Gesture Learning in Human Computer Interaction

by

Fraser Scott Anderson

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science
University of Alberta

Abstract

After decades of research, gestural interfaces are becoming increasingly commonplace in our interactions with modern devices. They promise natural and efficient interaction, but suffer from a lack of affordances and thus require learning on the part of the user.

This thesis examines the declarative and procedural components of learning gestural interaction, and how designers can best support gesture learning within their interfaces. First, we show that user-defined gestures are not always consistent, even when the same user is defining a gesture for the same task, indicating that even when the user is able to select their own gestures some amount of gesture learning still may be necessary. Next, we present two studies that help us better understand the role of visual feedback, finding that it has a dramatic effect on the degree to which gestures are learned. Next, we examine the procedural component of gesture learning by varying the scale, location, and animation of visual feedback presented during training. We also show that evaluation using a retention and transfer paradigm is more appropriate for evaluating gestures than the other methodologies used previously. Lastly, we present YouMove, a full-body gesture training system that incorporates the lessons learned from the present work on stroke-based gestures.

Preface

This thesis is an original work by Fraser Anderson. The research projects, of which this thesis is a part, received research ethics approval from the University of Alberta Research Ethics Board, Pro00027438: "Design of Interfaces for Gesture Learning", 2012-2014, and Pro00033336: "Evaluation of Human-Computer Interaction Devices and Techniques", 2012-2014.

The research presented in Chapter 2 and 3 was performed in collaboration with Peter Lenkic, David Wu, and Dr. Alan Kingstone at the University of British Columbia, as well as my supervisor, Dr. Walter Bischof. We collaboratively designed the studies, I wrote the experimental software, and they collected the data. I also performed the data analysis and wrote the manuscript.

The research in Chapter 4 was performed under the supervision of Dr. Walter Bischof, and has been published as Anderson, F and Bischof, W. F., "Learning and Performance with Gesture Guides" in Proceedings of the ACM Conference on Human Factors in Computing Systems, 2013, pp. 1109-1118. I designed and ran the user study, analyzed the results, and wrote the manuscript under the guidance of Dr. Bischof.

The research presented in Chapter 5 was in collaboration with Tovi Grossman, Justin Matejka, and George Fitzmaurice while I was an intern at Autodesk Research. The majority of this chapter has been published as Anderson, F., Grossman, T., Matejka, J., and Fitzmaurice, G. "YouMove: Enhancing Movement Training with an Augmented Reality Mirror." In the Proceedings of User Interfaces and Software Technology, 2013, pp. 311-320. They provided guidance throughout the course of the project, and I implemented the system, conducted the user study, analyzed the data, and wrote the manuscript.

Acknowledgements

Walter. You taught me how to science and write English goodly. You were always there to support me and provide last-minute manuscript corrections (even in my final hours of thesis-ing), and supply the lab with toys and other widgets. More importantly, you showed me that research is fun. You are caring, smart and encouraging - an excellent mentor that I will strive to emulate. I will always be your student and will bother you for grammar and statistics corrections for many more years.

Justin, Tovi and George, you made coming in to 'work' fun. You made the 'big ass mirror' possible, and made it publishable. You also took me back for a second time (you silly geese) and gave me an opportunity to work on ridiculous things. I hope we can continue to work together on increasingly ridiculous projects.

Alan, I appreciate you taking me into your lab and letting me weasel my way into a variety of projects. You are a brilliant scientist, and I will always have a latent fear of presenting you with data that I think I understand. David and Peter, you are both incredibly clever, and this thesis wouldn't be done without your help.

Ken, Bill, and Michel, you encouraged me to try new ideas, and adapt existing ones. The guidance you provided and the MSR experience will stay with me forever.

Pierre, Brian, Christine, Craig, and Ehud, you challenged me to think about gestures differently, shifted my point of view, and offered insight I couldn't find anywhere else. You helped me better frame and balance the science and engineering aspects of research.

Mom and Dad, you've always been there for me and put me first. You are supportive and put everyone before yourselves. You're constant role models for what it means to be a good person.

Doreen and Dan, sorry … Mr. and Mrs. Annett, you've taken me in and provided me with a home away from home. I truly appreciate everything you've done for me (and will likely continue to do for me, despite my protests).

Michelle: You're alright, I guess.

I thought about leaving it like that, but then also thought you might hit me a bunch for years to come if I did. I wouldn't be me without you. You pushed me, worked harder than me, opened doors, read my garbage (some of it was good though), encouraged my stupid ideas and put up with my perpetual nonsense. We've built stuff and broke stuff, discovered new things (and disappointingly discovered old things that we thought were new), and dealt with long (and very short) distance relationships for years at a time. We can talk about anything/everything together, and help each other with all of our projects. I'm not sure what will happen in the future, but if we're together I figure it'll all be ok. Everything is cool when you're part of a team!

Table of Contents

CHAPTER 1 INTRODUCTION 1

1.1. BACKGROUND 2

1.1.1. EFFICIENCY BENEFITS 2

1.1.2. TYPES OF GESTURES 3

1.1.3. NECESSITY OF GESTURE LEARNING 5

1.2. THESIS OBJECTIVES 7

1.3. A FRAMEWORK FOR GESTURE LEARNING 7

1.3.1. LEVERAGING EXISTING KNOWLEDGE 8

1.3.2. SUPPORT FOR DECLARATIVE LEARNING 9

1.3.3. SUPPORT FOR PROCEDURAL LEARNING 10

1.4. THESIS ORGANIZATION 11

CHAPTER 2 RELATED WORK 13

2.1. TYPES OF MEMORY 13

2.1.1. SHORT AND LONG TERM MEMORY 13

2.1.2. PROCEDURAL AND IMPLICIT MEMORY 14

2.1.3. DECLARATIVE AND EXPLICIT MEMORY 14

2.2. FACTORS IN LEARNING 15

2.2.1. FACTORS IN LEARNING PROCEDURAL MEMORIES 15

2.2.2. FACTORS IN LEARNING DECLARATIVE MEMORIES 16

2.3. MEASURING LEARNING 18

2.3.1. MEASURING LEARNING FOR PROCEDURAL MEMORIES 18

2.3.2. MEASURING LEARNING FOR DECLARATIVE MEMORIES 19

2.4. LEARNING AND MEMORY IN GESTURAL INTERACTION 20

2.4.1. APPROACHES TO TRAINING 20

2.5. EVALUATION METHODS 23

CHAPTER 3 SELF-CONSISTENCY OF USER-DEFINED GESTURES 25

3.1. User Defined Gestures **25**

3.2. Gesture Passwords **27**

3.3. Experiment 1: Influence of Instruction **28**

3.3.1. Participants 28

3.3.2. Equipment and Application 29

3.3.3. Procedure 29

3.3.4. Measures 30

3.3.5. Results 32

3.4. Experiment 2: Influence of Environment **34**

3.4.1. Participants 35

3.4.2. Equipment and Apparatus 35

3.4.3. Procedure 35

3.4.4. Measures 35

3.4.5. Results 35

3.5. Discussion **38**

3.5.1. Analysis of Gesture Form 40

3.5.2. Implications for Gesture Passwords 42

3.5.3. Implications for Gesture Learning 43

3.5.4. Future Work 43

3.6. Summary **43**

3.6.1. Relevance to Gesture Learning 44

3.6.2. Limitations 44

CHAPTER 4 THE COGNITIVE ADVANTAGE OF GESTURES **45**

4.1. Learning the Command-Action Mapping **45**

4.2. Experiment 1: Comparing Gestures and Pointing **46**

4.2.1. Participants 47

4.2.2. Equipment and Apparatus 47

4.2.3. Procedure 47

4.2.4. Measures 50

4.2.5. Results 50

4.2.6. Discussion 51
4.3. Experiment Two: Effects of Visual and Motor Components 52
4.3.1. Participants 52
4.3.2. Equipment and Apparatus 52
4.3.3. Procedure 53
4.3.4. Measures 54
4.3.5. Results 54
4.3.6. Discussion 57
4.4. Cross-Experiment Comparisons 58
4.4.1. Stroking 58
4.4.2. Pointing 59
4.4.3. Discussion 59
4.5. Summary 60
4.5.1. Relevance to Gesture Learning 61
4.5.2. Limitations 61

CHAPTER 5 LEARNING AND PERFORMANCE WITH GESTURE GUIDES 62

5.1. Learning with Gesture Guides 62
5.2. Gesture Guide Experiment 63
5.3. Methods 63
5.3.1. Participants 63
5.3.2. Apparatus and Gestures 63
5.3.3. Guide Types 64
5.3.4. Procedure 66
5.4. Results 68
5.4.1. Gesture Similarity 68
5.4.2. Duration 71
5.5. Discussion 73
5.5.1. Guide Design 73
5.5.2. Evaluation Paradigm 76
5.6. Summary 77

5.6.1. Relevance to Gesture Learning 77
5.6.2. Limitations 78

CHAPTER 6 FULL BODY GESTURE TRAINING WITH AN AUGMENTED REALITY MIRROR 79

6.1. Learning Full Body Gestures 79
6.2. Insights on Traditional Instruction 81
6.3. Design Guidelines 81
6.3.1. Leverage Domain Knowledge 81
6.3.2. Motivate the User 81
6.3.3. Simple Presentation, Low Cognitive Load 82
6.3.4. Adaptive Guidance 82
6.3.5. Average Feedback 82
6.3.6. User Driven learning 82
6.4. Augmented Reality Mirror Implementation 82
6.4.1. Configuration 83
6.4.2. Lighting 83
6.4.3. Interaction 84
6.5. YouMove Implementation 84
6.5.1. YouMove Authoring System 85
6.5.2. YouMove Training System 86
6.5.3. Demonstration 89
6.5.4. Post-Stage Feedback 92
6.6. Evaluation 93
6.6.1. Participants 93
6.6.2. Study Movements 93
6.6.3. Conditions 94
6.6.4. Design 94
6.6.5. Procedure 94
6.6.6. Results 95
6.6.7. Stage Usage Analysis 96
6.7. Discussion 97

6.7.1. ANALYSIS OF LEARNING 97
6.7.2. MIRROR INTERACTION 98
6.7.3. QUERY BY EXAMPLE 98
6.7.4. AUTHORING 98
6.7.5. TRAINING STAGES 99
6.8. FUTURE WORK 99
6.9. SUMMARY 99
6.9.1. RELEVANCE TO GESTURE LEARNING 100
6.9.2. LIMITATIONS 100

CHAPTER 7 THESIS CONCLUSIONS 101

7.1. SUMMARY 101
7.2. CONTRIBUTIONS 102
7.3. FUTURE WORK 103
7.3.1. UNDERSTANDING GESTURE LEARNING 103
7.3.2. APPLICATIONS 104

BIBLIOGRAPHY 106

List of Tables

Table 5.1. ANOVA results for Accuracy.......... 70

List of Figures

Figure 1.1. a) Example of a marking menu, in which users implicitly learn gestures associated with menu items Kurthenbach & Buxton (1993). b) Scriboli gestural interface in which selection and action are combined into a single, fluid movement. Images from Hinckley, Baudisch, & Ramos (2005). 3

Figure 1.2. Examples of the Graffiti (top) and EdgeWrite (bottom), in which each symbolic gesture bears a resemblance to the letter it represents. Images from Castellucci and Mackenzie (2008) and Wobbrock, Myers, & Kembel (2003). 5

Figure 1.3. Examples of abstract gestures in which each gesture has a recognizable form, but does not correspond to a particular stimulus or action. Image from Zhai et al. (2010). 5

Figure 1.4: Framing of gestural interaction which outline the effects of pre-existing knowledge and the various mechanisms that provide support for procedural and declarative learning. The combined effect of these components manifests itself in users' recall and execution of gestures. 8

Figure 2.1. A) Swype keyboard, in which users trace over the keys using a single fluid gesture. With enough iterations, users should implicitly learn the motor patterns required for frequent words. Image from wirelesszone.com. B) Octopocus is a dynamic guide that allows the user to directly trace out gestures and updates dynamically as the user completes the stroke. As the cursor moves from the position on the left to that on the right, the guide is updated to reflect the improbability of the 'paste' command being the target action. Image from Bau & Mackay (2008). 21

Figure 2.2. A) Example of a type of crib note, in which each gesture is depicted next to the corresponding command. Image from Brandl et al. (2008) B) Gestural interface providing interactive help in which users can retrieve hints and practice gestures in the menu bar of the program. Image from Bragdon et al. (2010). 22

Figure 3.1. Experimental setup with the touch-screen placed vertically (left) and horizontally (right) in front of the participant. Note that the horizontal condition was only used in Experiment 2. 29

Figure 3.2. Mean gesture length by instruction. Error bars represent the standard error of the mean. 32

Figure 3.3. Frequency of starting location for each instruction. A. Easy; b. Hard to guess; c. Hard to steal. 33

Figure 3.4. Gesture similarity by instruction and comparison-type. Note the increased similarity found within the 'easy' condition, and that self-similarity is consistently higher than group-similarity. Error bars represent the standard error of the mean. 34

Figure 3.5. Length of resulting gestures across the four conditions. Error bars represent the standard error of the mean. 36

Figure 3.6. Frequency of each starting location by instruction, collapsed across orientation. a) Easy, and b) Hard. 36

Figure 3.7. Frequency of each starting location by orientation, collapsed across instruction. a) Horizontal and b) Vertical. 37

Figure 3.8. Similarity (self and group) between the gestures created under all four conditions. Note the consistently high similarity in the easy condition and the higher similarity in the horizontal conditions. Error bars represent the standard error of the mean. 38

Figure 3.9. Variations used in constructing hard to steal gestures: a) Crossovers, b) Repetition of points, and c). Minimization of space. 39

Figure 3.10. Frequency of use of gestures. Note that the first bar is truncated for clarity, the true value is 708, indicating that 708 gestures were totally unique and generated only once across both experiments. 41

Figure 3.11. Most popular gestures generated by participants, with the top left being the most frequently used gesture, with frequency decreasing to the right, then down. 42

Figure 4.1. Experimental setup showing a participant learning a sequence in the pointing condition. The dots illuminated in sequence (in the top portion of the screen) and participants would touch them in the same sequence on the bottom portion of the screen. 48

Figure 4.2. (Left) The learning phase in the stroking condition displayed lines connecting the dots in the sequence during the demonstration as well as during user input. (Right) The screen presented to participants during the recognition task required participants to hold the bottom-most button until the probe gesture appeared. Then participants quickly touched the Yes or No button to indicate their response. 48

Figure 4.3. Response time by condition. Error bars show standard error of the mean. 50

Figure 4.4. Accuracy by condition. Error bars show standard error of the mean. 51

Figure 4.5. Efficiency by condition. Error bars show standard error of the mean. 52

Figure 4.6. Participant demonstrating the learning phase during the pointing condition just prior to touching the final target. 53

Figure 4.7. Response time for stroking and pointing, by length of sequence. Error bars represent standard error of the mean. 55

Figure 4.8. Accuracy for stroking and pointing, by length of sequence. Error bars represent standard error of the mean. 56

Figure 4.9. Efficiency for stroking and pointing, by length of sequence. Error bars represent standard error of the mean. 56

Figure 4.10. Efficiency for stroking and pointing, by length of sequence. Error bars represent standard error of the mean. 57

Figure 5.1. The behavior of the guides while performing the Send gesture during training trials for a) crib-notes, b) static-tracing, and c) dynamic-tracing. Note that adaptive guide is not shown as its behavior is identical to the 'static-tracing' guide, except the guide is removed partway through the trial. 65

Figure 5.2. Inter-trial screen showing KR (score in top right) and KP (trajectory overlay). 67

Figure 5.3. Error for training, retention and transfer, for each guide type. There is an apparent tradeoff between performance during training, and the amount of learning, measured by retention and transfer. 69

Figure 5.4. Error for training, retention, and transfer, for each gesture. There are no appreciable differences between the performances of each gesture. Error is primarily due to the type of training guide used. 69

Figure 5.5. Duration of stroke during training, retention, and transfer, for each guide type. Guide type has little effect on the speed, allowing the tracing-based guides to provide high accuracy without a decrease in input speed. 72

Figure 5.6. Duration of stroke during training, retention, and transfer, for each gesture. The 'Send' and 'Find' gestures are performed slightly faster than 'Build' and 'Choose'. 72

Figure 6.1. YouMove allows users to record and learn physical movement sequences. An augmented reality mirror provides graphic overlays for guidance and feedback. Note that for this photo the virtual viewpoint was vertically repositioned to account for the offset of a head-mounted camera, and floor lighting was used to reduce glare. 80

Figure 6.2. Overview of system design showing projector, layered screen, dynamic lighting, Kinect and user location. 83

Figure 6.3. The editing interface allows authors to specify keyframes and global movement parameters. Each keyframe specifies the important joints for that moment, and can be associated with a recording that provides detail. 86

Figure 6.4. The movement gallery allows users to change profiles, query by example, and select a movement. 87

Figure 6.5. Left) Stage selection interface allowing users to begin one of the unlocked stages. Right) Demonstration stage. 89

Figure 6.6. The posture guide requires trainees to maintain a stable posture, matching the position of the trainer. Errors in joint position are indicated by red circles. The callout (top right of figure) resolves depth ambiguities. 90

Figure 6.7. The movement guide encourages the trainee to match the trainer's movements at full speed, using green ribbons (inset) to cue upcoming movements. 91

Figure 6.8. The Post-Stage Feedback screen, showing trainees their overall score, average position (skeleton), and video for each keyframe. 93

Figure 6.9. Performance on each trial, for each of the conditions, averaged over all 8 subjects. 95

Figure 6.10. Pre-test, training, and post-test scores for the YouMove and Video conditions. 96

Figure 6.11. Frequency of use for each guide type for all participants during training with the YouMove system. 97

Chapter 1

Introduction

Mobile devices and interactive surfaces are the primary driver of recent developments in gestural interfaces. Mobile phones have become ubiquitous, with approximately six billion mobile phones in use worldwide, one billion of which are smart phones (Kafka, 2012). These devices have the processing capabilities of general-purpose computers, but their input is often constrained to 2D finger input on a touch screen. Tablets, touch screens, and interactive surfaces have also found increased adoption among consumers. Many of these devices have severely constrained input, sensed only through touch or pen-based interaction with the display surface. For these devices, gestures offer a promising alternative to traditional input with on-screen buttons and widgets. They allow users to provide input to the device without having to select a number of small on-screen targets, navigate through hierarchies of menus, or interact directly with the on-screen content.

Despite their potential for use on an increasing number of devices, gestures have remained relatively primitive. This is due, in part, to the difficulty with gestural interfaces lacking clear affordances, and their largely hidden functionality. The gesture-action mapping that is necessary for interacting with an application is often hidden and users must expend considerable effort to learn which gestures are available as well as learning how to perform the gestures. There are several existing research efforts that address various aspects of this problem (e.g., making them more approachable (Bragdon et al., 2010), or making it more convenient to access the guide (Bau and MacKay, 2009)), but there lacks a systematic analysis of gesture learning and an identification of the various components that affect how well a user learns the gestures.

This thesis examines how users learn gestures, and how we can best support that learning with the design of our interfaces. The thesis contributes a novel framework which identifies the factors that designers can leverage in their interfaces to enhance users' gesture recall and execution. Specifically, it identifies the user's pre-existing knowledge, the interface's support for declarative learning, and the interface's support for procedural learning as being central to affecting gesture recall and execution. This thesis samples problems from each of these three top-level components and attempts to provide answers to open questions. With respect to pre-existing knowledge, the thesis examines how reliable users' self-defined gestures are when the context of use changes. Next, the thesis examines the declarative component of gesture learning by analyzing the cognitive advantage that gestures have when encoding sequences and investigating whether that advantage is due to a visual or motor process. With regards to procedural learning, the thesis examines appropriate ways to train and evaluate gestural guides to ensure that the user maximizes learning. Lastly, the thesis presents a full-body movement training system that extends what is known about two dimensional stroke gestures to a more complex scenario to understand the generalizability of the presented principles.

1.1. Background

1.1.1. Efficiency Benefits

Gestural interaction also offers efficiency benefits over other input modalities. Marking menus, for example, allow users to execute commands using the physical actions associated with accessing menus, without visually searching for the target items (Kurtenbach, Sellen, & Buxton, 1993; Kurtenbach, Moran, & Buxton, 1994; Kurtenbach & Buxton, 1993; Figure 1a). With proper design, gestural interfaces allow for chunking and phrasing (Buxton, 1986), which provides cognitive benefits and increased input bandwidth. One such system is FlowMenu (Guimbretiere & Winograd, 2000), which allows for the simultaneous specification of command and parameter. Scriboli implements chunking and phrasing by allowing selection and action to be specified using a single, fluid movement (Hinckley, Baudisch, & Ramos, 2005; Figure 1b). SimpleFlow pushes efficiency benefits even further and allows users to input partial gestural commands, enabling the system to 'auto-complete' a gesture when it has been sufficiently distinguished (Bennett et al., 2011). While these systems have not matured into widespread commercial offerings, they demonstrate the potential for effective gestural input.

The efficiency benefits provided by gestures and the widespread use of interactive displays have driven the development of gesture interfaces for a wide variety of tasks. Text entry, for example, can be accomplished using Graffiti (Fleetwood et al., 2002), Unistroke (Mackensize & Soukoreff, 2002), EdgeWrite (Wobbrock, Morris, & Wilson, 2003) or SHARK (Zhai & Kristensson, 2003) gestures. The Android and Windows 8 operating systems provide support for gesture-based passwords for fast and safe logins (Microsoft, 2012; Niu & Chen, 2012). Mozilla Firefox also has several add-ons that allow users to navigate webpages using mouse gestures (Gomita, 2012). Wigdor et al.'s (2011) Rock and Rails multi-touch gestures enable precise and efficient manipulation of content on large interactive surfaces. Gestural interaction can also be used to navigate interfaces in video games and entertainment systems (Segen, 1998). Autodesk Maya and 3DSmax use gesture shortcuts to allow designers to change tools, navigate, and select options quickly (Autodesk, 2014; Kurtenbach, 1993). Several diagram editors have also been developed with gestural support to allow natural specification of visual elements (Sutherland, 1964; Zeleznik et al., 2008).

a) b)

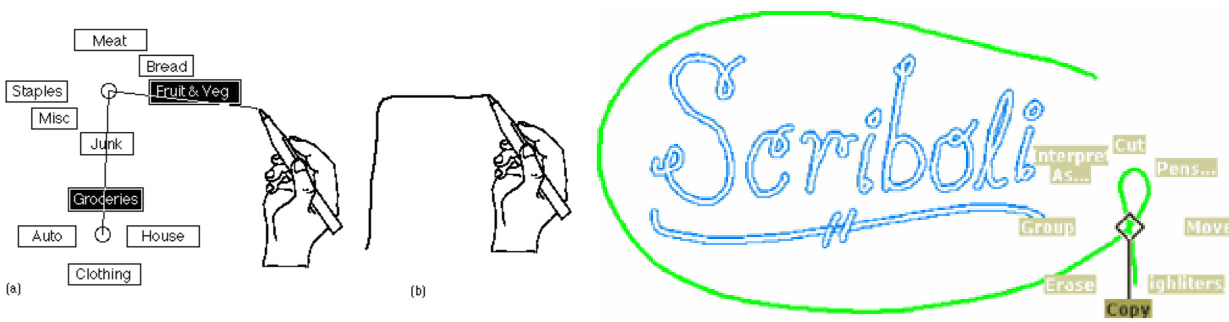

Figure 1.1. a) Example of a marking menu, in which users implicitly learn gestures associated with menu items Kurthenbach & Buxton (1993). b) Scriboli gestural interface in which selection and action are combined into a single, fluid movement. Images from Hinckley, Baudisch, & Ramos (2005).

1.1.2. Types of Gestures

The term ‘gesture’ is very broad, describing many interactions with devices today. Some designers consider small unit operations like “tap” or “press and hold” to be gestures. However, these actions are of little interest as they are simple to perform but have very low input bandwidth. More complex gestures, such as stroke gestures or 3D free-space gestures are able to convey much more information, but are not as user-friendly as the more primitive gestures.

Stroke gestures have attracted substantial attention in both research and commercial scenarios. Such gestures are composed of a single contact event (e.g., a finger or pen contact), movement of that point in 2D space, and are terminated when the finger or pen is lifted from the surface. These gestures are particularly interesting, as many properties (gesture form, user interface support, etc.) generalize across input modalities (e.g., pen, mouse, touch; Tu, Ren, & Zhai, 2012), and they are a very expressive method for specifying input on touchscreen devices.

Many of the more complex modes of gestural interaction (e.g., 3D free-space, multi-touch, etc.) have gained acceptance due to the development of new sensing hardware and the emergence of new applications for interactive technology. These gestures allow interaction with devices from a distance, provide high input bandwidth, and allow for more natural methods to specify actions. Many of the more complex gestures share similarities with complex movements found in everyday life, such as dancing or sports. As such, much of the knowledge learned from those domains can benefit gestural interaction, just as developments in gestural training have the potential to benefit those domains in return.

The composition of a gesture set varies widely depending on the task. Many gesture sets are designed to be symbolic analogues to the actions or items they are mapped to. For instance, many text entry gestures resemble corresponding letters (Figure 1.2; Fleetwood et al., 2002; Wobbrock, Myers, & Kembel, 2003). Other applications use gestures that resemble the first letter of the intended action, e.g., an 'S' shape to create a String object (Zhai et al., 1995; Chatty & Lecoanet, 1996). However, the number of possible commands quickly exceeds the available letter-based gestures, restricting the adoption of this technique. Other symbolic gestures include those that exploit prior knowledge or those are an iconic representation of an action, such as a scribbling gesture to delete or using a lasso to select multiple objects (Bragdon et al., 2008). The use of symbolic gestures is also limiting, as users often disagree on what gesture is representative of a given action (Wobbrock, Morris, & Wilson, 2003), especially when it comes to more abstract actions such as 'Insert Phrase' (Wolf & Morrel-Samuels, 1987). To avoid these issues, many systems and researchers use abstract gestures (Figure 1.3) with no obvious symbolic mapping (Bau & Mackay, 2008; Freeman et al., 2009). While this removes any bias users may have to a particular gesture, each study tends to devise their own set of gestures, making it difficult to compare results across studies.

