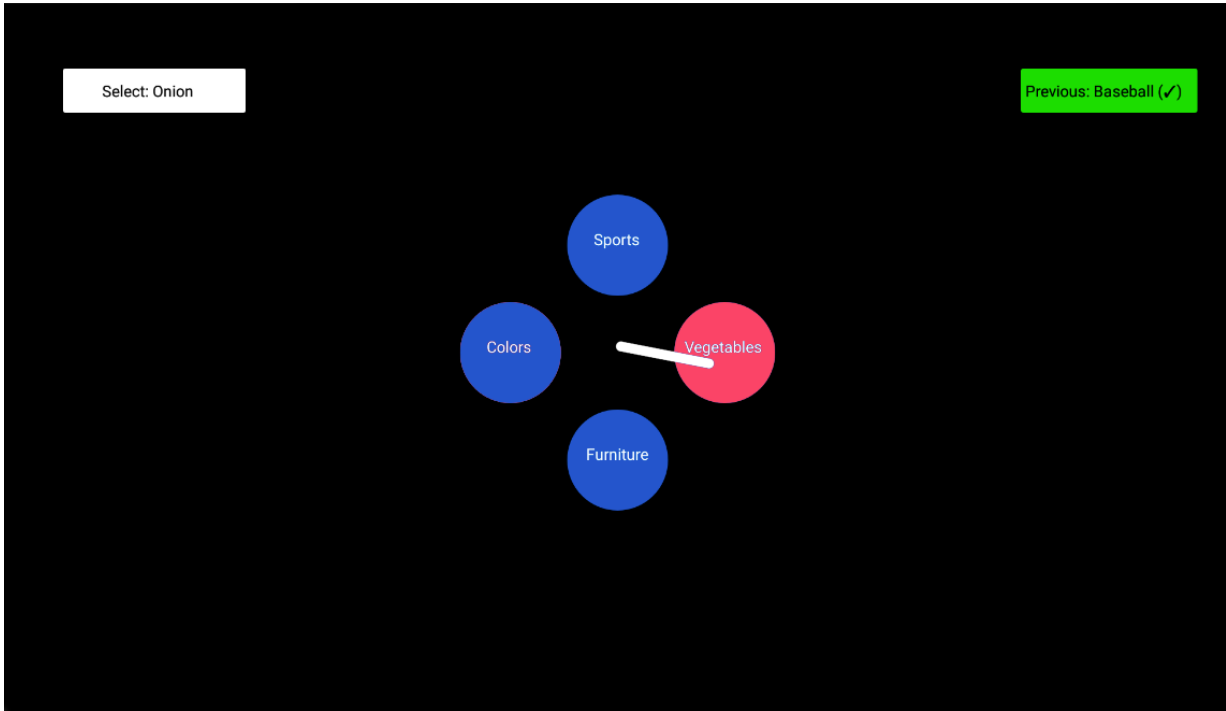


Figure 4.3: Interface for the MID-AIR condition on a monitor (external display), showing the movement motion path (i.e. pointer path) from the mobile device.

4.3.1 Error Rate

Figure 4.5 shows the fraction of gestures performed correctly in each condition. When participants train to perform mid-air gestures via mid-air training, 89% of gestures are performed correctly; interestingly, with cross-modal or touch training for mid-air gestures, the rate of correct gestures is slightly *higher*, 94%. We performed a χ^2 analysis of error rate and found that the difference was not significant ($p = 0.45$), indicating no significant difference in error rate for mid-air versus touch training of mid-air gestures.

4.3.2 Time

We ran an independent samples t-test on the mean time from prompt to selection and mean time from beginning a gesture to selection for participants in our study. Figures 4.6 and 4.7 show the length of time from prompt to command selection and from beginning of gesture (as measured by displacement of the device) and command selection. Differences

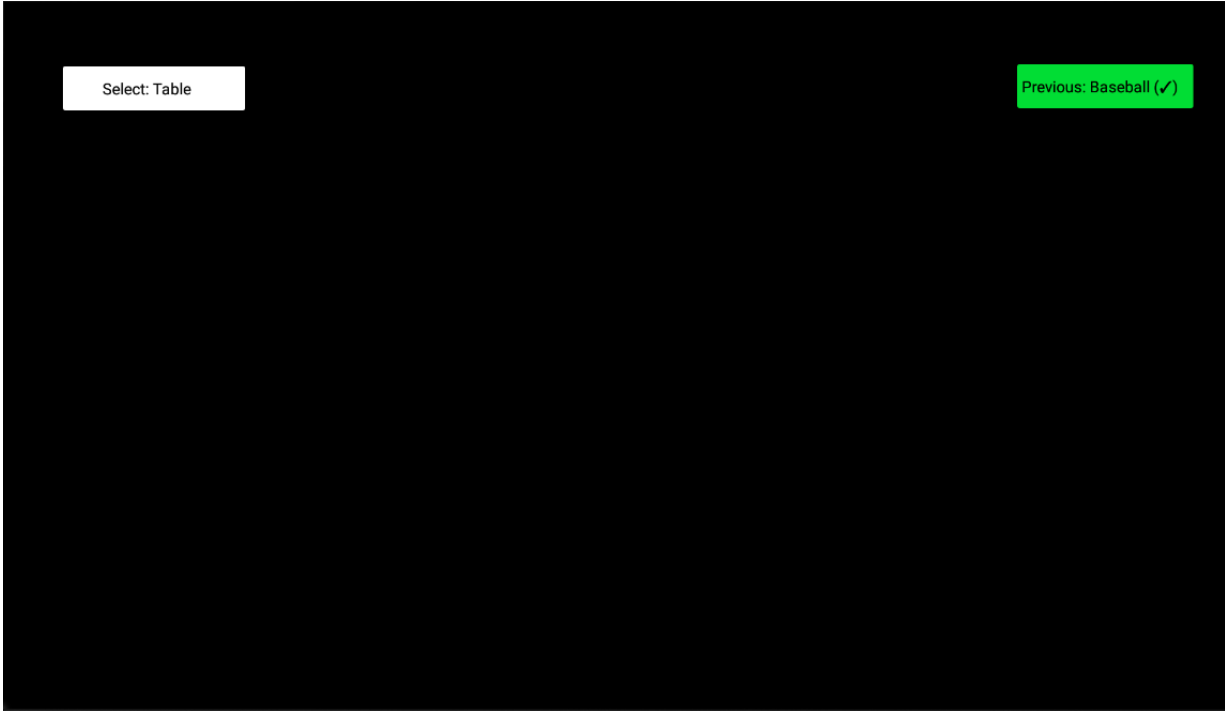


Figure 4.4: Experiment phase interface, always in MID-AIR on a monitor (external display). No visual guidance is provided.

were not significant ($T_{12} = -0.076, p = 0.94$ and $T_{12} = 0.085, p = 0.93$ respectively). Upon observation, time from prompt to gesture was slightly shorter and time to gesture slightly longer for the touch training condition, but neither measure exhibits statistical significance.

We also analyzed gesture time per block. One factor that does seem to differentiate touch-based training of mid-air gestures is that the first mid-air gesture performed in the experimental block is significantly slower (longer time) than subsequent blocks. Figure 4.8 demonstrates this effect, a result of the initial cost of switching modalities [50]. However, participants quickly converged on equivalent performance, and, by the second and subsequent blocks, performance was indistinguishable.

4.3.3 NASA TLX

Recall after each phase in the user study, training and experiment, we asked participants to complete the six category NASA Task Load Index [72]. We performed a multivariate

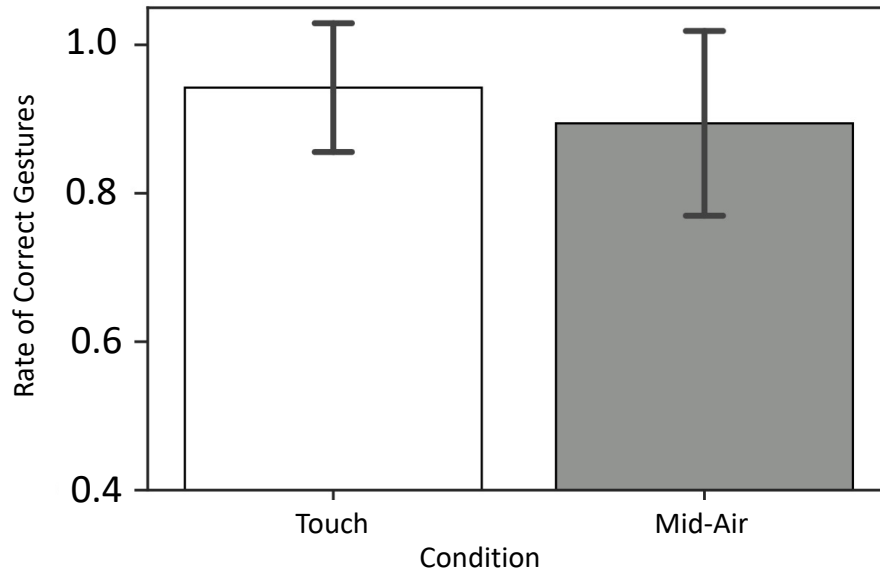


Figure 4.5: Rate of correct selections across conditions (error bars indicate SD).

analysis (MANOVA) on reported TLX scores and found no significance across each study phase (*training* or *experiment*) and each condition (TOUCH or MID-AIR). These results are depicted in Figure 4.11. Between subjects tests indicated significance for reported physical demand scores ($F_{3,24} = 3.869, p < 0.05$). Post-hoc analysis using Tukey’s HSD indicated a difference in physical demand between the *training* and *experimental* phase for participants who trained using touch and between *training* via touch and the *experiment* in mid-air (after mid-air training).

4.4 Discussion

With quantitative analyses as presented in results, it is often desirable to examine hypotheses, but, in this work, our goal was slightly different. We expected touch-based training to be worse than mid-air training for mid-air gestural input, and our goal was to quantitatively assess a qualitative question: *How much worse is touch-based training for learning mid-air input?*

Surprisingly, our results argue that touch training is as effective as mid-air training to learn mid-air motion gesture input. There is no statistically significant difference between

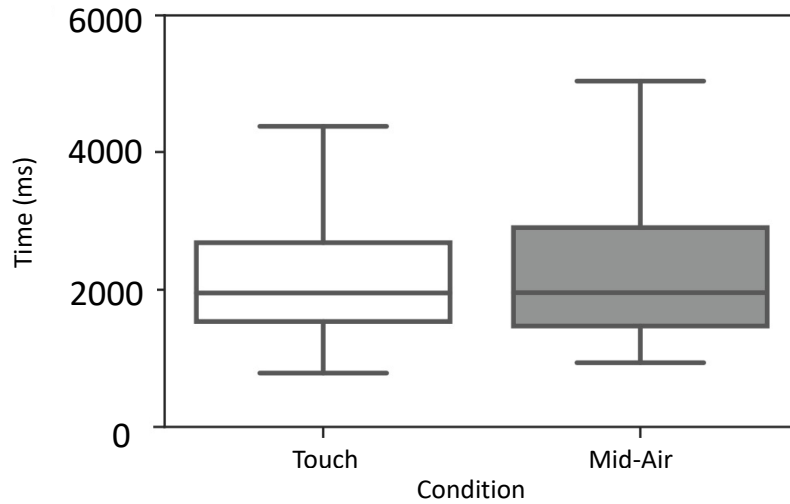


Figure 4.6: Time from prompt appearing to selection across conditions.

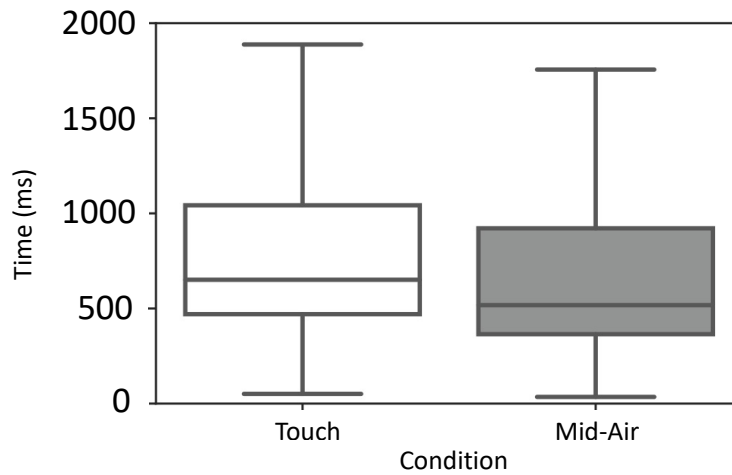


Figure 4.7: Time from beginning a gesture to selection across conditions.

error rate and time, and, qualitatively, overall values of performance appear similar in the experimental phase: touch training results in slightly higher accuracy scores and both training mechanisms exhibit similar interaction time. Stated more succinctly, we found that participants were able to effectively learn marking menu gestures through an alternative mechanism (touch).

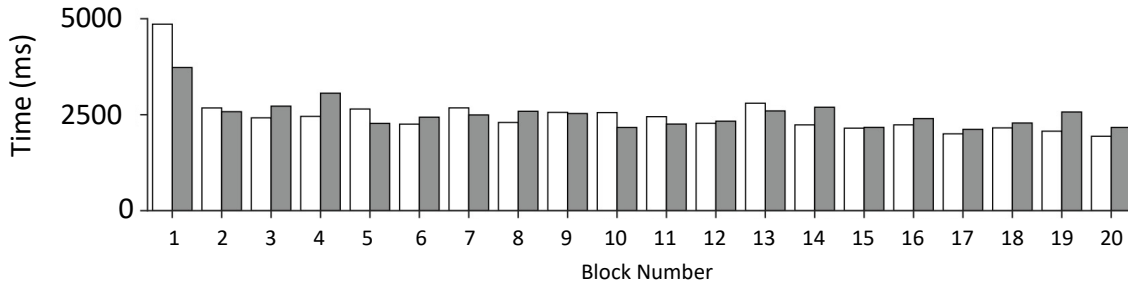


Figure 4.8: Time from prompt appearing to selection across blocks by condition.

Our control condition, of teaching mid-air gestures from scaffolding on a distant display, echoes findings of previous literature – users are able to learn bodily motions, gestures, in particular – through a representation of the required movement path on an external display [10, 110, 9, 1, 177, 184]. Our result – of cross-modal learning – echoes findings of Kamal et al., who showed that an alternative mechanism of revealing a gesture (on-device video plus recognizer feedback) was as effective as instruction via an external display representation of the gesture [110]. We note the difficulty in comparing our findings with the literature as we are unaware of prior studies that observe learning free-space (mid-air) gestures via touch.

In our view, mid-air gestural input as we have formulated the problem has much in common with keyboard accelerators. Users typically use one, sub-optimal modality (e.g. a menu or toolbar) to perform commands. However, use of these sub-optimal commands provides them with an awareness of an expert mode - a keyboard accelerator - that can sufficiently enhance performance. Cockburn et al. [50], in studying this learning of expert performance, note that moving from a novice to expert mode may have a performance cost, and we do see a brief performance cost in our experimental phase during the first block of eight gestures as participants habituated themselves to the new input modality. However, after that one habituation block, participant performance converged to being qualitatively indistinguishable for the remaining blocks in the experimental phase.

There are some potential slight differences in NASA TLX ratings that might merit further investigation. In Figure 4.11, it is observed that training in mid-air resulted in the lowest overall average TLX score. Participants who trained in the touch condition provided some insight of why this may be, for instance, “the occlusion of the menu by my finger” [P16, trained via touch]. Since they were aware that they would have to later perform gestures in mid-air, one participant noted “learning the gestures on the phone in part 1 was somewhat stressful” [P10, trained via touch].

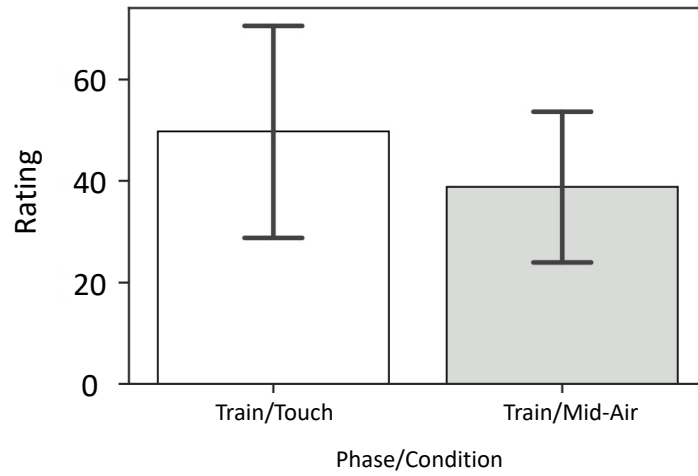


Figure 4.9: Training phase.

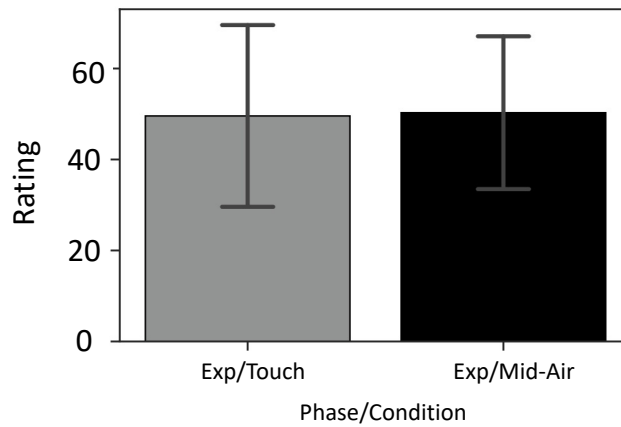


Figure 4.10: Experimental phase.

Figure 4.11: Overall NASA TLX scores across phases and conditions (error bars indicate SD). Training phase conditions are performing via touch or mid-air. Experimental phase is always performed in mid-air, conditions are mode which training phase was completed.

4.4.1 Future Work

While our results demonstrate that we can leverage touch-based training to teach in-air marking menus, the marks represented in marking menus and evaluated in our study are comprised only of two straight-line segments where the two segments vary only in direction. One area of future work is to assess cross-modal gesture learning on more complex gesture sets, including ideographic, alphanumeric and multi-stroke gesture sets [201] to see if our findings hold. Furthermore, alongside mid-air gestures that can be rendered on a 2D plane (i.e. planar gestures, as in the gestures by Siddhpuria et al. [201]), one could imagine teaching 3-dimensional rotation gestures with a hand-held device, where the user could rotate their finger or add an additional finger to indicate a rotation and we could explore mappings to mid-air such that non-planar mid-air gestures could be trained via touch. Another area extending from the current work is the incorporation of haptic feedback with our cross-modal learning technique [177, 197]. Exploration of this space could include haptic feedback while learning on a touch surface in an effort to further ingrain gestures into a users memory, providing haptic effects while interacting in mid-air, or using both of these in conjunction to create a more congruent mapping between modalities.

4.5 Limitations

In our study, the participant sample size is limited. To counter this, we performed a sample size power estimate to ensure that our sample generates sufficient statistical power to identify discrepancies at the level we wish. Alongside this, we also note that, qualitatively, sample size is a highly questionable critique given that the error rate for touch training is actually lower than the error rate for mid-air training and given that temporal profiles are so similar – again touch training resulting in lower prompt to selection timing. Thus, in conjunction with our sample size power estimate, it would be highly unlikely to identify significant differences with a higher sample size, meaning that our likelihood of type 2 error is quite low.

Alongside this obvious limitation, it is true that we do not explore delimiter costs in mid-air input. However, we note that, first, delimiters are a cost for mid-air input whether training is done mid-air or by touch, so this is not germane to our research question. Second, if we were to evaluate delimiter cost for mid-air versus remaining in touch modality for input, we would also need to measure, in some fashion, the cost of screen input in touch being restricted during touch input (i.e. we need to provide touch on-screen so other apps must be closed) versus the use of, for example, a delimiter motion gesture to switch the

phone into gesture input mode [187] before accessing a command via a mid-air marking menu.

4.6 Conclusion

One challenge with mid-air gesture input is that gestures are not self-revealing, so to teach users gestures, it is common to use an external display and/or tracking to provide guidance and feedback to the user. Revisiting RQ 2, that is *under what circumstances do users need to transition from a novice mode to a secondary mode?*, we shed light on the challenge of providing novice guidance to non-self revealing interactions – such as mid-air input. In this work, as self-revelation is unavailable, we examine whether or not we can leverage another modality, touch, to teach users a mid-air gesture set. Leveraging the paradigm of free-space marking menus, and a smartphone, as a motion-gesture-style input device, we explore how well learning in touch transfers to mastery in mid-air input when compared to learning and performing in mid-air. Our results argue that transferring spatial knowledge of marking menus between touch and mid-air exhibits similar performance as obtaining such knowledge in mid-air, with only minimal, short-term performance cost.