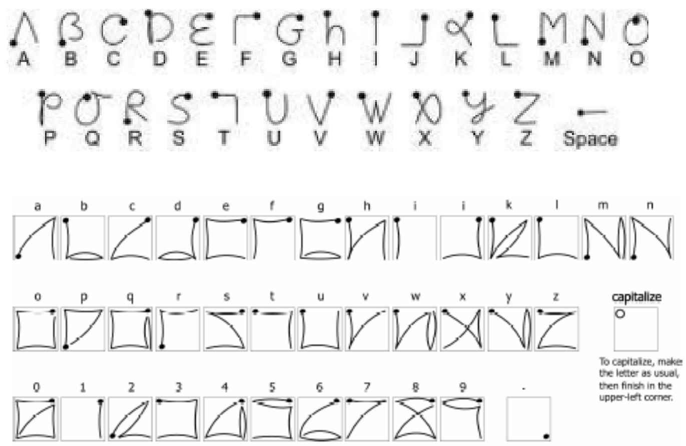

Figure 1.2. Examples of the Graffiti (top) and EdgeWrite (bottom), in which each symbolic gesture bears a resemblance to the letter it represents. Images from Castellucci and Mackenzie (2008) and Wobbrock, Myers, & Kembel (2003).

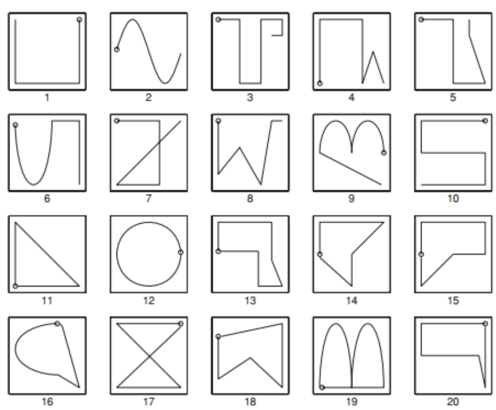

Figure 1.3. Examples of abstract gestures in which each gesture has a recognizable form, but does not correspond to a particular stimulus or action. Image from Zhai et al. (2010).

1.1.3. NECESSITY OF GESTURE LEARNING

The difficulty with gestural interfaces, and one of the primary reasons behind their slow adoption, is that gestural interfaces are not 'self-revealing' (Baudel & Beaudouin-Lafon, 1993; Bragdon et al., 2008). Users are required to learn and practice each gesture to become efficient enough to use it in place of other input. This problem has yet to be solved, and many current gestural interfaces rely on a small set of simple gestures (i.e., swipes, taps, and pinches) to avoid the problem of learning gestures. With these interactions, it is sufficient to write instructions for end-users in the manual or on screen (e.g., 'swipe to unlock') as users will be able to execute the simple actions with little training. To achieve effective interaction with gestural devices, however, such simple interactions are not sufficient. A rich gesture vocabulary requires gesture languages that must be learned by users, either implicitly or explicitly (Norman, 2010).

Learnability of gesture sets involves two factors. The first is the cognitive mapping between the desired task or command and the required gesture. This declarative component of learning is typically studied in human computer interaction (HCI)-focused research. It is easy to measure with recall tasks (using proportion correct) and it is intuitively important (users must know which gesture to execute before they perform it). The second, equally important aspect of gestural interactions is the procedural component of gesture learning, which involves the ability to perform a gesture accurately. Bau and MacKay (2008) recognize the importance of gesture execution, stating that users must "master the details of drawing the shape to improve recognizer accuracy". This component of gestural interaction becomes increasingly important as the use of gestural interfaces continues to grow and devices rely solely on gestural input. In the case of experts, many of their input sequences are largely automatic, relying primarily on responses from the motor system. Motor performance is important for novices as well. As the size of gesture sets is increasing (e.g., to 40 targets (Ouyang & Li, 2012)), both novices and experts have to perform gestures with increasing accuracy for the recognizer to distinguish them from other, potentially similar gestures. It is also foreseeable that future interfaces will allow users to modify parameters of commands by producing variations on gestures, which again would require substantial skill to perform.

Recently, several researchers have proposed that users should be able to define their own gestures for interaction rather than using a designer-defined set (Nacenta et al., 2013). Studies have shown that there can be high agreement on the gesture-to-action mapping between users, especially for actions that are more concrete (Wobbrock et al., 2009). Other systems have leveraged crowd-based definitions of gestures, enabling users to input gestures without defining them, relying on the similarity of their gesture to other users' gestures to determine the intended action (Ouyang & Li, 2012). While these approaches offer learning-free gestural input, it is not clear whether they scale to more abstract actions (Ruiz, Li & Lank, 2011). Additionally, no studies have examined the *self-consistency* of users' choice of gesture under different conditions, that is, the degree to which the same user generates the same gesture for the same task. If users vary their chosen gesture, then they may need support for learning the appropriate gesture to use within the given context.

1.2. Thesis Objectives

Though there is existing research that analyzes various aspects of the learnability of gestures, there is no focused contribution that identifies the constituent elements that affect a users' ability to learn and perform gestures. Thus, this thesis seeks to provide a novel framework for gestural interaction, as well as work towards answers to several important questions within gestural interaction. In the subsequent chapters, the following questions are addressed:

Chapter 3: To what degree is gesture learning necessary? Can gesture learning be avoided by implementing user-defined gestures?

Chapter 4: Does gesturing offer a learning advantage over traditional input methods? If so, are these advantages due to the motor or visual component of gesturing?

Chapter 5: How can users be trained to gesture efficiently, and how should we evaluate such learning?

Chapter 6: How well does knowledge of 2D stroke gestures extend to movement scenarios that are more complex?

1.3. A Framework For Gesture Learning

Although there has been much focus and attention devoted to the learning of gestures, and many novel techniques have been developed to aid in gesture learning, there has yet to be a clear understanding of how and when to support the learning of gestural interaction. This thesis presents an examination of gesture learning, detailing the factors that affect gesture learning. While some of these factors have been leveraged previously in gestural interaction, we identify many new factors and provide a categorization from which other work can build upon. Thus, we make the novel contribution of the gesture-learning framework, depicted in Figure 1.4.

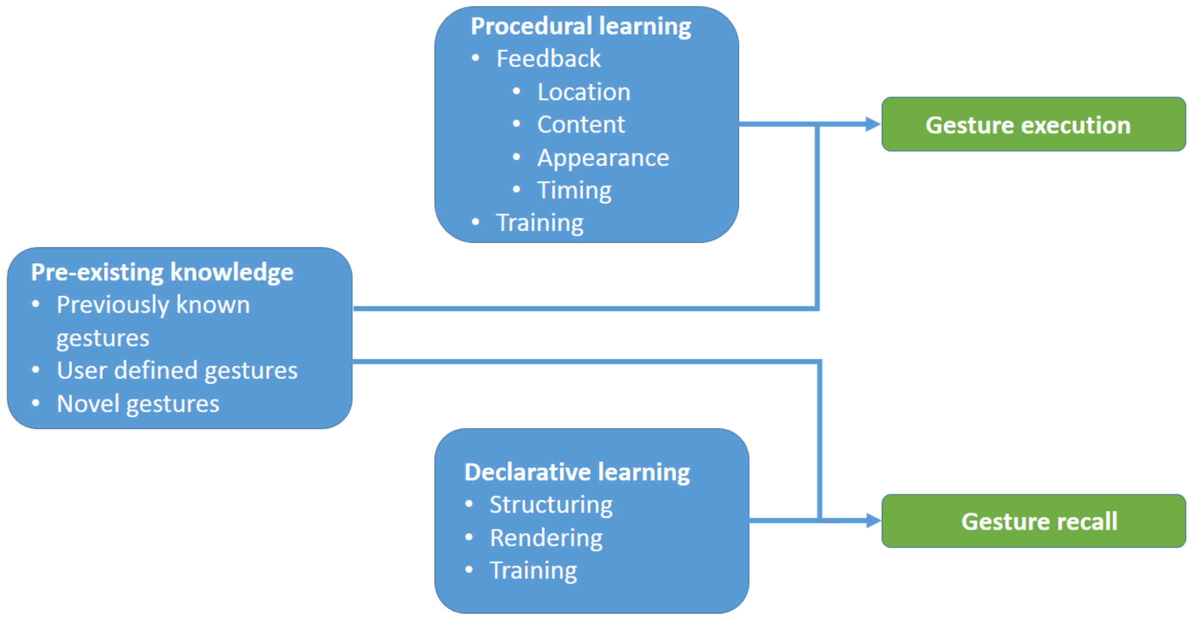

Figure 1.4: Framing of gestural interaction which outline the effects of pre-existing knowledge and the various mechanisms that provide support for procedural and declarative learning. The combined effect of these components manifests itself in users' recall and execution of gestures.

If gestures are supported within the interface, then the gestures must be selected in a way that minimizes their need for learning by *leveraging existing knowledge*. If a gesture set must be learned, then appropriate support for learning the *declarative component* of the gesture (i.e., the recall of the correct gesture) as well as support for learning the *procedural component* of the gesture (i.e., the articulation of the gesture) must be provided. Figure 1.4 illustrates how this framework relates to the resultant recall and execution of the gestures within the user interface.

1.3.1. Leveraging Existing Knowledge

Once gestural interaction has been identified as an input mechanism, the designer must consider the form of the gestures themselves and how they can be designed to minimize the need for learning. The degree to which gestural support is required is dependent on whether gestures are *previously known, user-defined*, or completely *novel*.

Some gestures may be *previously known* and thus require little or no training. A select group of gestures is culturally engrained through marketing campaigns and product dominance such that they have become widely known (e.g., swipes). Alternatively, gestures may be mapped to physical affordances (e.g., pinches, or rotations); in these instances, little learning support is likely required as users rely on known metaphors.

If the functionality of the interface is primarily rooted in operations that are concrete in nature, then user-defined gestures may provide users with the ability to choose memorable gestures that do not require extensive training to recall and perform. Prior work has shown that user-defined gestures are easier to remember (Nacenta et al. 2013), and may thus require minimal training. It is still unknown, however, if users consistently choose the same gestures for the same actions performed in different contexts. If this is the case, then the amount of training required may be more than previously expected.

If a gesture-based user interface has a large number of functions, operations based on abstract commands, or a high degree of command parameterization, then it is likely that the system will need to leverage *novel* gestures. In this case, it will need to provide infrastructure for instructing users on the proper selection and execution of the gestures. When possible, completely novel gesture sets should be avoided. If, however, they must be used, a number of strategies (as outlined in the following sections) can help ease learning and reduce the burden on the user.

It is also important to consider what aspects of the gesture the user may already know. The declarative component may be well known, for example, if you are using gestures that represent alphanumeric characters. In this case, the user may still need to learn how to execute the gesture accurately enough for the system to recognize it.

When deciding on the degree and form of the gesture learning support it is important to consider the resulting usability of your system. If the user must undergo extensive training before using the system then they may be discouraged from using the product. If too little support is provided then users may struggle to achieve proficiency and can become frustrated during interaction. An ideal support system would scaffold novice users, allowing them to focus on their primary task while simultaneously implicitly teaching them the declarative and procedural components of their gestural interactions.

1.3.2. SUPPORT FOR DECLARATIVE LEARNING

Interfaces may support the declarative component of gesture learning by *structuring* the gestures effectively, modifying the *rendering* of gestures, or by providing *explicit training*. As gestural interfaces do not have visible affordances, users must be informed of the available gesture set in some other manner.

Consideration for learning the declarative component could also be achieved through careful *structuring* of the gestures themselves (e.g., the hierarchical structure of Autodesk's Maya software (2014)). By structuring the menu in a logical manner and grouping related items, the interface can take advantage of the hierarchy to aid in the recall of actions. Other structures can be possible depending on the nature of the interface, and could potentially rely on abstract categorization or spatial mapping, for example.

Systems may also modify the *rendering* of gestures to make them more unique and easily remembered. This can be achieved by changing the form (e.g., the visual appearance) of the gesture or by rendering the gestures using additional modalities (e.g., haptic, colour mapping, or audio pairing) to provide some of the benefits seen in dual coding studies (Paivio & Kalman, 1973).

Support for the declarative component can also be achieved through an *explicit training* phase (e.g., the training sandbox of Bragdon et al. (2010)). With this approach, users get the benefits of repeated rehearsals without the worry of unintended consequences on their work environment. The training system could be designed to take advantages of many of the factors known to affect learning (e.g., distributing practice, drawing attention to the pairing itself, etc.). Within a 'live' user interface, many of these approaches would not be available, as they would interfere with the operation of the system itself.

1.3.3. Support for Procedural Learning

Interfaces can support the learning of the procedural component of a gesture using appropriate *feedback* or *explicit training*. This is necessary so that users can perform the gesture accurately enough for recognition by the system. For example, with handwriting recognition software, the user invariably knows which letters they intend to convey but their writing is often not precise enough for the system's algorithms to recognize the intended character. This problem is compounded with gestural interfaces, as gestural interfaces become more complex, allowing for a multitude of commands and parameters to be expressed in a single stroke.

The designers of gestural interfaces thus need to provide *feedback* on the users' performance, not only by relaying the recognized action, but also by supplying useful information to improve future performances of the gesture and improve the communication between the user and gesture recognizer. Such feedback could be provided by many mechanisms and at various points throughout the interaction. Before and during the interaction, "feedforward" (Bau and Mackay, 2009) can provide users with a guide that informs them of the correct actions. Following interaction, a system can provide feedback regarding which gesture was recognized. This information allows the user to compare their input to what the system was expecting.

There are many considerations to the type of feedback and guidance provided. The *location*, *content*, *appearance*, and *timing* of the feedback are of prime consideration. Excess feedback can hinder learning, and poorly designed feedback may go unnoticed. The motor learning literature has examined some of these issues, but it is not immediately evident how to adapt their findings to the specific needs of gestural interaction that must also consider usability.

As with declarative learning, users can perform *explicit training* prior to using the interface to improve their ability to perform gestures. With an explicit training phase, the system could leverage methods or modalities of feedback that may be too intrusive to leverage within a live system (e.g., summary feedback after a number of gesture attempts).

1.4. Thesis Organization

This thesis presents several contributions in the area of gestural learning with the goal of better understanding how users learn gestures and how to better enrich their training. With the increase in complexity and the adoption of gestural interfaces for a variety of tasks, it is critical to have methods and systems that scaffold users as they begin to use new gesture-based systems.

Chapter 2 outlines relevant work from the human-computer interaction and motor learning domains to frame our understanding of how movements are learned and the applicability of various learning methods to gestural interaction.

In Chapter 3, we analyze gesture learning within the context of user-defined gestures. The purpose of these studies is to establish whether user-defined gestures may be a viable alternative to learned gestures, mitigating the need for gesture training. With two experiments, the consistency of gesture creation was observed, as high-level tasks and environmental context was manipulated. The studies use gesture-passwords as a testing sandbox and provide insights into the strategies that users employ when defining secure passwords for gesture based authentication on mobile devices. This chapter addresses how pre-existing knowledge can be leveraged in the design of gestural interfaces, and whether or not user-defined gestures are consistent across various contexts.

In Chapter 4, we analyze whether the use of gestural input or traditional 'pointing' input aids in the encoding of information and declarative memory. If gestures are proven to encode information more readily, then it will likely be easier for novice users to learn the association between a gesture and a command than it would be for them to navigate a traditional button-based interface. Following this, we analyze the respective roles of the visual and motor component within gesture learning. Prior work within HCI has typically ignored the distinction, but the respective roles of each modality are important to consider when designing gesture-based user interfaces. This chapter provides insight into how the visual system contributes to the learning of the declarative component of gesture learning.

Chapter 5 explores how the form of the visual feedback used during training impacts the learning of the procedural component of gestural interaction. Using a retention and transfer paradigm from the motor learning literature, three guides from the existing literature, and a novel, adaptive guide were evaluated. The use of the retention and transfer paradigm revealed properties of gesture guides often overlooked in previous works. This chapter analyzes how gestural interfaces can be designed and evaluated to best support the procedural learning of gesture execution outlined in the previous section.

In Chapter 6, we present YouMove, a training system for complex, full-body gestures. By integrating findings from previous studies, the system uses an augmented-reality mirror to overlay visual feedback directly over top of the user's reflection providing an intuitive and natural guide. The system supported a wide variety of movement domains and abstract movements, as well as more concrete movements from the dance domain. YouMove demonstrated the applicability of the results found in Chapters 3, 4, and 5 to more complex, higher-dimensional gestures.

Lastly, we conclude with a review of how the presented work fits within the gesture learning framework, and outline directions for future avenues of research.

Chapter 2

Related Work

Human memory involves a complex set of interconnected processes, the details of which are beyond the scope of this thesis. However, there are several fundamental concepts that are important with respect to gesture learning. Of particular relevance are a basic understanding of long term and working memory, the distinction between procedural and declarative memories, and the factors affecting learning, as well as how learning is evaluated for each of these components. We also examine relevant work within the HCI literature on existing systems and methods that support gesture learning.

2.1. Types of memory

Long term memory can be categorized along a number of dimensions according to the length of the memory (short or long term), and the characteristics of the memory (procedural or declarative, and implicit or explicit).

2.1.1. Short and Long Term Memory

Memory can be logically divided into short term memory (i.e., working memory) and long term memory (i.e., permanent storage). The capacity of working memory is relatively small, with room for approximately seven unique items at a time (Baddeley, 1994). Items in working memory are thus available for less than a minute (Luck & Vogel, 1997). This capacity can however be expanded through the use of chunking (Chase & Simon, 1973), in which distinct, logical objects can be grouped into a single, coherent group. For example, the string 'ogd' is unlikely to have meaning to a person, whereas one can remember the single word 'dog' quite easily. Combining the letters into a logical whole allows for one item, instead of three separate items, being encoded. Chunking can be can be an extremely useful method of learning large amounts of information, but requires that the information have some structure or meaning (Gobet & Simon, 1998).

Baddeley and Hitch (1947), described a useful conceptual model of working memory in 1947. This model consists of a *central executive*, *phonological loop*, and the *visuo-spatial sketchpad*. The central executive is responsible for allocating attention, processing information, and accesses information stored in long-term memory (as well as the other components of working memory).

The phonological loop is the short-term memory store responsible for processing auditory and verbal information, such as speech, music, and the rehearsal of words. The visuo-spatial sketchpad is the component of working memory that is responsible for processing images, color, and spatial information. Within gestural interaction, this type of memory is typically used only when referencing the guide or other learning material (e.g., consulting a crib sheet).

While items in short term memory are typically forgotten after a minute, items in long term memory can be remembered for a much longer period, potentially years (Rohrer et al., 2005). Long term memory acts as a relatively permanent, limitless, store of memory, whereby memories are stored via consolidation. Sleep is believed to play a large role in consolidation, with memories being 'strengthened' during sleep (Stickgold, 2005).

2.1.2. Procedural and Implicit Memory

Remembering a particular sequence of actions, or steps to achieve a goal, is accomplished using procedural memory. In general, procedural (or non-declarative) memories are memories that are difficult to describe (Squire, 1992). The ability to tie shoelaces or ride a bicycle are due to stored procedural memories. Procedural memories that relate to motor movements are of particular interest to gestural interfaces, which require movements that can be somewhat complex and occasionally unnatural.

Many procedural memories are learned implicitly, i.e., memories are generated without the conscious awareness of the learner (Roediger, 1990). This type of memory is believed to operate using an entirely separate process from explicit memory (Cohen et al., 1985). Implicit learning is important to gestural interfaces, as the form of a gesture is rarely the focus of the interaction, yet it must be learned for efficient gestural interaction.

2.1.3. Declarative and Explicit Memory

Remembering the pairing between a desired action and corresponding gesture falls within the scope of declarative memory. That is, it relates to a memory that can be described (Tulving & Markowitsch, 1998). The work verbal associate learning is of most relevance to gesture learning. In this type of learning, participants associate pairs of words, or a word with a corresponding action (Bower, 1970). Uses of declarative memory are common in everyday life, for example, remembering telephone numbers, or remembering that the alphorn and yodeling are icons of Swiss music.

Declarative memories are commonly learned explicitly, i.e., the learner is aware that they are being learned (Berry and Broadbent, 1988). Explicit learning requires active and conscious involvement on the part of the learner. Studying for a test or researching the history of the clock are common examples of explicit learning.

2.2. FACTORS IN LEARNING

2.2.1. FACTORS IN LEARNING PROCEDURAL MEMORIES

Several factors affect the learning and performance of a movement. Unsurprisingly, the amount of practice has been shown to improve learning and performance. For many complex movements it is generally accepted that the practice must be deliberate, and not simply repetition of learned movement (Ericsson, Krampe, & Tesch-Romer, 1993). The distribution of practice over time has also been shown to affect the amount of learning, with increased learning when practice is distributed over time (Donovan & Radosevic, 1999). While these factors are important to skill learning, they are not as relevant to gesture learning as the feedback given via knowledge of results and knowledge of performance.

Wulf and Shea (2004) provide an excellent summary of many of the known effects of augmented feedback. For example, they outline how the delay between performance and feedback can affect learning, what type of feedback (qualitative or quantitative) should be presented based on the user's performance, and how summarizing and aggregating feedback can improve learning. While all of these elements are relevant to gestural learning, a full examination is beyond the scope of this thesis. However, this, along with other work (Wulf and Shea, 2004), provides an excellent review of the various parameters known to influence the learning of movements. A few key factors directly relevant to the work in this thesis are presented next

Knowledge of results (KR) is information regarding the success or failure of a movement. The presence and frequency of KR has substantial consequences on the amount of learning that occurs during practice. When KR is too frequent, it hinders learning, as users become dependent on it to make small 'corrective' movements (Salmoni, Schmidt, & Walter, 1984; Schmidt, 1991). With respect to gestural interaction, KR can detail the gesture form that was recognized, or include the similarity of the performed gesture to other gestures.

Knowledge of performance (KP) is information regarding how the performed movement differed from the target movement. As with KR, the presence and frequency of the target movement can affect the degree to which the movement is learned (Park, Shea, & Wright, 2000). Within the context of gestural interaction, KP can be presented by displaying the user's trajectory with the target trajectory, potentially highlighting discrepancies between the trajectories. Within gesture guides, this type of information has been recognized as important (e.g., the use of 'feedforward' and 'feedback' by Bau & Mackay (2008)), but the existing research from the motor learning literature has been largely ignored.

Though the field of motor learning studied many fundamental issues such as how movements are learned and performed, it is not clear which research is directly applicable to gestural interaction. Research in motor learning typically makes no consideration for the usability of systems, so directly implementing their findings could result in systems that are not user-friendly and unpredictable. In addition, many studies within motor learning used simple, static one-dimensional positioning tasks that do not reflect the complex nature of gestures or the dynamic environment that modern devices support.

2.2.2. Factors in Learning Declarative Memories

Several factors determine the degree to which something is learned and remembered. As with procedural learning, repetition plays a large role, with more repetitions aiding in learning. Similarly, the structure of practice has an impact, with distributed practice resulting in better learning than massed practice (Pashler et al., 2007). These two factors alone do not regulate the degree to which items are learned; when designing gestural interactions, there are several factors that affect the rate of learning. It is therefore important to understand the potential impacts these factors have on the usability and learnability of gestural interfaces.

One factor that influences the memorability of items is the degree to which items are elaborated on (Cohen & Aphek, 1980). Rather than simply rehearsing each item, elaboration involves constructing mental associations between the new item and existing knowledge. By situating new information within a person's existing mental schema, information becomes more strongly encoded and memorable. Related to the idea of elaboration is Craik and Lockhart's (1972) levels-of-processing effect. Craik and Lockhart observed that items that were processed superficially (e.g., based on their sensory components) were not remembered as well as when more semantic processing was involved (e.g., when participants thought about the meaning of the items). Within gestural interaction, the gestures could be designed such that they relate to some symbolic meaning associated with the action.

Another factor that influences the strength of a memory is the generation of an item. When subjects are able to generate their own stimuli, or portions of the stimuli, they remember them better than when the stimuli are given to them (Slackmecka & Graf, 1978). This generation effect has been shown to extend beyond improving memory for the stimuli itself. For example, Marsh, Edelman and Bower (2001), had participants either read or generate a list of 30 words, and those words were presented either on paper or on a computer monitor. Not only did participants remember the generated words better than the read words, but they were also better able to remember the context in which the generated words were presented.

The organization of information also affects its memorability. If items can be structured into a logical order, e.g., a hierarchy, this tends to aid learning (Bower, 1970). For instance, Dowling (1973) found that participants were better able to recognize melodies derived from a single, previously heard group than melodies that spanned two previously heard groups. This provides evidence for organizational chunking in long-term memory, allowing for more efficient storage when items have structure. Within gesture learning, marking menus take advantage of this type of organization by structuring the commands in a hierarchy, and associating directional strokes with each level of the hierarchy.

Interference can also play a large role in how items are remembered or forgotten. Retroactive interference occurs when previously learned information cannot be recalled due to new information being learned (Baddley & Dale, 1966). Conversely, proactive interference occurs when old information prevents new information from being learned (Kane & Engle, 2000). Associative interference may also lead to problems during recall, as it occurs when multiple, similar items are trying to be remembered. A large number of similar items results in greater interference and decreased learning (Ellenbogen, 2006). In all types of interference, the similarity between the pieces of information regulates the amount of interference that occurs, with more similarity resulting in more interference and difficulty during recall. Increasing the distinctiveness of each piece of information can reduce the interference, though in some cases this is at odds with constructing a meaningful organization of the material. Within gestural interfaces, interference can result from different contexts requiring the same gesture, or by having similarly formed gestures mapped to distinct actions. Designers should attempt to separate the gesture forms as much as possible, not only to decrease cognitive interference but to increase the accuracy of the gesture recognizer.

2.3. Measuring Learning

Due to differences in how memories are processed in the brain and the different contexts they are used in, procedural and declarative memories are studied using different paradigms. Procedural memory is often studied within the context of motor skills, involving the physical practice of a particular skill. Following practice, performance for that skill is measured using retention and transfer tests to assess learning. While studying declarative memory, a number of methods can be used. Recall tasks following training are common, but are sometimes not sensitive enough to small effects or are not applicable in some scenarios. As such, evaluations using mental chronometry or recognition tasks are common.

2.3.1. Measuring Learning for Procedural Memories

The field of motor learning has established methods for assessing the ability to learn and execute movements, ranging from simple pointing and grasping movements (Chapman et al., 2010) to complex skills such as surgical movements (Brydges et al., 2007) or sports (Helsen et al., 2000). The motor learning literature acknowledges a critical difference between *performance* and *learning* (Schmidt & Lee, 2011). Performance is the production of a specific action, whereas learning is the relatively permanent acquired capabilities that facilitate improved performance of that action. Within gestural interfaces, performance would refer to the production of a gesture, whereas learning would refer to one's increased ability to recall a gesture and perform it more efficiently.

Empirical studies that separate performance from learning commonly involve a *training* phase followed by a *retention* and then *transfer* component. In the retention component, participants perform the task at a common level of the independent variable, typically 24 to 48 hours after training (Shea & Morgan, 1979). In the transfer component, participants perform a novel variation on the task they were trained on, e.g., performing the task with the other limb.

The use of retention tests is standard in the motor learning literature, as they allow the researcher to separate the effects of the performance factors from the learning factors. Performance factors have an effect only for a short time, whereas learning factors have an effect much longer after training. Tests are frequently performed after at least one full night of sleep (e.g., 24 hours), as sleep has been shown to play an important role in the consolidation of motor skills (Savion-Lemieux & Penhune, 2005). The task performed during retention tests is usually similar to the task performed during training, but all participants are moved to the same level of the independent variable, which is often the removal of feedback (Schmidt & Lee, 2011).

In the transfer component, participants perform a novel variation on the task they were trained on, e.g., performing the task with the other limb. For instance, participants might perform a learned skill at a different scale or angle (Albaret & Thon, 1998). There has also been substantial work within the motor learning field on bilateral transfer, i.e., the degree to which a skill learned with one hand transfers to the other (Annett and Bischof, 2013; Panzer et al., 2010; Sainburg & Wang, 2002). These studies show that transfer takes place even when participants are not ambidextrous. Transfer tests are another way to assess learning, as the mental changes associated with learning one skill are frequently generalizable to another, very similar skill. These tests can also show how well a learned skill generalizes to a new context (Shea & Morgan, 1979). Though previous research in the gesture learning literature does not use retention and transfer paradigms, we use them to examine the effects of the guidance hypothesis within the context of gesture learning (Chapter 5).

2.3.2. Measuring Learning For Declarative Memories

Various methods of testing memory and learning have been employed to understand the cognitive component of memory. An obvious method of testing memories is using free recall, wherein participants recall as much information as they can remember (Squire, 1992). A variant on this method is cued recall, wherein participants are given a cue and are asked to recall specific information related to that cue (Ellenbogen et al., 2006). This method is most relevant to gesture learning, as gestures are often paired with specific commands and those commands can be used as cues.

Another method to evaluate the degree of learning is the use of 'forgetting curves' (Haist, Shimamura, & Shea, 1992). Measuring the ability to remember information in the days following the learning of the information allows researchers to measure how quickly the information is forgotten, which correlates negatively with the strength of learning. Shallower curves reflect information that was initially learned to a greater degree. In cases of overlearning, people are unlikely to forget the information, as it is deeply encoded with strong memory traces (Walker, 1986). Related to this, researchers can also measure memory 'savings', which represent the cost to re-learn items which have previously been learned but forgotten. When items have been previously learned, they will become remembered more quickly with re-training than unseen items (Roediger, 1990).

Finally, mental chronometry can assess the degree to which something has been learned. Mental chronometry is the use of response times to assess how quickly something can be retrieved from memory (Squire & Zola, 1996). Faster response times are indicative of a memory that has a simpler trace, or stronger connections to other memories, indicating a greater learning. Mental chronometry can be a powerful tool to measure learning, as it removes many potential sources of variability, including the verbalization or motor performance used to express learned information. Mental chronometry is often used in conjunction with forced choice paradigms, where participants must select whether an item has been seen before, or if it is a new, unseen item (Dudukovic & Wagner, 2007). While mental chronometry has not been previously used in the gesture learning literature, we adapt it and use it to measure the efficiency of encoding in the recognition of gestures (Chapter 4).

2.4. Learning and Memory in Gestural Interaction

Within the context of human computer interaction, gestures have been seen as particularly suited for niche applications, for example, where input is otherwise constrained. These types of interfaces rely on both declarative and procedural memory, and typically require substantial amounts of training to master. There have not been studies on how the fundamental aspects of memory influence gesture learning, but there have been a number of systems and methods developed to improve the learnability and usability of gestural input. These systems often incorporate many features aimed at improving learning, making it difficult to establish the contribution of each relevant factor.

2.4.1. Approaches to Training

Researchers in human computer interaction have tended to view gesture learning as a problem to be solved rather than studied, with most research focusing on the development and evaluation of new systems rather than a systematic evaluation of the constituent factors.

2.4.1.1. Implicit Memory in Gestural Interaction

Several gestural systems are designed such that the guide is always available and the user implicitly learns the gestures that correspond to frequently used actions. Examples of this can be found with marking menus (Kurtenbach, 1993), where the user performs directional strokes on a radial menu to indicate selections. The availability of the menu allows users to perform a visual search for the desired menu item if they are not able to recall it. Several extensions of this idea have been proposed to increase the bandwidth or ability to phrase commands while still maintaining the learning benefits of the structured menu (Bailly, Lecolinet, & Nigay, 2008; Zhao & Balakrishnan, 2004). A similar concept can also be found with gesture keyboards (Figure 2.1a), e.g., SHARK (Kristensson & Zhai, 2004), or Swype for the Android operating system. With these keyboards, users draw strokes directly over top of the desired letters to input text on touch-screen devices. In both systems, it is assumed that users will perform the same command repeatedly, i.e., access the same menu items or input the same words, thus implicitly learning the corresponding gestures. While these systems may be effective for their specific use case, they do not generalize across applications, e.g., a marking menu system does not function well on a small touch screen due to the limited input space and high occlusion, and Swype-based interactions require substantial screen real-estate and only provide alpha-numeric input.

a) b)

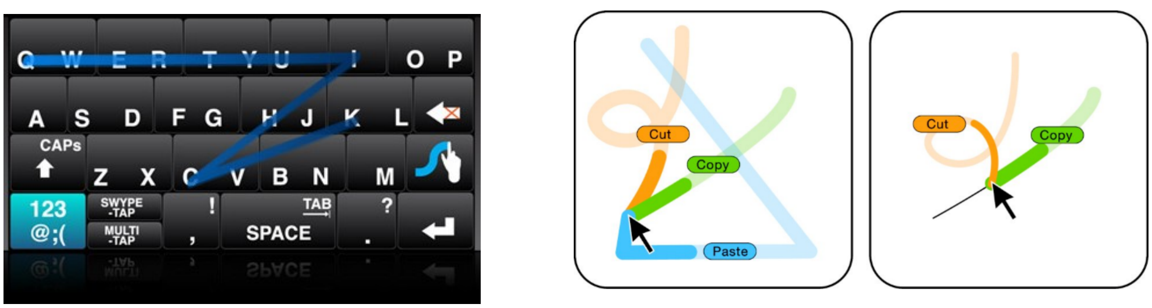

Figure 2.1. a) Swype keyboard, in which users trace over the keys using a single fluid gesture. With enough iterations, users should implicitly learn the motor patterns required for frequent words. Image from wirelesszone.com. b) Octopocus is a dynamic guide that allows the user to directly trace out gestures and updates dynamically as the user completes the stroke. As the cursor moves from the position on the left to that on the right, the guide is updated to reflect the improbability of the 'paste' command being the target action. Image from Bau & Mackay (2008).

Recently, systems offering dynamic, real-time guidance have been proposed (Bau & Mackay, 2008; Bennett et al., 2011; Freeman et al., 2009; Kristensson & Denby, 2011). These systems provide the user with information to guide the execution of a gesture, such as a traceable depiction of the gesture (Figure 2.1b). These guides are believed to improve performance, as "feedforward and feedback facilitates learning and execution of complex gesture sets" (Bau & Mackay, 2008). The guide reflects the current state of the recognizer, allowing users to receive immediate feedback as they are performing the gesture to help them complete the remainder of the gesture.

2.4.1.2. EXPLICIT MEMORY IN GESTURAL INTERACTION

The simplest method for teaching gestures is to present the user with a list (e.g., crib notes, Figure 2.2a) of potential actions and a depiction of the corresponding gestures (Bau & Mackay, 2009; Brandl et al., 2008). These depictions can be simple trajectories that should be copied by the user, or complex pictograms describing hand configuration and movement (Baudel & Beaudouin-Lafon, 1993). Kurtenbach, Moran, and Buxton (1994) developed animated crib notes to assist users in learning to perform gestures. While crib notes alone would be sufficient to aid users in the recall of gestures, the addition of in-context animations provide extra cues that help users learn the dynamics of a movement. Extending this concept is the use of video demonstrations, where the required movement is pre-recorded and played on-demand for the user (Freeman et al., 2009; Vogel & Balakrishnan, 2004). In each of these cases, the guide is separated from the input, leading to a less cohesive interaction and task-interruption when the user accesses the guide. These types of guides are thought to be less user-friendly and less effective at training gestures (Bau & Mackay, 2009; Bragdon et al., 2008; Freeman et al., 2009).

a) b)

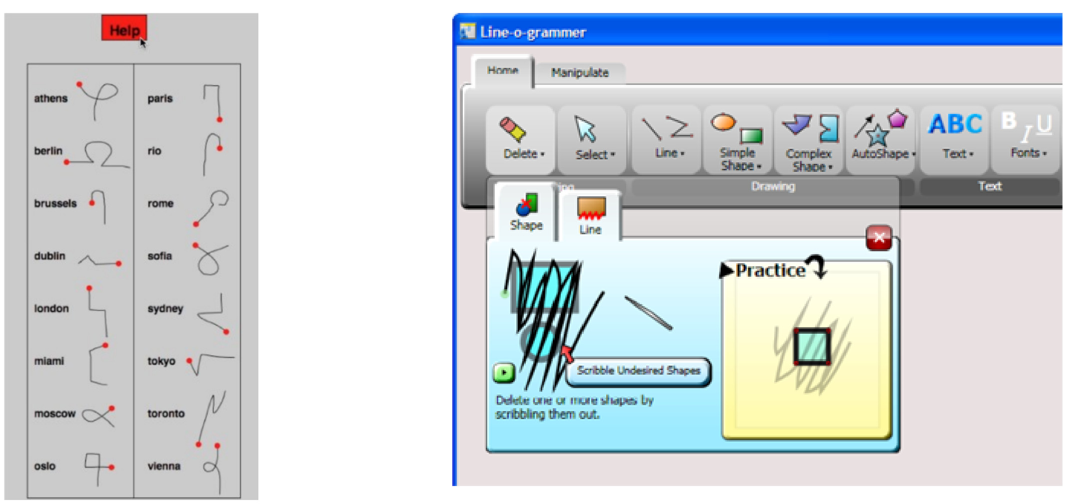

Figure 2.2. a) Example of a type of crib note, in which each gesture is depicted next to the corresponding command. Image from Brandl et al. (2008) b) Gestural interface providing interactive help in which users can retrieve hints and practice gestures in the menu bar of the program. Image from Bragdon et al. (2010).

Bragdon et al. (2008) designed an explicit method for teaching gestures, aimed at making gestural interfaces more approachable. In their 'GestureBar' (Figure 2.2b), users practice and explore gestures within the menu bar before using them within an application. While GestureBar was more effective than traditional help menus, it requires additional training time prior to use and distracts the user from their primary task. A similar concept is found within the 'Gesture Play' system (Bragdon et al. 2010), where users perform multi-touch gestures that mimic physical actions in a separate sandbox before using them.

While each of these techniques proposed and evaluated ways to improve the learnability of gestural systems, none evaluated the underlying mechanisms that influence the degree of learning. While each system may be better than traditional approaches, it is impossible to know how much each of the constituent features influenced the resultant learning.

2.5. Evaluation Methods

There is currently no standard method for evaluating the learning of gestural interaction. One common approach is to analyze behavior while participants are using the gesture system. In such studies, researchers analyze the frequency with which the gestures are used, the rate of gesture input, or user preference with the gesture system (Bragdon et al., 2010; Lepinski, Grossman, & Fitzmaurice, 2010). Appert and Zhai (2009) analyzed preference and memorability for keyboard shortcuts and gestures after training. They found that users did not have to consult the help menu system as often with gestures, and the use of gestures resulted in faster and more accurate recall of menu commands. To evaluate their menu-based gestural learning system, GestureBar, Bragdon et al. (2008) analyzed the number of correct gestures and the number of attempted gestures as participants used a gestural diagram editor. Kurtenbach et al. evaluated performance improvements over time as participants learned to use marking menus (1993). With each of these systems, user behavior was evaluated while users were actively engaged with the system, and did not separate performance from learning.

Another common approach to evaluate gesture systems is to measure participant's ability to recall specific gestures after training (Bradgon et al., 2010). To evaluate their dynamic and traceable gestural guide, Octopocus, Bau and Mackay (2008) compared participants' ability to recall gestures before and after training with a gesture system and with a traditional help window. In the evaluation of a multi-touch gestural guide system, ShadowGuides, participants recalled gestures immediately following a training phase with ShadowGuides or a video-based guide (Freeman et al., 2009). Zhai and Kristensson (2003) extended this evaluated the ability of participants to recall 100 gestures over a period of days. Though the aim of some of these systems is to assist users in the performance or execution of the gesture, they tend to focus on the cognitive component of gesture learning as measured using recall. While recall is a useful measure to assess the degree to which the action-gesture pairing was learned, these studies did not analyze the motor component of the gesture and were not able to isolate performance from learning.

Due to the lack of standards in gestural interaction and the ad-hoc nature of many empirical studies, it is difficult to characterize the rate of gesture learning. For instance, Appert and Zhai (2009) found that participants could accurately recall approximately 80% of a 14-gesture set after 10 exposures to each gesture. Freeman et al. (2009) found participants could recall 67% of a 16-gesture set after 8 exposures. When testing Gesture Play, Bragdon et al. (2010) found approximately 88% recall of a set of 16 gestures. Bau and Mackay (2008) compared video guides and their dynamic guide and found between 57 and 73% recall on a 16-gesture set after 9 exposures to each gesture. From the wide variance in findings and number of gestures used, it is clear that standardized, generalizable evaluation methods need to be developed if the field is to move forward.

Chapter 3

Self-Consistency of User-Defined Gestures[1]

Within the framework defined in Section 1.3, it is important to identify the degree to which gesture learning is required, and the degree to which the system can leverage users' existing knowledge. It is widely believed that if users select their own gestures, then this dramatically decreases the need for gesture learning. In this chapter, we examine the degree to which user-defined gestures can lessen the need for gesture learning by probing the consistency of user-defined gestures.

As our study employs a gesture-password creation paradigm, we contribute descriptions of common gesture-choice strategies, as well as a method to compute similarity between two gesture password sequences. An understanding of how users create gestures passwords can help to recognize insecure gestures, can provide guidelines on the types of instructions provided during the gesture creation phase, and inform the design of gesture password input interfaces.

3.1. User Defined Gestures

User choice in gestural interfaces has been studied extensively in the context of command-based gestures. Wobbrock, Morris, and Wilson (2009) studied users' choice of gestures for 27 separate commands on a multi-touch tabletop. They found that users produced similar gestures for 'concrete' commands, but agreement decreased for commands that were more abstract. Further investigation by Morris, Wobbrock, and Wilson (2010) showed that users preferred gestures designed by end-user consensus to those developed by experts, indicating that there is a common basis for gesture design. In the context of mobile devices, Kray et al. (2010) examined how users chose gestures when their phone was interacting with different devices. They found that gestures

[1] The majority of this chapter is currently under review at the Journal of Experimental Psychology: Applied

involving two phones were associated with greater movement than those involving a phone and a fixed display, demonstrating an interaction between environmental context and movement. Ruiz, Li, and Lank (2011) performed a gesture elicitation study using mobile devices to determine how users design motion gestures on mobile phones. Their results demonstrated that users preferred natural gestures, real-world metaphors, and direct manipulation. These research efforts indicate that users may choose natural and simple gestures, but as there is no concrete command or action on which to map the gesture, it is not obvious what a natural or simple gesture password may be.

While commands that are more abstract are not consistent between users, there is still hope that a majority of functionality can be accessed through user-defined gestures. Recently, researchers have proposed systems that support user-defined gestures by querying a crowd-sourced database of gestures (Ouyang & Li, 2012). With such a system, users input what they believe is a natural gesture to accomplish an action and the system finds the nearest match based on other users' gestures. If effective, this could mean that users need not learn gestures at all. However, for commands that have no match in the crowd-sourced database, the user must learn or define the appropriate action. Research by Nacenta et al. (2013) supports the adoption of user-defined gestures, and shows that they are more memorable than pre-defined and randomly assigned gestures. However, even this study includes a training phase where users had the gestures reinforced prior to being tested.

Additionally, it is not clear if users are consistent with themselves when producing a gesture for an action multiple times. In typical desktop environments, the display, input devices, and visual feedback remain relatively constant from operation to operation. However, as technology moves to mobile, wearable, and ubiquitous interfaces, the environments are in a constant state of flux. It is unknown whether the desired action is the only factor influencing gesture choice in these scenarios, or if the screen location, orientation or other factors may also affect the users' choice.

Research into embodied cognition has found evidence that the environment impacts high-level cognitive processes and gestural choice may be influenced by the current context of the device. Embodied cognition theory posits that cognition is situated in the environment (Wilson, 2002). That is, high-level thoughts are grounded in the physical world. Thus, contextual and spatial factors may become particularly important when users cannot anchor gestures with personal or task-based meanings. For instance, pushing movements are more closely associated to negative judgments, and pulling movements are more closely associated to positive judgments (Markman

& Brendl, 2005). In this theory, we may expect users to create spatially anchored gestures, but the gestures may be more prone to influence by external factors.

To examine the consistency of user-defined gestures when high-level instruction and environmental context were manipulated, we conducted two studies using a gesture-password generation paradigm. Gesture passwords provide a test-bed that allows for simple manipulations of the independent variables of interest while providing results that are relevant to emerging issues in security and authentication techniques for mobile devices. We examine two external factors that may influence the creation of gesture passwords: high-level instructions and device orientation.

3.2. Gesture Passwords

Gesture passwords provide an efficient method of authentication for mobile phones and tablets (Niu & Chen, 2012). To authenticate with a gesture password, users must slide their finger through a grid of buttons in a pre-set sequence. Such passwords are memorable, quick to perform, and require little cognitive overhead. In contrast to traditional numerical PIN authentication schemes, gesture passwords leverage motor and visual memory to provide memorability and high input speed. Gesture passwords have been popularized by the Android operating system, which uses them as the default authentication method. These experiments not only gave us insight into how the environment affects the choice of gestures, but it also enlightens us as to how users construct secure and memorable passwords in the absence of alphanumeric anchors.

The types of gesture passwords users create may be related to their practices with other authentication mechanisms. Bonneau (2012) analyzed over 70 million passwords from Yahoo! users, finding that most passwords effectively provide fewer than 10 bits of security, despite the password space being much larger. Extracting 4 and 5 digit numeric passwords from the database allowed Bonneau to analyze PINs in-the-wild, though it is not clear if the use of numeric passwords in an alpha-numeric context represents real-world usage of PINs. Stanekova and Stanek (2013) analyzed numeric passwords and provided methods for users to remember PINs from randomly generated sequences easily. An analysis of leaked in-the-wild PINs revealed that users tend to use very simple, non-unique sequences when defining PINs (DataGenetics, 2012). This analysis found that more than 10% of the PINs they analyzed were '1234', and the twenty most popular PINs (0.2% of the password space) represented more than 25% of the PINs used by users. A survey by Bonneau, Preibusch, and Anderson (2012) found that 7% of users chose PINs based on their birthdays. This desire for convenience in authentication has been repeatedly identified in several studies (Clarke et al., 2002; De Luca, Langheinrich, & Haussmann, 2010). While the existing work on PIN choice uses in-the-wild data and provides thorough analyses, it is not clear if any of the results transfer to gesture passwords.

While the common gesture password input space is spatially similar to a PIN pad, the numerical labels are not present so users will likely employ different strategies in their password design. Gesture passwords lack the content or meaning that users can rely on for PINs (e.g. Birthdays). It remains to be seen how users select passwords when there are no alphanumeric values or metaphors available to anchor their selections.

3.3. Experiment 1: Influence of Instruction

The first experiment was aimed at understanding the influence of an instruction on participants' gesture password creation. Prior research on gesture instruction has found that the modality of instruction can influence the accuracy of the gesture performed by the user being trained (Fothergill et al., 2012). We were specifically interested in how the form and design of a password changed when participants were encouraged to design passwords motivated by internal goals (memorability) versus external demands (security from attackers). To explore this issue, participants created gestures for three scenarios: easy for them to remember, hard for someone to guess, and hard for someone steal by watching. These three scenarios parallel the change in task that commonly used with gesture elicitation studies. We hypothesized that participants would create simpler passwords when only the internal motivation of memorability was a factor. When external factors, such as hypothetical attackers were introduced, we hypothesized that participants would vary their passwords more and make them more complex.

3.3.1. Participants

Thirty university students (M = 19.9 years, SD = 2.3 years, range = 18 - 27 years) were recruited for the experiment. Twenty-three participants were female and 14 had experience with gesture passwords. Participants were naive to the purpose of the study. All participants had normal or corrected-to-normal vision and were treated according to the APA ethical guidelines. The experiment lasted 30 minutes.

3.3.2. Equipment and Application

Participants were seated in an adjustable computer chair in front of a Dell SX2210T 21.5" touchscreen monitor with a resolution of 1920 x 1080 pixels and a refresh rate of 60 Hz. The monitor was oriented in an upright, vertical position, approximately 30 cm from the participant, within their comfortable reaching range (Figure 3.1).

The monitor was connected to a PC that ran a custom WPF application that displayed the stimuli and recorded each gesture. The application divided the screen into thirds, with the top third reserved for the instruction, and the bottom two thirds used to display a grid of 3 x 3 targets.

3.3.3. Procedure

At the beginning of the experiment, participants were seated in front of the touch screen and were informed that they had to create a number of gesture passwords, like those used on some tablets and mobile phones today. To create a password, participants had to draw a stroke through at least four grid targets. Whenever the participant's finger crossed through a target, the target changed color to indicate a selection. As the finger moved towards the next target, an elastic line was rendered from the last target location to the current finger location. Once the finger was lifted from the screen, the stroke disappeared and the targets returned to their original color.

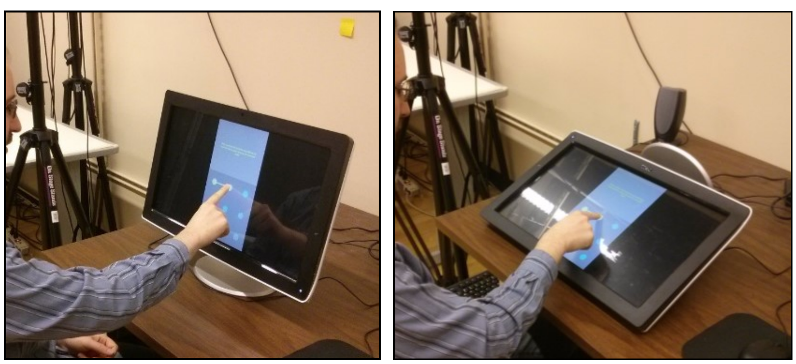

Figure 3.1. Experimental setup with the touch-screen placed vertically (left) and horizontally (right) in front of the participant. Note that the horizontal condition was only used in Experiment 2.

After participants created a password, they were asked to enter it again for validation. If the two passwords did not match, a tone sounded, indicating that the password was not valid and that they would need to enter it again. If participants made a mistake during the first two or three target selections, they were instructed to lift their finger, thereby erasing the stroke, and playing a tone that indicated the gesture was not accepted.

Participants were told to pay attention to the instruction on the screen before making each password, as the instructions changed during the experiment. Three different instructions were provided. In the easy condition, participants were prompted to "*Enter a gesture that you can remember easily*". For the hard to guess condition, participants were asked to "*Enter a gesture that would be very difficult for someone else to guess, but you can remember easily*". Lastly, in the hard to steal condition, participants were asked to "*Enter a gesture that you would remember easily, but would be secure if someone was watching you enter it*". These instructions probed how internal versus external motivation and task influenced the gestures created.

Participants were instructed to create seven different gestures for each condition, resulting in twenty-one unique gesture passwords. The presentation order of the three conditions was counterbalanced across participants.

3.3.4. MEASURES

To quantify the influence of motivation and instruction on the gestures, several measures were computed from the recorded touch data. Gesture length serves as a simple method to measure complexity of gesture, with longer gestures generally representing passwords that are more intricate. The starting location provides quantification of participants' strategy, as well as examining how the spatial layout affects gesture choice. Finally, gesture similarity provides a method to measure how much participants vary their passwords, as well as measuring how unique a participant's gestures are in comparison to other participants.

3.3.4.1. GESTURE LENGTH

To quantify the complexity of gestures, the length or number of targets that composed each gesture was computed. While other measures could have been used, such as the size of resulting bounding box, number of 'corners' in the gesture, or number of unique points, the total number of points provides a simple, direct measure of gesture complexity. While it is possible to make long gestures that are simple, and short gestures that are complex, such a measure accurately reflected the complexity of the majority of gestures produced by participants.

3.3.4.2. STARTING LOCATION

The starting location was chosen to understand the strategies participants used when creating gestures. We hypothesized that the starting location would be influenced by the complexity of the gesture, as well as the strategy used to define the gesture. The starting location was defined as the first target used in the creation of each gesture.