Chapter 5

Typing On The Thigh for HMDs

While we can conceptualize the transfer of simple unistroke gestures between modalities, the question then arises of whether complex unistroke gestures can transfer to new modalities in the same way? Due to the intricate nature of complex gestures, as opposed to providing an in-study training task, we wished to look at whether *we can leverage existing interface expertise to assist in transition to a secondary mode in a new modality?* (RQ 3).

Utilizing the complexity of word-gesture input, in this chapter, we introduce STAT, a mobile touch typing technique for HMD that leverages smartphone screen located at the thigh. Through a controlled laboratory study, we compare users' ability to transfer pre-existing QWERTY expertise to perform tap text input and word gesture text input via a new modality: STAT. We then explore whether users can then transfer expertise gained via rehearsal with STAT to perform the technique within an enclosed pocket. Lastly, we present design recommendations for the opportunistic use of transferring keyboard expertise to use a personal touchscreen device positioned at a user's thigh for HMD text entry.

5.1 Introduction

Over the past decade, there has been a surge in popularity of head-mounted displays (HMDs) for presenting an augmented or virtual reality to the wearer. Many HMDs, such as smartglasses, are designed to be ubiquitous displays for providing a personalized, always available, augmented view, without requiring external hardware. A challenge arising from these HMDs, which has subsequently become a roadblock in their widespread adoption (e.g. smartglasses), is the lack of an input mechanism for their control [153].

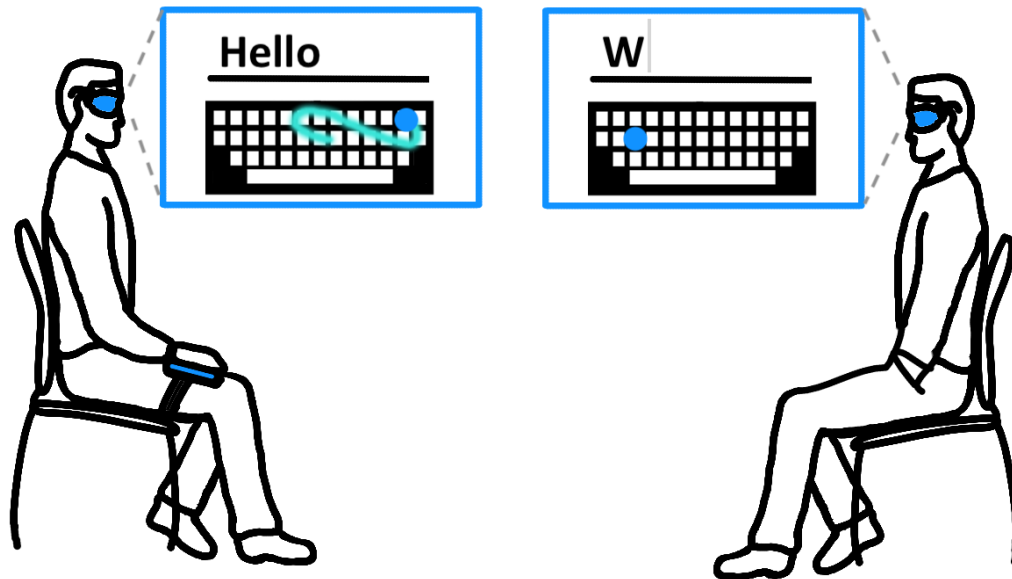


Figure 5.1: Sample interactions using STAT techniques – *STATSwype* on-thigh (left) and *STATTap* in-pocket (right).

One primary input mechanism we require for HMDs is some facility for text entry. There is an extensive body of research on techniques for text entry in wearables, including HMDs (e.g. [130, 145, 232, 247]). In general, text entry is a challenge in ubiquitous computing as input either requires a button or key associated with each character, or some form of gesture or chord to describe characters or words. This, in turn, may require specialized devices [145], additional sensors [232], or learning a new input mapping [247] to effectively input text. Gaze eliminates the need for specialized devices, but is perceived to be “complex, strenuous and slow” [7], and both speech and gaze suffer from issues of social acceptability, especially when compared with on-device interaction [181]. While it is possible to type on a virtually displayed keyboard [205], this requires tracking of finger position and, without a physical surface, it is challenging for users to localize keys—potentially reducing the speed and accuracy of such a technique. Thus, a large amount of research has been dedicated to optimizing text input when using a virtual display—resulting in increased performance via novel interaction techniques [4, 84, 205, 247, 246]. Many of these techniques, however, require specialized hardware or physical controllers that often encumber the user’s hands during interaction.

In recent work, Akkil et al. [7] note that, for users of smartglasses and other HMDs, mobile phones are considered to “complement” HMDs, particularly for functions where the HMDs are lacking, such as text entry. As users have become proficient in text entry [180] on mobile touchscreens, we propose integrating this existing proficiency with HMDs. Leveraging a state-of-the-art mobile keyboard (SwiftKey), we introduce a soft keyboard variant we dub STAT, Subtle Typing Around the Thigh, a low-cost, mobile, touch typing technique. STAT supports both tap and word gesture text entry, leveraging a smartphone screen at a user’s front thigh area. This chapter describes the implementation of STAT and a controlled within-subjects study of the technique. The results indicate that STAT can reach average speeds of 13.15 words-per-minute (WPM) for word-gesture input and 13.37 WPM for tap-based text entry after minimal training. The main contributions of this work are:

- (1) an innovative, low-cost solution to text entry for HMDs; and
- (2) a comprehensive laboratory study contrasting users’ ability to transfer pre-existing keyboard knowledge to word-gesture typing and to tap-based typing (in and out of an enclosed pocket)

Our results indicate users are capable of transferring their existing keyboard knowledge to a new, unfamiliar interaction modality and we argue for the feasibility of leveraging a personal smartphone placed on the users thigh to support text entry for HMDs.

5.2 Related Work

5.2.1 Around Thigh and In-Pocket Interaction

When a person’s hand or arm is at resting state, either seated or standing, it is most often placed on or around the thigh, as depicted in Figure 5.2. Thus, on-leg and in-pocket interaction is an active area of research, as an ideal location for subtle, unobtrusive, low-fatigue input [141, 201]. In-pocket techniques such as Tap [182] and Whack [104] leverage the on-device IMU to capture quick, gestural commands. Other in-pocket systems leverage touch-sensing fabric [94, 111] or augment the smartphone to capture touch events through fabric [189]. In contrast to these, alongside the Nintendo Ring Fit’s leg strap [163], other researchers have also looked at strap-on controllers to capture at-thigh interactions [141].

On-thigh input is particularly opportunistic because, as Thomas et al. note [208], the front-of-thigh seems an optimal location for high precision input – even in contrast to other on-body locations such as the wrist or forearm – without compromising user comfort. However, care must be taken when supporting interactions near the waist, as these interactions can have low social acceptability ratings [178].

With respect to HMDs, two proposed around-thigh techniques specifically applied to touch interaction are Belt [63] and PocketThumb [64]. In Belt, Dobbelstein et al. [63] added metal divets to a leather belt for touch sensing capabilities, allowing a large horizontal surface for input near a user’s waist and found that interaction near the front pocket was preferred for touch input in general (particularly for longer interactions of 10s or more), and interactions near the belt buckle (in the middle) were less desired. Leveraging this idea, PocketThumb employs a dual-sided touch surface, on the inside of a pocket, for the user’s thumb and index finger, where the thumb is used as a cursor and the index finger to tap indicating selection (i.e. a pinch gesture). In a target selection task they determined the dual-sided interaction was more efficient than a single-sided touch interaction [64]. One challenge with the systems built for around-thigh and in-pocket interactions is that, while effective, they all require additional hardware for facilitating touch input around the user’s thigh. In contrast, a system such as the Nintendo Ring Fit’s leg strap makes use of a pre-existing controller that the user already owns (as part of the Nintendo Switch system), and the controller is strapped to the user’s thigh [163].

Text Entry on Constrained Touch Interfaces for HMDs

We define a *constrained touch interface* as a device with limited space for providing input. Previous studies have worked on improving touch typing interactions on constrained touch interfaces such as devices with ultra-small interfaces or small interaction space. Within this space, Ahn et al. [4] explored various techniques that leverage a smartwatch’s touch screen. TipText [243] uses small finger-tip gestures to capture text input (but requires augmentation of the fingertips). Researchers have also used the surface of an HMD for text input, e.g. the arm of smartglasses [80, 135, 247]. Typically, these small screen text input techniques reach input speeds of between 8 and 11 WPM. Alongside constraints of screen size, physical restriction can further constrain the use of touch interfaces—e.g. when a user’s hand is in their pocket. Zhong et al. [256] presented a subtle pressure-based text input technique that leveraged an off-the-shelf iPhone with (now discontinued) pressure sensing. Users entered text by varying pressure via their finger on a smart phone touch screen. While they note in-pocket interaction as a use-case, they do not explicitly evaluate in-pocket performance.

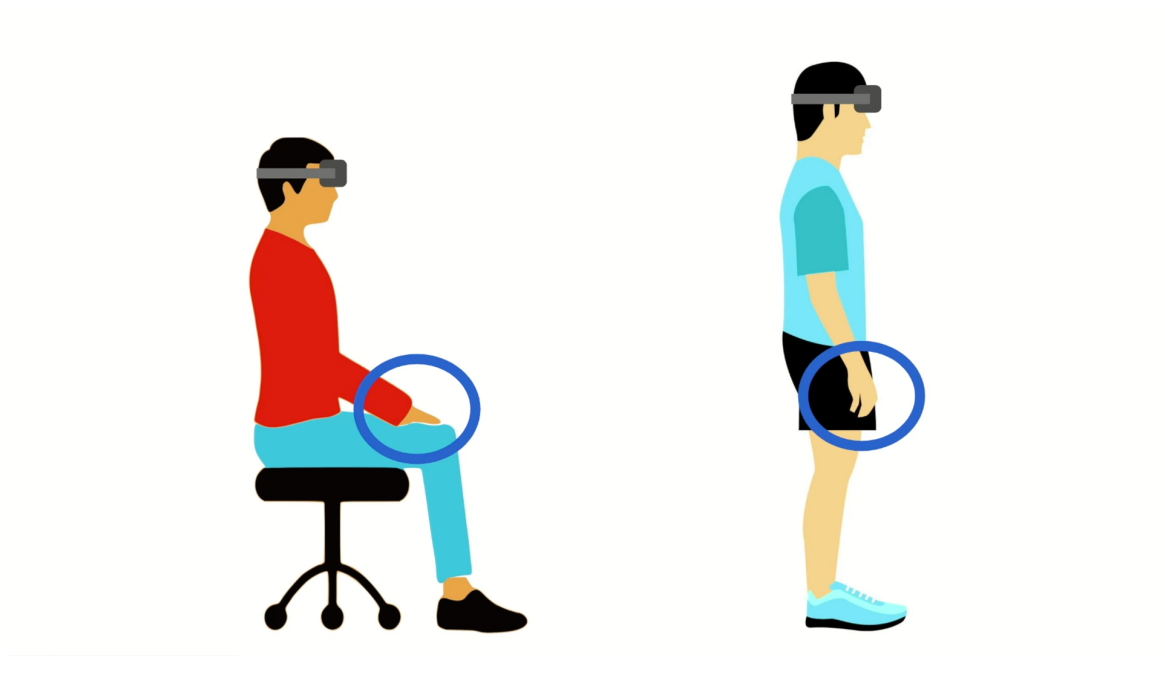


Figure 5.2: Position of the arm or hand at resting state.

5.2.2 On Thigh Gestural Text Entry for HMDs

To synthesize, we revisiting the question in Section 2.4 of: *how far can this knowledge transfer be pushed to new contexts?*. Taking into consideration the requirement for text entry in head-mounted displays, the natural integration mobile devices can provide for wearable devices, and the seamless input space on thigh interaction provides, we ask: *can the spatial knowledge of a keyboard layout be leveraged to provide word-gesture text input to a thigh mounted device?*

5.3 STAT Design

STAT is designed to be a subtle technique to facilitate text entry while wearing an HMD. The subtlety of the technique lies in its positioning [141]: the controller is mounted to the front of the user’s thigh, the typical location of a user’s hand/arm at rest, when seated or standing. The technique is implemented using two components: the controller and the display.

To implement STAT, a Huawei Nexus 6P running Android 8.0.0 was used for the display, with screen dimensions 2560×440 (landscape), encased in a MoGo Cinema2Go headset [142]. The headset allows for near-display viewing, i.e. the phone display remains the same but appears as a large screen close to the user’s eyes in the HMD. The smartphone for text entry was an LG Nexus 5 running Android 6.0.1, with screen dimensions 1080×1920 , mounted to the user’s thigh either using a Velcro strap (portrait), as shown in Figure 5.3a or inside a simulated pocket attached to the user’s clothing, Figure 5.3b. A “simulated” pocket was chosen both for internal validity (to control pocket size for consistent measurement) and to ensure inclusivity of participants (regardless of wardrobe preferences/size). Both the HMD and the smartphone for text entry were connected to a Macbook Pro (OSX 10.11.6) and information was wired through USB between the two devices and sent via tcp/adb forwarding.

5.3.1 STAT Controller Design and Input

During the design phase, we evaluated a series of potential interactive designs via pilot studies. We explored screen layout mechanisms and input paradigms, including whether a Word-Gesture-Keyboard (WGK) or a Tap-Based-Keyboard should be used.

Two-State versus Three-State Input and Screen Orientation

One challenge with mobile phone touch-screen based input is that mobile phones are a two-state input device (versus a three state model [38]) due to the absence of a tracking state. Furthermore, because of the presence of an HMD, the user’s eyes are focused on the HMD. As a result, the user is typing eyes-free relative to the smartphone screen.

In Zhu et al.’s I’sFree [257] developed a shifting QWERTY layout to support eyes-free typing, where they synthesize a deformation model for WGKs that they then apply to infer word gestures. However, one challenge we found in applying an eyes-free technique was the inverted mapping which alters spatial perception. As well, while Zhu et al.’s technique can effectively be used for WGKs, it is unclear how accurate the technique would be for character-by-character entry, and, given that character-by-character entry appears to be the most common smartphone-based text entry paradigm [168], we felt it important to support both character-by-character and WGK-based text entry. Character-by-character text entry was plausible with Lu et al.’s eyes-free technique [143], however, without spatial perception of where your fingers are tapping in relation to the edges of the phone, as possible while holding a phone, this becomes a challenge. While we considered

performing additional analysis to explore inverted eyes-free tap typing, in pilot testing, another option presented itself—the use of multi-touch input to support a 3-state input model. We highlight the ability to support text entry without spatial perception – via 3-state input – as an integral difference in our work in comparison to Blindtype [143] and I’sFree [257], which leveraged 2-state input.

To capture 3-state input, STAT takes advantage of the multi-touch nature of the smartphone screen by dividing the screen into a touchpad and a button. Given the position of the smartphone on the thigh, the “top” section (dimensions 1080×608) of the smartphone is used as an absolutely mapped trackpad for a cursor on the HMD. This “top” section is positioned further from the waist (so nearer the user’s knee). The bottom section (dimensions 1080×1312) of the smartphone is a button to indicate an action (either being a gesture or tap on a character, Figure 5.3c) and is positioned nearer the user’s waist. To use the smartphone for text input, the user positions their hand on top of the screen; the index, middle or ring finger can be used on the trackpad as a cursor, and the thumb is used to press the button for an action.

Screen separation was chosen for several reasons. First, when mapping the text entry smartphone to HMD, the HMD smartphone was landscape oriented and thigh positioned smartphone portrait; this complicated mapping for our participants, leading us to divide the screen so that mapping was more natural. Second, our pilot studies highlighted an advantage in dedicating the lower section to state-switching. Consider, for example, if the user navigates to the right edge of the thigh-mounted phone with their index finger and attempts to tap with their middle finger they will miss the screen, whereas the thumb will always be placed on the lower portion of the screen (closer to the user’s belt) regardless of index finger position. Thus screen separation and mapping ensures the thumb is always ideally positioned to manipulate input state. Finally, the separation of sections allows for navigation with only minimal movements of the navigational (index) finger. In a constrained pant-pocket, the deeper the hand is, the more restricted movement becomes, and in our technique, the navigational finger (placed deeper in the pocket) does not require lifting/tapping at all, while the thumb (placed nearest to the pocket entrance) is at the easiest location for lifting/tapping (to switch states).

Gesture versus Tap-Based Text Entry

Given the above 3-state model, Gesture versus Tap text entry can be supported. Actions differ subtly based on two different implementations of the STAT technique: *STATTap* or *STATSwype*.

- *STATTap*: This implementation utilizes tapping on each individual key to type. Finger position on the track pad in the top section of the controller is depicted as a cursor on the HMD. The user moves the cursor using their finger to the letter they wish to type, and presses the button in the bottom section of the controller using their thumb to select.
- *STATSwype*: This implementation utilizes word gesture typing. Again, finger position on the track pad is depicted as the cursor, and the user presses the button with their thumb to start a word gesture. To end the word gesture, the user can lift their finger that is on the track pad or press the button again in the bottom section with their thumb. In the case a word gesture is not recognized, to type a single letter the user can press the button with their thumb twice in a row (double-tap), or press the button with their thumb once and release the cursor finger (lift-cursor-finger).

Alongside character-by-character and WGK typing, both auto-correction and word-completion are also commonly used techniques to assist users in fast, accurate typing [168]. In order to provide these features, many researchers make use of state-of-the-art keyboards that incorporate language models such as the Google or SwiftKey keyboards for gesture recognition [155]. In our work, we make use of the SwiftKey keyboard [155]. Events were injected using Android NDK [61] to the SwiftKey keyboard on the HMD. The smartphone used for text entry was in incognito mode to prevent confounds introduced by the learning of user input. Participants were permitted to use predictive text and auto-complete during text entry. Corrections were completed by tapping backspace, and participants could only backspace a single character at a time.

Finger Movement Mapping

One primary design decision that must be made in STAT is how best to map finger motion on the display to cursor motion in the HMD. Consider Figure 5.3b, where the participant is standing with the controller smartphone in the simulated pocket. The keyboard could either be mapped such that gestures toward the waist, gestures that are “up” relative to the ground, map to “up” in the HMD display (upright mapping), or gestures that move toward the waist, away from the ground, could be mapped “down” from the perspective of the user in the HMD (inverted mapping).

We performed a series of pilot studies, and in all cases, participants preferred inverted mapping, where gestures toward the waist map as down. Because this perspective was the most natural for end users, we adopted it for our STAT evaluation/implementation.

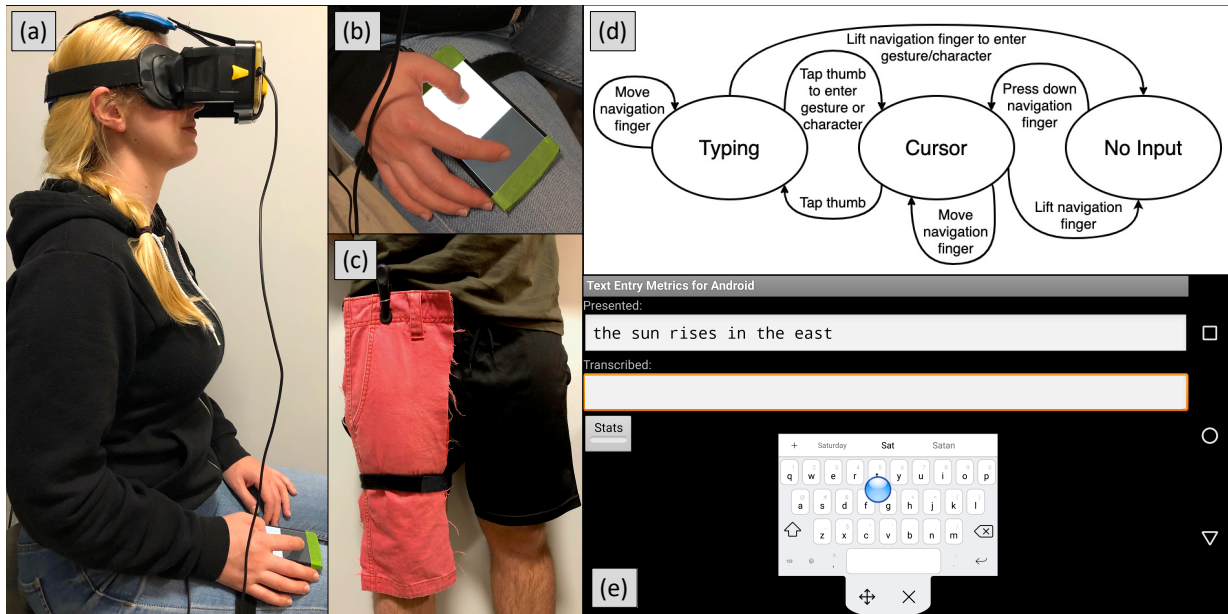


Figure 5.3: (a) A participant using STAT for text entry on the HMD; (b) A closeup of the STAT controller on a user’s thigh, with the index finger being used on the top section trackpad as a cursor, and the thumb on the bottom section to press the button for an action; (c) The simulated pocket used for in-pocket interaction; (d) State diagram for user input using STAT; (e) The experimental interface used for performing text entry.