3.3.4.3. Gesture Similarity

While there are several algorithms to compute the similarity of stroke gestures (e.g., $1 (Wobbrock, Wilson, & Li, 2007), Rubine's (1991), Li's Protractor (2010), etc.), there are no published methods for computing the similarity of gesture passwords defined on a discrete grid. This is because, for grid-based gestures, similarity is often irrelevant, as the application is only interested if the input sequence matches the template sequence exactly. In contrast to this, we are interested in the relative similarity of non-identical gestures. Such a measure should accurately reflect minor variations in shape, as well as being translation invariant and robust to mirroring operations.

The gesture similarity of two sequences was computed by first simplifying the gesture such that one of three possible states represented each directional change in the sequence: horizontal, vertical or diagonal. Thus, a four-point gesture in the shape of an 'L' would be represented as the sequence: vertical, vertical, horizontal. To convert this representation into a numerical value of similarity, the Levenshtein (1966), or edit distance, was computed between the two simplified sequences. The gesture similarity measure G_s was computed as:

$$G_s = \frac{1}{1 + D_L(A,B)}$$

Where $D_L(A,B)$ is the Levenshtein distance of the simplified gesture sequences A and B. When gestures are identical, $G_s=1$; as gestures become less similar, G_s tends towards zero. Thus, G_s is bounded in the interval (0, 1].

In the analysis of gesture similarity, two variants were considered: self-similarity and group-similarity. Self-similarity averaged the similarity of each participant's gestures to the other gestures they created for the same experimental phase. This provided a measure of how each participant varied his or her own gestures. Group-similarity averaged the similarity of each participant's gestures with the gestures that all other participants created for the same experimental phase. This represented the uniqueness of the participants' gestures amongst the set of gestures collected from all participants.

3.3.5. Results

Each of the measures was analyzed separately and is presented independently below.

3.3.5.1. Gesture Length

A one-way repeated measures ANOVA compared the effect of Instruction (i.e., easy, hard to guess, and hard to steal) on the gesture length participants used. The analysis revealed that there was a significant effect of Instruction ($F_{2, 58} = 10.21$, $p < .005$; Figure 3.2). Post-hoc comparisons using Bonferroni-corrected paired t-tests indicated that the mean gesture length for the easy condition (M = 5.82 points, SEM = 0.12 points) was significantly lower than the difficult to guess instruction (M = 7.02 points, SEM = 0.20, $p < .001$) and difficult to steal instruction (M = 7.03 points, SEM = 0.19 points, $p < .001$). No significant difference was found between the gesture length used with the hard to guess and hard to steal instructions ($p = .96$). The results thus suggest that the instruction or prompt influences the complexity of the created gestures. Instructions that encourage memorability alone result in shorter gesture passwords, whereas instructions that suggest the need for increased security or privacy result in longer, more complex gesture passwords.

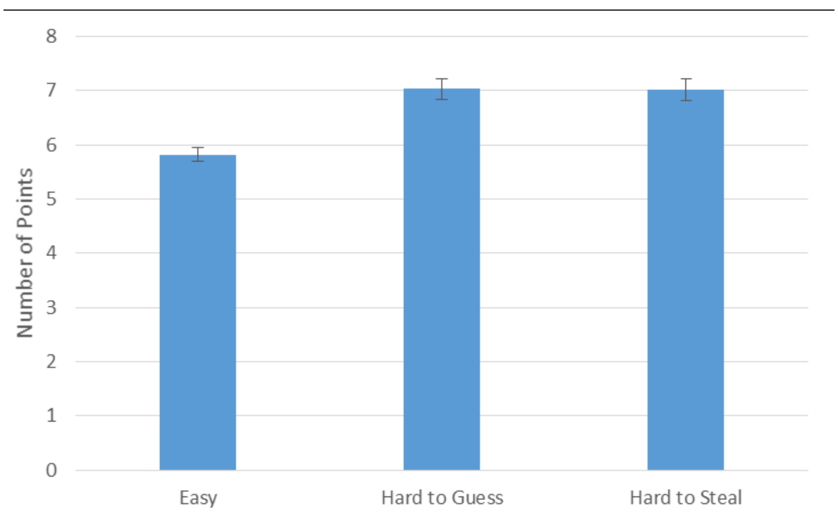

Figure 3.2. Mean gesture length by instruction. Error bars represent the standard error of the mean.

3.3.5.2. Starting Location

A Pearson's Chi-squared test of independence examined the relation between Starting Location (i.e., 1-9) and Instruction (i.e., easy, hard to guess, hard to steal). The analysis found that

Instruction significantly influenced the starting location of the gesture passwords ($\chi^2(16) = 33.47$, $p < .01$; Figure 3.3). 'Easy' gestures started in the top-center location more frequently (31 compared to 16 and 15), whereas the 'hard to guess' gestures started more frequently in the bottom-right location (19 compared to 6 and 9).

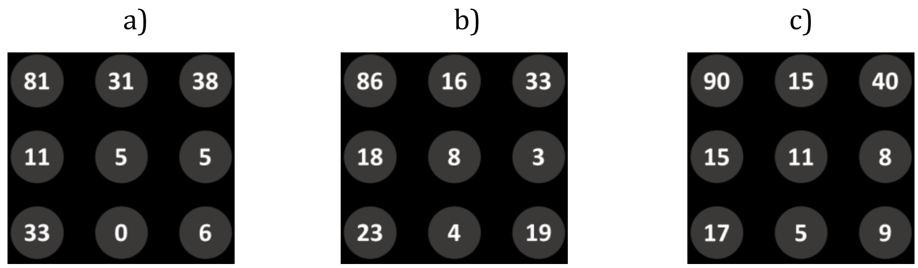

Figure 3.3. Frequency of starting location for each instruction. a) Easy; b) Hard to guess; c) Hard to steal.

3.3.5.3. Gesture Similarity

A repeated-measures ANOVA with Instruction (i.e., easy, hard to guess, hard to steal) and Comparison-Type (i.e., self, group) determined how similar participant's gestures were. The analysis found a main effect of Instruction ($F_{2, 28} = 20.7$, $p < .001$), and Comparison-Type ($F_{1, 29} = 19.8$, $p < .001$; Figure 3.4). Post-hoc comparisons using Bonferroni-corrected paired-t tests indicated that the easy instruction resulted in significantly more similar gestures (M = 0.27; SEM = 0.011) than the hard to guess (M = 0.22; SEM = 0.008; $p < .001$) and hard to steal (M = 0.22; SEM = 0.010; $p < .001$) instructions. There was no significant difference between the hard to guess and hard to steal instructions ($p = 0.74$). The main effect of Comparison-Type additionally demonstrated that self-similarity measures (M = 0.26; SEM = 0.012) were significantly greater than the group-similarity measures (M = 0.22; SEM = 0.005; $p < .001$). This suggests that even when participants were instructed to generate hard gestures, they still produced a set of gestures that were more similar to each other, than they were to gestures created by other participants. No interaction was found between Instruction and Comparison-Type ($p = 0.98$).

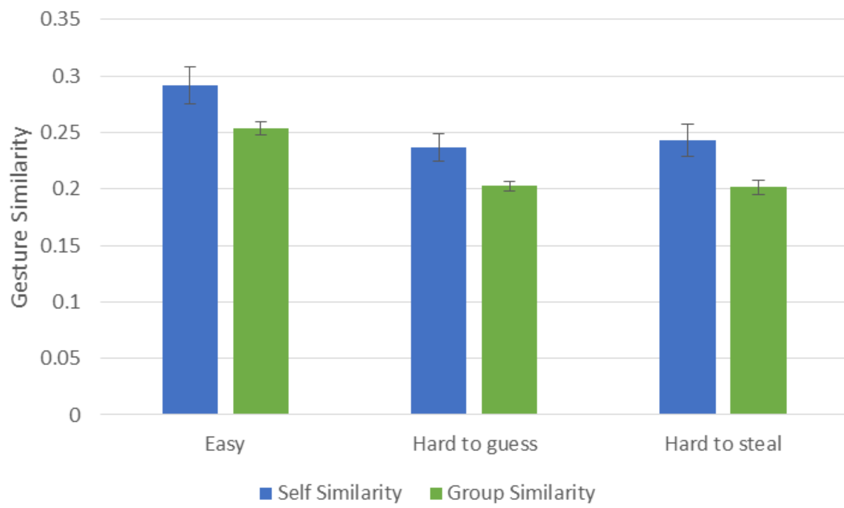

Figure 3.4. Gesture similarity by instruction and comparison-type. Note the increased similarity found within the 'easy' condition, and that self-similarity is consistently higher than group-similarity. Error bars represent the standard error of the mean.

The results indicate that participants were creating gestures that were less self-similar when instructed to create 'hard' gestures, as the self-similarity decreased between the easy condition and the two hard conditions. The length of the gestures increased from the easy to the hard conditions, which may account for some of the decrease in self-similarity. While this may be addressed by a length-normalized similarity measure, such a normalization may artificially discount the difference between gestures of vastly different lengths. Note that the self-similarity is higher than group-similarity in all cases (i.e., there is no significant interaction between the instruction factor and the comparison-type factor, p = 0.61). Thus, even when participants are trying to create difficult to steal gestures, they tended to re-use the same patterns.

3.4. Experiment 2: Influence of Environment

Given that instructions influenced the creation of gestures, we sought to identify other factors affecting gesture choice. The second experiment explored how device orientation affected the choice of gesture passwords. In this experiment, the orientation of the touch screen was either vertical or horizontal.

Due to the similarities found in Experiment 1 between the hard-to-guess and hard-to-steal instructions, the hard-to-steal instruction condition was omitted from Experiment 2.

3.4.1. Participants

Twenty-eight university students were recruited (M = 22 years, SD = 3.2, range = 15 - 27), 17 of which were female. All participants had normal or corrected-to-normal vision and were treated according to the APA ethical guidelines. Eleven had experience with gesture passwords. The experiment lasted 30 minute. Participants were divided into a *vertical* condition, where the touchscreen was vertically upright as in Experiment 1 (Figure 3.1), and a *horizontal* condition, where the touchscreen was placed horizontally on the table in front of the participant.

3.4.2. Equipment and Apparatus

The experimental setup was the same as in Experiment 1, except that the touchscreen was placed horizontally for half of the participants.

3.4.3. Procedure

Participants were asked to generate a password suitable for use on a mobile device. Each participant generated seven unique passwords for the two instruction conditions: *easy* - *"Enter a gesture that you can remember easily"* and *hard* - *"Enter a gesture that would be very difficult for someone else to guess, but you can remember easily"*. This resulted in 14 unique gestures created by each participant. Each participant completed each instruction condition when the touchscreen was either vertical or horizontal, as device orientation was a between-subjects factor. The presentation order of instructions and orientations were counterbalanced across participants.

3.4.4. Measures

The same measures used in Experiment 1 were also used in the analysis of Experiment 2.

3.4.5. Results

As in Experiment 1, each measure was analyzed independently and is presented separately.

3.4.5.1. Gesture Length

A mixed-design ANOVA was conducted using a 2 (Instruction: easy, hard; within-subjects) x 2 (Orientation: horizontal, vertical; between-subjects) design. The analysis revealed a significant effect of Instruction on the length of the gesture generated by participants ($F_{1, 26}$ = 27.6, p < .001, with the gestures created when the easy instruction was provided (M = 5.42, SEM = 0.32) being shorter than those generated when the hard instruction was provided (M = 7.06, SEM = 0.18).

Orientation was not found to influence the length of gestures created ($F_{1, 26}$ = 2.14, p = 0.16)), nor was there an interaction between Instruction and Orientation (p = 0.08).

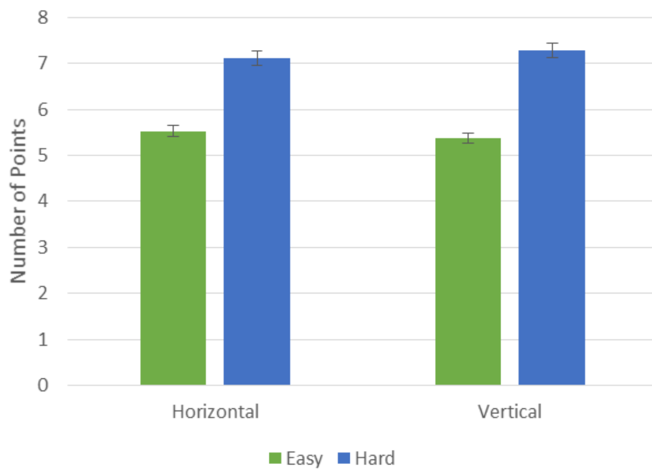

Figure 3.5. Length of resulting gestures across the four conditions. Error bars represent the standard error of the mean.

3.4.5.2. Starting Location

A Pearson's Chi-squared test analyzed the influence of Instruction (i.e., easy, hard) on the Starting Location (i.e., 1-9) of each gesture that was created. The analysis determined that gestures generated when the hard instruction was provided started in the bottom right corner more often than those generated when the easy instruction was given ($\chi^2(8)$ = 25.14, p < .01; Figure 3.6). With both instructions, gestures started in the top left hand corner three times more often than any other location. With the hard instruction however, gestures started in the top left corner less often than with the easy instruction (i.e., 73 to 89) and more often in the bottom right corner than with the easy instruction (p < .05, 9 to 0). The top-center location was also used more often in the creation of easy gestures, corroborating the findings of Experiment 1.

a) b)

Figure 3.6. Frequency of each starting location by instruction, collapsed across orientation. a) Easy, and b) Hard.

Screen Orientation also significantly influenced starting position ($\chi^2(8) = 18.2$, $p < .05$, Figure 3.7). With both instructions, the top left was the most popular starting location, with gestures in the vertical condition starting in the top left significantly more often than in the horizontal condition (i.e., 99 to 63, $p < .05$). Similarly, gestures created in the vertical condition started less often in the bottom left and bottom right corner than in the horizontal condition (i.e., 5 to 13 and 2 to 7, respectively). These values were marginally significant due to the small cell frequencies.

a) b)

Figure 3.7. Frequency of each starting location by orientation, collapsed across instruction. a) Horizontal and b) Vertical.

3.4.5.3. Gesture Similarity

A three factor (Orientation: vertical, horizontal; Instruction: easy, hard; Comparison-Type: self, group), mixed design ANOVA was conducted to understand how similar the participant generated gestures were (Figure 3.8). The analysis revealed a significant effect of Instruction on the uniqueness of the gestures ($F_{1, 26} = 12.1$, $p < .005$). Post-hoc comparisons showed that the mean similarity for the easy instruction (M = 0.29, SEM = 0.009) was significantly higher than the hard instruction (M = 0.23, SEM = 0.015, $p < .001$). Comparison-Type was found to be significant ($F_{1, 26} = 6.26$, $p < .05$), with self-similarity (M = 0.28, SEM = 0.016) significantly higher than group-similarity (M = 0.24, SEM = 0.004, $p < .05$). Orientation also significantly affected the similarity of gestures ($F_{1, 26} = 5.29$, $p < .05$), with gestures created in the horizontal condition having significantly higher similarity (M = 0.268, SEM = 0.009) than those in the vertical condition (M = 0.239; SEM = 0.009). There was no significant interaction between any of the factors ($p > 0.50$ in all cases).

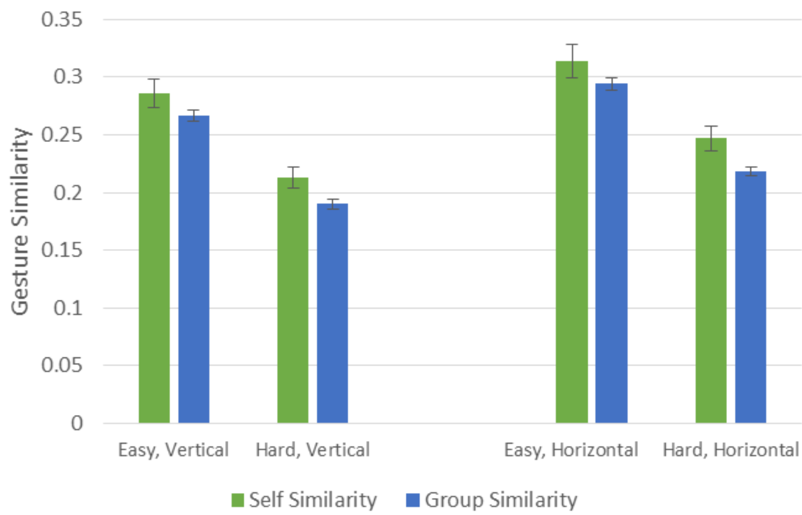

Figure 3.8. Similarity (self and group) between the gestures created under all four conditions. Note the consistently high similarity in the easy condition and the higher similarity in the horizontal conditions. Error bars represent the standard error of the mean.

3.5. Discussion

The instruction clearly influences participants' choice of gestures. When told to create an 'easy to remember' gesture, participants typically created simple spatial patterns. However, when the external factors were introduced, e.g., the instruction was 'hard to guess', participants lengthened their gestures and generated more complex gestures. The impact of instruction is clearly visible in the gesture length, which demonstrated a significant difference between the easy and two hard conditions.

When asked to create 'hard to steal' passwords, participants lengthened and increased the complexity in most cases, as with the 'hard to guess' condition. We noticed, however, qualitative differences between the 'hard to steal' and 'hard to guess' conditions in terms of the types of gestures created and the strategies used. Three strategies were observed when participants were designing gestures that were difficult to steal: *crossovers* (Figure 3.9a), *repetition of points* (Figure 3.9b) and *minimization of space* (Figure 3.9c). When using crossovers, participants generated gestures passwords that were long, and contained many overlapping strokes (sometimes using an arcing motion to skip over points deliberately). The example shown in Figure 3.9b is an extreme example of repetition of points, in which participants included a single point multiple times in the same gesture, likely so that the imagined attacker could not simply memorize the sequence locations. Lastly, many participants approached the creation of 'hard to steal' gestures by producing smaller gestures which could be completed quickly without making large, overt movements, making them harder for an attacker to observe.

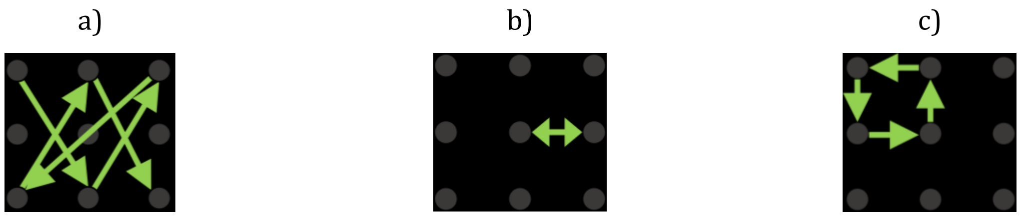

Figure 3.9. Variations used in constructing hard to steal gestures: a) Crossovers, b) Repetition of points, and c). Minimization of space.

Regarding the starting location, there was a shift away from the top-center location when creating 'hard' gestures, indicating that some participants were perhaps conscious about starting gestures in this 'obvious' location. This was emphasized in the increase in the use of the bottom right corner when creating 'hard to guess' instructions. However, the effect of starting location was minor in comparison to the drastic change in gesture length.

Gesture similarity provides further evidence that instruction can influence the design of gestures. Gestures created with the easy instruction were more self-similar than those with the hard instruction, indicating that participants varied their gestures more with the hard instruction. However, even with the hard instruction, participants still generated variations of their own gestures, resulting in self-similarity that was consistently higher than group-similarity. This suggests that an instruction to create hard gestures may not be enough to get users to appropriately vary their gesture passwords. Rather, when changing passwords, it may be beneficial for the system to compute the similarity of the new password to previous passwords and suggest a change if it is too similar.

With respect to the effects of instruction, the results of the second experiment mirror the first. Easy gestures tended to be shorter and simpler than hard gestures. Likewise, self-similarity was consistently higher than group-similarity, even when the hard instruction was provided. The orientation of the device did not have a significant effect on the measured complexity of the gestures.

With respect to the starting location, participants created more gestures starting at the bottom of the screen, which was physically closer to participants. This is evident in the increase in gestures starting in the bottom row more often for the horizontal condition, and the decrease in gestures that start in the top-left. This result emphasizes the interplay between motor movements and cognitive goals. The gestures used in the horizontal condition tended to minimize movements and could be completed using efficient maneuvers. Extending this reasoning to a mobile scenario, we would expect to find qualitative differences in gesture passwords generated when the tablet was flat on a surface or held at an angle. Likewise, passwords generated while using a device with one hand are likely to be qualitatively different from those defined the device is supported by one hand and the other hand is interacting, due to the substantial differences in movements required.

The higher similarity values (both self and group-similarity) in the horizontal condition re-iterate the influence of the physical input space. Further studies are needed to confirm the cause of the increased similarity, but we suspect that participants 'fell back' onto symbols and actions that they were familiar with due to prior experience with writing and sketching. As most writing and sketching is performed on a horizontal surface, familiar symbols and shapes (and variants on these patterns) may be more natural for users. Conversely, the vertical touch-screen is a relatively novel environment that may illicit patterns that are more novel.

3.5.1. Analysis of Gesture Form

From the data collected from both experiments, we inspected the gestures to determine the types of strategies used by participants. Across both experiments, 1022 gestures were recorded.

The uniqueness of gestures was analyzed by examining how many times each of the gestures was repeated across participants. There were 800 different gestures generated across participants with distribution shown in Figure 3.10. Of those, 708 were unique and only used once. On the other end of the spectrum, the most frequently used gesture, an 'L' shape (Figure 3.11), was independently generated by 20 participants.

The nine most popular gestures are depicted in Figure 3.11 and represent 13% of all generated gestures. From these samples, as well as a manual inspection of the rest of the generated gestures, it is clear that the overwhelming majority of easy-to-remember passwords were based on simple spatial arrangements. Simple shapes starting in the top-left corner were frequent, as were spatial variations on these patterns (e.g., translation, rotation, and mirroring).

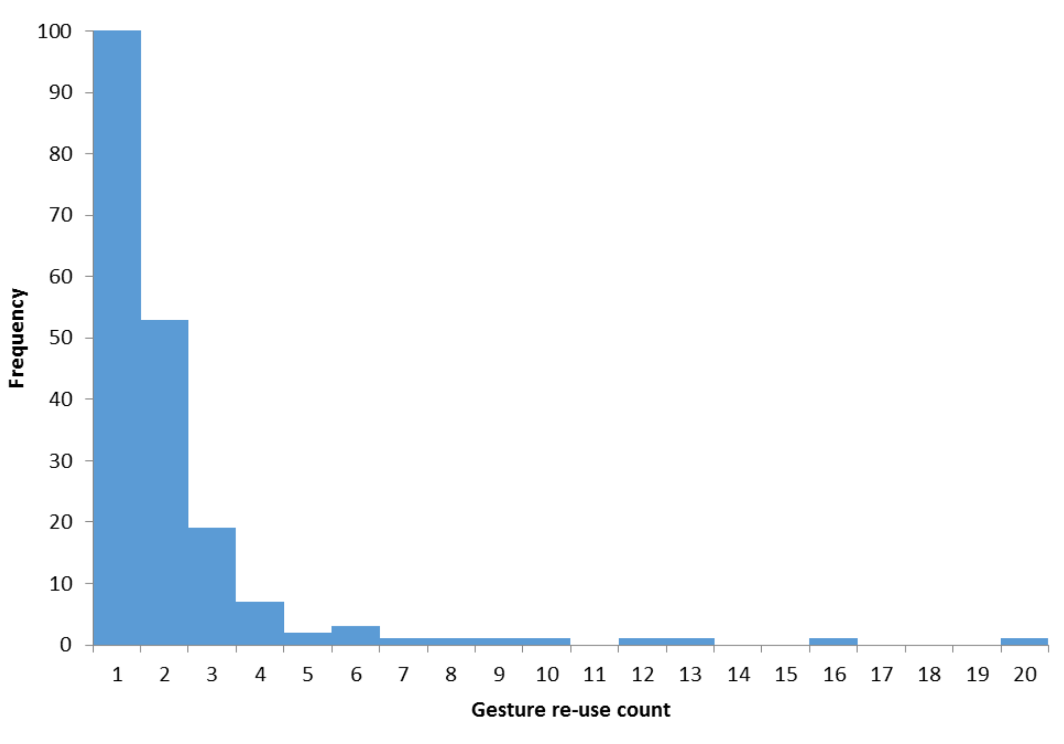

Figure 3.10. Frequency of use of gestures. Note that the first bar is truncated for clarity, the true value is 708, indicating that 708 gestures were totally unique and generated only once across both experiments.

Another common observation was the limited use of directional changes, i.e., participants often produced two subsequent strokes in the same direction (e.g., Figure 3.11, top row). This has a cognitive and motor advantage for users, as they can chunk a series of three points as a single, ballistic stroke. Consequently, this reduces the effective space of possible passwords and results in overall less secure passwords.

Participants did not appear to map numeric values onto the gesture positions (as in a PIN keypad), but rather treated them as a simple two-dimensional grid on which to draw shapes. This supports with our theory that gesture passwords may be anchored spatially when there is a lack of meaning associated with the input space.

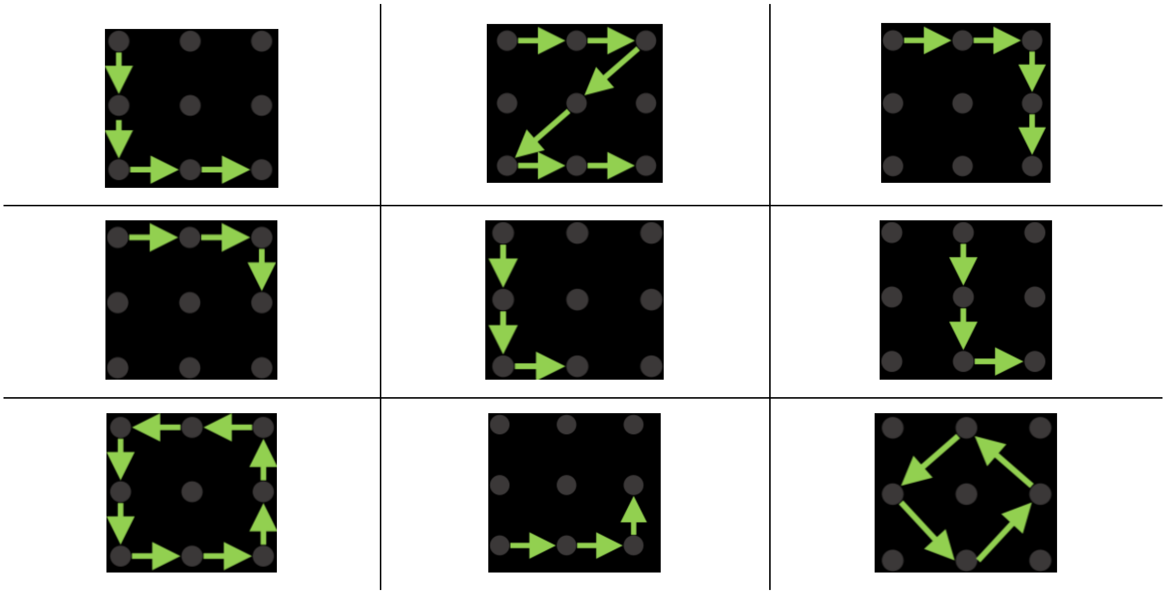

Figure 3.11. Most popular gestures generated by participants, with the top left being the most frequently used gesture, with frequency decreasing to the right, then down.

3.5.2. Implications for Gesture Passwords

For designers of gesture password interfaces, the results of this study can guide future implementations in several ways.

Designers should leverage the fact that the instructions given to the user can affect the strength or complexity of the user's passwords. Designers should seek to convey the importance of gesture security to users when they are prompting the user for their password. After a user enters a password, the system could run simple tests to validate the strength of the password. First, it could compare against a database of simple shapes (and spatial transforms of these shapes). Next, it could count the number of directional changes in the password (rather than just the length of the password) and warn if the user has only one directional change. Lastly, the interface could provide tips on creating 'hard' gesture passwords, such as varying the starting location, re-using points, and including numerous directional changes.

The similarity measure could prevent end-users from re-using variants on old passwords. This can reduce the chances that users could be re-exploited following a breach of their password. The similarity measure could also compare a user's password to database of known passwords to determine a password's uniqueness, and to encourage the user to create a more complex password if it is too similar to others.

3.5.3. Implications for Gesture Learning

The results of this study demonstrate that the high-level task can affect the types of gestures users choose. The results presented here with gesture passwords parallel the existing work on user-defined gestures that map to commands (Morris et al., 2006). What is novel, however, is the examination of the effects of manipulating the screen orientation in gesture definition. With the same high-level task, a change in the performance context can affect the resultant gestures for the same users.

The increasing adoption of wearable and ubiquitous interfaces has resulted in a highly dynamic computing environment. The location of interfaces, the pose of the body, and the required actions to perform a gesture will vary in this environment. If the system relies on user-defined gestures, there may be substantial confusion as users migrate from one gesture to another within the same task. Thus, it is clear that user-defined gestures are not a panacea for the lack of affordances presented by gestural interaction, and gesture sets in the foreseeable future will require learning on the part of the end user.

3.5.4. Future Work

This study was conducted on a touch-screen monitor, and the results reflect what users may do on a tablet rather than a mobile phone or wearable interface. We suspect that passwords generated on a mobile phone are qualitatively different due to the different movement cost associated with entering them. Future work will test this hypothesis and examine if there are other aspects that change when entering gestures on a variety of mobile devices.

Building on the spatial nature of the observed gestures, we plan to analyze how changes to the appearance of the input grid affect resulting passwords. Modifying the layout of the points with non-uniform spacing, or a circular layout may influence how users view and interact with the input space. In addition, by providing numerical or alphabetical anchors on the input grid we can examine how the presence of cognitive landmarks interacts with the spatial nature of gesture passwords, and how users generate memorable sequences when both mnemonic devices are available.

3.6. Summary

The results of the second experiment indicate that user-defined gestures are not a panacea for the problem of gesture learning. Simple changes, such as the orientation of the device, can affect how users create mappings between command and action. Therefore, interfaces that support user-defined mappings of gestures should provide support to scaffold learning.

3.6.1. Relevance to Gesture Learning

This chapter has provided evidence that users are not always consistent with the gestures that they produce, even when the task is held consistent. The metaphors that participants relied on changed as orientation changed, despite participants having similar intentions. Within the context of gestural interaction, these results point to a need to emphasize gestural learning even when users are able to select their own gestures. In particular, it demonstrates the need to support the declarative component of gesture learning within these scenarios. Users have to remember which action they intend to execute, even if the learning or performance context has changed. Systems must address the transfer of gesture learning across environmental conditions and minimize the cognitive interference that occurs when learning a variety of gestures in similar conditions. Systems may also need to integrate cues to prime the appropriate gesture to be recalled, or provide other mechanisms to help the users recognize or recall the appropriate action.

With respect to the framework outlined in Chapter 1, this study provides evidence that pre-existing knowledge may need to be supplemented by additional information. In cases where contextual interference is likely to occur (e.g., similar interfaces with different operations, or the same interface used in different environments), the user's choice of gesture may need to be reinforced by interface cues or other learning support.

3.6.2. Limitations

While this study was conducted in a lab with desktop hardware that was reconfigured, emerging interaction paradigms, such as wearable and ubiquitous computing, will have similar dynamic environmental changes, which may alter the user's perception of their functionality. Additionally, the use of gesture passwords as a test bed within this chapter has provided insights into the practical application of gestural interfaces. The preceding results can influence the design of future authentication mechanisms and improve security for mobile devices.

Chapter 4

The Cognitive Advantage of Gestures[1]

Complex gestural interfaces must support the user in learning the declarative mapping between their intent and the movements required to convey their intent to the system. Identified in Section 1.3, one method of providing this support is to modify how a gesture is rendered, in terms of the modality and the form within the modality. As identified in Chapter 2, many modalities and forms of feedback can be leveraged to support this learning. However, it is currently not clear how to best support such mappings.

In this chapter, we attempt to answer the question of whether or not gestures have a cognitive advantage over traditional input mechanisms, and why that may be the case. We explore gestures and traditional input using an ecologically focused experiment and examine the relative importance of visual and motoric actions when learning gestures. With a better understanding of how gestures are encoded, designers can build better training systems, which enable users to understand the gesture vocabulary of the system they are using quickly.

4.1. Learning the Command-Action Mapping

Some researchers have believed that gestures are relatively easy to learn as they leverage the picture superiority effect as well as motor memory (Weiss and De Luca, 2008). The picture superiority effect suggests that information is learned more readily if it is presented in picture form. As gestures are often displayed graphically as strokes, one could reason that it may benefit due to this effect. The belief that motor memory may facilitate improved retention of gestures likely stems from the familiar long-term and robust nature of learned motor skills. It may also arise from the dual encoding (Paivio & Kalman, 1973) of actions as visual and distinct motoric patterns. Or, it could be explained by the information packaging hypothesis (Kita, 2000) which predicts that gestures aid in the conceptualization of ideas. Similar work has shown gesturing

[1] The majority of this chapter is currently under review for publication at the Journal of Applied Cognitive Psychology.

while learning new information aids in the retention of that information (Cook, et al., 2008). This effect has been demonstrated in a number of domains, in particular the acquisition of language and mathematic skills (Goldin-Meadow et al. 2009; Iverson and Goldin-Meadow 2005).

Regardless of the explanation, both the visual and motor components are considered important aspects of gestural interfaces. Some of the earliest gestural interfaces, marking menus (Kurtenbach, 1993), were designed such that invoking the same menu command would require the same motor movement. Additionally, the design included a visual ‘mark’, which was in theory redundant as the menu selections provided visual feedback. However, the mark was considered important to the design and it is believed to be an integral part of the success of marking menus (Kurtenbach, personal communication, May 22, 2014).

Despite the prevalence of gestural interfaces, and the body of literature surrounding their design, there is no work confirming that gestures have a cognitive advantage over traditional input methods. There is also no work evaluating the relative effects of these components on gesture learning. To fully understand and be able to exploit the full potential of gestural interaction, it is essential that we determine what benefits gesture input offers, and what the causes of these benefits are.

In this chapter, we present two experiments that further our understanding of gesture learning. In the first experiment, participants learned sequences using either gesturing or pointing. This experiment was ecologically focused, with gesture input leveraging both of the hypothesized visual and motor advantages. In the second experiment, the visual component was fixed and participants learned sequences of varying length using gesturing or pointing.

4.2. Experiment 1: Comparing Gestures and Pointing

The purpose of the first experiment was to determine if gestures offer a cognitive advantage over traditional pointing methods with respect to the encoding and recall of pre-defined sequences. The study focused on addressing whether or not gestural interaction, as implemented in many interfaces, truly offers the advantage that many researchers claim. To that end, it was designed to be ecologically valid with the two conditions differing in both the visuals presented as well as the movements required. This experimental design is unable to assess the relative contribution of each aspect.

If the gesture condition is more efficient, it may be because fewer movements are required to articulate the sequence, which results in reduced movement complexity. Prior work has shown that movements that are more complex result in a longer response time when recalling the movement (Henry & Rogers, 1960). An alternative explanation may come from prior work spanning several domains that shows improvements in learning while gesturing (Kita, 2000; Goldin-Meadow et al. 2009). Lastly, gesture input may have an advantage due to the pictoral superiority effect (Paivio & Kalman, 1973), which would allow users to chunk the gestures into visually simpler shapes, which are easier to encode and recall. Alternatively, the use of pointing to enter sequences should result in movements that are more complex and have a longer movement time. This increases the user's exposure to the sequence and may subsequently aid in learning. Additionally, it requires the user to expend more effort during the learning phase and has been shown to be a factor in learning (Cockburn et al, 2007). Thus, it is important to assess which of these input modalities has an advantage in a real-world scenario.

4.2.1. Participants

Twenty university students (M = 20.1, *SD* = 2.2, range= 18-27 years; 16 female) were recruited to participate in 30 minute session. All participants had normal or corrected-to-normal vision and were treated according to the APA ethical guidelines.

4.2.2. Equipment and Apparatus

A 21.5" Dell SX2210T touch monitor (Figure 4.1) which was set in the upright position and used for the experiment. The monitor had a resolution of 1920 x 1080 pixels and a refresh rate of 60 Hz. The software was written in C# and WPF, and ran on a Windows 7 PC. The custom software was responsible for loading the current trial information, presenting the appropriate stimuli, and recording all touch events with their associated meta-data (e.g., time, position, and so on.).

4.2.3. Procedure

Each participant performed a training phase where they learned pairings between sequences of dots and different background colours. After a break, participants performed a two-alternative forced choice task where they responded as quickly as possible to whether or not the presented color and sequence matched one that they had learned in the training phase. While this approach does not require the user to perform the learned gesture, response time is typically more sensitive to smaller effects, providing a better understanding of the relative advantages of stroking and pointing.

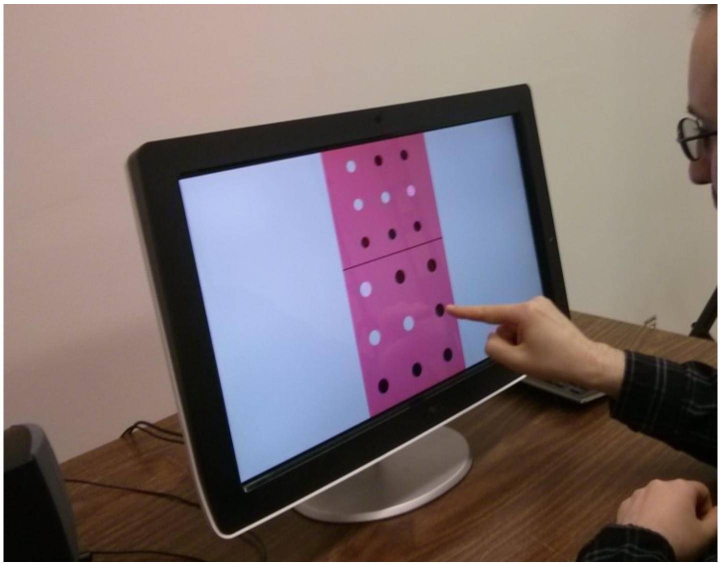

Figure 4.1. Experimental setup showing a participant learning a sequence in the pointing condition. The dots illuminated in sequence (in the top portion of the screen) and participants would touch them in the same sequence on the bottom portion of the screen.

Participants completed a learning phase, followed by a 5-minute distractor task, and finally the recognition phase. Participants were randomly assigned to one of two conditions, either *pointing* or *stroking*. The gesture grid for the learning and test phases consisted of a 3 x 3 grid of dots. This gesture grid was shown on a background of one of seven distinct colors.

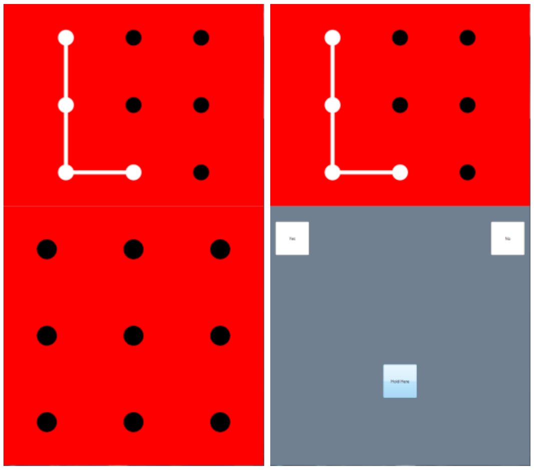

Figure 4.2. (Left) The learning phase in the stroking condition displayed lines connecting the dots in the sequence during the demonstration as well as during user input. (Right) The screen presented to participants during the recognition task required participants to hold the bottom-most button until the probe gesture appeared. Then participants quickly touched the Yes or No button to indicate their response.

In the learning phase of the pointing condition (Figure 4.1), the seven sequences to be learned were shown by illuminating dots one at a time at an interval of 200 milliseconds. The full sequence remained visible until the end of the trial. Participants then had to repeat that sequence on a separate input grid underneath the instruction grid by pointing to each of the dots in sequence one at a time using four separate movements. As participants touched each of the dots, they illuminated and remained illuminated until the end of the trial. If the participant did not repeat the sequence correctly, a tone sounded to indicate an incorrect response. The screen was cleared and the participant moved to the next trial. The stroking condition was similar to the pointing condition, but in addition to the dots being illuminated, a line between each of the dots was animated to connect the dots in the sequence providing the visual feedback of typical gestural interfaces (Figure 4.2). Further, participants in the stroking condition specified the sequence using a single continuous stroke through the dots rather than individual pointing movements.

Each participant performed both the pointing and stroking conditions, with the order counterbalanced across participants. Unique sequences and colours were used for each participant, but the same sequences and colours were used between participants and counterbalanced between the pointing and stroking condition. Each participant learned seven sequences during the learning phase. The learning phase consisted of three training blocks, each consisting of two sequential presentations of each gesture, resulting in six exposures to each of the seven gestures.

Once the learning phase was completed, the distractor task consisting of a personality questionnaire followed by mathematical questions was administered. Participants were timed with a stopwatch, and after 5 minutes were told to stop.

Finally, in the test phase, participants had 42 trials in which a gesture and background colour were shown, along with the instruction "Have you seen this gesture paired with this colour before?" as well as two buttons labelled "Yes" and "No" and a button at the bottom of the screen marked "Hold" (Figure 4.2). Participants rested their finger on the "Hold" button at which point a new gesture and background colour would be shown after a random interval between 500 to 3500 milliseconds. Participants were instructed to respond as quickly and accurately as possible after the gesture was shown, and responded by pressing either the "Yes" or "No" button. Of the 42 trials, half were pairings that were learned during the training phase, and half were unseen pairings. Of the unseen pairings, half were novel, unseen sequences, and half were learned sequences paired with the incorrect colour. No novel colours were shown during the testing phase. Accuracy as well as response time (i.e., the time that the gesture was shown on-screen to the time when a button was pressed) was recorded.

4.2.4. MEASURES

Three measures were used to evaluate the degree of learning of each of the sequences, *response time*, *accuracy*, and *efficiency*. The response time was measured as the interval between the time when the gesture appeared on the screen and the time the finger touched down on either the "Yes" or "No" buttons. Accuracy was measured as the proportion of correct responses to the recognition task. Efficiency was computed as the response time divided by the proportion of correct responses.

4.2.5. RESULTS

A two-sample paired t-test compared the influence of stroking and pointing on response time, accuracy, and efficiency.

4.2.5.1. RESPONSE TIME

Input condition was found to marginally influence response time ($t(18) = 1.9$, $p = 0.072$, $d = 0.85$), with stroking resulting in faster responses (M = 1914 milliseconds, SEM = 465.6 milliseconds) than pointing (M = 2672 milliseconds, SEM = 1163 milliseconds). This indicates that gesturing while learning sequences may result in retrieval that is more efficient.

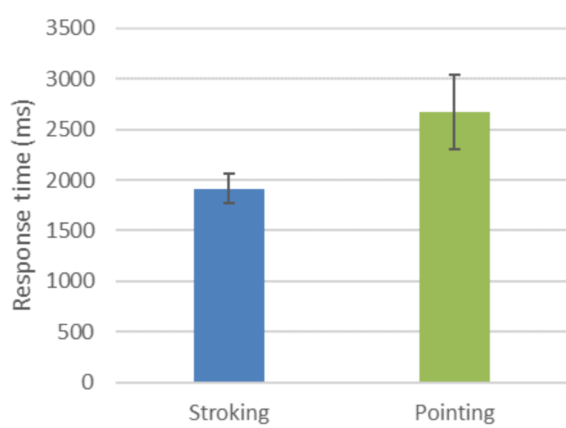

Figure 4.3. Response time by condition. Error bars show standard error of the mean.

4.2.5.2. Accuracy

Input condition was not found to significantly influence accuracy ($t(18) = 1.41$, $p = 0.176$, $d = 0.81$). While stroking, participants were slightly more accurate ($M = 0.84$, $SEM = 0.135$) than while pointing ($M = 0.73$, $SEM = 0.138$). This suggests that there may be a possible effect of stroking resulting in more memorable sequences than pointing.

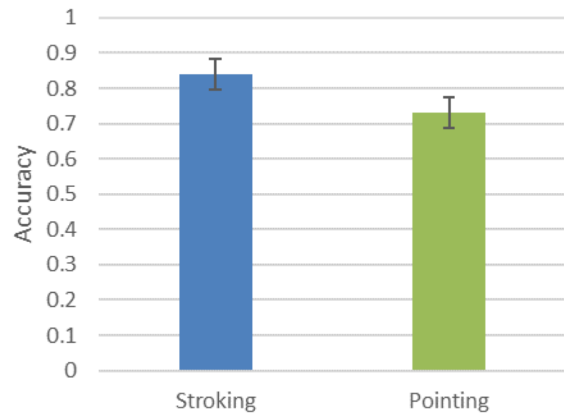

Figure 4.4. Accuracy by condition. Error bars show standard error of the mean.

4.2.5.3. Efficiency

Input condition was found to significantly influence efficiency, ($t(18) = 2.42$, $p = 0.027$; $d = 1.08$). The use of stroking resulted in more efficient responses ($M = 2493$ milliseconds, $SD = 1175$ milliseconds) than the use of pointing ($M = 3865$ milliseconds, $SD = 1626$ milliseconds). Combined with the accuracy and response time results, stroking shows a significant advantage over pointing. This indicates that gestures may be more readily learnable than traditional input.

4.2.6. Discussion

The results demonstrate a substantial advantage for gestural input in the ability to encode and retrieve gestures. Participants in the stroking condition had more accurate and faster responses than those in the pointing condition. This resulted in significantly better efficiency for the participants in the stroking condition.

While gestural input has an advantage, it is not clear what component of the gestural input is causing the advantage. The two conditions tested in this experiment differed in the movements required as well as the visual aspect so it is not possible to identify their relative contributions.

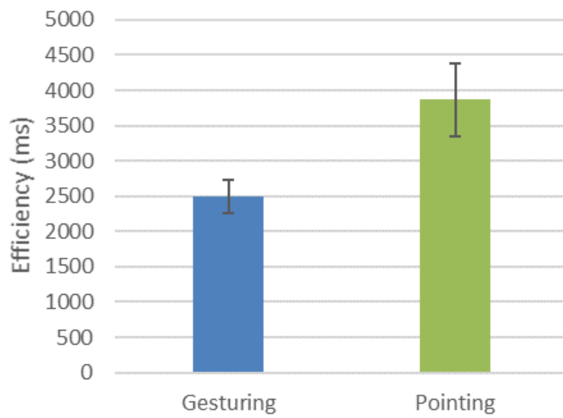

Figure 4.5. Efficiency by condition. Error bars show standard error of the mean.

4.3. Experiment Two: Effects of Visual and Motor Components

Experiment one demonstrated that gestures have an advantage over pointing, though it is not clear whether that advantage is due to the visual or motor components of gestural interfaces.

In this experiment, we sought to identify the contribution of the motor aspect to gesture learning. To that end, the visuals in both conditions were identical with visual strokes being rendered for both pointing and stroking conditions. The experiment was simplified by removing the colour pairing, and participants learned sequences of different lengths so that we could measure the 'cost' of adding points to sequences learned under both conditions.

4.3.1. Participants

Twenty-four university students (M = 20.5, *SD* = 2.4, range= 18-25; 18 female) were recruited to participate in a 60-minute session. All participants had normal or corrected-to-normal vision and were treated according to the APA ethical guidelines.

4.3.2. Equipment and Apparatus

The same apparatus and distractor tasks as used in Experiment 1 were used again in Experiment 2. The software used in the experiment was modified slightly to account for the minor change in experimental paradigm.

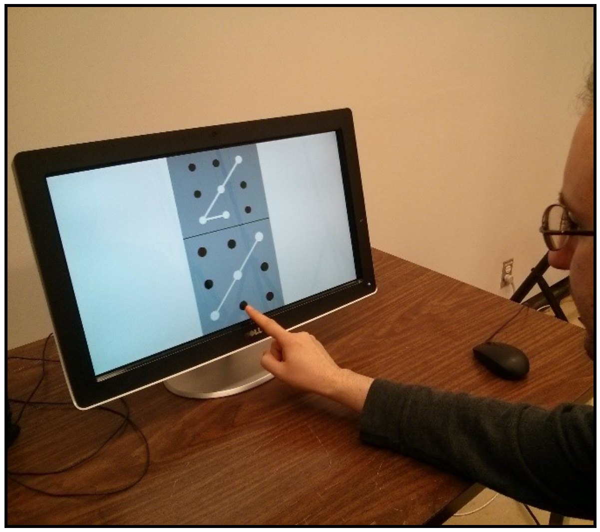

Figure 4.6. Participant demonstrating the learning phase during the pointing condition just prior to touching the final target.

4.3.3. Procedure

Participants performed a learning phase, followed by a distractor task, and finally a test phase, similar to Experiment 1.

During the learning phase, each participant learned a set of eight sequences of length 4, 5 or 6 depending on the condition. Learning was structured into three blocks of training trials, with participants being exposed to each sequence twice per block, resulting in six exposures to each sequence during the learning phase. Each sequence was displayed with a stroke animating through them, as in the stroking condition in Experiment 1. Participants then had to either stroke through the sequence or point at each dot in the sequence using discrete movements.

Following the learning phase, a 5-minute distractor task consisting of a personality questionnaire followed by mathematical questions was administered.

Following the distractor task, the test phase was completed. Participants were shown 32 sequences, each on a grey background, and determined if they had seen the gesture during the learning phase or not. The same response-time paradigm from Experiment 1 was used. Eight of the sequences presented during the test phase were trained sequences, eight were novel sequences, and each sequence was presented twice.

After the participant finished the testing phase there was a short break and the learning-distractor-test procedure was repeated for the remaining two sequence lengths. The order of sequence length and the distribution of pointing and stroking was counterbalanced across participants.

4.3.4. MEASURES

As in the first experiment, response time, accuracy, and efficiency are used to assess learning. In addition, the input duration provided a deeper understanding of the results. Duration was calculated as the time between the first 'touch down' event and the last 'touch up' event when inputting the sequence during training.

4.3.5. RESULTS

A 3 x 2 mixed-design ANOVA was conducted with factors of *length* (4, 5, 6; within-subjects) and *input type* (pointing, stroking; between subjects). Where appropriate, Bonferroni-corrected paired t-tests were used for post-hoc, pairwise comparisons.

4.3.5.1. RESPONSE TIME

Response time was analyzed using a mixed design ANOVA with factors of Length (4, 5, 6; within-subjects) and Input type (pointing, stroking; between subjects; (Figure 4.7). A significant main effect was found for Length ($F_{2,28}$ = 4.593; p = 0.015). No significant main effect was found for Input type ($F_{1,14}$ = 0.276; p = 0.605) nor was the interaction between Length and Input type found ($F_{2,28}$= 0.173; p = 0.842). It thus does not appear that pointing or stroking dramatically impacts the response time when the visuals remain fixed.

Post-hoc Bonferroni-corrected paired t-tests demonstrated that sequences of length six resulted in significantly longer response times (p = 0.007, d = 0.698; M = 2390 milliseconds, SD = 726 milliseconds) than those of length five (M = 1955 milliseconds, SD = 499 milliseconds). The other pairwise comparisons were not found to be significant (p > 0.1). It thus appears that increased sequence lengths result in longer response times, but a larger range of sequence lengths is needed to fully identify the relationship.