5.4 Experimental Protocol

In this section, we describe an evaluation of STAT. The purpose of the user evaluation was two-fold: first, to assess the validity of typing on the thigh, where the user’s hand naturally rests when seated, and where the user’s front pocket on the side of their dominant hand would be; and second, to compare implementations for interacting in this location, using either gestural text entry (*STATSwype*), or tapping text entry (*STATTap*).

Aside from these two primary questions, we wished to investigate whether or not users were capable of using the aforementioned techniques, once they had learned to type on their thigh, while carrying their phone in a more constrained environment (in a pocket or bag, where these devices are typically carried). To provide a preliminary investigation of the in-pocket interaction use-case (i.e. as suggested by Zhong et al. [256]), a pocket (taken from a pair of trousers so as to keep pocket size and tightness the same for every

participant) was clipped to the participant’s waistband and held in place with an adjustable elastic strap (Figure 5.3b). The study followed a within-subjects design, counter-balancing ordering of conditions. The apparatus was as-described in the previous section: i.e. a Huawei Nexus 6P in a MoGo Cinema2Go headset [142] for the display and an LG Nexus 5 at the user’s thigh (either externally or encased in the strap-on pocket) for text entry.

5.4.1 Participants

12 participants were recruited for the study and paid \$15 for the session. Average age was 24.92 (SD=2.27). Two participants identified as women and the remaining ten identified as men. All participants were post-secondary students from a technically-focused university. Each participant signed an informed consent form before starting the experiment. Participants were screened for motion sickness and whether or not glasses were required for normal vision, to reduce possible discomfort while wearing the HMD.

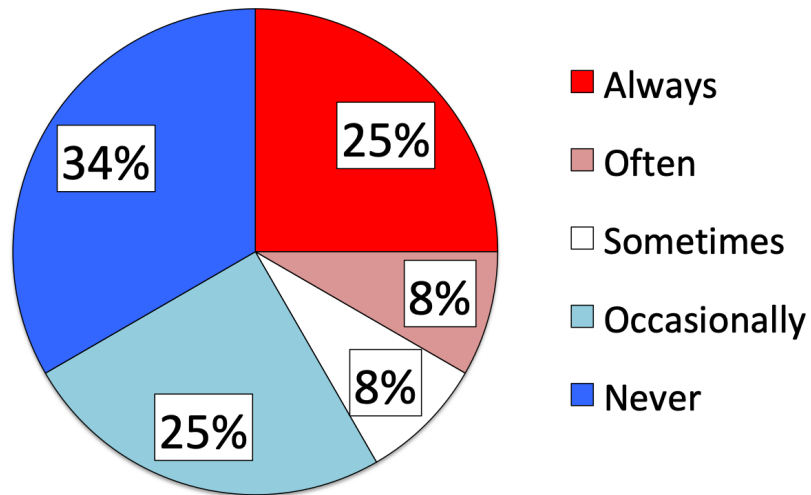


Figure 5.4: Distribution of participants’ self-reported usage of word-gesture typing.

5.4.2 Procedure

Prior to the study, participants were asked to self-report expertise with the QWERTY keyboard and word gesture typing (depicted in Figure 5.4), as well as general demographics

(e.g. gender, age, occupation, and handedness). Thereafter, participants were fitted with an adjustable elastic strap around their thigh on the leg on the side of their dominant hand, with a Velcro section facing the front. They sat on a chair for the duration of the study—between 1 and 1.5 hours. In order to get a baseline of participants’ mobile typing speed, before being given instruction on the upcoming typing technique to be used, participants were asked to type five phrases with either tap or word gesture typing, without the HMD and holding the device in their preferred manner, i.e. not attached to their thigh. Following this, the controller phone was mounted to the Velcro section of the strap on the participant’s thigh and the display phone fastened to the headset which was then placed on the participant’s head. Depending on the ordering of the conditions, participants were instructed to perform a series of text entry tasks (described in more detail in the next section) using either the *STATSwype* or *STATTap* implementation. To assess for potential learning of word-gesture text entry, upon completion of the *STATSwype* technique, participants were asked to do a second block of gestural text entry on the mobile device, holding it in their preferred manner without the HMD.

In the *STATSwype* condition, participants were told that they could ‘double-tap’ or ‘lift-cursor-finger’ on each individual character, if they were unable to type the correct word or phrase after the first attempt with gesture typing. This was made possible because, first, it is common for users to employ both tapping on individual keys and gesture writing in the same phrase (as mentioned in [84]) and second, to ensure that participants attempted gestural text entry at least once before abandoning the input method in preference for tap.

Task

We assessed the STAT technique using Castellucci and Mackenzie’s TEMA application [43], used for assessing text entry on android devices. The TEMA application presented random text phrases from the Mackenzie corpus [147] and participants were asked to transcribe these phrases. Each participant completed 5 trials (1 trial = 1 phrase) per block, with a total of 4 blocks for each condition. Upon completing each phrase, participants selected the enter button on the keyboard to continue. At the end of each block they were given the opportunity to take a break. Participants were told to focus on accuracy and speed while completing the task. If participants had an uncorrected error rate (UER) of 15% after a trial, they were required to re-do the trial. After completing the 4 blocks of one condition, participants were asked to complete the NASA Task Load Index (NASA-TLX) to measure perceived workload.

After the TLX, participants performed one block (transcribing 5 phrases) with the controller in the simulated pocket (Figure 5.3b). Then, participants were asked to comment

on the experience interacting in-pocket in comparison to out-of-pocket (mounted on the thigh). Once this was complete, participants repeated these steps in the second condition (either *STATTap* or *STATSwype*). At the end of the session, participants were asked which text entry method, tapping or swyping, they preferred in-pocket, and which out-of-pocket. Finally, participants were debriefed, asked for additional commentary and paid for their participation.

5.4.3 Measures

At a high level, our study reports on the following: *Performance* (measured with text entry and error rates), *Perceived Workload* (measured using the NASA-TLX), and *Subjective Preference* (measured through survey questions at the end of each condition and session). Text entry rate was measured using WPM, where a word is five characters (including spaces). Text entry duration for each trial began when the participant’s finger tapped the bottom section of the controller, and ended when the participant ends the final word (either by releasing their finger on the upper section of the controller or by tapping the bottom section of the controller). The error rates calculated were corrected error rate (CER), which considers rectified errors made during transcription, and uncorrected error rate (UER), i.e. errors left uncorrected.

The study employed a within-subjects design with the following factors and levels: condition (gestural text entry out-of-pocket, gestural text entry in-pocket, tap text entry out-of-pocket, and tap text entry in-pocket) and block (1-4).

In total, we collected:

$$\begin{aligned} &12 \text{ participants} \times ((5 \text{ phrases} \times 4 \text{ blocks} \times 2 \text{ conditions}) \\ &+ (5 \text{ phrases} \times 2 \text{ conditions})) \\ &= 600 \text{ phrase data points} \end{aligned}$$

5.5 Results

5.5.1 Gesture vs. Tapping - Out of Pocket

A repeated measures Analysis of Variance (RM-ANOVA) was conducted for text entry rate (WPM), uncorrected error rate (UER) and corrected error rate (CER) with two factors: condition (levels: gesture and tap), and block (1-4).

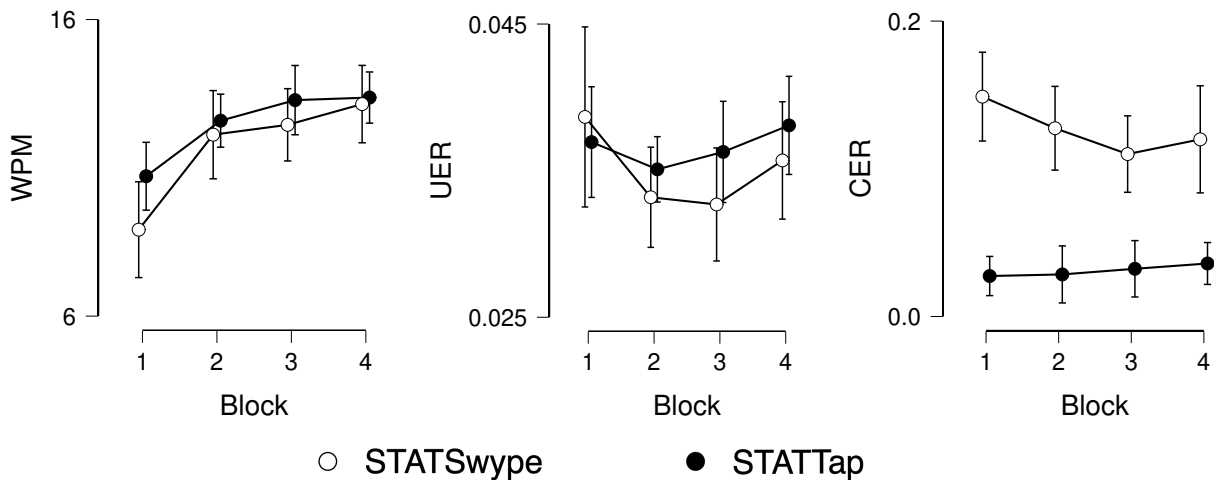


Figure 5.5: Each dependent measure across blocks and conditions (out of pocket). Error bars indicate a 95% confidence interval.

Text Entry Rate (WPM)

Figure 5.5 (a) shows the text entry rate in WPM across the 4 blocks, for both *STATTap* and *STATSwype*. There is no significant effect of condition on text entry rate, nor an interaction effect of condition and block. However, block does have a significant effect ($F_{3,33} = 25.662$, $p < .001$). Bonferroni post hoc tests indicate significant differences between block 1 and 2 (mean difference = -2.537 , $p < .001$), block 1 and 3 (mean difference = 3.051 , $p < .001$), and block 1 and 4 (mean difference = -3.443 , $p < .001$). Growth in WPM appears to slow from block 2 onward for both conditions—for *STATTap* performance seems to plateau, but continues slight growth in *STATSwype* from block 2 to 4. Mean WPM scores are depicted in Table 5.1.

Uncorrected Error Rate (UER)

Figure 5.5 (b) depicts the UERs over the 4 blocks. There is no significant effect of condition nor block; although we note a near significant effect on block ($F_{3,33} = 2.423$, $p < 0.1$). It may have been the case that some uncorrected errors were due to certain words not being present in the Swiftkey dictionary (as was found in prior work that used the same task setup [85]).

Condition	Block	WPM	UER	CER
SS out	1	8.92	0.04	0.15
SS out	2	12.12	0.03	0.13
SS out	3	12.45	0.03	0.11
SS out	4	13.15	0.04	0.12
SS out (mean)	-	11.66	0.035	0.1275
ST out	1	10.72	0.04	0.03
ST out	2	12.59	0.04	0.03
ST out	3	13.28	0.04	0.03
ST out	4	13.37	0.04	0.04
ST out (mean)	-	12.49	0.04	0.0325
ST in	1	12.25	0.04	0.03
SS in	1	12.31	0.04	0.11
HH T	1	36.30	0.04	0.02
HH S	1	22.68	0.04	0.09
HH S	2	26.14	0.04	0.08

Table 5.1: Summary of means by block and condition. (SS = *STATSwype*; ST = *STATTap*; in = in-pocket; out = out-of-pocket); HH = hand-held (T = tap, S = gesture). Note: Block 2 for regular on phone gestures is reported to control for those who were word-gesture typing novices.

Corrected Error Rate (CER)

Figure 5.5 (c) presents the CERs, showing a clear contrast between the two conditions. In fact, condition has a significant effect on CER ($F_{1,11} = 64.129$, $p < .001$), with a mean difference = 0.096. There is no effect of block or condition*block. Mean CERs are depicted in Table 5.1.

NASA Task Load Index

We found no significant effects across conditions via the NASA TLX. Results are summarized in Figure 5.6.

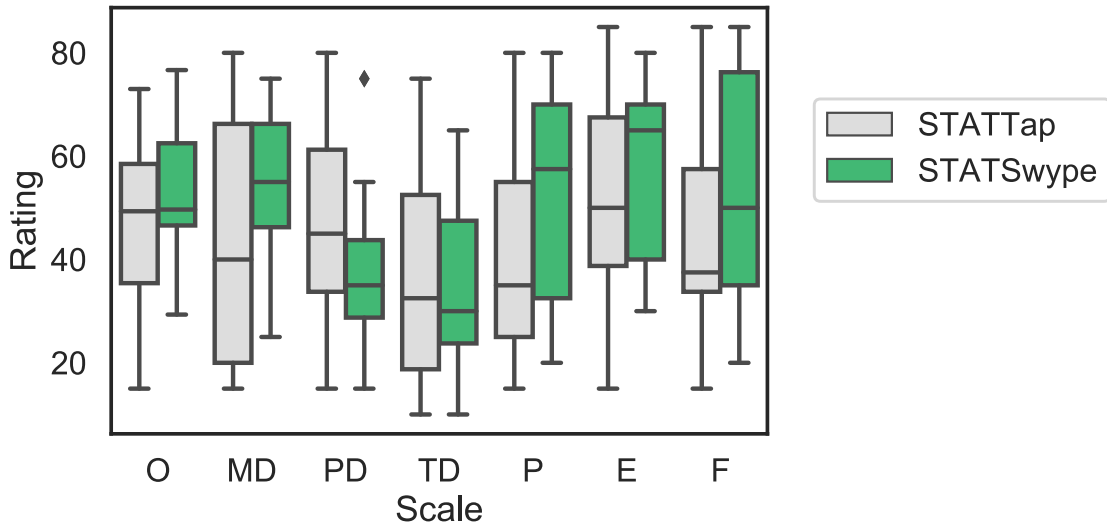


Figure 5.6: Categorical NASA TLX scores across conditions out of pocket. (O = Overall, PD = Physical Demand, TD = Temporal Demand, P = Performance, E = Effort, F = Frustration)

5.5.2 In-Pocket vs. Out-of-Pocket

A one-way RM-ANOVA was conducted for text entry rate, UER, and CER, with one factor: condition, with four levels (Block 4 of *STATSwype* out-of-pocket, Block 4 of *STATTap* out-of-pocket, *STATSwype* in-pocket, and *STATTap* in-pocket).

There was no significant effect of condition on text entry rate or uncorrected error rate (UER). Mauchly’s test of sphericity indicated the assumption of sphericity was violated for CER ($p < .05$). Using Greenhouse-Geisser correction, there was a significant effect of condition on CER ($F_{1.681,18.493} = 18.408, p < .001$). Bonferroni post-hoc tests (summarized in Table 5.2) indicate a significant difference between *STATSwype* in-pocket and *STATTap* in-pocket (mean difference = 0.080, $p < .001$); *STATSwype* in-pocket and *STATTap* out-of-pocket (mean difference = 0.079, $p < .001$); *STATSwype* out-of-pocket and *STATTap* in-pocket (mean difference = 0.085, $p < 0.01$); *STATSwype* out-of-pocket and *STATTap* out-of-pocket (mean difference = 0.084, $p < 0.005$). No significant difference was found between *STATSwype* in-pocket and *STATSwype* out-of-pocket; and between *STATTap* in-pocket and *STATTap* out-of-pocket. This indicates that once users had learned to type on their thigh, they were able to transfer this knowledge to a more constrained environment, with little loss in accuracy; further depicted in Table 5.1.

A correlation matrix was conducted for text entry rate (WPM) across the following

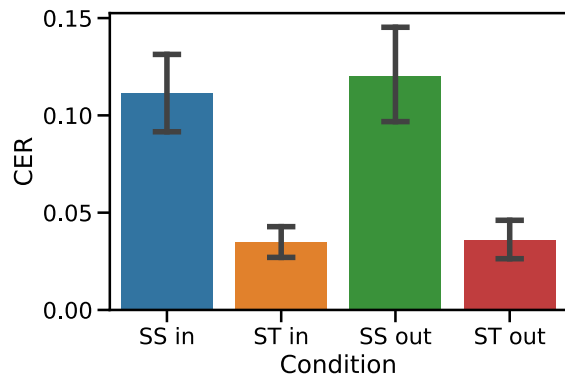
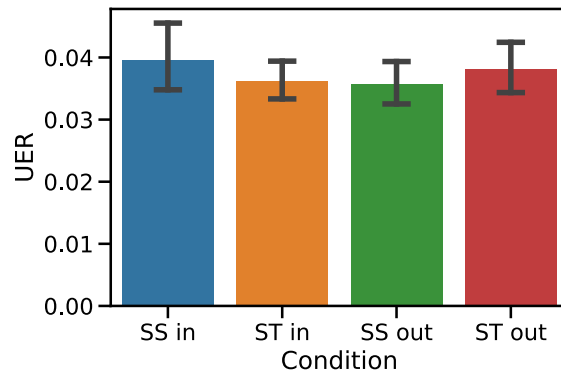
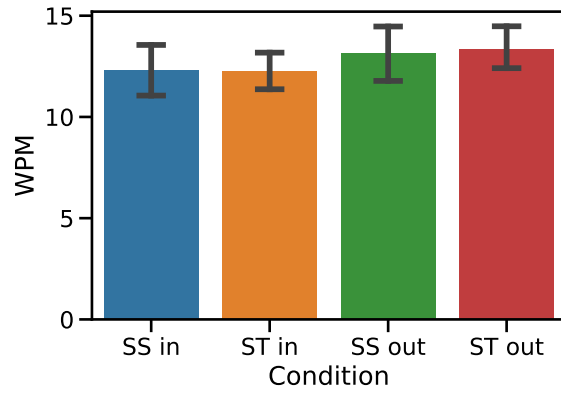


Figure 5.7: Each dependent measure for block 4 of out-of-pocket (SS out, ST out) and in-pocket conditions (SS in, ST in). Error bars indicate a 95% confidence interval.

	SS in	SS out	ST in	ST out
SS in	1	-0.005	*** 0.080	*** 0.079
SS out		1	** 0.085	** 0.084
ST in			1	$-8.69e^{-4}$
ST out				1

Table 5.2: Results of Bonferroni Post-hoc comparisons of CER for in-pocket vs. out-of-pocket. Mean differences (standard error) shown. ** indicates significance at the 0.01 level, and *** at 0.001. (Naming conventions follow Table 1).

conditions: Blocks 1 and 2 of regular handheld word-gesture text entry, Block 4 of out-of-pocket for *STATSwype* and *STATTap*, and both *STATSwype* and *STATTap* in-pocket. Significant results are summarized in Table 5.3.

Condition	Condition	Pearson's r	p
SS out	ST out	*0.641	< 0.05
SS out	HH S 2	**0.738	< 0.01
ST out	ST in	**0.743	< 0.01
SS in	ST in	*0.620	< 0.05
SS in	HH S 1	*0.629	< 0.05
SS in	HH S 2	*0.692	< 0.05
ST in	HH S 2	*0.599	< 0.05
HH S 2	HH T	*0.579	< 0.05

Table 5.3: Significant correlations between conditions for WPM. Naming conventions follow Tables 5.1 and 5.2. * indicates significance at the 0.05 level, ** at 0.01, and *** at 0.001.

5.5.3 Subjective Preferences

Preference for either STAT implementation did not seem to exhibit any strong trend. However, for out-of-pocket interaction, participants seemed to lean more toward word-gesture typing, with 7 participants preferring *STATSwype*, 3 preferring *STATTap* and 2 had no preference either way. For in-pocket, 5 participants preferred *STATTap* and 7 preferred *STATSwype*.