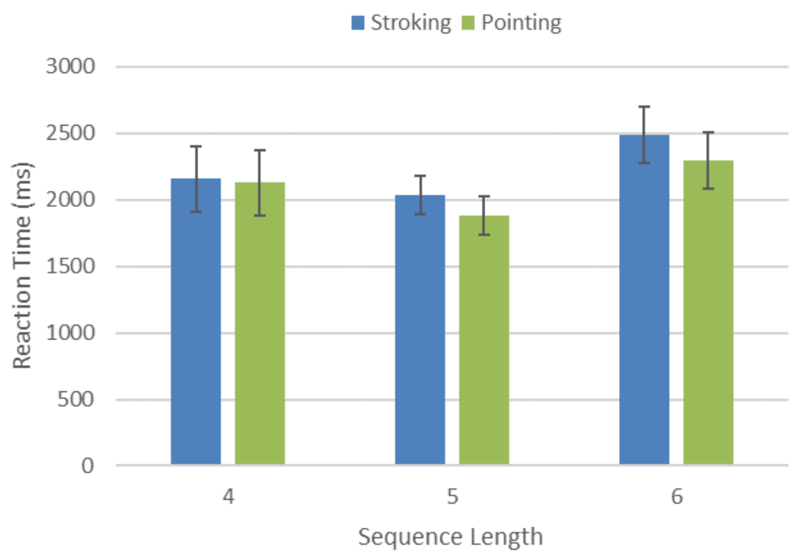

Figure 4.7. Response time for stroking and pointing, by length of sequence. Error bars represent standard error of the mean.

4.3.5.2. Accuracy

Accuracy was analyzed using a mixed design ANOVA with factors of Length (4, 5, 6; within-subjects) and Input type (pointing, stroking; between subjects; (Figure 4.8). No main effects were found for length ($F_{2,28}$ = 2.005; p = 0.147), input type ($F_{1,14}$ = 0.303; p = 0.588), or the interaction between both ($F_{2,28}$ = 0.432; p = 0.652). Thus, when the visuals are fixed, the accuracy of responses is not affected by sequence length or whether the participants were stroking or pointing.

4.3.5.3. Efficiency

Efficiency was analyzed using a mixed design ANOVA with factors of Length (4, 5, 6; within-subjects) and Input type (pointing, stroking; between subjects; (Figure 4.9). A significant main effect was found for Length ($F_{2,28}$ = 4.78; p = 0.013). No significant main effect was found for Input type ($F_{1,14}$ = 1.356; p = 0.257) now was the interaction between Length and Input type found to be significant ($F_{2,28}$ = 0.081; p = 0.923).

Post-hoc Bonferroni-corrected paired t-tests demonstrated that gestures of length 6 resulted in less efficient responses (p = 0.01, d = 0.95; M = 3141 milliseconds, SD = 931 milliseconds) than those of length 5 (M = 2394 milliseconds, SD = 618 milliseconds). All other pairwise comparisons were not significant (p > 0.1). Thus, efficiency does not appear to be affected by whether the participant strokes or points, but rather by the sequence length. For this experiment, efficiency seems to be heavily dominated by response time.

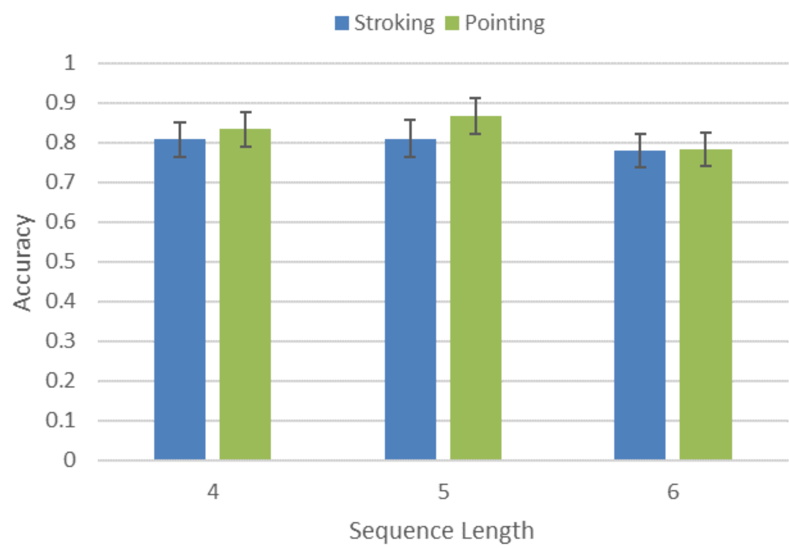

Figure 4.8. Accuracy for stroking and pointing, by length of sequence. Error bars represent standard error of the mean.

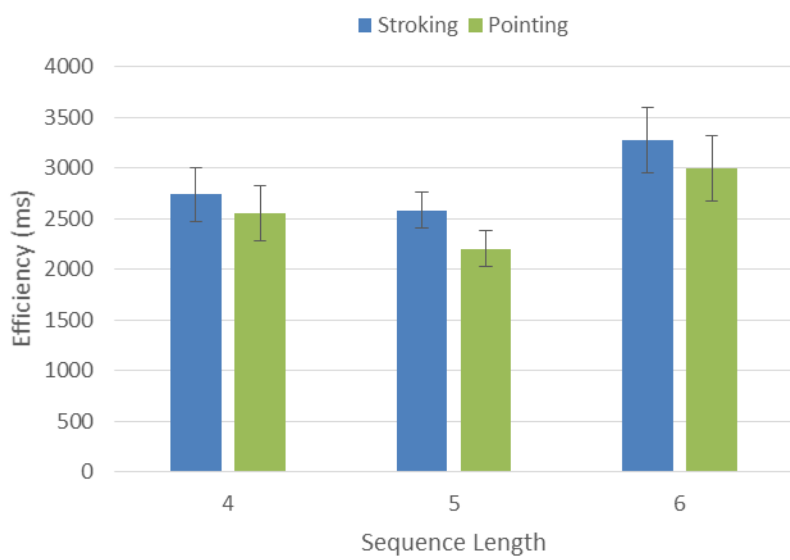

Figure 4.9. Efficiency for stroking and pointing, by length of sequence. Error bars represent standard error of the mean.

4.3.5.4. Duration

Duration was analyzed using a mixed design ANOVA with factors of Length (4, 5, 6; within-subjects) and Input type (pointing, stroking; between subjects; Figure 4.9). A significant main effect was found for Length ($F_{2,28} = 61.75$; $p < 0.001$) and Input type ($F_{1,14} = 63.50$; $p < 0.001$), as well as the interaction between Length and Input type ($F_{2,28} = 69.50$; $p < 0.001$).

Bonferroni-corrected post-hoc tests determined that there was a significant effect of Iinput type ($p < 0.001$, $d = 2.08$), with pointing resulting in longer durations (M = 1316 milliseconds; SD = 379.0 milliseconds) than stroking (M = 425 milliseconds, SD = 379.0 milliseconds). A significant difference was found between all post-hoc pairwise comparisons for length ($p < 0.01$) with sequences of length 4 (M = 664 milliseconds; SD = 213 milliseconds) less than those of length 5 (M = 868 milliseconds; SD = 257 milliseconds), and both being less than those of length 6 (M = 1079 milliseconds; SD = 369 milliseconds). There was, however, no significant difference between the different lengths of sequences for the stroking condition, only the pointing condition.

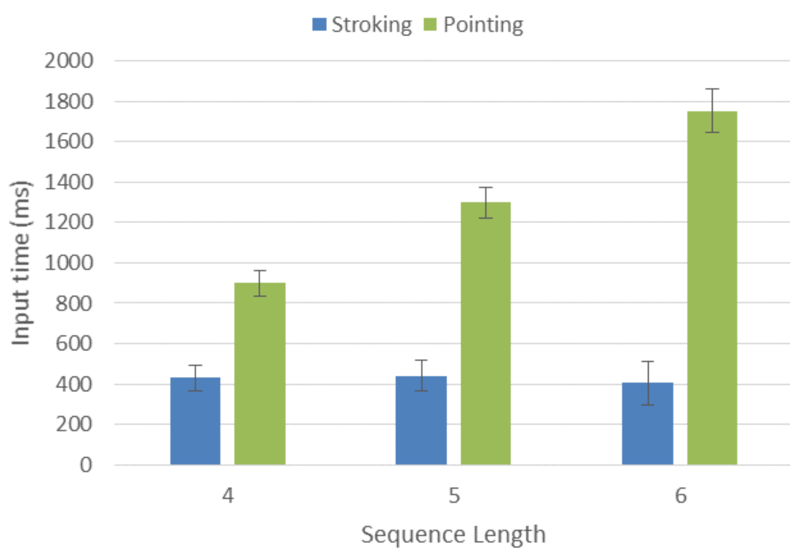

Figure 4.10. Efficiency for stroking and pointing, by length of sequence. Error bars represent standard error of the mean.

4.3.6. Discussion

Sequence length affects both response time and efficiency, with the longer sequence resulting in slower response times than the two shorter sequences. This is consistent with existing literature that demonstrates an increase in response time with an increase in movement complexity. While we did not see significant increases between all of lengths 4, 5 and 6, we suspect that with a larger sample size the pattern would become more consistent. Alternatively, testing sequence lengths that cover a greater range (e.g., 4, 8, and 12) would likely emphasize this result.

We did not see a significant or strong effect of movement type (i.e., *pointing* or *stroking*), or an interaction between movement type and length. In fact, the efficiency of responses to the sequences learned within the pointing condition was better than those learned with the stroking condition. This seems contradictory to Experiment 1. This indicates that the motor component of gesture learning is less important than the visual component.

The amount of time to input each sequence was much longer for participants within the pointing condition than those within the stroking condition. As such, the length of time that participants were 'exposed' to the sequence was longer and this may partially account for why the pointing condition results in better efficiency in the second experiment than the first.

To examine these issues further, we directly compared the results from the two experiments.

4.4. Cross-Experiment Comparisons

As there was no strong effect of gesturing in the second experiment, we compared data from the first experiment to those in the second experiment, for the condition where users learned sequences of length four. The experiments differed in the pairing of gestures, and the number of sequences learned as well as the presence of visual strokes during training in the 'pointing' condition. To determine the effect of visual strokes, we compare both the pointing condition and stroking condition between experiments. The stroking condition differed only by the experimental design (learning seven sequences and pairing them with colors), while the pointing condition differed by experimental design as well as the presence of visual strokes.

4.4.1. Stroking

A two-tailed t-test comparing response time for the 'stroking' condition between the experiments shows no significant difference ($t(20) = 1.2$, $p = 0.238$, $d = 0.52$), with Experiment 1 resulting in slightly lower response times (M = 1914 milliseconds, SD = 466 milliseconds) than Experiment 2 (M = 2156 milliseconds, SD = 462 milliseconds).

A two-tailed t-test comparing accuracy for the ‘stroking’ condition between the experiments shows no significant difference ($t(20) = 0.270$, $p = 0.79$, $d = 0.14$), with Experiment 1 resulting in lower accuracy ($M = 0.790$, $SD = 0.126$) than Experiment 2 ($M = 0.81$, $SD = 0.157$).

A two-tailed t-test comparing efficiency for the ‘stroking’ condition between the experiments shows no significant difference ($t(20) = 0.81$, $p = 0.428$, $d = 0.34$), with Experiment 1 resulting in less efficiency ($M = 2493$ milliseconds, $SD = 760$ milliseconds) than Experiment 2 ($M = 2739$ milliseconds, $SD = 663$ milliseconds).

4.4.2. Pointing

A two-tailed t-test comparing response time for the ‘pointing’ condition between the experiments shows no significant difference ($t(20) = 1.121$, $p = 0.276$, $d = 0.48$), with Experiment 1 resulting in higher response times ($M = 2672$ milliseconds, $SD = 1163$ milliseconds) than Experiment 2 ($M = 2128$ milliseconds, $SD = 1110$ milliseconds).

A two-tailed t-test comparing accuracy for the ‘pointing’ condition between the experiments shows a significant difference ($t(20) = 2.1$, $p = 0.049$, $d = 0.87$), with Experiment 1 resulting in lower accuracy ($M = 0.71$, $SD = 0.136$) than Experiment 2 ($M = 0.833$, $SD = 0.148$).

A two-tailed t-test comparing efficiency for the ‘pointing’ condition between the experiments shows a significant difference ($t(20) = 2.22$, $p = 0.038$, $d = 0.93$), with Experiment 1 resulting in less efficiency ($M = 3865$ milliseconds, $SD = 1627$ milliseconds) than Experiment 2 ($M = 2553$ milliseconds, $SD = 1139$ milliseconds).

4.4.3. Discussion

The stroking conditions in both experiments resulted in very similar results, with no significant differences between any of the measures. This indicates that the effects of the differences between experiments for this condition were relatively minor (i.e., pairing sequences with colors, and learning a different number of sequences). If this were not the case, then a comparison between the pointing conditions would not be valid.

The tests reveal a significant difference between pointing conditions between experiments. This is likely due to the presence of visual feedback in experiment two during the pointing task. Participants had faster response times, significantly more accurate responses, and significantly better efficiency, with such feedback The relatively large effect sizes ($d = 0.48$, 0.87 and 0.93) illustrate the impact that the presence of absence of the visual feedback has when learning sequences. These effects are similar to that of adding 2 points to the sequence (e.g., the effect size between length 4 and 6 on response time is d = 0.698).

Theories behind the picture superiority effect may explain why the visual component may play the larger role in gesture learning. The results of the presented experiment may support Nelson's semantic sensory theory, which states that pictures are remembered more readily because they are more distinct than one another (Nelson et al., 1977). Within the context of our experiment, the shapes formed by the lines of the gestures may have resulted in much higher visual dissimilarity between the sequences with connecting lines compared to the sequences of dots that are all visually similar. Alternatively, the results may support a dual-coding view of the picture superiority effect, as the visually-connected sequences can be encoded as shapes and perhaps be assigned some semantic label by the user, whereas the unconnected sequences of dots seems less likely to evoke a more abstract meaning. Regardless of the explanation, it is clear that presenting a visual stroke to represent the gestures can dramatically affect the degree to which the sequence is learned.

4.5. SUMMARY

Gestural input has become prevalent in part because of the belief that it leverages visual and motor memory to improve the encoding of input sequences. We have found that gestures do offer cognitive advantages over typical pointing-based input techniques, and that users respond faster to sequences learned via gesturing.

We have also shown that the visual and motor components of typical gesture input do not impact the memorability of gestures equally. The motor component appears to play a much less significant role than the visual trace connecting the points in the sequence. This finding is particularly important within the context of gesture learning systems, as it suggests that visual feedback may be leveraged to support learning the association between command and action. By repeatedly exposing users to the visual representation of a motion gesture, it may be encoded more readily and the declarative memory may be more easily recalled. Correspondingly, it does not appear to be sufficient to expose users to repeated movements in the hope that they will eventually learn the association between the movement and the desired action. The visual component is an essential aspect of gesture learning.

4.5.1. Relevance to Gesture Learning

With respect to the framework outlined in Section 1.3, we examined the declarative component of gesture learning. Specifically, the results indicate that the rendering of gestures affects the learnability of gestures. Additionally, the motor component does not appear to enhance the learning of the mapping between command and action, as previously thought. Designers should provide gestural interfaces which emphasize the visual component of gestures, as that appears to be the primary driver in memorability with respect to gesture recall.

4.5.2. Limitations

The presented studies focused on testing the declarative component of gesture learning using a two alternative forced-choice paradigm, and as such they did not require the participants to perform the gesture during the testing phase. Thus, the testing phase was primarily visual which likely has some measurable impact on the results. However, if the motor component is heavily involved in the ease-of-learning in gestural interfaces, then the paradigm used should still reflect some of those benefits. Further studies are needed to clearly identify the role of movements in gesture acquisition.

Chapter 5

Learning and Performance with Gesture Guides[1]

Support for learning the procedural component of gestures is crucial for enabling users to perform efficiently with a gesture-based interface. Once a user learns the command-action mapping, they need the ability to perform it, or to perform variants on it to convey additional parameters. To support this, systems need to provide appropriate feedback and guidance to support performance and learning. Additionally, designers of gesture guides need to be aware of the difference between learning and performance and how they apply to gestural interactions. This chapter explores how we can best train the procedural component of gesture learning and how it can be evaluated.

5.1. LEARNING WITH GESTURE GUIDES

As mentioned in section 2.3.1, performance refers to the execution of an action, whereas learning refers to the long-term changes associated with the ability to perform that action. The difference between learning and performance becomes very important when considering the guidance hypothesis (Schmidt and Lee, 1991). The guidance hypothesis states that excessive guidance during training can hinder learning, as the user can become reliant on the guidance. Guidance can take the form of knowledge of results (KR), which is information regarding the success or failure of a movement, or knowledge of performance (KP), which is information regarding how the participant and target movements differ. The amount of guidance provided to a user is an important consideration, as many new gesture guides provide concurrent or real-time feedback (or 'feedforward') to help the user execute the gesture.

[1] The majority of this chapter has been published as Anderson, F and Bischof, W. F. "Learning and Performance with Gesture Guides" in the Proceedings of the ACM Conference on Human Factors in Computing Systems, 2013, pp. 1109-1118.

To date, most studies have evaluated the learnability of gestural interfaces by comparing performance measures taken during, or shortly after, the training phase of an experiment (Appert & Bau, 2010; Appert & Zhai, 2009; Bradgon, et al., 2008). These often include gesture recall, frequency of gesture use, or input speed. Although this evaluation is direct and intuitive as it mimics real world use cases, it does not necessarily evaluate how well the participants learned the gestures; it measures how well they performed them.

This chapter makes three main contributions to the literature on gesture learning. First, we introduce the use of the retention and transfer experimental paradigm within the context of gesture learning. Second, we analyze four different gestural guides with this paradigm. Third, we introduce an adaptive guide that mitigates this tradeoff, and provides a smooth transition from novice to learned user.

5.2. Gesture Guide Experiment

The purpose of this study was to evaluate the degree to which the design of the feedback affected the degree to which the procedural component of the gesture was learned. To assess this, participants learned a set of four gestures using one of four different types of guides, each of which varied how they presented feedback. Following this, participants performed retention and transfer tests, as well as delayed retention and transfer tests 24 hours later.

5.3. Methods

5.3.1. Participants

Thirty-six subjects participated in the study (M=25 years, SD=11 years, range = 18-77 years, 15 male). All participants were right-handed, as determined by the Edinburgh handedness inventory (Oldfield, 1971). Each participant was assigned to one of four gesture guides: *crib-notes*, *static-tracing*, *dynamic-tracing*, or *adaptive* which are described below. Before beginning the training phase, each participant was informed that there would be a follow-up test, but were not informed of the nature of this test.

5.3.2. Apparatus and Gestures

The experiment used a pen-based Cintiq 21UX from Wacom, with the screen positioned directly in front of the participant at a 10° incline. The software was developed using the Windows Presentation Framework, and ran full-screen at 1600 x 1200 pixels. For reference, the display size of the screen is (43 cm x 32.5 cm), resulting in a mapping of 1 pixel = 0.27 mm. The buttons on the pen were not used; all interaction was accomplished through the contact and motion of the pen on the screen.

Each of the four gestures (see Figure 5.1b) was composed of one or two simple line or curve segments and paired with an arbitrary, unrelated verb. All gestures were the same length and defined by a single, computer-generated template rather than as a series of user-generated examples. The gestures were designed to cover a wide range of possible gestures, as all gestures can be described as a series of curves, lines, and corners (Cao & Zhai, 2007). The initial angle of each gesture was rotated such that it did not coincide with a major axis or a diagonal. While typical gesture systems use more than four gestures, this study is intentionally restricted to analyzing only four. If more gestures are added, participants have a difficult time learning the pairing between gesture and command, i.e., they struggle at the 'cognitive' phase of Fitts and Posner's (1967) model rather than progressing through to the associative or autonomous phases. As the intent is to study the production and form of the gesture, the use of four gestures allows participants to learn the pairing quickly so they could then better learn the form.

The red gesture, paired with '*Choose*', was two straight lines of equal length joined by an obtuse angle. The green gesture, paired with '*Send*', was a curve of constant radius, sweeping out a 180° arc. The blue gesture, paired with '*Build*', was a curve of constant radius connected to a straight line. Lastly, the orange gesture, paired with '*Find*', was composed of a long straight line connected to a short straight line at a 90° angle.

5.3.3. GUIDE TYPES

We evaluated four types of gesture learning systems. Three of the guides have been previously described in the literature or are very similar to previously described guides (*crib-notes*, *static-tracing*, and *dynamic-tracing*), while the fourth (*adaptive*) is a novel contribution.

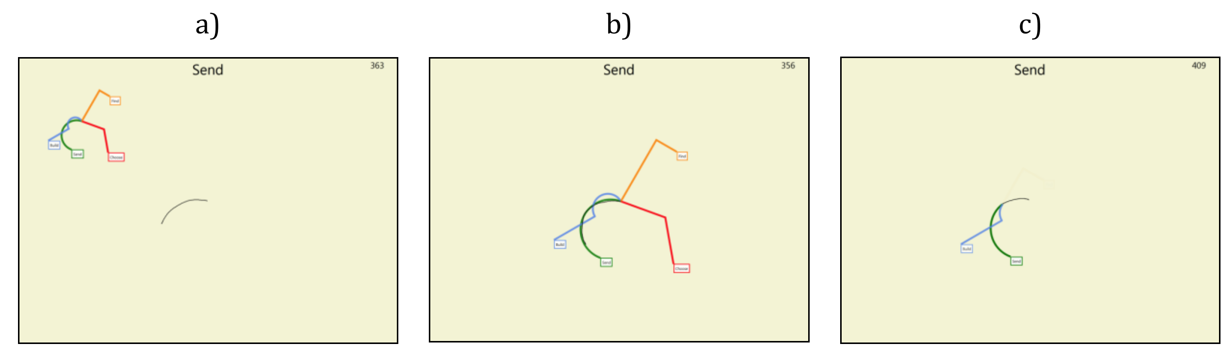

Figure 5.1. The behavior of the guides while performing the Send gesture during training trials for a) crib-notes, b) static-tracing, and c) dynamic-tracing. Note that adaptive guide is not shown as its behavior is identical to the 'static-tracing' guide, except the guide is removed partway through the trial.

The *crib-notes* guide used a half-scale depiction of the gestures placed in the top-left corner of the screen (Figure 5.1a). Participants using this system were not informed of the scale relation between the guide and the target gesture, and learned the appropriate scale through the KP provided after each trial. This guide provides the least guidance, as participants cannot directly compare their current trajectory to the template and the template does not adapt to their movement.

The *static-tracing* guide used a full-scale depiction of each of the template gestures, radiating from the initial pen location (Figure 5.1b). This guide allowed the participant to trace over the target gesture. As the participant drew their stroke, the guide was not updated. The use of this guide allowed us to examine what effects the continuous updating has on the learning and performance of the gestures.

The *dynamic-tracing* guide (referred to as 'dynamic guide' in Bau and Mackay (2008) used a full-scale depiction of the gestures, as with the static-tracing guide, but as the participant moved the pen, the guide dynamically updated to reflect the state of the recognizer (Figure 5.1c). As the participant drew their stroke, the opacity of each of the four gestures was mapped to a function of the similarity between the participant's trajectory and the template of the target gesture. Gesture similarity during training was measured by computing the RMSE between the participant's trajectory and an equivalent path length from each of the target gestures. In addition to modifying the opacity of the guide strokes, the initial segment of each template gesture was removed (an amount equal to the current participant's stroke length), and the result is appended at the current pen location. This procedure effectively provided the '*feedforward*' information to help guide the participant to the correct performance.

The *adaptive* guide provided a traceable guide identical to the one used in the static-tracing condition, but the guide disappeared at some point in time during the trial. The current trial as well as the current length of the participant's stroke determined when the guide disappeared. For the first trial, the guide disappeared once the participant's stroke had the same path length as the target gestures. Midway through the trials, the guide disappeared once the participant's stroke was half the path length of the target gestures. By the end trial, the gesture guide did not appear at all. This approach let participants initially trace the gestures with high accuracy and usability, but eventually required them to draw the gestures without the guide. The adaptive guide provides a form of faded feedback (Sherwood, 1983; Wulf, Shea & Matschiner, 1998; Wulf & Schmidt, 1989) which has shown to be an effective method of presenting feedback to enhance learning. While the implementation of this guide for the lab study is straightforward, as the number of trials is known, the implementation in a real-world scenario is potentially more difficult. Various methods of implementing an adaptive guide in a real-world scenario are described in the Discussion.

None of the guides used in this study were dynamic in the sense that they changed scale or orientation in response to the user's strokes, as in other recent guide designs (Appert & Bau, 2010; Panzer et al., 2010). This is an intentional choice, as it allows control over the exact gesture learned by the participants. This decision allows more precision in studying the effects of the guide on learning a particular gesture. It is highly unlikely that the ability to change scale or orientation will have any effect on the degree to which the user is guided, or subsequently learns the gesture. Once a user determines or selects a particular scale and orientation, they will likely use the guide to continue drawing the gesture at that particular scale or orientation. That is, the users would not just be guided to the same degree, they would also be guided to a different target gesture.

5.3.4. Procedure

Participants were shown where to place the pen on the screen to activate the guide and where their score would appear. They were told to accrue as many points as possible and that their score was derived from the similarity to the target gesture, with additional points for faster performance. To compute the points, the training system awarded the user with points proportional to an execution time under four seconds and an accuracy error under 220 pixels. For instance, if the gesture was completed in two seconds with 30 pixels of error (i.e., average crib notes performance halfway through the trials) the participant received 197 points.

The training phase consisted of 60 trials for each gesture, with the presentation order randomized such that no gesture appeared in more than two successive trials. At the start of each trial, the target word appeared at the top of the screen. The current score, as well as the target circle, were also visible. To begin the trial, the participant placed the pen tip in the target circle, which caused the gesture guide to become visible immediately. The participant then drew the gesture on the screen, which left a visible 'ink trail'. When the pen was lifted, all on-screen content was hidden for 1000 milliseconds. The participant was then provided with KP and KR, consisting of the target gesture along with the participant's trajectory overlaid and the score for the current stroke Figure 5.2). After 1500 milliseconds of exposure to the KP and KR, the screen went blank for 1000 milliseconds, and then the next target word appeared. All training was performed with the right hand.

Participants performed retention and transfer tests 15 minutes after completing the training phase. The retention test was similar to the training, with each participant performing 16 trials (4 per gesture, with the order randomized), but no guide (KR or KP) was provided. Participants were instructed to draw the gestures from memory. They were also reminded of the four target words and told to take as much time as needed before drawing the gestures. The participants were not shown any of the gestures. The transfer test was identical to the retention test (16 trials with no guide, KP, or KR), but performed with the left hand. Approximately 24 hours later, each subject completed the retention and transfer follow-up tests again.

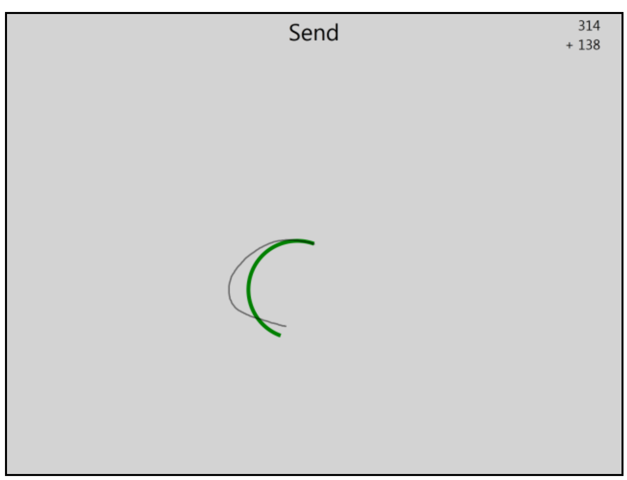

Figure 5.2. Inter-trial screen showing KR (score in top right) and KP (trajectory overlay).

5.4. RESULTS

Training, retention, and transfer data was analyzed with three mixed-design ANOVAs, and post-hoc tests were conducted using Tukey's HSD. Prior to each analysis, trials where the participant performed the incorrect gesture were discarded, as the focus of this study was on the form of the gesture, not on gesture recall. As we were interested in errors due to performance (i.e., slips) rather than cognitive errors (i.e., mistakes), these errors were removed. These errors were spread evenly across all guides, came primarily from the training data, and resulted in less than 1% of data being removed from training, retention, and transfer phases.

5.4.1. GESTURE SIMILARITY

The similarity of each stroke to the template was computed by resampling the template and the participant strokes to 128 evenly spaced points, then computing the root mean square error (RMSE) of the Euclidean distance between corresponding points. This method is sensitive to both scale and rotation, as participants matched the template gestures along those dimensions as well as shape. A graphical representation of the similarity data separated by GuideType is shown in Figure 5.3, and similarity by Gesture is shown in Figure 5.4.

While RMSE is not the most popular method in gesture recognition, there are several reasons that make it a good choice for analyzing accuracy of motor production. First, the use of RMSE gives a direct and accurate measurement of the participant's ability to produce the target gesture. Secondly, it does not rely on a collection of high-level features (see, e.g., Rubine's algorithm (1991)), the selection of which will change with the next advancement in gesture recognition. That is, our results are independent of the current state of the art in gesture recognition. It is also worth noting that the error was also analyzed using the error measure used for the $1 recognizer (Wobbrock, Wilson, & Li, 2007), as well as by using the number of 'points' awarded on each trial, and the same patterns emerged from the resulting data.

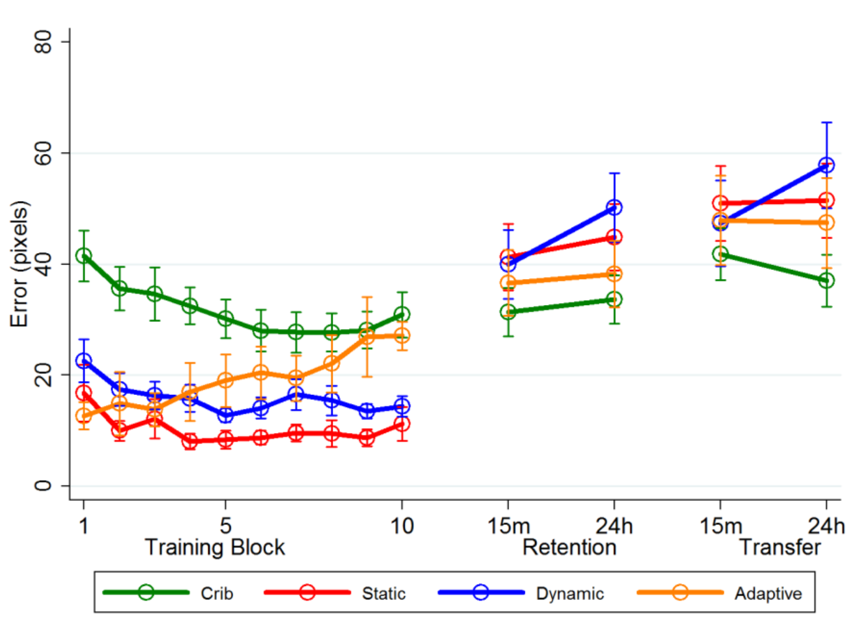

Figure 5.3. Error for training, retention and transfer, for each guide type. There is an apparent tradeoff between performance during training, and the amount of learning, measured by retention and transfer.

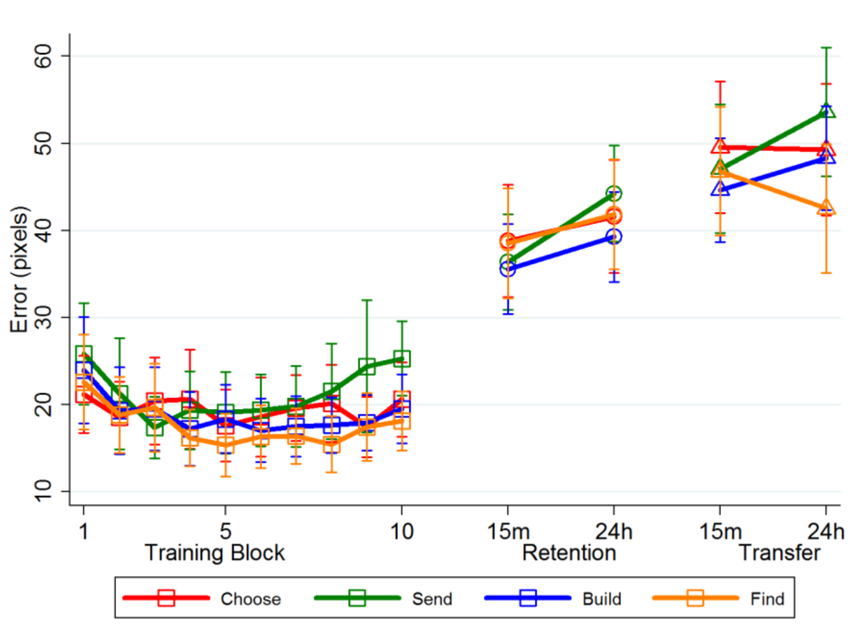

Figure 5.4. Error for training, retention, and transfer, for each gesture. There are no appreciable differences between the performances of each gesture. Error is primarily due to the type of training guide used.

5.4.1.1. Training

The training data was blocked by averaging six consecutive trials for each gesture, resulting in 10 training blocks. As the error distributions were skewed, a log transform was applied to the RMSE values to normalize them before conducting the ANOVA. The same log transform was applied to all RMSE values before the analysis in subsequent sections. A three (GuideType) x four (Gesture) x ten (Block) mixed-design ANOVA was conducted with Gesture and Block as within-subjects factors and GuideType as between-subjects factor (summarized in Table 5.1).

Table 5.1. ANOVA results for Accuracy.

Factor	F	p	ω^2
GuideType	$F_{3,32} = 34.34$	0.00	0.45
Gesture	$F_{3,96} = 4.20$	0.01	0.01
GuideType x Gesture	$F_{9,96} = 1.44$	0.18	0.01
Block	$F_{9,288} = 5.08$	0.00	0.05
Block x GuideType	$F_{27,288} = 7.45$	0.00	0.06
Block x Gesture	$F_{27,864} = 0.15$	0.12	0.00
GuideType x Block x Gesture	$F_{81,864} = 1.08$	0.31	0.00

Post-hoc tests on GuideType showed that all guide types produced significantly different scores during training. From lowest to highest error produced, the guides are: ‘Static’, ‘Dynamic’, ‘Adaptive’, ‘Crib’. Participants generally improved during training, with the error in the first block being significantly higher than the last block. The exception to this is with the adaptive guide, where participants progressively decreased in performance, due to the guide being removed earlier in the trial as they progressed through the training.

The main effect of gesture shows that the ‘*Find*’ gesture was significantly easier to perform than the ‘*Choose*’ and ‘*Send*’ gestures, and *‘Build’* was easier to perform than *‘Send’*, but the effect sizes were very small, and therefore were not analyzed further.

5.4.1.2. Retention

To analyze the retention data, the four trials for each gesture were averaged per participant and analyzed using a four (GuideType) x four (Gesture) x two (Delay) mixed-design ANOVA with GuideType as a between-subjects factor and Gesture and Delay as within-subjects factors. Gesture was not found to significantly affect retention scores ($F_{3,\,96}$ = 0.30, p = 0.83), so this factor was pooled and the ANOVA was re-computed.

Both Delay ($F_{1,\,248}$ = 8.90, p = 0.0031, ω^2 = 0.02) and GuideType ($F_{3,\,32}$ = 2.93, p = 0.049, ω^2 = 0.09) were found to significantly affect retention scores. Participants trained with crib-notes or the adaptive guide had significantly lower retention scores than those trained with the traceable guides. There was no significant difference between the retention scores of participants trained with either of the traceable guides. There was also no significant difference in the retention scores of participants trained with the adaptive guide or the crib notes. Performance on the 24-hour follow-up was poorer across all participants, compared to the 15-minute follow-up.

5.4.1.3. Transfer

To analyze the transfer data, all four trials for each gesture were averaged and analyzed using a four (GuideType) x four (Gesture) x two (Delay) mixed-design ANOVA with GuideType as between-subjects factor and Gesture and Delay as within-subjects factors. Again, the Gesture factor was not significant ($F_{3,\,96}$ = 1.23, p = 0.30) and was pooled in the reported results.

The transfer results mimic the same pattern as the retention results, as evidenced by a Pearson's correlation (ρ = 0.85, p < 0.001). While the ANOVA did not report significant main effects, this similarity in results to the retention results demonstrates the potential utility of transfer scores. One reason for the lack of significant main effects is the performance improvement in the crib-notes trained participants following the 24-hour rest period, contrasted with the decreased performance of the participants trained with the dynamic guide.

5.4.2. Duration

Duration was comptued as the time from the pen's initial contact with the screen to when the pen left the screen. This measure includes any time the participant spent consulting the guide as well as the time to draw the gesture. Duration data for the training, retention, and transfer phases are shown with results separated by GuideType in Figure 5.5, and results separated by Gesture in Figure 5.6.

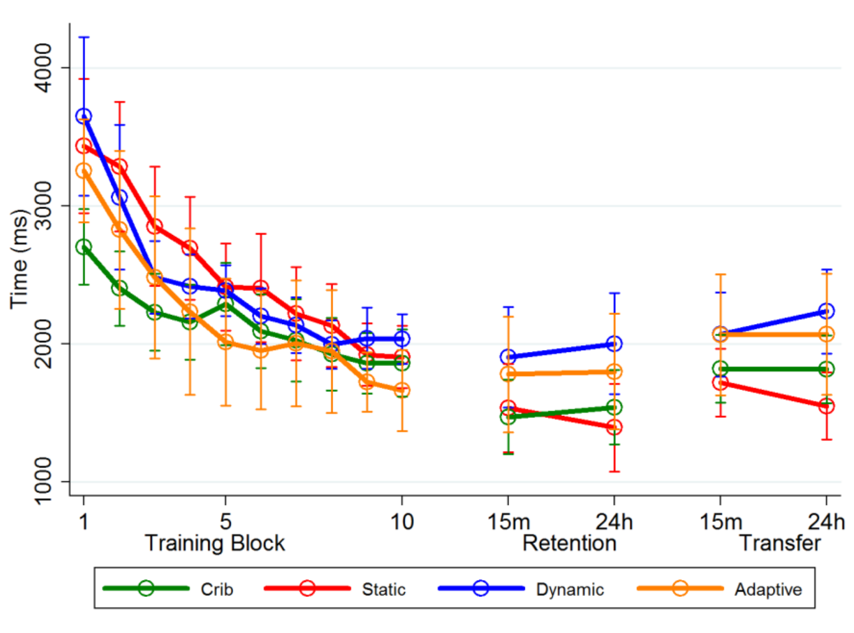

Figure 5.5. Duration of stroke during training, retention, and transfer, for each guide type. Guide type has little effect on the speed, allowing the tracing-based guides to provide high accuracy without a decrease in input speed.

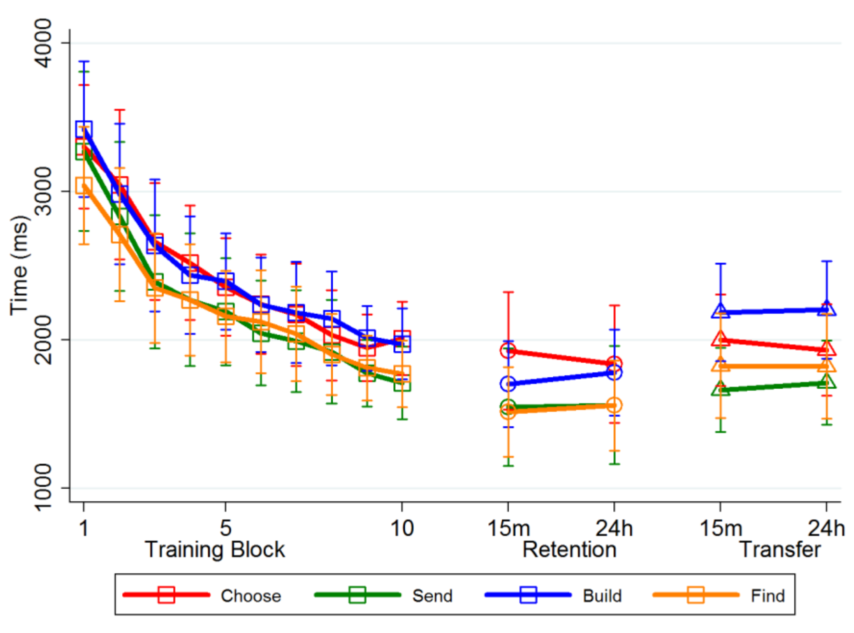

Figure 5.6. Duration of stroke during training, retention, and transfer, for each gesture. The ‘Send’ and ‘Find’ gestures are performed slightly faster than ‘Build’ and ‘Choose’.

5.4.2.1. Training

The mixed design ANOVA showed a main effect of Block ($F_{9,\,216} = 25.5$, $p < 0.001$, $\omega^2 = 0.13$). There was a significant decrease in duration with nearly every block. There was also a main effect of Gesture ($F_{3,\,72} = 19.3$, $p < 0.001$, $\omega^2 = 0.01$). The 'Send' and 'Find' gestures were performed significantly faster than the 'Choose' and 'Build' gestures, but the effect size is very small. GuideType had no effect on duration ($F_{2,\,24} = 0.53$, $p = 0.59$, $\omega^2 = 0.02$).

5.4.2.2. Retention

The retention data shows only a main effect of Gesture ($F_{3,\,96} = 3.95$, $p = 0.01$, $\omega^2 = 0.02$). 'Choose' is performed significantly slower than 'Send' and 'Find', but again, the effect size is quite small. As with the training data, the guide type has no effect on duration ($F_{3,\,32} = 0.60$, $p = 0.62$, $\omega^2 = 0.03$).

5.4.2.3. Transfer

There was a main effect of Gesture during transfer ($F_{3,\,96} = 15.86$, $p < 0.001$, $\omega^2 = 0.04$). Post-hoc tests revealed that all gestures were significantly different from each other. In increasing order of duration, the gestures were: 'Send', 'Find', 'Choose', and 'Build'. These results were very similar to the training durations, except that the variance between gestures is increased, resulting in higher significance.

5.5. Discussion

The type of guide used during training has clear effects on the behavior during training, as well as the retention and transfer scores. With high performance during training, there is little learned, but with low performance the participants retained substantially more. Additionally, by adapting the guide over time, the participants are able to balance performance and learning. These results have important implications for both the design of gestural guides, as well as for how they are evaluated.

5.5.1. Guide Design

The three 'traditional' gesture guides (i.e., *static-tracing, dynamic-tracing*, and *crib notes*) showed performance improvements with training. There was an obvious and significant difference during training between crib-notes and the traceable guides with the crib-notes-trained participants performing much worse. Looking exclusively at the training data, it appears that the traditional guide types provided equal amounts of learning, but the baseline performance for crib-notes was worse. However, this was not the case. The retention scores, which mimic an expert-usage scenario showed the lowest error for crib-notes.

The newly proposed adaptive guide shows a much different result. While the traditional gesture guides lead to improved performance during training, the adaptive guide shows a gradual decrease in performance. This performance loss is easily explained. As the participant completed more trials, the guide disappeared earlier and earlier during gesture execution, forcing them to perform more of the gesture without the guide in place. In contrast to the traceable guides (static and dynamic), the adaptive guide actually provides a relatively smooth transition from novice to expert; there is not a substantial decrease in performance when the guide is removed. In contrast to the crib-notes guide, the adaptive guide provides a much more usable interface to novices, allowing direct tracing and high accuracy at the beginning of the training phase.

With respect to the three traditional guides, it seems that the more guidance given during training, the worse the learning. The participants using traceable guides had the most guidance and the worst performance in retention and transfer. Conversely, the participants using crib-notes had the least guidance during training but the best performance during retention and transfer. These results are explainable by the guidance hypothesis, and only become apparent within the retention and transfer paradigm. This shows an apparent tradeoff between immediate performance and learning. In addition, it appears that the dynamic guide has no benefit for performance or learning. The dynamic-traceable guide resulted in worse performance during training than the static guide, and users of this guide showed little to no learning of the gestures during retention and transfer. This is interesting, but not entirely surprising. That is, anytime the dynamic guide updated, it would necessarily deviate from the template trajectory. Thus, participants who attempted to trace the guide would also deviate from the template. However, unlike with crib notes, participants did not attempt to use the post-trial feedback (KP) to correct their previous errors and learn the correct performance.

It is clear that the adaptive guide provides a balance between initial usability and long-term learning. Implementing the adaptive guide in practice, however, is not necessarily straightforward. In the experiment, the number of trials was fixed, and a simple linear model allowed us to gradually remove the guide. In practice, the user is continuously interacting with an interface for an unknown length of time. A simple way to implement the adaptive guide would be to monitor the number of times each gesture was accessed, and provide less guidance each time the guide is accessed, up to some pre-determined threshold.

Other approaches to achieving 'adaptive' style guides are possible as well. One simple way would be to add an access cost to the guide (e.g., a delay (Bau & Mackay, 2008)), and let users self-regulate the appearance of the guide. This approach was tested in pilot studies, but did not have the desired effect. Many users simply accepted the delay as an intrinsic cost of using the interface, even when the delay was long (e.g., over 1 second). This behavior was present even through it was clearly explained that they did not have to wait for, and use the guide.

In contrast to previous studies, the increased learning observed with crib-notes or the adaptive guides cannot be attributed to increased effort. In fact, the crib-notes guide requires the least effort, as drawing the gesture does not require careful tracing. This was evident with the duration data, which showed the crib-notes had marginally faster input time during training, especially in the earlier trials. If effort were the major determinant of learning, one would expect the dynamic-tracing guide to have the best learning outcomes, as it requires more cognitive effort (and time) to follow the constantly updating trajectory.

Overall, the type of gesture guide has little impact on duration during training, retention, or transfer. This is consistent with Bau and Mackay's (2008) work that found no difference in the input time between a help menu and a dynamic guide Participants became faster over time as they became more familiar with the gestures. This is an important find, as it seems to violate the speed-accuracy tradeoff as long as the guide is available. That is, accuracy during training using traceable guides is substantially greater than crib-notes, but the speed of execution is similar, particularly after the first few training blocks.

With these results, it is important to recognize that not all gesture-based interactions target expert-level skill acquisition. Many interfaces are used on an infrequent or casual basis, where the user is not expected to perform at maximum efficiency. For these situations, heavy guidance (with ease of use) is more desirable than limited guidance (with better learnability). However, if efficient expert use is of concern, one may consider a type of guide that initially leads to lower performance, but increases learning and speeds the progression from novice to expert.

5.5.2. Evaluation Paradigm

During retention and transfer, the performance of participants trained with crib-notes was significantly better than all the other participants. This demonstrates a severe limitation of the training-only evaluation methods, as the performance changed drastically when guidance was removed. If only the performance data were considered, as in previous studies, one would have reported that the traceable guides were superior with a very large effect size of $\omega^2 = 0.45$. However, when looking at the retention data to assess the learning that occurred, crib-notes proved more effective.

In addition to the focus on learning, another important difference of this work is the use of gesture accuracy as the outcome measure. Previous evaluations tend to use recall as the primary measure (Bau & MacKay, 2008; Zhai & Kristensson, 2003). However, as gesture sets become more complex, with a variety of hand shapes and strokes, the ability to articulate precise movements will be a very important measure of user proficiency. Additionally, analyzing the execution of a gesture allows performance improvements to be seen for each block, providing a more detailed look at how users improve with each system.

The performance of nearly all participants suffered after a 24-hour break, with participants that used the traceable guides suffering the greatest performance loss. In general, this indicates users were ‘forgetting’ how to replicate the gesture precisely. When learning occurs, these losses are much less dramatic. These findings demonstrate the utility of using delayed retention and transfer, as the separation between participants who learned and those who did not becomes greater after a night of sleep (Savion-Lemieux & Penhune, 2005). This makes it easier to pinpoint the factors influencing learning.

This study has important implications for the evaluation of future gesture-based interfaces or interactions involving relatively complex movements. When reporting on the effectiveness of various gesture guides, it is imperative that learning be properly assessed, so that designers are aware of the implications (both short and long term) of using an interface. While this places an additional burden on the investigator, it is critical when making claims about the learnability of these interfaces.

It is also important to note that experts do not tend to reproduce the template precisely. In general, they make simplifications that deviate from the template and allow them to produce the gesture more efficiently, but still be recognizable to the system. While this behavior would not be represented by the strict RMSE measure accurately, our experiments were focused on the reproduction of specific gestures. Similar to the use of a guide that does not adapt to scale or orientation, this simplification allowed us to study the effect of guidance in a controlled fashion. While we anticipate that expert gestures will show more error in a real-world scenario, we do not believe that will negate the guidance hypothesis as it applies to gesture learning.

5.6. SUMMARY

The results presented here show that we should have learning at the forefront when designing gesture guides, and leverage as much existing knowledge as possible. We demonstrated that excessive guidance hinders learning, and that this can be mitigated in a user-friendly manner by introducing faded feedback. We also showed how new experimental paradigms can reveal some of the existing problems with gestural interfaces. Lastly, we hope to have demonstrated the utility of drawing from existing knowledge from the motor control and motor learning domains. There is thus a plethora of knowledge from these areas that can be adapted for the design of gestural interfaces.

With gesture systems that target efficient, expert usage, it is critical to consider the long-term learning of gestures and the effects that guidance can have on both performance and learning. Using a novel 'adaptive' guide, we demonstrated that detrimental effects of heavy guidance can be overcome, and that gesture guides can produce a smooth transition from novice to expert. We have also shown that traditional evaluation techniques can be ineffective in measuring learning itself, and that gesture guides may have unforeseen consequences. We have used an established technique from the area of motor learning to evaluate a representative set of gesture guides. We hope researchers and practitioners will use these results in the design and evaluation of future gestural interfaces. These findings are important to gesture learning, as they demonstrate improved methods of training the procedural component of gesture learning and provide a clear mechanism that can be used by designers and researchers to evaluate the degree to which new guides improve this aspect of learning.

5.6.1. RELEVANCE TO GESTURE LEARNING

Within the framework outlined in Section 1.3, we have demonstrated the importance of feedback in learning the correct execution of the gesture. We demonstrated that providing excessive feedback during training could cause the user to rely on the feedback rather than learn the appropriate movements and achieve independence from the guides. We intentionally limited the size of the gesture set such that learning the declarative component of the gestures would not be a problem and participants would be able to focus solely on learning the procedural component. In this way, we can study those two aspects of gesture learning independently.

5.6.2. LIMITATIONS

This study is limited in the small number of gestures that were studied, and the complexity of the gestures. While a real-world study involving the learning of dozens of gestures would provide more valid results, such a study is impractical to conduct. With this being said, it is likely that the results of this study will remain true even with more complex 2D stroke gestures. From the training data, performance plateaus after the 5th or 6th block, so participants are already 'overlearning' the gestures to a degree, yet the effect is still clear. Secondly, the guidance hypothesis that underlies the results has been demonstrated in a variety of tasks (including timing, force production, and figure drawing). While this does not guarantee that it will extend to gestures that are more complex, there is nothing to indicate the contrary. Lastly, the gestures were chosen to represent a wide range of potential motor actions. It is not clear, however, that the results would generalize to a 3D stroke gesture scenario, where the task is fundamentally more complex and would benefit more from guidance.

Chapter 6

Full Body Gesture Training with an Augmented Reality Mirror[1]

As gesture learning is not limited to two-dimensional stroke gestures, we were interested to see how well the gesture-learning framework applied to movements that are far more complex. To that end, we examined the training of full-body, 3D gestures. We leveraged prior results and emphasized visual feedback to encourage learning of the declarative component of gesture memory. We also leveraged adaptive, real-time feedback as well as summary feedback to encourage rapid learning of the procedural components of gestures. This chapter provides evidence that the framework outlined in Section 1.3 can be extended to complex interactions.