5.6 Discussion

This work demonstrates and evaluates the STAT technique for HMDs. Two implementations of STAT were tested: *STATTap* and *STATSwype*, and two levels of constraint: in and out of an enclosed pocket. At a high level, our results indicate: (1) STAT is comparable to prior work in text entry for HMDs, (2) *STATSwype* and *STATTap* exhibit similar performance and combining their usages would likely improve the technique, and (3) while out-of-pocket is usually preferred, in-pocket text entry is feasible under certain circumstances (unrestrictive pockets).

5.6.1 Comparison of Performance with Prior Work

Since our study measured 4 blocks of 5 phrase trials, we focus on the novice stages of top performing related techniques of each category (findings outlined in Table 5.4) at the closest reported WPM measure to 20 phrase trials. Surprisingly, our technique was able to substantially outperform prior cursor based techniques that had an average rate of 7.66 WPM vs. the current techniques’ average rates of 12.49 and 11.66 WPM.

The techniques that out-perform STAT are head pointing [246], controller pointing [205], and techniques optimizing hand-held smartphone typing [143, 244, 257]. Considering head and controller pointing, we note that, while speed is high, gaze is perceived to be strenuous [7], and may suffer from issues of social acceptability [181]; in contrast, while specialized handheld controllers may exhibit stronger overall performance, they are yet another device to locate, and, as Akkil et al. note [7], mobile phones are considered a natural complement to HMDs for functions such as text entry – particularly, since users can easily transfer their pre-existing soft-keyboard knowledge to the new modality.

This, then, leads to smartphone-based techniques for HMD text entry. While blind tapping [143] and eyes-free gesturing [244, 257] have higher reported input rates, it is important, first, to note that Lu et al.’s evaluation was performed on a touchpad oriented with the screen and Yang et al.’s [244] and Zhu et al.’s [257] techniques are restricted to gesture-typing. As well, all were evaluated such that user’s can monitor the position of the touchpad/smartphone via inter-hand proprioception and peripheral vision (both of which simplify spatial correspondence targeting [171]). In our pilot evaluations, the inverted and strapped nature of the device increased the complexity of the targeting problem. We highlight these phenomena as a trade off revealed when introducing the usage of a truly eyes-free technique, such as STAT, where users have more limited proprioception and peripheral vision of the touch screen input device (than prior approaches discussed [143, 244, 257]).

Finally, in contrast to these prior approaches [143, 244, 257], positional correction for eyes-free tap-typing and gestural typing both depend on dictionaries; *STATTap* can handle out-of-dictionary words due to its ability to select characters deterministically.

This is not to say that past work in blind-tapping and eyes-free gesture typing are flawed in any way; we believe that there is a second trade-off between hands-encumbered techniques such as those of Lu et al., Yang et al., and Zhu et al., and techniques that use more subtle forms of input via specialized controllers or restricted input spaces [4, 85, 243, 246, 247] as highlighted in Table 5.4. After minimal training, and considering Table 5.4, STAT’s performance recommends it as a useful addition to the suite of techniques for text-based input on HMDs. Though a useful comparison, we do acknowledge limitations due to the varying number of phrases in each study.

Technique	# of phrases	WPM
<i>STATTap</i>	20	12.49
<i>STATSwype</i>	20	11.66
Eyes-free WGK [257]	10	22.44
Indirect Touch WGK [244]	10	19.40
Eyes-free Tapping [143]	30	17-23
Head Pointing WGK [246]	8	17.04
Controller Pointing [205]	5	15.40
Index + thumb tapping [243]	40	11.90
Smartwatch [4]	60	10.24
Wrist Rotation (via ring, WGK) [85]	20	9.20
Arm of SmartGlasses [247]	18	8.84
Eyes-free Cursor [143]	15	7.66

Table 5.4: Text entry speed of related techniques (WPM) in comparable (novice) learning stages. We note a challenge in this direct comparison with differing # of phrases. (WGK = Word Gesture Keyboards).

5.6.2 Design Implications

We conclude this discussion section by addressing issues of gesture versus tap text entry and in- vs. out-of-pocket use.

Gesture vs. Tap Text Entry

Our results indicate comparable performance for our two technique variations in terms of speed, with *STATSwype* reaching 13.15 WPM on average in block 4 and *STATTap* reaching 13.37 WPM on average in block 4. Both conditions exhibited learning over time, as depicted in Figure 5.5 (a). *STATSwype* may have a slightly steeper learning curve than *STATTap*, and, while both converge on a similar speed, *STATTap* appears to be plateauing sooner than *STATSwype*, so results for *STATSwype* could be pessimistic. One participant noted this, stating: “I think that gesture could be better in both scenarios if I had more time to practice more and become more comfortable with this method of typing” [P9].

A notable difference between techniques is the CER—while *STATSwype* exhibits comparable speed to *STATTap*, a significantly higher amount of corrections, thus increased usage of the backspace key, were required to input the same phrase. This highlights an important trade-off in comparing the techniques: small deviations, gesture collision between similar words, and out-of-dictionary words in gestural text entry will be more detrimental to recognition of the intended word or phrase, but in-dictionary words that do not collide will require less effort (not having to tap for each individual character), a higher risk, higher reward input scenario. However, since we restricted correction to deleting each character (as opposed to word-deletion), optimizing of correction/backspacing is likely to increase speed of the *STATSwype* variation and increase its external validity – particularly since users tend to spend more time on correction in real-world mobile-typing than in laboratory studies [118].

Considering real-world gestural text entry on smartphones, while we restricted participants to at least try the gesture first before employing the secondary tap technique in *STATSwype*, a combination of the two (as on modern smartphones) would likely exhibit better performance, giving the user the choice to tap or gesture if they think a word will not be correctly recognized. [P9] observed that “smaller words are much more difficult than larger words in the gesture method”—thus for words they believed would not be recognized, they would prefer tapping. One advantage of STAT in real-world use is that, as with modern smartphone-based WGKs, both tap and gesture typing can elegantly co-exist. While our chosen task for evaluation was transcription, due to ease of comparison with related works, a composition task [220] would assist in assessing these real-world scenarios in future work, and has the potential to reveal additional benefits of incorporating out of dictionary text in tapping conditions.

In vs. Out of Pocket

Our results for in vs. out of an enclosed front pants pocket exhibited similar performance (see Figure 5.7), but typing out-of-pocket was approximately 1 WPM faster in comparing each condition to their in-pocket counterpart. As anticipated, some participants noted pitfalls of typing in-pocket: the hand posture was more challenging [P5, P7], the cloth made dragging difficult [P3], and there were instances of accidental activation/tapping [P5, P7]. Additionally, some participants noted that in-pocket interaction may be difficult to do if wearing pants with tighter pockets [P4, P8]. While we controlled pocket size/flexibility, the interaction would not be plausible if pockets were too restrictive to contain both the user’s hand and phone, while allowing for small motions of the hand; an issue likely to arise for stiff, “skinny” pants – which clothing companies have begun to combat with increasingly more flexible fabric.

We expected that users would prefer STAT outside of an enclosed pocket rather than inside for two reasons. First, prior literature notes that social acceptability increases when it is obvious that the user is interacting with computation [181]; we assumed an explicit controller near the thigh would be perceived as more socially appropriate than subtle touches on one’s body near the waist or movement of fingers within one’s pocket. As well, the physical constraints of the pocket on the user’s hand might make input more challenging. However, surprisingly, multiple participants indicated a preference for in-pocket interaction: i.e. “inside felt better because I was getting some additional support from the pocket walls which made me feel less fatigued” [P3]; “[it] felt almost the same, but [a] bit easier, since I felt that the phone was more stable [in-pocket]” [P9]; the edges of the pocket were effective boundaries [P8, P11]; and “inside the pocket seemed more practical [...] I would use it if it was available” [P11]. Others noted there was “no difference” [P4] or it was “the same” [P6] as out-of-pocket, and that it was “trickier but not as much as I expected it to be” [P5]. These comments by participants suggest that on-body interaction near the user’s waist may not always be perceived as less socially acceptable [178]. Further, these results indicate that physical restriction may not necessarily be a limiting factor to the usability of STAT in constrained spaces.

5.6.3 Limitations

First, considering study design, we evaluated participants primarily in a seated position. This decision was driven by two aspects of our study configuration: the specific HMD used and study length. Considering our HMD, as noted in the Experimental Protocol, we used a MoGo Cinema2Go headset with a Nexus 6P smartphone. This headset, while

allowing a headlocked (egocentric) display solution for mobile device screens, lacks the comfort of other common VR HMDs, such as HTC’s Vive or Oculus Rift. Thus, for participant comfort, in an extended time frame wearing the headset (1-1.5 hours), our research ethics protocol was restricted to a seated position. While we initially piloted interaction in a standing position using *STATSwype*, we found the technique exhibited comparable performance to a seated position; thus, we determined tap (*STATTap*) vs. gestural text entry (*STATSwype*) to be a more useful contribution of the experiment. While standing in a stationary position in lab may exhibit similar performance, in mobile situations we would undoubtedly anticipate a degradation in performance. It becomes a challenge to compare how walking and/or running would impact performance across text input techniques, primarily because past at-side text input techniques such as Twiddler [145] and “eyes-free” input techniques [4, 84, 143, 205, 244, 246] were also evaluated in fixed, stationary contexts. However, an exploration of text input while moving serves as an interesting avenue for future work of the current technique, as well as others cited.

Next, while our method of strapping a capture device to a user’s leg echoes past work on subtle input [141], we acknowledge that this is somewhat unrealistic. We replicate prior evaluations [141] by strapping an input device to the thigh as a mechanism to evaluate the potential of forthcoming input mechanisms, such as pants pockets that allow transmission of touch input through fabric or interactive and touch-sensitive fabrics that transmit input to personal devices. Considering our use of a simulated pocket, it is the case that, unless users choose to wear pockets that have sufficient space or flexibility, in pocket text entry (or even carrying a smartphone in pocket) may not be desired. Past research [84, 141, 145, 208] addressed this by simply wearing the input device at belt or side, as in our strap-on condition. In the end, we chose to include a comparison of in- and out-of-pocket, as improving touch typing interactions on constrained touch interfaces is an area of ongoing research [4, 63, 64, 84, 130, 243, 256] with applications that include in-pocket text entry [256]. Given that many people do carry phones in their pants pockets (21 out of 23 of our survey respondents), we felt both in- and out-of-pocket had merit for exploration, but – while out-of-pocket text entry could be achieved through a strap on a users leg (e.g. Nintendo’s Ring Fit [163]), a belt clip [145] or, through fabric that permits text entry atop pants pockets – in-pocket text entry requires small hand movements [256]. While screen separation (see Section 3.1.1) does permit touch input in restricted spaces, both the use of a strap-on sensor and the use of a simulated pocket is one factor that may impact generalizability of our evaluation to real-world contexts.

Finally, we recognize a critique of the work may be the chosen sample size or number of repetitions. Our selection of sample size was initially motivated by prior text entry work [122, 143, 145, 257], and HCI literature, in which 12 participants was the most common

sample size reported [40]. As recommended by Caine [40], to ensure that sample size limitations were considered, we performed a power analysis on the 95% confidence interval of sample size 12 for moderate effect size (0.3), means of 12.5 and 11.7 (WPM), S.D. of between 3.0 and 3.1, which yielded power estimates above 0.95 for repeated measures analysis due to highly correlated speed and error across conditions (note that the results of power analysis – while not reported in past work – may be one reason for sample size selection in past work). Also, while we show that both *STATTap* and *STATSwype* are effective implementations for text entry, with only 20 repetitions of phrase entry it may be that our results are pessimistic estimates of the potential of STAT. A longitudinal study may provide more accurate final performance estimates.

5.7 Conclusion

In this work, we aim to determine whether users *can leverage existing interface expertise to assist in transition to a secondary mode in a new modality?* To address this, we assess two variations of Subtle Typing Around the Thigh (STAT), a text entry technique that allows for subtle, low-cost, unencumbered text entry for head-mounted displays (HMDs). Through a controlled laboratory evaluation of the technique, we validate whether users can transfer their pre-existing expertise with the QWERTY layout on a soft keyboard to STAT, to both a secondary mode, the word-gesture variation *STATSwype*, and the consistent mode, tap-based text entry variation *STATTap*. Atop this, we investigate how well the users can then transfer their new skill acquisition to inside a more constrained interaction environment: inside a user’s front pocket – and find only a minimal performance cost to doing so. While the expertise transfer from QWERTY soft keyboard experience to STAT shows promise by exhibiting comparable results to the literature for HMD text entry, the modality transition does result in a substantial initial performance and ceiling cost.

Chapter 6

Lessons in Mode and Modality Transfer

When we interact with computing systems, there often exists multiple interaction mechanisms that we can apply to complete the same task. As an illustration, we will use the example of a user wanting to zoom in on a PDF document displayed in macOS’s Preview application on a laptop, as there are at least four different methods they can use. First, they could right click on the document, open a contextual menu and navigate to zoom in (Figure 6.1). They could also use the menu bar, navigate to “view”, then to zoom in (Figure 6.2). Though these first two methods are within the same modality, that is, mouse based interaction, they are different modes of accomplishing the same task. You could also leverage two different modalities for identical command invocation: by pressing a hot key of “*Command +*” (Figure 6.3), or, by gesturing *pinch-to-zoom* on the trackpad (Figure 6.4). Though transferring from one of the more novice modalities, likely the manual menu navigation technique to the hot key modality (or “expert” modality), could exhibit an *intermodal* transfer style as characterized by Scarr et al. [195], this characterization is limited to these strict novice to expert transitions. It’s less clear how other modes or modalities may transfer to each other.

Kurtenbach notes, in the early 90s, that typically, good interfaces provide two modes of operation: novice and expert [127]. In reality, with the amount of multi-modal, cross-device, and truly pervasive computing today, interaction is far more nuanced. For instance, what if a user is moving from PDF annotation on a laptop to a mobile device? It’s likely that the mobile device would be less efficient, however, in mobile or ubiquitous scenarios, this type of interaction can be extremely useful. Thus, one could imagine benefits introduced by transferring expertise between these two modalities.

Leveraging the specific use case of unistroke gestural input and findings outlined in chapters three through five, this chapter is dedicated to characterizing mode and modality transfer both within and outside of the context of novice to expert scenarios.

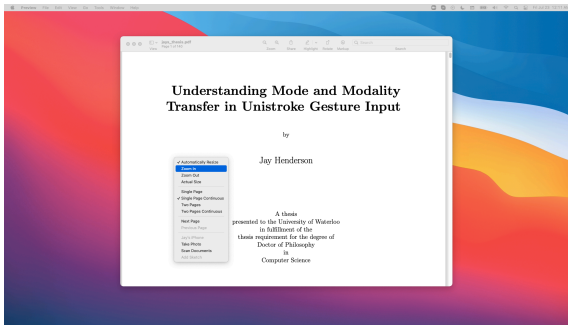


Figure 6.1: Contextual menu zoom in.

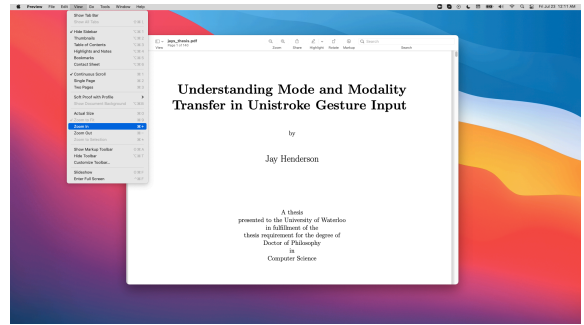


Figure 6.2: Menu bar zoom in.



Figure 6.3: Hot key zoom in.



Figure 6.4: Trackpad pinch to zoom in.

6.1 Should we force users to transfer modalities?

First, we will take a deeper look into our initial research question of whether or not users should be penalized, or forced, into transferring modalities. As mentioned in section 2.2.3, in his doctoral dissertation, Kurtenbach introduces the principle of *rehearsal* — that novice actions should mimic those of an expert to “smoothen” transition to expert interaction

techniques – the foundation of what marking menus are built on [127]. To encourage transition to the expert interaction technique (mark mode), the novice user is artificially penalized via delay. However, if the novice action is virtually identical to the expert action, should novice performance be penalized? In the specific case of rehearsal-based interfaces, this introduces a trade-off of whether penalties should be applied to optimize highly practiced use.

Taking our results of contrasting a zero delay marking menu to the traditional marking menu into account, we illustrate a characterization of the performance trade-off in mode/modality transfer for rehearsal-based, novice to expert transitions, depicted in Figure 6.5. As in Scarr et al.’s Dips and Ceilings, novice to expert mode or modality transfer is characterized by two power curves and a switch between the initial mode to the second mode results in a performance dip – which Kurtenbach’s principle of rehearsal aims to mitigate [127]. The first mode or modality relies on recognition, or guidance within an interface, and the second relies on recall.

For the purpose of the current work, as is typical in the HCI literature, we define performance as a function of time (including preparation, execution, selection), and rate of correctness. In our characterization, we introduce a non-penalized, truly unimodal curve, where the user can always rely on recognition, or scaffolded guidance – shown in red – as opposed to recall.

As you can see, for the novice user, there is an increase in temporal performance; this is illustrated as the area between the unimodal, non penalized curve, U , and the novice mode curve, N , from initial performance ($t = 0$) to the switch in mode/modality (s) (presented in orange). By implementing a penalty, this region of untapped performance is lost. Thus, if we recall that area between two curves is the integral of the the lower function subtracted from the upper function, we can conceptualize novice mode difference (α) as:

$$\alpha = \int_0^s U(t) - N(t) dt \tag{6.1}$$

For the expert technique, there is also a potential increase in temporal performance – if, in the expert mode area, the area between the expert mode curve, E , and the unimodal curve, U , is positive. We will refer to this as expert mode difference (β), defined as, the area between the unimodal curve and the expert mode curve (E) from the switch in mode/modality (s) to the point of intersection (v), represented in green, subtracted from

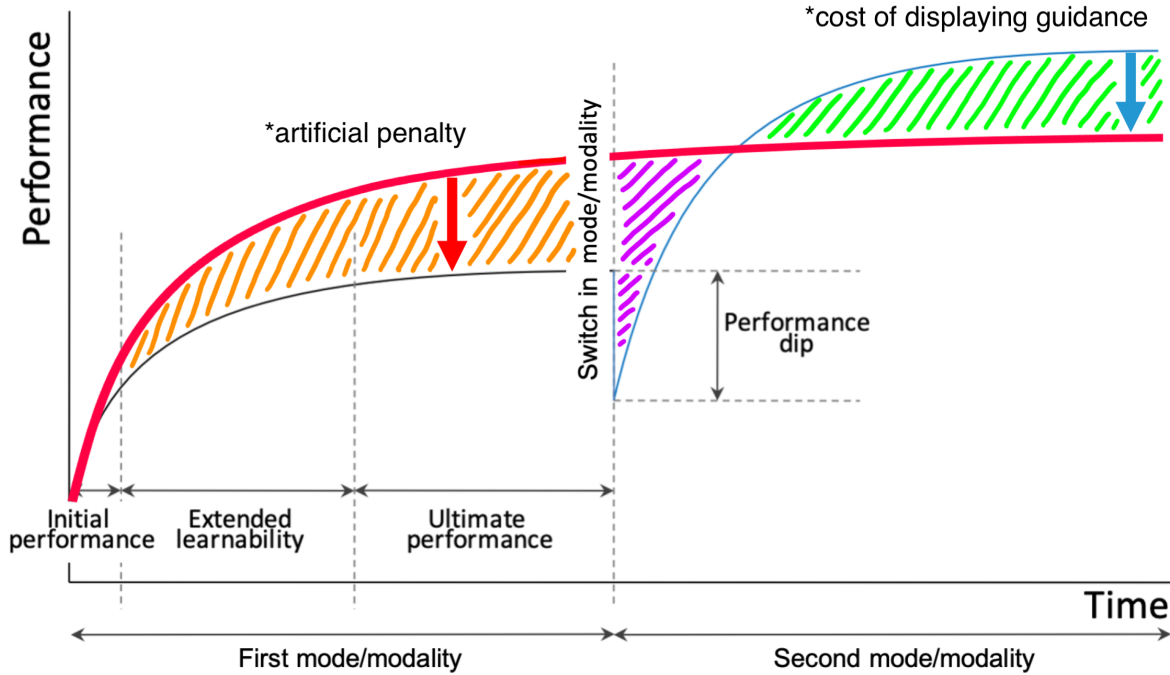


Figure 6.5: Mode/modality transfer characterized by removing penalty for relying on recognition in rehearsal based interfaces.

the area between the unimodal curve and the expert mode curve from the intersection (v) to the maximum time spent using an interaction technique (max), depicted as the section in purple.

$$\beta = \int_v^{max} E(t) - U(t) dt - \int_s^v U(t) - E(t) dt \quad (6.2)$$

So, for any particular command, there is a temporal benefit to a penalty if:

$$\beta > \alpha$$

Note that this characterization will apply to each command individually. So, the penalty must be advantageous for *more potentially invoked commands* than not. I say

potentially invoked commands, as not all commands will be used as frequently as one another, e.g. the copy hot key on macOS (Command C) is likely used more often than the table of contents hot key (Option Command 3). Additionally, not every command has an equal stake in terms of speed. For instance, if commands are in a gaming environment where temporal factors are of a higher degree of importance, a shooting command likely has a higher temporal requirement than an open menu command. Thus, for a system leveraging penalty to benefit from expert temporal performance, the following must be true:

$$\sum_{c=0}^n \rho_c k_c \beta_c > \sum_{c=0}^n \rho_c k_c \alpha_c \quad (6.3)$$

where:

c is each individual command

k is the number of times a command, (c), will be invoked

n is the total number of commands

ρ is the time sensitivity of a command | $0 < \rho < 1$

One might assume that since the expert curve appears to continue infinitely, that β will always be greater than α , so the penalty would always be useful. However, that is dependent on how long the user continues to use the mode or modality to invoke a command, in other words, how far along the time axis before the user stops using a command or system. This also applies to the novice interaction or first modality, the user may never interact with a system long enough or with enough repetitions to even reach the transition to expert mode. This has been evident in commercial implementations of marking menus – Kurtenbach noted in our email correspondence that “some users never used marks” and “many people have to be told about mark mode. They don’t seem to discover it”. Taking this into consideration, there seems to be a disconnect between research literature and implementations of these interaction styles in practice.

Though the first experiment in Chapter 3, reveals ambivalent benefit of the use of penalty in marking menus, Cockburn et al. [51] argue that designers should consider explicitly increasing mental effort, as they show that greater effort from the user will benefit learning spatial tasks in graphical user interfaces. However, they do note that this is of particular interest when the main objective is to train users to interact with interfaces that are highly dependent on spatial properties – including gesture shapes or

keypad layouts. Their findings are echoed by Lewis et al. [132], who show that greater penalties, specifically when using delay, increased the usage of expert interaction techniques in the case of marking menu and FastTap [90] style interfaces. However, they also found that the greater penalties also decreased accuracy, so they discourage the use in high-stakes interaction. We, too, observe this in Chapter 3, where participant error increases as forced recall increases. Understanding if fully autonomic behaviours can persist when more than a small number of menu items are to be accessed is an outstanding question. Furthermore, while we observed that visual disruption may cause small delays in fully autonomic action, it may be the case that, with increased experience, this slight delay vanishes because users are able to overlook the disruption. In general, it is difficult to draw strong conclusions about the benefit of delay for expert use, and it is clear that delay is a significant impediment to throughput during learning [133].

The results from Cockburn et al. [51] and Lewis et al. [132] are somewhat expected due to the guidance hypothesis, which suggests that augmented feedback geared to improve early performance through guidance may impair retaining the performed skills once the guidance is removed [196]. However, if removing guidance is not necessary, do we really need to rely on recall? Or is recognition sufficient? As the benefits are purely temporally related, and as recognition lowers error rate overall (as we discuss in Chapter 3 and as found by Lewis et al. [132]), if relying on recognition is possible, it isn't detrimental to performance. We therefore predict, and our results suggest, that less cognitive resources being spent on skill acquisition in interaction, will allow for a greater degree of focus on the particular task at hand. This is not to say that recall methods aren't useful, but, rather, that benefits should be carefully weighed as to whether or not they should not be forced upon users by discouraging reliance on guidance mechanisms. In addition, other methods that do not temporally impact users could be used to activate recall methods, such as a hot key, in order to not penalize novice users, but to preserve the shortcut mode/modality.

So, from our characterization, penalties should be reserved for **highly specific, time sensitive, frequently used interactions**, and it is vital to weigh the costs and benefits before introducing penalty.

Experiment one and two in Chapter 3 solidify the left-most and right-most portions of Figure 6.5; that is, for initial stages of skill acquisition, we see a substantial decrease in performance (orange region). In the *autonomous* case, as described by Fitts and Posner [67], we see an increase in performance (green region). However, the more intermittent stages of skill acquisition, or the *associative* stage, is less understood, and will require additional research for further understanding.

6.2 When do we need to transfer modes/modalities?

Our above characterization, and our results from Chapter 3, posit that reliance on recognition in the general use case for interaction can be beneficial to users. If this is the case, our second proposed research question was: *Under what circumstances do users need to transition from a novice mode to a secondary, recall, mode?*

These situations can be categorized as guidance free interactions – where it is either impractical or impossible for the user to rely on recognition, and must rely on recall. As guidance is usually provided via a visual mechanism [1, 10, 9, 47, 58, 59, 70, 110, 177, 222], these are generally either *eyes-free* interactions, interactions that do not encumber the user’s visual perception, or interfaces that do not have display capabilities, examples including:

- **Mid-air input:** Mid-air input is not-self revealing [25], so without some form of external display to guide users on how to perform input to a system, there is no natural mechanism to introduce interaction techniques to the user.
- **In-vehicle input:** When operating a vehicle, e.g. a car, drivers’ attention should ideally be dedicated to the road, and not to a guidance mechanism presenting interface controls.
- **Eyes-free mobile touch input:** Prior works have introduced eyes-free touch input on mobile devices [143, 257], so that (1) the screen can be freed to hold other content and/or (2) the user can visually focus elsewhere.
- **Imaginary touch interfaces:** Imaginary touch interfaces are spatial, non-visual, touch interfaces on any display-less objects or on the user them self [87, 88]

In these cases, since there is a desire to focus on recall during performance, we build upon the concept of using a separate modality that is capable, or allows for, visual guidance, in an effort to implicitly train users how to perform such inputs. In other words, we suggest leveraging an input mechanism that can trivially supply guidance, for mode or modality transition, to an input mechanism that requires recall. This is in congruence with Gustafson’s Imaginary Phone placed on the palm, where users transfer spatial knowledge from a mobile device touch screen to perform touch input on an ambiguous display-less surface [87] and Vatavu’s suggestion that gestures in a novel modality (such as mid-air) should be familiar to users from prior interactive modalities [214].

As presented in chapter 4, we zero in on the particular use case of mid-air marking menu gestures, that falls into the first example noted above. From our findings, we characterize

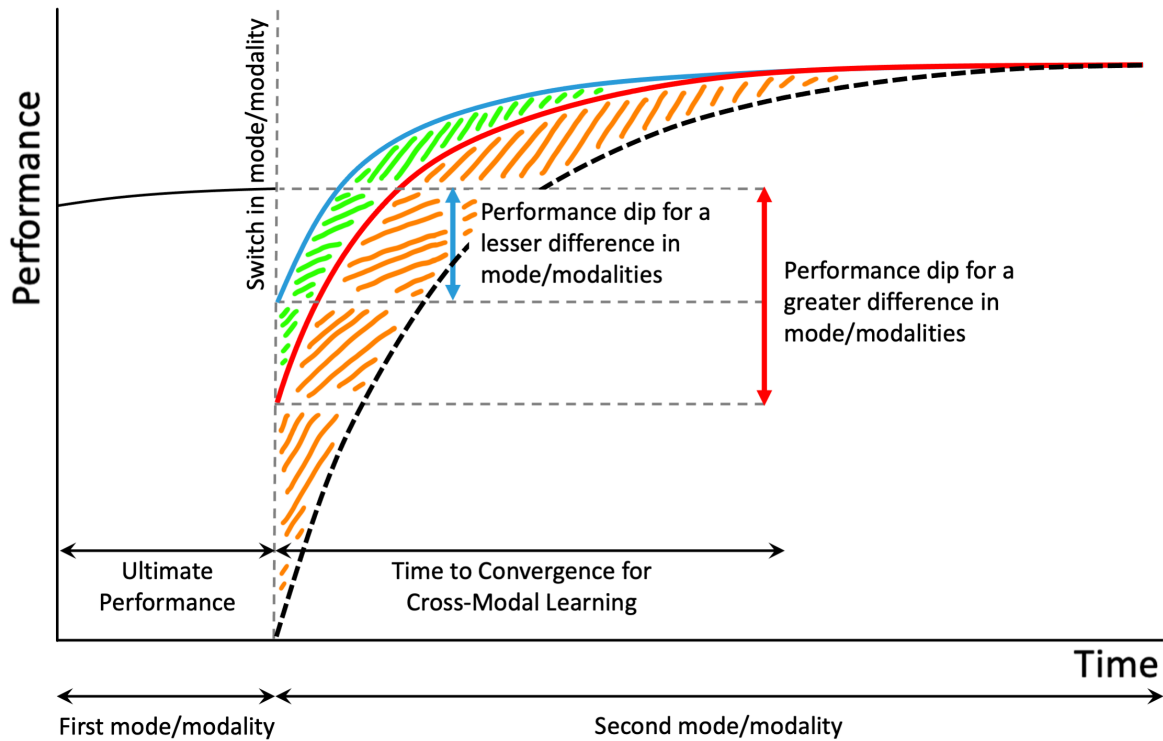


Figure 6.6: Characterization of transferring between recognition and recall modes across modalities.

this type of modality transfer as shown in Figure 6.6. The characterization consists of four power law performance curves. The first curve, depicted in solid black to the left of the mode or modality switch, is representative of the training or novice modality, as it reaches ultimate performance. The three performance curves after the switch in mode or modality are as follows:

- The blue curve – depicting optimal performance – is exhibited when there is the smallest possible difference between the novice modality and the target modality (second modality). In chapter 4, this would be indicative of performance after training in mid-air for a second mode also in mid-air – the same modality. As can be seen, there is a dip in performance due to the switch in mode (visual guidance to no visual guidance), but the dip remains relatively small due to the modality consistency.
- The next best is the novice modality that has a greater difference to the target modality

- the red curve. From chapter 4, this would represent the performance in mid-air after training via touch. The dip in performance is greater than that of the consistent modality training, but once the necessary adjustments are understood by the user, performance converges.
- Lastly, the baseline scenario, depicted as a dotted black curve, means no training at all. The user can eventually converge on ultimate performance, but the process would likely be slow and erroneous, due to relying on trial and error.

The area between the consistent modality performance curve and the cross modal performance curve, in green, represents the net performance increase in learning via the same modality as performance versus learning from an alternative modality. Between the cross-modal curve and baseline curve is the net performance increase for learning across modalities versus no training at all. I’ll also note, that this characterization aims to represent a spectrum of modalities that can be used as an alternative training modality, for example for mid-air interactions, this could be video representations, mouse input, touch input, etc. We then suggest, that the *less different* the training (first) modality is to the target (second) modality, the (1) smaller the performance dip (PD), and (2) less time to converge to ultimate performance (TC) of a second modality.

Thus, we theorize:

$$TC \propto \Delta modality \tag{6.4}$$

$$PD \propto \Delta modality \tag{6.5}$$

6.3 Transferring to less efficient modalities

As I mentioned in the initial portion of the chapter, mode or modality transfer may not always be devoted to a performance increase. Though the prior cases were geared more toward performance increase through the requirement of recall over recognition, there are other scenarios that may benefit from transferring between modalities – even when guidance mechanisms are still present for users. In this section, we will take a deeper look at how existing expertise with a mode or modality, can increase performance and ubiquity of an interactive technique.

Transferring the spatial knowledge of a desktop QWERTY keyboard layout to a mobile QWERTY keyboard is one of the most widespread interaction techniques that has stemmed

from the concept of leveraging existing expertise in another modality, for a less efficient, more convenient modality. Palin et al. [168] found that soft keyboard mobile typing was 15 WPM slower than physical desktop typing, and participants left more errors uncorrected on mobile devices. Even though there is a performance cost, these interactions are still valuable and have been commercially successful – as they introduce a number of benefits in comparison to desktop environments, such as portability and flexibility in display content. This transfer of keyboard layout expertise was also a culprit in prior modalities for text input on mobile being largely discontinued, such as T9 or multi-tap.

Within the HCI literature, this knowledge transfer of the QWERTY keyboard layout has been continuously used for creating intuitive interactions in novel modalities [85, 143, 243, 247, 246] – all of which suffer a deterioration in performance in comparison to the original desktop form factor and even to the mobile soft keyboard. In the same way, looking purely at performance, we see a substantial decline in comparison to our initial modality of soft-keyboard typing to the implementations of the STAT technique. However, as illustrated in Table 5.4, STAT is in the same performance vicinity as other text entry techniques designed for head-mounted displays, demonstrating its advantages.

Taking lessons from these works, and chapter 5, we present a final characterization of mode or modality transfer to less efficient second mode/modalities (Figure 6.7), that may pose contextual benefits. First, we note that in this case, and potentially the prior cases, the first mode or modality can be viewed as an accumulation of all prior experiences with a particular, or multiple related, input technique(s). While in our specific user study, we focus purely on the use case of QWERTY expertise, the characterization is aimed to generalize beyond this context. Within this model, there are two performance curves after switching to the target modality:

- The dotted black curve depicts a baseline case, where the user has no prior knowledge of the interaction technique before engaging in performance – therefore, the curve begins at 0 on the vertical axis.
- The red curve indicates a new target modality that leverages existing expertise, thus, it exhibits an initial performance increase in comparison to the baseline case, but a dip in performance relative to the existing modality expertise.

This reiterates Vatavu et al.’s suggestion that designers should allow users to interact with new modalities in familiar methods from preceding interactions [214]. The area between the baseline and the second modality curve (from existing expertise) is indicative of the net performance increase from using familiar techniques. Setting this characterization

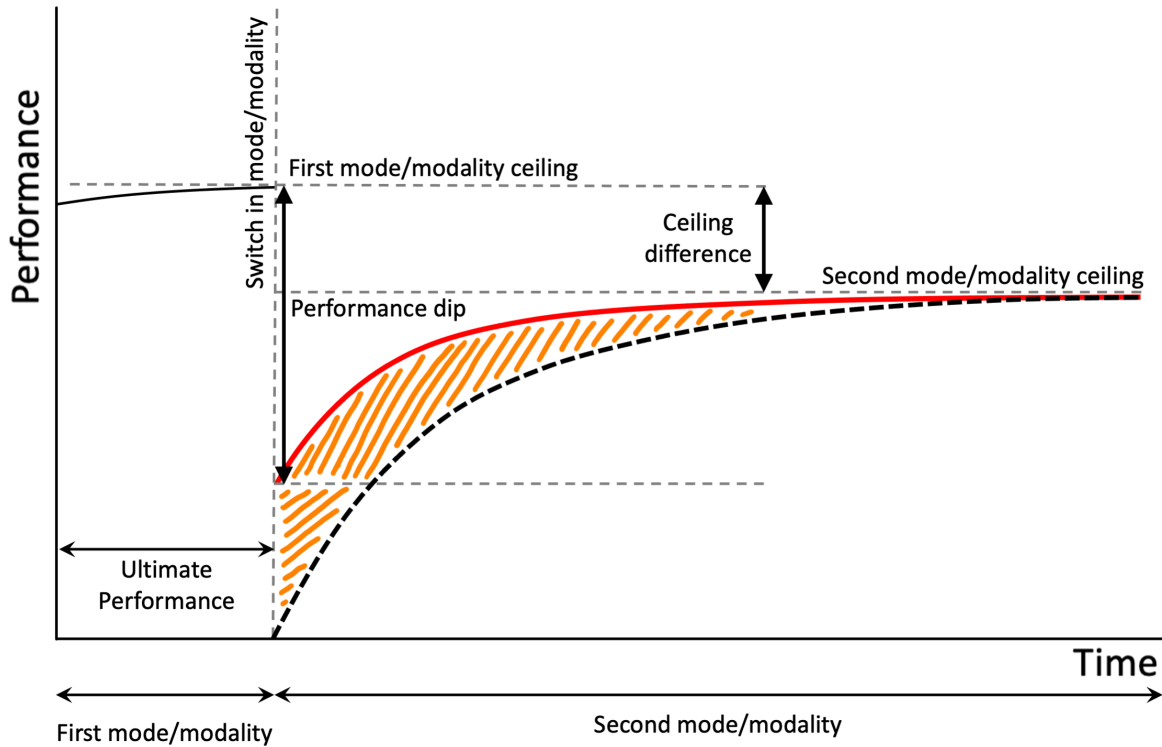


Figure 6.7: Characterization of transferring expertise from a higher ceiling modality to a lower ceiling modality.

apart from the others, is the ceiling difference at a lower level of performance than the previous modalities — thus, exhibiting both a ceiling difference and a performance dip. Similar to the cross-modal characterization introduced in section 6.2, we hypothesize that both the ceiling difference (CD) and performance dip (PD) will be proportional to the difference in the first modality and the second or target modality; as follows:

$$CD \propto \Delta modality \quad (6.6)$$

$$PD \propto \Delta modality \quad (6.7)$$

We revealed these characteristics in chapter 5, to start, when the user transitioned from their pre-existing QWERTY keyboard expertise to use the STAT technique. As the existing

expertise was substantially different and more challenging than STAT, the performance dip and ceiling difference were greater than those exhibited in a more similar interaction to the prior expertise (as Zhu et al. found in their displayless word-gesture keyboard [257]). However, when transferring from STAT atop the thigh to in-pocket, as the interactions were quite similar, only in a more constrained environment, so the performance dip and ceiling performance revealed a lesser degree of difference.

6.4 Transfer distance between modes/modalities

In section 2.2.1, we define *transfer* as the ability to perform a motor skill in an environment or method separate from the context of which it was acquired. This requires the user to conceptualize the difference between the new method and the originally acquired skill in order to perform the newly transferred skill, but this distance between a first mode or modality to a secondary mode or modality can differ between scenarios. In Table 6.1, I present a few examples of transfer distance between interaction methods.

Transfer Distance	First Mode/Modality	Second Mode/Modality
LOW	Performing a touch screen gesture with visual guidance	Performing the same touch screen gesture with no visual guidance
MED	Performing a touch screen gesture	Performing the same gesture with the user’s bare hand in mid-air
HIGH	Performing a touch screen gesture	Performing the same gesture in virtual reality using foot rotation

Table 6.1: Examples of relative skill transfer distances from low to high.

To illustrate this in terms of the current works, in chapter 3, the transfer distance between modes is minimal or low – the motor requirements remain relatively consistent, with the exception of no visible interface. The same is true for transferring on-thigh STAT to in-pocket STAT (chapter 5); the interaction is almost identical, the display remains on the same HMD, the controller is a mobile device placed around the thigh, the interaction remains a three-state-cursor based technique. The difference is the device is placed in an enclosed pocket, so the motor requirements may slightly change due to the additional constrained environment.

There is, however, substantial difference in transferring expertise from traditional QWERTY tap or gesture typing on a mobile device to performing it using STAT, which we define as a high transfer distance. For example, the mobile device is inverted, the user must incorporate both their index finger and thumb in the three-state-cursor technique (as opposed to one or the other likely in typical mobile device input), and the interaction becomes indirectly mapped with output on a HMD.

6.5 Conclusion

In this chapter, based on results from chapter 3 through 5, we present three characterizations for transferring expertise between modes or modalities, extended from Scarr et al.'s *Dips and Ceilings* [195] and Cockburn et al.'s *Supporting Novice to Expert Transitions* [50].

While we are optimistic about the generalizability of these models, we note an obvious limitation of our user studies is that we focused on one particular use case of mode/modality transition – that is, for transferring to *symbolic-abstract unistroke gesture input*. Though this is a very specific use case, we felt since marking menus served as the introductory HCI piece for the concept of transferring expertise through rehearsal, and that word-gesture keyboards are one of the most commonly available gesture inputs techniques, that these were credible starting points. Furthermore, because we use a wide variety of input mechanisms in our user studies; including mouse, touch screen, in-air, and input for head-mounted displays, we encompass a large body of interactive equipment and methodologies. However, we acknowledge the need for confirmation and welcome contradiction or extension of our characterizations – especially considering the multifaceted nature of mode/modality transitions.

Chapter 7

Conclusion

In this chapter, we discuss the implications of our findings, and revisit proposed research questions introduced in Chapter 1.

The vast majority of related work on mode or modality transitions has been devoted to strict, novice to expert transitions for optimizing performance in a user interface. The fundamental goal of this thesis, is to build upon these existing theories of mode or modality transfer by either questioning existing, accepted, hypotheses or applying them to new contexts. Using symbolic-abstract unistroke gestures: marking menus and word-gesture keyboards, we provide insights to the following questions.

7.1 Should interaction designers force users to transition to a secondary mode?

The seminal rehearsal-based interface, the marking menu, introduced a penalty to encourage users to switch from a recognition mode to a potentially more efficient interaction technique that relies on recall [127]. Despite its adoption in the research community, the necessity of applying a penalty, often deployed through a temporal delay, had never been questioned. In Chapter 3, we contrast the original marking menu with a no delay marking menu at varying stages in the skill acquisition process, revealing performance trade-offs for separating modes via delay. We conduct a deeper dive into whether penalty should be applied to motivate the use of a secondary, recall-based, mode or modality, and present a framework indicating when designers should implement penalization in Chapter 6.

7.2 Under what circumstances do users need to transition to a secondary mode?

Since in our initial study (Chapter 3), we revealed little evidence for relying on a recall mode over recognition mode, specifically in marking menus, we theorize that, situations where displaying visual guidance is impractical could pose advantages for recall mode or modalities. In the use case of mid-air interactions, since guidance is challenging, we propose the use of guided touch screen interactions to develop spatial expertise in performing displayless mid-air interactions. In the context of marking menus, after a short-term performance dip, this type of transfer resulted in similar performance regardless of whether the user rehearsed using a guided touch screen or in mid-air through an external display. We further characterize these cross-modal transfers in Chapter 6, introducing a spectrum of alternative mechanisms for rehearsal and recognize that the degree of interactive difference will influence performance dips and time to convergence.

7.3 Can we leverage existing interface expertise to assist in transition to a new mode in a new modality?

After determining the efficacy of transferring expertise across modalities (for touch to in-air), we study the user’s ability to transfer pre-existing expertise to new modalities. In a complex unistroke gesture interaction — the QWERTY keyboard layout, that requires extensive prior exposure, we present an analysis of users abilities to conduct word-gesture and tap-based text entry in a novel context: touch input to head-mounted display, when the interface is mounted to their thigh. We found the technique exhibited comparable performance to other HMD text input techniques, suggesting the validity of the expertise transfer. Additionally, once users rehearsed the on-thigh technique, they were able to perform in an enclosed pant pocket without intervention, contributing to our cross-modal transfer characterization in Chapter 6. As expected, the new context exhibited a lower degree of performance in comparison to the initial modality (soft-keyboard typing), so we present an additional characterization for inter-modal transfer outside of novice to expert transitions, also in Chapter 6, and suggest the performance dip and secondary modality performance ceiling is proportional to the difference in interaction.

7.4 Recommendations

Based on findings of our three research questions and our characterization, we make the following recommendations for interaction design:

1. *Consider whether the performance increase in a recall mode justifies penalizing the novice in a recognition mode.* Revisiting equation 6.1, there needs to be an advantage for a significant amount of command usage to give merit to penalizing the novice user. Penalty should be reserved for specific, frequently used, temporally demanding, command selection.
2. *Spatial knowledge transfer can occur across modalities.* If a system requires reliance on spatial recall, the guidance mode can be presented in an alternative modality, to allow for spatial learning. This approach can provide similar benefits for spatial learning to a consistent modality. Particularly, mid-air interactions could be an ideal target modality for touch-based instruction.
3. *When possible, leverage existing expertise to increase expertise in novel modalities.* Similar to the previous point, consistency is key. While this is suggested in prior works, existing interaction methods can and should be included in novel modalities. This would allow for reliance on implicit, rather than explicit, learning to ease transfer to and adoption of new technology.

7.5 Limitations

While we feel our research protocol is sufficient to draw upon some recommendations and shed light on new areas for research within mode/modality transfer, our work is inherently limited in scope. To start, we focus on a niche use case for these transfers: *symbolic-abstract unistroke gestures*. However, there are many other types of interfaces that rely on rehearsal-based transfer, such as MarkPad [71], FastTap [89, 90], and ExposeHK [150]. While MarkPad relies on unistroke gesture input, the latter two use chording or keypresses.

Secondly, each of our experiments are limited to a relatively small sample size ($n \approx 12-16$), within a limited population (for the most part, university students ages 20-30 from a technical institution). This is a constraint in much of the human-computer interaction literature, due to the reach of recruitment in a university setting. Our population sample generally grew up in the later part of the information age, where personal computers,

smartphones, and WiFi have become commonplace. It's less clear if or how different populations, such as older adults, people with motor impairments, or less experienced computer users, would transfer expertise across modes or modalities. For instance, in Chapter 3, a different population may never discover or switch to using the *expert* mode, diminishing the temporal benefit for switching. In addition, STAT (Chapter 5), requires extremely fine-grained motor skills, so regardless of pre-existing expertise using variants of QWERTY, the skill may never transfer. Taking these limitations into consideration, an important piece of follow-on work is understanding if and how these results may generalize to (1) separate populations and (2) other rehearsal-based interfaces.

7.6 Future Work

Undertaking our research agenda in the space of transfer between modes and modalities has shed some light on additional areas for exploration.

7.6.1 Penalty to other rehearsal-based interfaces

First, as discussed, Chapter 3 largely discourages the use of a delay penalty. However, we only explore a single type of rehearsal-based interface, marking menus [127, 124]. Marking menus have spawned an entire area of rehearsal-based interfaces, so we question if our results can generalize to each of these. Of particular interest is MarkPad [71], that deploys directional gestures on a trackpad, and FastTap [90, 89], that utilizes chords on a touchscreen; both of these utilize a guided mode and a recall mode – separated by a delay.

Outside of this, if there are advantages to an expert mode in other rehearsal-based interfaces, we question whether a technique outside of delay should be used to delimit the recognition and recall modes. For example, could we use an additional key press (like “Shift”), or an double-tap to enter novice mode? How would these techniques impact performance in comparison to delay?

7.6.2 How far can expertise be transferred?

Since we focused solely on basic, directional stroke gestures for transfer across touch to in-air modalities, we question whether other spatial unistroke gesture systems will exhibit similar results. For instance, ideographic gestures, such as drawing a square, or alphanumeric gestures, like drawing an “X”. How complex can gestures be when transferring across

touch and in-air before there is no benefit to cross-modal training? How many gestures can be transferred? Other than unistroke (static pose and path) gestures, can other types of gestures, such as dynamic poses, be transferred?

Another area we wished to explore is whether contexts makes a difference. In-vehicle interactions appeared an opportunistic area to apply cross-modal transfer to, as discussed in Chapter 6, because they require consistent attentional resources from the operator. If users can be guided via surface gestures of how to invoke particular commands, we question if they can then transfer that knowledge to perform mid-air gestures as a shortcut – thus, leaving more cognitive resources to the primary task of driving.

7.6.3 Quantifying the difference in modes/modalities

Both section 6.2 and 6.3 posit that performance after transition relies on the difference between the modes or modalities. That being said, while we can observe surface level differences (discussed in section 6.4), how exactly to quantify these differences is an open research question. Tu et al. quantified the differences between varying complexities of unistroke gestures produced on 2D surfaces, using various algebraic and geometric features. We propose using a similar experimental protocol to determine differences in gesture productions between modalities, such as: mouse, trackpad, touch, in-air controller, or in-air barehand. Analyzing these modality differences can also be of value when transferring gestures between contexts, as the more representative a trained gesture is to the required performance gesture, the more likely a system’s produced output will be indicative of the user intention.

7.7 Concluding Remarks

Expertise transfer across modalities are more common than you think — this field includes hot keys, consistent gestures across devices (such as pinch to zoom), and physical to soft keyboard layouts. Throughout my research, it has become abundantly clear that these types of interaction expertise transitions extend far beyond the novice-to-expert contextualization proposed in the literature. While I acknowledge that our work is a preliminary investigation of deepening the understanding of these phenomena, limiting our user evaluations to *symbolic-abstract unistroke gestures*, I feel the findings and subsequently outlined characterizations will be invaluable for implementing transferable expertise in future interaction design.

References

- [1] Christopher Ackad, Andrew Clayphan, Martin Tomitsch, and Judy Kay. An in-the-wild study of learning mid-air gestures to browse hierarchical information at a large interactive public display. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '15*, pages 1227–1238, New York, NY, USA, 2015. ACM.
- [2] David Ahlström, Andy Cockburn, Carl Gutwin, and Pourang Irani. Why it's quick to be square: Modelling new and existing hierarchical menu designs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10*, pages 1371–1380, New York, NY, USA, 2010. ACM.
- [3] David Ahlström, Khalad Hasan, and Pourang Irani. Are you comfortable doing that?: Acceptance studies of around-device gestures in and for public settings. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services, MobileHCI '14*, pages 193–202, New York, NY, USA, 2014. ACM.
- [4] Sunggeun Ahn, Seongkook Heo, and Geehyuk Lee. Typing on a Smartwatch for Smart Glasses. In *In Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces - ISS '17*, pages 201–209, 2017.
- [5] Sunggeun Ahn, Stephanie Santosa, Mark Parent, Daniel Wigdor, Tovi Grossman, and Marcello Giordano. Stickypie: A gaze-based, scale-invariant marking menu optimized for ar/vr. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, CHI '21*, New York, NY, USA, 2021. Association for Computing Machinery.
- [6] Roland Aigner, Daniel Wigdor, Hrvoje Benko, Michael Haller, David Lindbauer, Alexandra Ion, Shengdong Zhao, and Jeffrey Tzu Kwan Valino Koh. Understanding

mid-air hand gestures: A study of human preferences in usage of gesture types for hci. Technical Report MSR-TR-2012-111, November 2012.

- [7] Deepak Akkil, Andrés Lucero, Jari Kangas, Tero Jokela, Marja Salmimaa, and Roope Raisamo. User expectations of everyday gaze interaction on smartglasses. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, NordiCHI '16, New York, NY, USA, 2016. Association for Computing Machinery.
- [8] Jason Alexander, Teng Han, William Judd, Pourang Irani, and Sriram Subramanian. Putting your best foot forward: Investigating real-world mappings for foot-based gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1229–1238, New York, NY, USA, 2012. ACM.
- [9] Florian Alt, Sabrina Geiger, and Wolfgang Höhl. Shapelineguide: Teaching mid-air gestures for large interactive displays. In *Proceedings of the 7th ACM International Symposium on Pervasive Displays*, PerDis '18, pages 3:1–3:8, New York, NY, USA, 2018. ACM.
- [10] Fraser Anderson, Tovi Grossman, Justin Matejka, and George Fitzmaurice. Youmove: Enhancing movement training with an augmented reality mirror. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 311–320, New York, NY, USA, 2013. ACM.
- [11] Daniel Ashbrook, Patrick Baudisch, and Sean White. NENYA: Subtle and Eyes-Free Mobile Input with a Magnetically-Tracked Finger Ring. In *Proceedings of the 29th international conference on Human factors in computing systems - CHI 11*, pages 2043–2046, 2011.
- [12] İlhan Aslan, Ida Buchwald, Philipp Koytek, and Elisabeth André. Pen + mid-air: An exploration of mid-air gestures to complement pen input on tablets. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, NordiCHI '16, pages 1:1–1:10, New York, NY, USA, 2016. ACM.
- [13] Shiri Azenkot and Shumin Zhai. Touch behavior with different postures on soft smartphone keyboards. *MobileHCT'12 - Proceedings of the 14th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 251–260, 2012.
- [14] Gilles Bailly, Eric Lecolinet, and Yves Guiard. Finger-count and radial-stroke shortcuts: 2 techniques for augmenting linear menus on multi-touch surfaces. In *Proceed-*

ings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10, pages 591–594, New York, NY, USA, 2010. ACM.

- [15] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. Wave menus: Improving the novice mode of hierarchical marking menus. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction*, INTERACT'07, pages 475–488, Berlin, Heidelberg, 2007. Springer-Verlag.
- [16] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. Flower menus: A new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, pages 15–22, New York, NY, USA, 2008. ACM.
- [17] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. Visual menu techniques. *ACM Comput. Surv.*, 49(4):60:1–60:41, December 2016.
- [18] Gilles Bailly, Jörg Müller, Michael Rohs, Daniel Wigdor, and Sven Kratz. Shoesense: A new perspective on gestural interaction and wearable applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1239–1248, New York, NY, USA, 2012. ACM.
- [19] Gilles Bailly, Robert Walter, Jörg Müller, Tongyan Ning, and Eric Lecolinet. Comparing free hand menu techniques for distant displays using linear, marking and finger-count menus. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II*, INTERACT'11, pages 248–262, Berlin, Heidelberg, 2011. Springer-Verlag.
- [20] Ravin Balakrishnan and I. Scott MacKenzie. Performance differences in the fingers, wrist, and forearm in computer input control. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 303–310, New York, NY, USA, 1997. ACM.
- [21] Matthias Baldauf, Florence Adegeye, Florian Alt, and Johannes Harms. Your browser is the controller: Advanced web-based smartphone remote controls for public screens. In *Proceedings of the 5th ACM International Symposium on Pervasive Displays*, PerDis '16, pages 175–181, New York, NY, USA, 2016. ACM.
- [22] Nikola Banovic, Koji Yatani, and Khai N. Truong. Escape-Keyboard: A Sight-Free One-Handed Text Entry Method for Mobile Touch-Screen Devices. In *International Journal of Mobile Human Computer Interaction*, volume 5, pages 42–61, 2013.

- [23] Raju S Bapi, Kenji Doya, and Alexander M Harner. Evidence for effector independent and dependent representations and their differential time course of acquisition during motor sequence learning. *Experimental Brain Research*, 132(2):149–162, 2000.
- [24] Olivier Bau and Wendy E. Mackay. Octopocus: A dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, pages 37–46, New York, NY, USA, 2008. ACM.
- [25] Thomas Baudel and Michel Beaudouin-Lafon. Charade: Remote control of objects using free-hand gestures. *Commun. ACM*, 36(7):28–35, July 1993.
- [26] Sebastian Baumgärtner, Achim Ebert, Matthias Deller, and Stefan Agne. 2d meets 3d: A human-centered interface for visual data exploration. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '07, pages 2273–2278, New York, NY, USA, 2007. ACM.
- [27] Michel Beaudouin-Lafon. Instrumental interaction: An interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, page 446–453, New York, NY, USA, 2000. Association for Computing Machinery.
- [28] Louis-Pierre Bergé, Emmanuel Dubois, Laurence Boudet, and Michel Rodriguez. Menu linéaire, contextuel, et circulaire: cas d’application pour la création de situation tactique (sitac). In *Résumés étendus de la 31e conférence francophone sur l’Interaction Homme-Machine (IHM 2019)*, pages 4–1, 2019.
- [29] Mathieu Berthelley, Elodie Cayez, Marwan Ajem, Gilles Bailly, Sylvain Malacria, and Eric Lecolinet. Spotpad, locipad, chordpad and inoutpad: Investigating gesture-based input on touchpad. In *Proceedings of the 27th Conference on L’Interaction Homme-Machine*, IHM '15, pages 4:1–4:8, New York, NY, USA, 2015. ACM.
- [30] Xiaojun Bi and Shumin Zhai. Ijqwerty: What difference does one key change make? gesture typing keyboard optimization bounded by one key position change from qwerty. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 49–58, New York, NY, USA, 2016. Association for Computing Machinery.
- [31] Richard A Bolt. *“Put-that-there”: Voice and gesture at the graphics interface*, volume 14. ACM, 1980.

- [32] Matthew N. Bonner, Jeremy T. Brudvik, Gregory D. Abowd, and W. Keith Edwards. No-look notes: Accessible eyes-free multi-touch text entry. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6030 LNCS:409–426, 2010.
- [33] David Bonnet, Caroline Appert, and Michel Beaudouin-Lafon. Extending the vocabulary of touch events with thumbrock. In *Proceedings of Graphics Interface 2013*, GI '13, pages 221–228, Toronto, Ont., Canada, Canada, 2013. Canadian Information Processing Society.
- [34] Doug Bowman, Chadwick Wingrave, Joshua Campbell, V. Ly, and C. Rhoton. Novel uses of pinch glovesTM for virtual environment interaction techniques. *Virtual Reality*, 6:122–129, 10 2002.
- [35] Heather Wilde Braden, Stefan Panzer, and Charles H Shea. The effects of sequence difficulty and practice on proportional and nonproportional transfer. *The Quarterly Journal of Experimental Psychology*, 61(9):1321–1339, 2008.
- [36] Andrew Bragdon, Robert Zeleznik, Brian Williamson, Timothy Miller, and Joseph J. LaViola, Jr. Gesturebar: Improving the approachability of gesture-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2269–2278, New York, NY, USA, 2009. ACM.
- [37] Stephen A. Brewster, Peter C. Wright, and Alistair D. N. Edwards. The design and evaluation of an auditory-enhanced scrollbar. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, page 173–179, New York, NY, USA, 1994. Association for Computing Machinery.
- [38] William Buxton. A three-state model of graphical input. In *Human-computer interaction-INTERACT*, volume 90, pages 449–456, 1990.
- [39] William Buxton. Touch, gesture & marking. In *Readings in Human Computer Interaction: Toward the Year 2000*, chapter 7, pages 469–482. San Francisco: Morgan Kaufmann Publishers, 1995.
- [40] Kelly Caine. Local standards for sample size at chi. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, page 981–992, New York, NY, USA, 2016. Association for Computing Machinery.

- [41] Xiang Cao and Ravin Balakrishnan. Visionwand: Interaction techniques for large displays using a passive wand tracked in 3d. *ACM Trans. Graph.*, 23(3):729–729, August 2004.
- [42] John M. Carroll and Caroline Carrithers. Training wheels in a user interface. *Commun. ACM*, 27(8):800–806, August 1984.
- [43] Steven J. Castellucci and I. Scott MacKenzie. Gathering text entry metrics on android devices. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, 2011.
- [44] Pew Research Center. Mobile Fact Sheet, 2019.
- [45] Xiang Anthony Chen, Tovi Grossman, and George Fitzmaurice. Swipeboard: A text entry technique for ultra-small interfaces that supports novice to expert transitions. *UIST 2014 - Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 615–620, 2014.
- [46] Xiang 'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott E. Hudson. Air+touch: Interweaving touch & in-air gestures. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 519–525, New York, NY, USA, 2014. ACM.
- [47] Christopher Clarke and Hans Gellersen. Matchpoint: Spontaneous spatial coupling of body movement for touchless pointing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pages 179–192, New York, NY, USA, 2017. ACM.
- [48] A. Cockburn and B. McKenzie. Evaluating spatial memory in two and three dimensions. *Int. J. Hum.-Comput. Stud.*, 61(3):359–373, September 2004.
- [49] Andy Cockburn. Revisiting 2d vs 3d implications on spatial memory. In *Proceedings of the Fifth Conference on Australasian User Interface - Volume 28*, AUIC '04, pages 25–31, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
- [50] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. Supporting novice to expert transitions in user interfaces. *ACM Comput. Surv.*, 47(2):31:1–31:36, November 2014.
- [51] Andy Cockburn, Per Ola Kristensson, Jason Alexander, and Shumin Zhai. Hard lessons: Effort-inducing interfaces benefit spatial learning. In *Proceedings of the*

- SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1571–1580, New York, NY, USA, 2007. ACM.
- [52] Andy Cockburn and Bruce McKenzie. Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 203–210, New York, NY, USA, 2002. ACM.
- [53] Philip R. Cohen, Michael Johnston, David McGee, Sharon Oviatt, Jay Pittman, Ira Smith, Liang Chen, and Josh Clow. Quickset: Multimodal interaction for distributed applications. In *Proceedings of the Fifth ACM International Conference on Multimedia*, MULTIMEDIA '97, pages 31–40, New York, NY, USA, 1997. ACM.
- [54] Céline Coutrix and Nadine Mandran. Identifying emotions expressed by mobile users through 2d surface and 3d motion gestures. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 311–320, New York, NY, USA, 2012. ACM.
- [55] Fergus Craik and Robert S. Lockhart. Levels of processing: A framework for memory research. 11:671–, 12 1972.
- [56] Fergus IM Craik and Robert S Lockhart. Levels of processing: A framework for memory research. *Journal of verbal learning and verbal behavior*, 11(6):671–684, 1972.
- [57] Sarah E Criscimagna-Hemminger, Opher Donchin, Michael S Gazzaniga, and Reza Shadmehr. Learned dynamics of reaching movements generalize from dominant to nondominant arm. *Journal of neurophysiology*, 89(1):168–176, 2003.
- [58] William Delamare, Céline Coutrix, and Laurence Nigay. Designing guiding systems for gesture-based interaction. In *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '15, pages 44–53, New York, NY, USA, 2015. ACM.
- [59] William Delamare, Thomas Janssoone, Céline Coutrix, and Laurence Nigay. Designing 3d gesture guidance: Visual feedback and feedforward design options. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, pages 152–159, New York, NY, USA, 2016. ACM.
- [60] Android Developers. Android NDK, 2018.

- [61] Android Developers. Android NDK. 2019.
- [62] Tilman Dingler, Rufat Rzayev, Alireza Sahami Shirazi, and Niels Henze. Designing consistent gestures across device types: Eliciting rsvp controls for phone, watch, and glasses. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 419:1–419:12, New York, NY, USA, 2018. ACM.
- [63] David Dobbstein, Philipp Hock, and Enrico Rukzio. Belt: An Unobtrusive Touch Input Device for Head-worn Displays. *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems - CHI '15*, pages 2135–2138, 2015.
- [64] David Dobbstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. Pocket-Thumb: a Wearable Dual-Sided Touch Interface for Cursor-based Control of Smart-Eyewear. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2), 2017.
- [65] Yuan Du, Haoyi Ren, Gang Pan, and Shjian Li. Tilt & touch: Mobile phone for 3d interaction. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, pages 485–486, New York, NY, USA, 2011. ACM.
- [66] Brian D. Ehret. Learning where to look: Location learning in graphical user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 211–218, New York, NY, USA, 2002. ACM.
- [67] Paul M Fitts and Michael I Posner. Human performance. 1967.
- [68] Jérémie Francone, Gilles Bailly, Eric Lecolinet, Nadine Mandran, and Laurence Nigay. Wavelet menus on handheld devices: stacking metaphor for novice mode and eyes-free selection for expert mode. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pages 173–180. ACM, 2010.
- [69] Dustin Freeman, Hrvoje Benko, Meredith Ringel Morris, and Daniel Wigdor. Shadowguides: Visualizations for in-situ learning of multi-touch and whole-hand gestures. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 165–172, New York, NY, USA, 2009. ACM.
- [70] Euan Freeman, Stephen Brewster, and Vuokko Lantz. Do that, there: An interaction technique for addressing in-air gesture systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 2319–2331, New York, NY, USA, 2016. ACM.

- [71] Bruno Fruchard, Eric Lecolinet, and Olivier Chapuis. Markpad: Augmenting touchpads for command selection. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 5630–5642, New York, NY, USA, 2017. ACM.
- [72] Sandra G. Hart and L E. Stavenland. Development of nasa-tlx (task load index): Results of empirical and theoretical research, 12 1988.
- [73] Varun Gaur, Md. Sami Uddin, and Carl Gutwin. Multiplexing spatial memory: Increasing the capacity of fasttap menus with multiple tabs. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '18, pages 22:1–22:13, New York, NY, USA, 2018. ACM.
- [74] Emmanouil Giannidakis, Gilles Bailly, Sylvain Malacria, and Fanny Chevalier. Iconhk: Using toolbar button icons to communicate keyboard shortcuts. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 4715–4726, New York, NY, USA, 2017. ACM.
- [75] Alix Goguey, Sylvain Malacria, Andy Cockburn, and Carl Gutwin. Reducing error aversion to support novice-to-expert transitions with fasttap. In *Actes de la 31e conférence francophone sur l'Interaction Homme-Machine (IHM 2019)*, pages 1:1–10, Grenoble, France, 2019. ACM.
- [76] David Goldberg and Cate Richardson. Touch-typing with a stylus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 80–87, New York, NY, USA, 1993. ACM.
- [77] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang'Anthony' Chen, Xiaojun Bi, and Xing-Dong Yang. WrisText: One-handed Text Entry on Smartwatch using Wrist Gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018.
- [78] Gabriel González, José P. Molina, Arturo S. García, Diego Martínez, and Pascual González. Evaluation of Text Input Techniques in Immersive Virtual Environments. In *New Trends on Human-Computer Interaction: Research, Development, New Tools and Methods*, pages 1–161. 2009.
- [79] Mitchell Gordon, Tom Ouyang, and Shumin Zhai. WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, pages 3817–3821, 2016.

- [80] Tovi Grossman, Xiang Anthony Chen, and George Fitzmaurice. Typing on Glasses: Adapting Text Entry to Smart Eyewear. In *In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '15*, pages 144–152, 2015.
- [81] Tovi Grossman, Pierre Dragicevic, and Ravin Balakrishnan. Strategies for accelerating on-line learning of hotkeys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1591–1600, New York, NY, USA, 2007. ACM.
- [82] Tovi Grossman, Daniel Wigdor, and Ravin Balakrishnan. Multi-finger gestural interaction with 3d volumetric displays. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, page 61–70, New York, NY, USA, 2004. Association for Computing Machinery.
- [83] François Guimbreti re and Terry Winograd. Flowmenu: Combining command, text, and data entry. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, pages 213–216, New York, NY, USA, 2000. ACM.
- [84] Aakar Gupta and Ravin Balakrishnan. DualKey: Miniature Screen Text Entry via Finger Identification. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, (Figure 1):59–70, 2016.
- [85] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. Roto-Swype: Word-Gesture Typing using a Ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, 2019.
- [86] Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. Imaginary interfaces: Spatial interaction with empty hands and without visual feedback. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 3–12, New York, NY, USA, 2010. ACM.
- [87] Sean Gustafson, Christian Holz, and Patrick Baudisch. Imaginary phone: Learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 283–292, New York, NY, USA, 2011. Association for Computing Machinery.
- [88] Sean G. Gustafson, Bernhard Rabe, and Patrick M. Baudisch. *Understanding Palm-Based Imaginary Interfaces: The Role of Visual and Tactile Cues When Browsing*, page 889–898. Association for Computing Machinery, New York, NY, USA, 2013.

- [89] Carl Gutwin, Andy Cockburn, and Benjamin Lafreniere. Testing the rehearsal hypothesis with two fasttap interfaces. In *Proceedings of the 41st Graphics Interface Conference, GI '15*, pages 223–231, Toronto, Ont., Canada, Canada, 2015. Canadian Information Processing Society.
- [90] Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. Faster command selection on tablets with fasttap. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems, CHI '14*, pages 2617–2626, New York, NY, USA, 2014. ACM.
- [91] Faizan Haque, Mathieu Nancel, and Daniel Vogel. Myopoint: Pointing and clicking using forearm mounted electromyography and inertial motion sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, pages 3653–3656, New York, NY, USA, 2015. ACM.
- [92] Chris Harrison and Scott Hudson. Using shear as a supplemental two-dimensional input channel for rich touchscreen interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pages 3149–3152, New York, NY, USA, 2012. ACM.
- [93] Chris Harrison, Julia Schwarz, and Scott E. Hudson. Tapsense: Enhancing finger interaction on touch surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST '11*, pages 627–636, New York, NY, USA, 2011. ACM.
- [94] Florian Heller, Stefan Ivanov, Chat Wacharamanotham, and Jan Borchers. Fabri-Touch: Exploring Flexible Touch Input on Textiles. *Proceedings of the 2014 ACM International Symposium on Wearable Computers - ISWC '14*, pages 59–62, 2014.
- [95] Jay Henderson, Sachi Mizobuchi, Wei Li, and Edward Lank. Exploring cross-modal training via touch to learn a mid-air marking menu gesture set. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '19*, pages 8:1–8:9, New York, NY, USA, 2019. ACM.
- [96] Okihide Hikosaka, Hiroyuki Nakahara, Miya K Rand, Katsuyuki Sakai, Xiaofeng Lu, Kae Nakamura, Shigehiro Miyachi, and Kenji Doya. Parallel neural networks for learning sequential procedures. *Trends in neurosciences*, 22(10):464–471, 1999.
- [97] Okihide Hikosaka, Kae Nakamura, Katsuyuki Sakai, and Hiroyuki Nakahara. Central mechanisms of motor skill learning. *Current opinion in neurobiology*, 12(2):217–222, 2002.

- [98] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1063–1072, New York, NY, USA, 2014. ACM.
- [99] Ken Hinckley. The handbook of multimodal-multisensor interfaces. chapter A Background Perspective on Touch As a Multimodal (and Multisensor) Construct, pages 143–199. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA, 2017.
- [100] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. A survey of design issues in spatial input. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST '94, page 213–222, New York, NY, USA, 1994. Association for Computing Machinery.
- [101] Ken Hinckley and Hyunyoung Song. Sensor synaesthesia: Touch in motion, and motion in touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 801–810, New York, NY, USA, 2011. ACM.
- [102] Jonggi Hong, Seongkook Heo, Poika Isokoski, and Geehyuk Lee. SplitBoard: A Simple Split Soft Keyboard for Wristwatch-sized Touch Screens. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, pages 1233–1236, 2015.
- [103] Don Hopkins. The design and implementation of pie menus. *Dr. Dobb's J.*, 16(12):16–26, December 1991.
- [104] Scott E. Hudson, Chris Harrison, Beverly L. Harrison, and Anthony LaMarca. Whack gestures: Inexact and inattentive interaction with mobile devices. *TEI'10 - Proceedings of the 4th International Conference on Tangible, Embedded, and Embodied Interaction*, pages 109–112, 2010.
- [105] Simon Ismair, Julie Wagner, Ted Selker, and Andreas Butz. Mime: Teaching mid-air pose-command mappings. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, pages 199–206, New York, NY, USA, 2015. ACM.
- [106] Mikkel R. Jakobsen, Yvonne Jansen, Sebastian Boring, and Kasper Hornbæk. Should i stay or should i go? selecting between touch and mid-air gestures for large-display interaction. In Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe

- Palanque, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2015*, pages 455–473, Cham, 2015. Springer International Publishing.
- [107] Anthony Jameson and Per Ola Kristensson. The handbook of multimodal-multisensor interfaces. chapter Understanding and Supporting Modality Choices, pages 201–238. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA, 2017.
- [108] Brett Jones, Rajinder Sodhi, David Forsyth, Brian Bailey, and Giuliano Maciocci. Around device interaction for multiscale navigation. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, pages 83–92, New York, NY, USA, 2012. ACM.
- [109] Eleanor Jones, Jason Alexander, Andreas Andreou, Pourang Irani, and Sriram Subramanian. Gestext: Accelerometer-based gestural text-entry systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2173–2182, New York, NY, USA, 2010. ACM.
- [110] Ankit Kamal, Yang Li, and Edward Lank. Teaching motion gestures via recognizer feedback. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, pages 73–82, New York, NY, USA, 2014. ACM.
- [111] Thorsten Karrer, Moritz Wittenhagen, Leonhard Lichtschlag, Florian Heller, and Borchers Jan. Pinstripe: Eyes-free continuous input on interactive clothing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*, pages 1313–1322, 2011.
- [112] Keiko Katsuragawa, Ankit Kamal, Qi Feng Liu, Matei Negulescu, and Edward Lank. Bi-level thresholding: Analyzing the effect of repeated errors in gesture input. *ACM Trans. Interact. Intell. Syst.*, 9(2-3):15:1–15:30, April 2019.
- [113] Keiko Katsuragawa, Krzysztof Pietroszek, James R. Wallace, and Edward Lank. Watchpoint: Freehand pointing with a smartwatch in a ubiquitous display environment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, pages 128–135, New York, NY, USA, 2016. ACM.
- [114] Steven W Keele, Peggy Jennings, Steven Jones, David Caulton, and Asher Cohen. On the modularity of sequence representation. *Journal of Motor Behavior*, 27(1):17–30, 1995.

- [115] J. Kim, W. Delamare, and P. Irani. ThumbText: Text entry for wearable devices using a miniature ring. *Proceedings of Graphics Interface 2018 - GI '18*, 2018-May:18–25, 2018.
- [116] Kenrick Kin, Björn Hartmann, and Maneesh Agrawala. Two-handed marking menus for multitouch devices. *ACM Trans. Comput.-Hum. Interact.*, 18(3), August 2011.
- [117] Pascal Knierim, Anna Maria Feit, Niels Henze, Valentin Schwind, and Florian Nieuwenhuizen. Physical Keyboards in Virtual Reality: Analysis of Typing Performance and Effects of Avatar Hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 2018.
- [118] Andreas Komminos, Mark Dunlop, Kyriakos Katsaris, and John Garofalakis. A glimpse of mobile text entry errors and corrective behaviour in the wild. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, MobileHCI '18, page 221–228, New York, NY, USA, 2018. Association for Computing Machinery.
- [119] Attila J Kovacs, Thomas Mühlbauer, and Charles H Shea. The coding and effector transfer of movement sequences. *Journal of Experimental Psychology: Human Perception and Performance*, 35(2):390, 2009.
- [120] John W Krakauer, Pietro Mazzoni, Ali Ghazizadeh, Roshni Ravindran, and Reza Shadmehr. Generalization of motor learning depends on the history of prior action. *PLoS biology*, 4(10):e316, 2006.
- [121] Brian Krisler and Richard Alterman. Training towards mastery: Overcoming the active user paradox. In *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges*, NordiCHI '08, page 239–248, New York, NY, USA, 2008. Association for Computing Machinery.
- [122] Per-Ola Kristensson and Shumin Zhai. Shark2: A large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 43–52, New York, NY, USA, 2004. ACM.
- [123] Gordon Kurtenbach and William Buxton. Issues in combining marking and direct manipulation techniques. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology*, UIST '91, pages 137–144, New York, NY, USA, 1991. ACM.

- [124] Gordon Kurtenbach and William Buxton. The limits of expert performance using hierarchic marking menus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 482–487, New York, NY, USA, 1993. ACM.
- [125] Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 258–264, New York, NY, USA, 1994. ACM.
- [126] Gordon Kurtenbach, George W. Fitzmaurice, Russell N. Owen, and Thomas Baudel. The hotbox: Efficient access to a large number of menu-items. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 231–237, New York, NY, USA, 1999. ACM.
- [127] Gordon Paul Kurtenbach. *The Design and Evaluation of Marking Menus*. PhD thesis, Toronto, Ont., Canada, Canada, 1993. UMI Order No. GAXNN-82896.
- [128] Benjamin Lafreniere, Carl Gutwin, Andy Cockburn, and Tovi Grossman. Faster command selection on touchscreen watches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4663–4674, New York, NY, USA, 2016. ACM.
- [129] Kris MY Law, Victor CS Lee, and Yuen-Tak Yu. Learning motivation in e-learning facilitated computer programming courses. *Computers & Education*, 55(1):218–228, 2010.
- [130] Luis A. Leiva, Alireza Sahami, Alejandro Catala, Niels Henze, and Albrecht Schmidt. Text entry on tiny QWERTY soft keyboards. *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems - CHI '15*, 2015-April:669–678, 2015.
- [131] Sören Lenman, Lars Bretzner, and Björn Thuresson. Using marking menus to develop command sets for computer vision based hand gesture interfaces. In *Proceedings of the Second Nordic Conference on Human-Computer Interaction*, NordiCHI '02, page 239–242, New York, NY, USA, 2002. Association for Computing Machinery.
- [132] Blaine Lewis. Longer delays in rehearsal-based interfaces increase expert use. Master's thesis, University of Waterloo, 2019.
- [133] Blaine Lewis and Daniel Vogel. Longer delays in rehearsal-based interfaces increase expert use. *ACM Trans. Comput.-Hum. Interact.*, 27(6), November 2020.

- [134] Clayton Lewis and John Rieman. Task-centered user interface design. *A Practical Introduction*, 1993.
- [135] Frank Chun Yat Li, Richard T Guy, Koji Yatani, and Khai N Truong. The 1line keyboard: A QWERTY layout in a single line. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, pages 461–470, 2011.
- [136] Hai-Ning Liang, Cary Williams, Myron Semegen, Wolfgang Stuerzlinger, and Pourang Irani. User-defined surface+motion gestures for 3d manipulation of objects at a distance through a mobile device. In *Proceedings of the 10th Asia Pacific Conference on Computer Human Interaction, APCHI '12*, pages 299–308, New York, NY, USA, 2012. ACM.
- [137] Hyunchul Lim, Jungmin Chung, Changhoon Oh, SoHyun Park, Joonhwan Lee, and Bongwon Suh. Touch+finger: Extending touch-based user interface capabilities with "idle" finger gestures in the air. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology, UIST '18*, page 335–346, New York, NY, USA, 2018. Association for Computing Machinery.
- [138] Zhi Han Lim and Per Ola Kristensson. An evaluation of discrete and continuous mid-air loop and marking menu selection in optical see-through hmds. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '19*, pages 16:1–16:10, New York, NY, USA, 2019. ACM.
- [139] Robert W. Lindeman, John L. Sibert, and James K. Hahn. Towards usable vr: An empirical study of user interfaces for immersive virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '99*, page 64–71, New York, NY, USA, 1999. Association for Computing Machinery.
- [140] Frank Linton, Deborah Joy, and Hans-Peter Schaefer. Building user and expert models by long-term observation of application usage. In *UM99 User Modeling*, pages 129–138. Springer, 1999.
- [141] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. Gunslinger: Subtle Arms-down Mid-air Interaction. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*, pages 63–71, 2015.
- [142] Cinema2Go LTD. MOGO Travel, 2018.

- [143] Yiqin Lu, Chun Yu, Xin Yi, Yuanchun Shi, and Shengdong Zhao. Blindtype: Eyes-free text entry on handheld touchpad by leveraging thumb’s muscle memory. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(2), June 2017.
- [144] Yuexing Luo and Daniel Vogel. Pin-and-cross: A unimanual multitouch technique combining static touches with crossing selection. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, UIST ’15, pages 323–332, New York, NY, USA, 2015. ACM.
- [145] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E W Looney. Twiddler typing: one-handed chording text entry for mobile phones. In *Proceedings of the 2004 conference on Human factors in computing systems - CHI ’04*, volume 6, pages 671–678, 2004.
- [146] I. Scott MacKenzie. Fitts’ law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7(1):91–139, 1992.
- [147] I. Scott MacKenzie and R. William Soukoreff. Phrase sets for evaluating text entry techniques. *CHI ’03 extended abstracts on Human factors in computing systems - CHI ’03*, 2003.
- [148] I Scott Mackenzie and R William Soukoreff. Text Entry for Mobile Computing: Models and Methods, Theory and Practice. *Human-Computer Interaction*, 17(December 2014):37–41, 2011.
- [149] I. Scott MacKenzie and Shawn X. Zhang. The immediate usability of graffiti. In *Proceedings of the Conference on Graphics Interface ’97*, page 129–137, CAN, 1997. Canadian Information Processing Society.
- [150] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. Promoting hotkey use through rehearsal with exposehk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’13, pages 573–582, New York, NY, USA, 2013. ACM.
- [151] Anders Markussen, Mikkel Ronne Jakobsen, and Kasper Hornbæk. Vulture: A Mid Air Word Gesture Keyboard. In *In Proceedings of the 2014 CHI Conference on Human Factors in Computing Systems - CHI ’14*, pages 1073–1082, 2014.
- [152] Nicolai Marquardt, Ricardo Jota, Saul Greenberg, and Joaquim A. Jorge. The continuous interaction space: Interaction techniques unifying touch and gesture on and above a digital surface. In Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno

- Nunes, Philippe Palanque, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2011*, pages 461–476, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [153] Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. A dose of reality: Overcoming usability challenges in vr head-mounted displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 2143–2152, New York, NY, USA, 2015. Association for Computing Machinery.
- [154] Joanna McGrenere, Ronald M. Baecker, and Kellogg S. Booth. A field evaluation of an adaptable two-interface design for feature-rich software. *ACM Trans. Comput.-Hum. Interact.*, 14(1), May 2007.
- [155] Microsoft. SwiftKey Keyboard for Android. 2019.
- [156] Sarah Morrison-Smith, Megan Hofmann, Yang Li, and Jaime Ruiz. Using audio cues to support motion gesture interaction on mobile devices. *ACM Trans. Appl. Percept.*, 13(3):16:1–16:19, May 2016.
- [157] Aske Mottelson, Christoffer Larsen, Mikkel Lyderik, Paul Strohmeier, and Jarrod Knibbe. Invisiboard: Maximizing display and input space with a full screen text entry method for smartwatches. *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2016*, pages 53–59, 2016.
- [158] Jörg Müller, Gilles Bailly, Thor Bossuyt, and Niklas Hillgren. Mirrortouch: Combining touch and mid-air gestures for public displays. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services, MobileHCI '14*, pages 319–328, New York, NY, USA, 2014. ACM.
- [159] Lisa M Muratori, Eric M Lamberg, Lori Quinn, and Susan V Duff. Applying principles of motor learning and control to upper extremity rehabilitation. *Journal of Hand Therapy*, 26(2):94–103, 2013.
- [160] Miguel A. Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. Memorability of pre-designed and user-defined gesture sets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1099–1108, New York, NY, USA, 2013. ACM.

- [161] Mathieu Nancel, Stéphane Huot, and Michel Beaudouin-Lafon. Un espace de conception fondé sur une analyse morphologique des techniques de menus. In *Proceedings of the 21st International Conference on Association Francophone D'Interaction Homme-Machine*, IHM '09, pages 13–22, New York, NY, USA, 2009. ACM.
- [162] Laurence Nigay. Design space for multimodal interaction. In Renè Jacquart, editor, *Building the Information Society*, pages 403–408, Boston, MA, 2004. Springer US.
- [163] Nintendo. Ring fit adventure for nintendo switch, 2020.
- [164] Ian Oakley and Junseok Park. A motion-based marking menu system. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '07, pages 2597–2602, New York, NY, USA, 2007. ACM.
- [165] Ian Oakley and Junseok Park. Motion marking menus: An eyes-free approach to motion input for handheld devices. *Int. J. Hum.-Comput. Stud.*, 67(6):515–532, June 2009.
- [166] Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. ZoomBoard: A Diminutive QWERTY Soft Keyboard Using Iterative Zooming for Ultra-Small Devices. *Proceedings of the 2013 CHI Conference on Human Factors in Computing Systems - CHI '13*, pages 2799–2802, 2013.
- [167] Alexander Pacha. Sensor fusion for robust outdoor augmented reality tracking on mobile devices. 2013.
- [168] Kseniia Palin, Anna Maria Feit, Sunjun Kim, Per Ola Kristensson, and Antti Oulasvirta. How do people type on mobile devices? observations from a study with 37,000 volunteers. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '19, New York, NY, USA, 2019. Association for Computing Machinery.
- [169] Stefan Panzer, Melanie Krueger, Thomas Muehlbauer, Attila J Kovacs, and Charles H Shea. Inter-manual transfer and practice: Coding of simple motor sequences. *Acta psychologica*, 131(2):99–109, 2009.
- [170] Krzysztof Pietroszek, Anastasia Kuzminykh, James R. Wallace, and Edward Lank. Smartcasting: A discount 3d interaction technique for public displays. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design*, OzCHI '14, pages 119–128, New York, NY, USA, 2014. ACM.

- [171] Krzysztof Pietroszek and Edward Lank. Clicking blindly: Using spatial correspondence to select targets in multi-device environments. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '12, page 331–334, New York, NY, USA, 2012. Association for Computing Machinery.
- [172] Krzysztof Pietroszek, Liudmila Tahai, James R. Wallace, and Edward Lank. 3d interaction with networked public displays using mobile and wearable devices. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, UbiComp/ISWC'15 Adjunct, pages 787–788, New York, NY, USA, 2015. ACM.
- [173] Krzysztof Pietroszek, James R. Wallace, and Edward Lank. Tiltcasting: 3d interaction on large displays using a mobile device. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 57–62, New York, NY, USA, 2015. ACM.
- [174] Beryl Plimmer, Andrew Crossan, Stephen A. Brewster, and Rachel Blagojevic. Multimodal collaborative handwriting training for visually-impaired people. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 393–402, New York, NY, USA, 2008. ACM.
- [175] Henning Pohl, Andreea Muresan, and Kasper Hornbæk. Charting Subtle Interaction in the HCI Literature. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, 2019.
- [176] Stuart Pook, Eric Lecolinet, Guy Vaysseix, and Emmanuel Barillot. Control menus: Execution and control in a single interactor. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '00, pages 263–264, New York, NY, USA, 2000. ACM.
- [177] Otniel Portillo-Rodriguez, Oscar O. Sandoval-Gonzalez, Emanuele Ruffaldi, Rosario Leonardi, Carlo Alberto Avizzano, and Massimo Bergamasco. Real-time gesture recognition, evaluation and feed-forward correction of a multimodal tai-chi platform. In Antti Pirhonen and Stephen Brewster, editors, *Haptic and Audio Interaction Design*, pages 30–39, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [178] Halley Profita, James Clawson, Scott Gilliland, Clint Zeagler, Thad Starner, Jim Budd, and Ellen Yi-luen Do. Don't Mind Me Touching My Wrist : A Case Study

- of Interacting with On-Body Technology in Public Social Acceptability of Wearable Technology. In *Proceedings of the 2013 International Symposium on Wearable Computers (ISWC '13)*, pages 89–96, 2013.
- [179] Hanae Rateau, Yosra Rekik, Laurent Grisoni, and Joaquim Jorge. Talaria: Continuous drag & drop on a wall display. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces, ISS '16*, pages 199–204, New York, NY, USA, 2016. ACM.
- [180] Shyam Reyal, Shumin Zhai, and Per Ola Kristensson. Performance and User Experience of Touchscreen and Gesture Keyboards in a Lab Setting and in the Wild. *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems - CHI '15*, pages 679–688, 2015.
- [181] Julie Rico and Stephen Brewster. Gesture and voice prototyping for early evaluations of social acceptability in multimodal interfaces. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI '10*, New York, NY, USA, 2010. Association for Computing Machinery.
- [182] Sami Ronkainen, Jonna Häkkinä, Saana Kaleva, Ashley Colley, and Jukka Linjama. Tap input as an embedded interaction method for mobile devices. *TEI'07: First International Conference on Tangible and Embedded Interaction*, pages 263–270, 2007.
- [183] Anne Roudaut, Gilles Bailly, Eric Lecolinet, and Laurence Nigay. Leaf menus: Linear menus with stroke shortcuts for small handheld devices. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part I, INTERACT '09*, pages 616–619, Berlin, Heidelberg, 2009. Springer-Verlag.
- [184] Gustavo Rovelo, Donald Degraen, Davy Vanacken, Kris Luyten, and Karin Coninx. Gestu-wan - an intelligible mid-air gesture guidance system for walk-up-and-use displays. In *INTERACT*, 2015.
- [185] Quentin Roy. Marking menu implementation in javascript. <https://github.com/QuentinRoy/Marking-Menu>, 2018.
- [186] Quentin Roy, Sylvain Malacria, Yves Guiard, Eric Lecolinet, and James Eagan. Augmented letters: Mnemonic gesture-based shortcuts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13*, pages 2325–2328, New York, NY, USA, 2013. ACM.

- [187] Jaime Ruiz and Yang Li. Doubleflip: A motion gesture delimiter for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2717–2720, New York, NY, USA, 2011. ACM.
- [188] Jaime Ruiz, Yang Li, and Edward Lank. User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 197–206, New York, NY, USA, 2011. ACM.
- [189] T. Scott Saponas, Chris Harrison, and Hrvoje Benko. PocketTouch: Through-fabric capacitive touch input. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*, 2011.
- [190] William Saunders and Daniel Vogel. Tap-kick-click: Foot interaction for a standing desk. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, DIS '16, pages 323–333, New York, NY, USA, 2016. ACM.
- [191] Joey Scarr, Andy Cockburn, and Carl Gutwin. Supporting and exploiting spatial memory in user interfaces. *Found. Trends Hum.-Comput. Interact.*, 6(1):1–84, December 2013.
- [192] Joey Scarr, Andy Cockburn, and Carl Gutwin. Supporting and exploiting spatial memory in user interfaces. *Found. Trends Hum.-Comput. Interact.*, 6(1):1–84, December 2013.
- [193] Joey Scarr, Andy Cockburn, Carl Gutwin, and Andrea Bunt. Improving command selection with commandmaps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 257–266, New York, NY, USA, 2012. ACM.
- [194] Joey Scarr, Andy Cockburn, Carl Gutwin, and Sylvain Malacria. Testing the robustness and performance of spatially consistent interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3139–3148, New York, NY, USA, 2013. ACM.
- [195] Joey Scarr, Andy Cockburn, Carl Gutwin, and Philip Quinn. Dips and ceilings: Understanding and supporting transitions to expertise in user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2741–2750, New York, NY, USA, 2011. ACM.

- [196] Richard A. Schmidt, Douglas E. Young, Stephan Swinnen, and Diane C. Shapiro. Summary knowledge of results for skill acquisition: Support for the guidance hypothesis. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15(2):352–359, 1989.
- [197] Christian Schönauer, Kenichiro Fukushi, Alex Olwal, Hannes Kaufmann, and Ramesh Raskar. Multimodal motion guidance: Techniques for adaptive and dynamic feedback. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction, ICMI '12*, pages 133–140, New York, NY, USA, 2012. ACM.
- [198] Abigail J. Sellen. Speech patterns in video-mediated conversations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '92*, pages 49–59, New York, NY, USA, 1992. ACM.
- [199] Charles H Shea, Attila J Kovacs, and Stephan Panzer. The coding and inter-manual transfer of movement sequences. *Frontiers in psychology*, 2:52, 2011.
- [200] John Shea and Robyn Morgan. Contextual interference effects on the acquisition, retention, and transfer of a motor skill. *Journal of Experimental Psychology: Human Learning and Memory*, 5:179, 03 1979.
- [201] Shaishav Siddhpuria, Keiko Katsuragawa, James R. Wallace, and Edward Lank. Exploring at-your-side gestural interaction for ubiquitous environments. In *Proceedings of the 2017 Conference on Designing Interactive Systems, DIS '17*, pages 1111–1122, New York, NY, USA, 2017. ACM.
- [202] Shaishav Siddhpuria, Sylvain Malacria, Mathieu Nancel, and Edward Lank. Pointing at a distance with everyday smart devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18*, pages 173:1–173:11, New York, NY, USA, 2018. ACM.
- [203] Rajinder Sodhi, Hrvoje Benko, and Andrew Wilson. Lightguide: Projected visualizations for hand movement guidance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pages 179–188, New York, NY, USA, 2012. ACM.
- [204] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. In-air gestures around unmodified mobile devices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, UIST '14*, page 319–329, New York, NY, USA, 2014. Association for Computing Machinery.

- [205] Marco Speicher, Anna Maria Feit, Pascal Ziegler, and Antonio Krüger. Selection-based Text Entry in Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 2018.
- [206] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20:1371–1375, 1998.
- [207] David J. Sturman and David Zeltzer. A design method for “whole-hand” human-computer interaction. *ACM Trans. Inf. Syst.*, 11(3):219–238, July 1993.
- [208] B. Thomas, K. Grimmer, J. Zucco, and S. Milanese. Where does the mouse go? An investigation into the placement of a body-attached touchpad mouse for wearable computers. *Personal and Ubiquitous Computing*, 6(2):97–112, 2002.
- [209] Theophanis Tsandilas, Caroline Appert, Anastasia Bezerianos, and David Bonnet. Coordination of tilt and touch in one- and two-handed use. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 2001–2004, New York, NY, USA, 2014. ACM.
- [210] Huawei Tu, Xiangshi Ren, and Shumin Zhai. A comparative evaluation of finger and pen stroke gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 1287–1296, New York, NY, USA, 2012. Association for Computing Machinery.
- [211] Huawei Tu, Xiangshi Ren, and Shumin Zhai. Differences and similarities between finger and pen stroke gestures on stationary and mobile devices. *ACM Trans. Comput.-Hum. Interact.*, 22(5), August 2015.
- [212] Md. Sami Uddin, Carl Gutwin, and Benjamin Lafreniere. Handmark menus: Rapid command selection and large command sets on multi-touch displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 5836–5848, New York, NY, USA, 2016. Association for Computing Machinery.
- [213] Ultraleap. Tracking: Leap motion controller.
- [214] Radu-Daniel Vatavu. Nomadic gestures: A technique for reusing gesture commands for frequent ambient interactions. *J. Ambient Intell. Smart Environ.*, 4(2):79–93, April 2012.

- [215] Radu-Daniel Vatavu. User-defined gestures for free-hand tv control. In *Proceedings of the 10th European Conference on Interactive TV and Video*, EuroITV '12, pages 45–48, New York, NY, USA, 2012. ACM.
- [216] Radu-Daniel Vatavu. A comparative study of user-defined handheld vs. freehand gestures for home entertainment environments. *J. Ambient Intell. Smart Environ.*, 5(2):187–211, March 2013.
- [217] Radu-Daniel Vatavu and Stefan Pentiu. Interacting with gestures: An intelligent virtual environment. pages 293–299, 2006.
- [218] Radu-Daniel Vatavu and Ionut-Alexandru Zaiti. Leap gestures for tv: Insights from an elicitation study. In *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video*, TVX '14, pages 131–138, New York, NY, USA, 2014. ACM.
- [219] Keith Vertanen, Dylan Gaines, Crystal Fletcher, Alex M. Stanage, Robbie Watling, and Per Ola Kristensson. VelociWatch: Designing and Evaluating a Virtual Keyboard for the Input of Challenging Text. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, 2019.
- [220] Keith Vertanen and Per Ola Kristensson. Complementing text entry evaluations with a composition task. *ACM Trans. Comput.-Hum. Interact.*, 21(2), February 2014.
- [221] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. VelociTap: Investigating Fast Mobile Text Entry using Sentence-Based Decoding of Touchscreen Keyboard Input. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, pages 659–668, 2015.
- [222] David Verweij, Vassilis-Javed Khan, Augusto Esteves, and Saskia Bakker. Multi-user motion matching interaction for interactive television using smartwatches. In *Adjunct Publication of the 2017 ACM International Conference on Interactive Experiences for TV and Online Video*, TVX '17 Adjunct, pages 67–68, New York, NY, USA, 2017. ACM.
- [223] Willem B Verwey. A forthcoming key press can be selected while earlier ones are executed. *Journal of motor behavior*, 27(3):275–284, 1995.
- [224] Willem B Verwey. Buffer loading and chunking in sequential keypressing. *Journal of Experimental Psychology: Human Perception and Performance*, 22(3):544, 1996.

- [225] Willem B Verwey. Evidence for a multistage model of practice in a sequential movement task. *Journal of Experimental Psychology: Human Perception and Performance*, 25(6):1693, 1999.
- [226] Willem B Verwey. Effect of sequence length on the execution of familiar keying sequences: Lasting segmentation and preparation? *Journal of Motor Behavior*, 35(4):343–354, 2003.
- [227] Willem B Verwey. Processing modes and parallel processors in producing familiar keying sequences. *Psychological research*, 67(2):106–122, 2003.
- [228] Daniel Vogel and Ravin Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, page 33–42, New York, NY, USA, 2005. Association for Computing Machinery.
- [229] William S. Walmsley, W. Xavier Snelgrove, and Khai N. Truong. Disambiguation of imprecise input with one-dimensional rotational text entry. *ACM Transactions on Computer-Human Interaction*, 21, 2014.
- [230] Robert Walter, Gilles Bailly, and Jörg Müller. Strikeapose: Revealing mid-air gestures on public displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 841–850, New York, NY, USA, 2013. ACM.
- [231] Robert Walter, Gilles Bailly, Nina Valkanova, and Jörg Müller. Cuenesics: using mid-air gestures to select items on interactive public displays. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, pages 299–308. ACM, 2014.
- [232] Cheng-Yao Wang, Wei-Chen Chu, Po-Tsung Chiu, Min-Chieh Hsiu, Yih-Harn Chiang, and Mike Y. Chen. Palmtree: Using palms as keyboards for smart glasses. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, page 153–160, New York, NY, USA, 2015. Association for Computing Machinery.
- [233] Yanqing Wang and Christine L. MacKenzie. The role of contextual haptic and visual constraints on object manipulation in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, page 532–539, New York, NY, USA, 2000. Association for Computing Machinery.

- [234] Daryl Weir, Henning Pohl, Simon Rogers, Keith Vertanen, and Per Ola Kristensson. Uncertain text entry on mobile devices. In *Proceedings of the 2014 CHI Conference on Human Factors in Computing Systems - CHI '14*, pages 2307–2316, 2014.
- [235] Wikipedia contributors. Graffiti (palm os) — Wikipedia, the free encyclopedia, 2021. [Online; accessed 18-May-2021].
- [236] Gerard Wilkinson, Ahmed Kharrufa, Jonathan Hook, Bradley Pursglove, Gavin Wood, Hendrik Haeuser, Nils Y. Hammerla, Steve Hodges, and Patrick Olivier. Expressy: Using a wrist-worn inertial measurement unit to add expressiveness to touch-based interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 2832–2844, New York, NY, USA, 2016. ACM.
- [237] Andrew Wilson and Steven Shafer. Xwand: Ui for intelligent spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 545–552, New York, NY, USA, 2003. ACM.
- [238] Andrew D. Wilson. Robust computer vision-based detection of pinching for one and two-handed gesture input. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 255–258, New York, NY, USA, 2006. ACM.
- [239] C. A. Wingrave, B. Williamson, P. D. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott, and J. J. LaViola. The wiimote and beyond: Spatially convenient devices for 3d user interfaces. *IEEE Computer Graphics and Applications*, 30(2):71–85, March 2010.
- [240] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 1083–1092, New York, NY, USA, 2009. Association for Computing Machinery.
- [241] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 159–168, New York, NY, USA, 2007. ACM.
- [242] Pui Chung Wong and Kening Zhu. FingerT9: Leveraging thumb-to-finger interaction for same-side-hand text entry on smartwatches. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 2018-April, 2018.

- [243] Zheer Xu, Pui Chung Wong, Jun Gong, Te-Yen Wu, Aditya Shekhar Nittala, Xiaojun Bi, Jürgen Steimle, Hongbo Fu, Kening Zhu, and Xing-Dong Yang. Tiptext: Eyes-free text entry on a fingertip keyboard. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, page 883–899, New York, NY, USA, 2019. Association for Computing Machinery.
- [244] Zhican Yang, Chun Yu, Xin Yi, and Yuanchun Shi. Investigating gesture typing for indirect touch. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(3), September 2019.
- [245] Hui-Shyong Yeo, Xiao-Shen Phang, Steven J. Castellucci, Per Ola Kristensson, and Aaron Quigley. Investigating tilt-based gesture keyboard entry for single-handed text entry on large devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 4194–4202, New York, NY, USA, 2017. ACM.
- [246] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. Tap, Dwell or Gesture?: Exploring Head-Based Text Entry Techniques for HMDs. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, pages 4479–4488, 2017.
- [247] Chun Yu, Ke Sun, Mingyuan Zhong, Xincheng Li, Peijun Zhao, and Yuanchun Shi. One-Dimensional Handwriting: Inputting Letters and Words on Smart Glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, pages 71–82, 2016.
- [248] Johannes Zagermann, Ulrike Pfeil, Daniel Fink, Philipp von Bauer, and Harald Reiterer. Memory in motion: The influence of gesture- and touch-based input modalities on spatial memory. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1899–1910, New York, NY, USA, 2017. ACM.
- [249] Shumin Zhai. *Human performance in six degree of freedom input control*. Citeseer, 1995.
- [250] Shumin Zhai and Per-Ola Kristensson. Shorthand writing on stylus keyboard. *Proceedings of the conference on Human factors in computing systems - CHI '03*, (5), 2003.
- [251] Shumin Zhai and Per Ola Kristensson. The Word-Gesture Keyboard: Reimagining Keyboard Interaction. *Communications of the ACM*, 55(9), 2012.

- [252] Cheng Zhang, Anhong Guo, Dingtian Zhang, Caleb Southern, Rosa Arriaga, and Gregory Abowd. Beyondtouch: Extending the input language with built-in sensors on commodity smartphones. In *Proceedings of the 20th International Conference on Intelligent User Interfaces, IUI '15*, pages 67–77, New York, NY, USA, 2015. ACM.
- [253] Shengdong Zhao, Maneesh Agrawala, and Ken Hinckley. Zone and polygon menus: Using relative position to increase the breadth of multi-stroke marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '06*, pages 1077–1086, New York, NY, USA, 2006. ACM.
- [254] Shengdong Zhao and Ravin Balakrishnan. Simple vs. compound mark hierarchical marking menus. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology, UIST '04*, pages 33–42, New York, NY, USA, 2004. ACM.
- [255] Jingjie Zheng, Xiaojun Bi, Kun Li, Yang Li, and Shumin Zhai. M3 gesture menu: Design and experimental analyses of marking menus for touchscreen mobile interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18*, pages 249:1–249:14, New York, NY, USA, 2018. ACM.
- [256] Mingyuan Zhong, Chun Yu, Qian Wang, Xuhai Xu, and Yuanchun Shi. ForceBoard: Subtle Text Entry Leveraging Pressure. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 2018.
- [257] Suwen Zhu, Jingjie Zheng, Shumin Zhai, and Xiaojun Bi. I'sFree: Eyes-free gesture typing via a touch-enabled remote control. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, 2019.