6.1. Learning Full Body Gestures

It is clear that dynamic visual feedback plays an important role in motor skill acquisition of 2D gestures, and that the form of the feedback can influence the degree to which a skill is learned. As there are many interfaces being developed which support 3D, free-space gestures for input (Sato et al. 2001; Frontier Developments, 2011), we were interested in developing a new technique for training these types of gestures. In addition, the system could be developed such that it also supports the training of more general full-body movements such as those found in dance, sports or other domains. Learning these types of motor skills can be challenging, requiring hours of training and repetitive practice (Ward et al., 2004). Often, learning involves enrolling in classes where an instructor leads a group through a set of exercises. Ballet dancers often use mirrors to receive visual feedback in addition to coaching (Dearborn & Ross, 2006). While there may be no replacement for expert coaching, less formal self-paced learning may be more desirable for in-home practice, to supplement professional coaching, or for hobbyists.

---

[1] The majority of this chapter has been published as Anderson, F., Grossman, T., Matejka, J., and Fitzmaurice, G. "YouMove: Enhancing Movement Training with an Augmented Reality Mirror." In the Proceedings of User Interfaces and Software Technology, 2013, pp. 311-320.

In this chapter, we present YouMove (Figure 6.1), a system for learning full body movements. YouMove is comprised of a Kinect-based recording system and a corresponding training system. The recording system is designed to be easy to use, so anyone can capture movement sequences and annotate them for learning, without the need for complicated motion capture hardware or software. The training system uses the recorded video and 3D movement data to guide the trainee through a series of interactive stages. The training system augments a traditional 'ballet mirror' experience by using a half-silvered mirror with graphic overlays for guidance and feedback. The use of a mirror allows for zero latency, high fidelity feedback. YouMove appears to be beneficial for several domains of movement, such as yoga, dance, and physical therapy, with more domains possible with further advances in sensing.

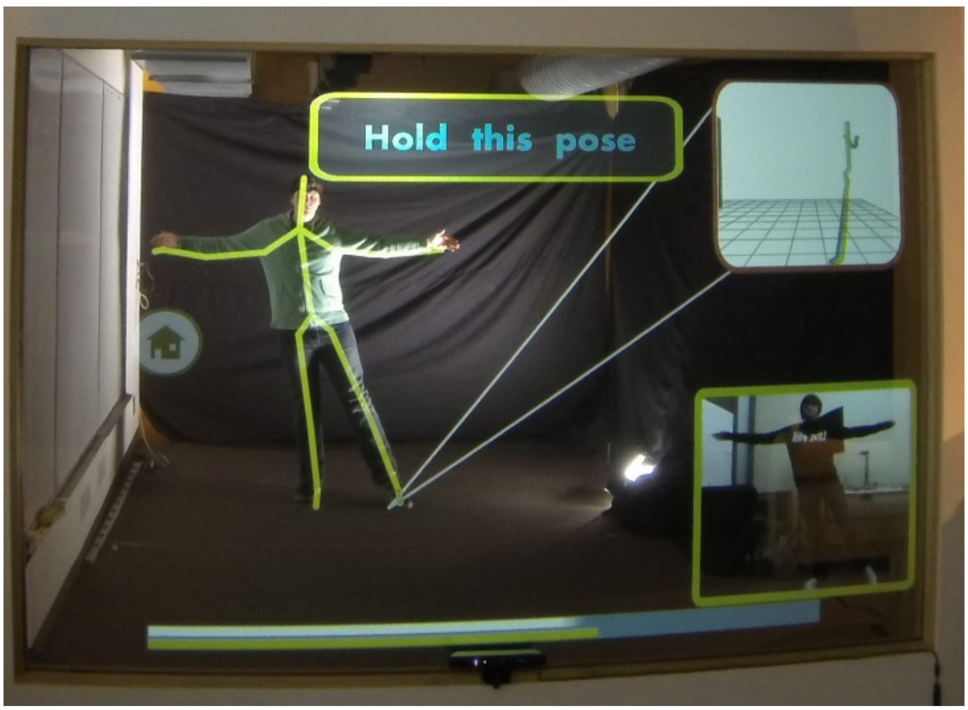

Figure 6.1. YouMove allows users to record and learn physical movement sequences. An augmented reality mirror provides graphic overlays for guidance and feedback. Note that for this photo the virtual viewpoint was vertically repositioned to account for the offset of a head-mounted camera, and floor lighting was used to reduce glare.

This chapter provides several contributions. First, we provide a generalized full-body movement training system, YouMove. Next, we provide design guidelines, implementation details, and interaction techniques for a whole-body, interactive, augmented reality mirror. Third, we provide an evaluation demonstrating YouMove's effectiveness over a traditional video approach. Last, we analyze learning and preferences with YouMove and provide directions for future work.

6.2. Insights on Traditional Instruction

To gain insights that could inform the design of YouMove, we observed professional yoga and ballet classes.

Our observations were consistent with the strategies discussed in the recent work of Velloso, Bulling, and Gellerson (2013). We observed a feedback loop between trainer and trainee consisting of demonstration, performance, and feedback. For example, both the ballet and yoga instructors would demonstrate the movement sequence, then have the trainees repeat the sequence as they made comments and physical corrections to the trainees' movements.

We also observed the ballet instructor using an adaptive guidance approach, reducing the amount of guidance as the class progressed. The trainees started at the barre, then moved to the mirror, and finished the class with more complex sequences without the use of the mirror. In both classes, instructors provided timing cues, motivation, and alternative metaphors for thinking about the movement.

6.3. Design Guidelines

Based on our observations of instruction strategies and a review of previous literature, we have developed a set of design goals to guide the development of the YouMove system. We believe these guidelines are applicable to movement training systems in general.

6.3.1. Leverage Domain Knowledge

Experienced trainers have more knowledge than what can be expressed in a recorded movement. Our observations indicated that feedback from instructors is based on domain knowledge. The experts know how to segment movement and which parameters of the movement are important. The authoring system should support the capture of this domain knowledge with minimal effort.

6.3.2. Motivate the User

Engagement and motivation is an important part of learning (Bederson, 2004). The training system should provide feedback on the user's progress, encourage the user to continue with their practice, and make it an enjoyable experience.

6.3.3. Simple Presentation, Low Cognitive Load

While practicing, the users should maintain a low cognitive load (Van Merriënboer & Sweller, 2005). The user's attention should be on the movement, not on interpreting user interface elements that are complex. A direct representation of the movement and simple scoring measures should focus the user on learning the movement.

6.3.4. Adaptive Guidance

Excessive guidance and video demonstrations can hurt learning, as users come to rely on the guides (Palmiter & Elkerton, 1991; Schmidt & Wulf, 1997). As the user learns the movement, guidance should be reduced.

6.3.5. Average Feedback

Feedback on individual movements can cause trainees to focus on small errors, rather than the larger systemic errors. Aggregating feedback from several performances allows trainees to see and correct systemic errors in their movement (Yao et al., 1994). While summary feedback (e.g., overlaying all performances rather than averaging them into a single skeleton) would result in improved learning (Lavery, 1962; Winstein & Schmidt, 1990), averaged feedback simplified the visual complexity.

6.3.6. User Driven learning

Trainees have varying skill levels and preferences that will dictate their training needs. The system should allow users to progress at their own pace to maximize learning (Wulf, Raupach, & Pfeiffer, 2005).

6.4. Augmented Reality Mirror Implementation

While there have been previous implementations of virtually enhanced physical mirrors (Doctorow, 2011; Microsoft Research, 2012), we are unaware of a technical setup similar to our own in the research literature. This section contributes a technical description of our Augmented Reality Mirror setup, which could have implications that go beyond the YouMove system.

The floor-to-ceiling mirrors often found in ballet studios inspired our use of the mirror for YouMove. These mirrors allow dancers to see their movements, providing them with immediate visual feedback (Dearborn & Ross, 2006). By augmenting a traditional mirror with interactive content, we provide additional information that helps trainees learn a movement.

6.4.1. Configuration

The configuration of the Augmented Reality Mirror is illustrated in Figure 6.2. The display consists of a 3.2m x 1.8m pane of glass with a half-silvered mirror film applied to one side, and a diffuse film applied to the other. The mirror film was applied to the surface facing the user. The film (Supreme Silver 20 – www.apexfilms.ca) transmits 16% of the visible light, and reflects 58%, resulting in a highly reflective surface while still allowing projected light to pass through. Several other mirror films were tested, but the others absorbed more of the projector light, or did not reflect the user's image as clearly. The diffuse film serves to diffuse the light from a rear-mounted projector (Mitsubishi FD630U, 1920x1080 pixels).

A Microsoft Kinect was mounted below the mirror to track the user. The location of the mirror was specified in the coordinate space of the Kinect. The position of the user's head and the corners of the mirror define an asymmetric projection matrix used to render on-screen content.

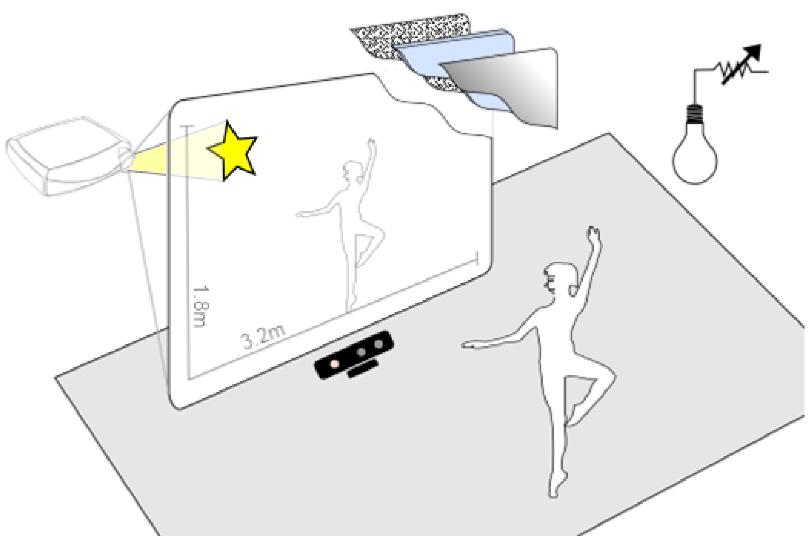

Figure 6.2. Overview of system design showing projector, layered screen, dynamic lighting, Kinect and user location.

6.4.2. Lighting

While the system is usable in varying ambient lighting conditions, the experience was improved by being in an environment with controlled lighting. With bright ambient light and a dark projection image, the user focuses solely on their reflection in the mirror. With low ambient light and a bright projection, the user's reflection disappears. With a moderate amount of ambient light, the user's reflection and the projected image are both visible (Figure 6.1). The ability to control lighting allows the system to shift the user's attention between the relevant images (virtual or reflected).

To manipulate the ambient light, a servo motor was mounted to a light's dimmer switch and connected to the PC through an Arduino over USB. The servo motor was mounted using modular building blocks (LEGO), allowing the motor to be easily removed and accurately replaced later. The physical actuation of existing dimmer switches would not require the end user to modify any existing wiring or manually adjust the lighting.

6.4.3. Interaction

Mirror-based augmented reality offers unique opportunities for interaction. The user's reflection directly activates on-screen components, allowing for direct manipulation of a 2D interface from 3D free-space. This *reflection selection* provides zero latency feedback on hand position, allowing quick positioning (Ng et al., 2012). Buttons are activated by dwelling the hand over the button. During the dwell period, the button expands, providing feedback and increasing the activation area in case the user's hand drifts.

We implemented two types of buttons. Global menu buttons are located on the left side of the mirror, so they will not be triggered accidently during training. The vertical location of the global buttons adapts to the user's height, so that menu items are not out of reach. Quick-access contextual buttons are presented near the user's head, and only appear when required. These buttons are convenient, and their body-centric positioning allows them to be activated by a 'gesture posture', as their location relative to the body is constant, similar to Virtual Shelves (Li, Dearman, & Troung, 2009).

6.5. YouMove Implementation

YouMove is composed of a simple program to record data, and a separate training system for playback of that data. The authoring and training system were designed to run independently. The recording software was written in C#, using the WPF framework and the Kinect SDK (v1.6). The training software is written in C++, using openFrameworks, OpenGL and the Microsoft Kinect SDK v1.6. All software was run on a Windows 7 PC, with 12 GB of RAM and a dual core processor.

The only calibration required for the training system is to specify the location of the mirror when it is first set up.

6.5.1. YouMove Authoring System

We developed custom software to capture a trainer's movement and domain knowledge. By simplifying capture, YouMove allows experts to contribute learning material without the need for complicated motion capture hardware or software. The recording system allows authors to record themselves performing the movement. The system captures video, audio and 3D skeleton movement data of the author.

6.5.1.1. Recording

After launching the software, the author is presented with a screen that has a single 'Record' button as well as the current video stream from the Kinect with a skeleton overlay. To capture movement, the author presses record, performs the movement, and then presses the stop button. Pressing stop takes the author to the editing interface.

6.5.1.2. Editing

The editing interface (Figure 6.3) allows authors to trim the recording to eliminate unwanted data – such as walking to or from the capture volume. The author can also specify global parameters for the movement, i.e., timing, smoothness, precision, or stability. These parameters were used by the training system when providing feedback.

Authors then specify keyframes for the recorded movement. Keyframes are postures within the movement that are particularly important for a trainee to match during the movement. For example, a keyframe for a baseball throw may be when the hand reaches peak extension.

Keyframes are specified by navigating to the desired frame and then directly clicking on joints to specify the *important joints* for that keyframe. When a joint is selected, the current frame becomes a keyframe. These keyframes and important joints will be used in the training system to provide tailored guidance and feedback.

Authors can also associate an additional audiovisual recording with individual keyframes, allowing the trainer to provide additional information regarding the movement. This annotation is done by clicking the 'record' and 'stop' buttons below each keyframe marker. Annotations provide information that may not be immediately obvious to a trainee from seeing the author's movement, or to discuss common pitfalls to avoid. For example, to throw a baseball, an author may annotate the keyframe with a short clip explaining that the elbow should be at approximately 90 degrees.

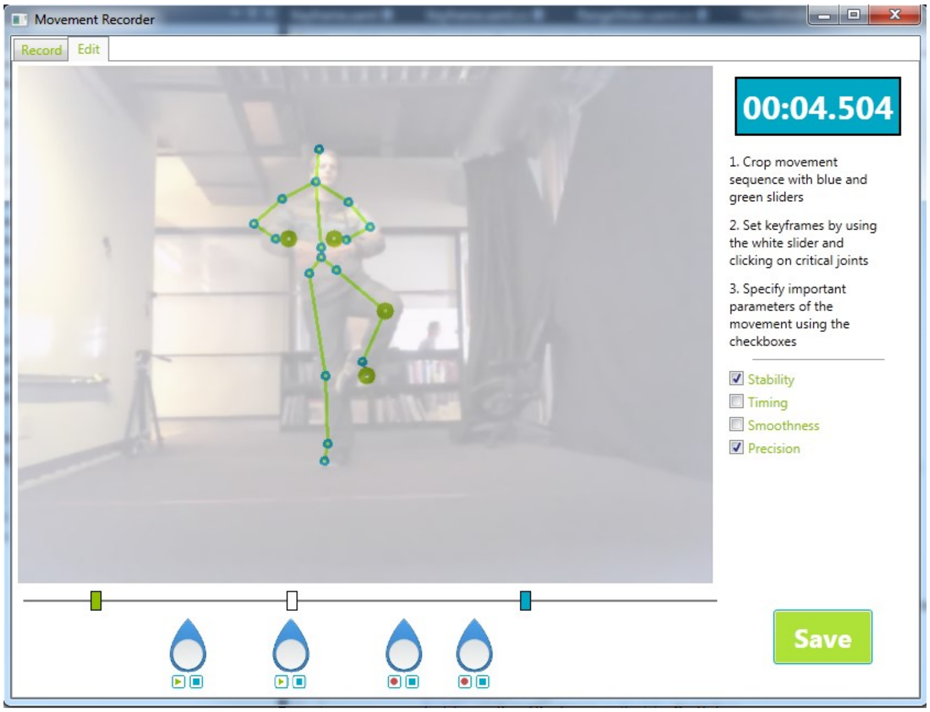

Figure 6.3. The editing interface allows authors to specify keyframes and global movement parameters. Each keyframe specifies the important joints for that moment, and can be associated with a recording that provides detail.

6.5.1.3. Saving and Sharing Captures

The user saves the capture by clicking the Save button. The data is saved as media files (.mp4 video and .wav audio) and plain-text files containing time stamped skeleton locations and keyframe metadata used to synchronize the data.

6.5.2. YouMove Training System

The training system is used to teach the movements previously recorded with the authoring system.

6.5.2.1. MOVEMENT GALLERY

The initial screen of the system presents a gallery of movements that the user may wish to learn (Figure 6.4). All buttons and icons throughout the entire system are selected using the reflection selection technique.

6.5.2.2. QUERY BY EXAMPLE

Selecting the magnifying glass allows users to search for a movement by example. The search screen instructs users to hold a representative posture of the desired movement. Once the user stays still, the system searches the movements in the library for the best match, presenting the most similar movements in a grid for the user to select.

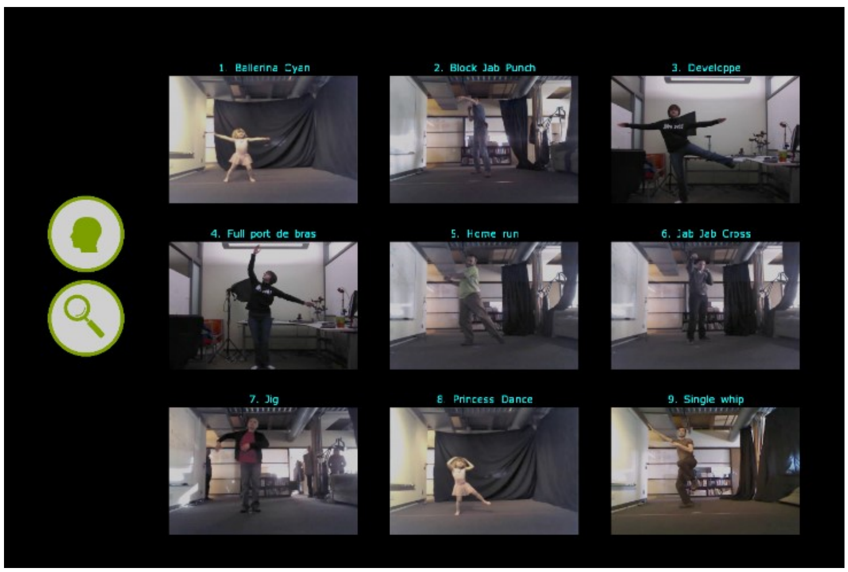

Figure 6.4. The movement gallery allows users to change profiles, query by example, and select a movement.

The posture similarity heuristic used for the search is the same as what is used for the scoring measure used in training. That is, it takes a 'snapshot' of the posture the user is performing, and compares it to each of the keyframes in each movement of the library. The ability to query by movement may also be useful to search for movements where the movement is emphasized, rather than the pose.

6.5.2.3. SKELETON ALIGNMENT

A fundamental feature of the YouMove training system is guidance and feedback based on a comparison of the author's and trainee's skeleton. The Kinect provides skeleton tracking, reported as 20 joints with 3D positions updated at 30Hz. The tracked joints include large body parts, such as the hands, arms, torso, legs, and the head, but fine movements (e.g., the fingers) are not tracked.

To properly calibrate the author's training skeleton to each user, it must be scaled and translated to match the user's size and position. This is necessary for the skeleton-driven feedback and for accurate scoring. The spatial alignment is done by aligning the hips of the author skeleton with the hips of the trainee. Orientation is not aligned, as it is assumed that the trainee and trainer will be performing the movements in the same orientation relative to the Kinect.

Scaling the skeleton is achieved by dynamically resizing each bone in the trainee skeleton to match the size of the corresponding bone in the trainer. This scaling is done hierarchically from the hips to propagate changes throughout the skeleton. This method is necessary as a simple uniform scaling would not accommodate users with proportionally different limb lengths. We have found that this skeleton algorithm works robustly, and allows the system to work well even when the trainee has a significantly different age, height, or weight from the author.

6.5.2.4. Scoring and Stage Progression

The YouMove system incorporates several elements of gamification designed to motivate the user (Li, Grossman, & Fitzmaurice, 2012). Training is composed of a series of stages, and each stage scores the user's performance based on the similarity between their movement and the target movement. If their performance is high enough, they get a gold star and are allowed to progress to the next stage. To avoid frustration, a stage is also unlocked if they repeat the stage twice.

Each keyframe is scored based on the joint with the maximum error, measured by Euclidean distance. Only *important joints* specified in the authoring tool are used to compute the score. To allow for small errors in timing, a window of 0.5s on each side of the target frame is searched to find the best matching posture. If the author has specified that timing is important in the authoring tool, this window is decreased to 0.25s. The maximal Euclidean distance is mapped to a score using a linear mapping, with 0 error being a perfect 10, and 0.15m of error resulting in the score of 7.5 needed to get a gold star and unlock subsequent stages. If precision is specified as a global parameter, this mapping is modified so 0.10m results in a score of 7.5. These values were determined by experimentation.

6.5.2.5. TRAINING STAGES

Once a movement is selected from the gallery, the system progresses through five stages: *Demonstration*, *Posture Guide*, *Movement Guide*, *Mirror*, and *On Your Own*. The user can navigate through the stages using a selection screen (Figure 6.5, left), although the locking mechanism forces the user to initially perform the stages in order.

The stages progressively introduce the movement to the trainee, and gradually reduce their reliance on guidance and feedback. Each stage presents the user with unique challenges and a different context to perform the movement in, reducing the negative impact of specificity of learning (Proteau, Marteniuk, & Levesque, 1992). The Posture Guide repeats twice, and the Movement Guide, Mirror and On Your Own repeat five times, allowing users to practice without being interrupted.

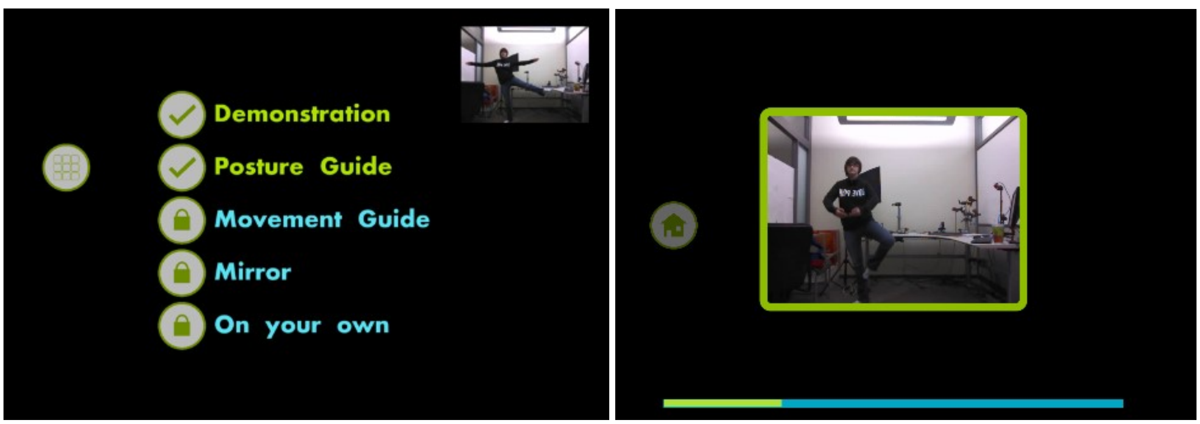

Figure 6.5. Left) Stage selection interface allowing users to begin one of the unlocked stages. Right) Demonstration stage.

6.5.3. DEMONSTRATION

The demonstration stage plays the recorded video for the trainee (Figure 6.5, right). This stage is designed to be simple and familiar, so the focus is on the movement to be learned.

Audio is played alongside the video to help with timing. At the start of the movement, a pre-recorded voice speaks the word 'and', and each keyframe is counted out in sequence, i.e., 'One', 'Two', etc. The use of numbers (rather than a metronome click) helps trainees remember each movement in the sequence. This counting is present in all other guides except for the 'On Your Own' guide. A progress bar on the bottom of the screen is synched to the video playback.

6.5.3.1. Posture Guide

The posture guide (Figure 6.6) helps trainees refine their body position by pausing the movement at each keyframe and providing real-time feedback on their errors. This stage is inspired by the tutorial system found in Pause and Play (Pongnumkul et al., 2011), as it halts the tutorial and allows trainees to work at their own pace. The stage pauses until the user holds a stable position for one second, or until five seconds have elapsed. An additional progress bar on the bottom of the screen represents the amount of time they have currently been stable. If stability is specified as a global parameter in the authoring tool, the threshold for detecting stable movement is lowered from 5 to 3 cm per frame.

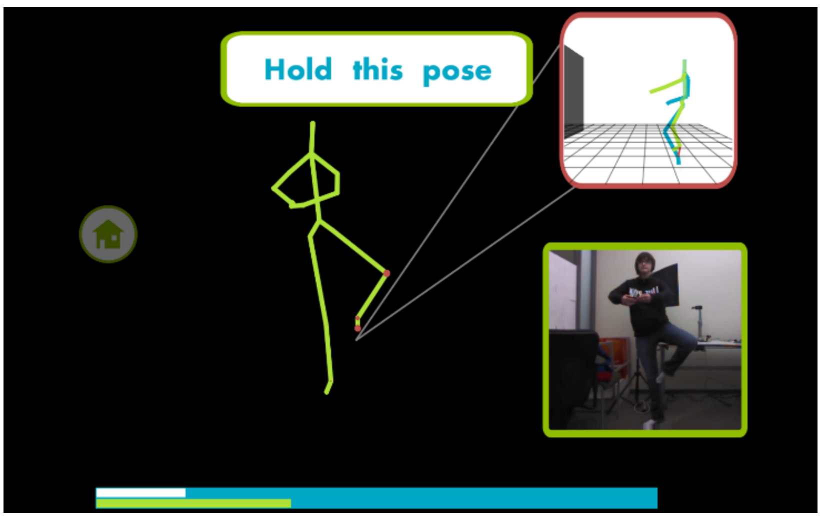

Figure 6.6. The posture guide requires trainees to maintain a stable posture, matching the position of the trainer. Errors in joint position are indicated by red circles. The callout (top right of figure) resolves depth ambiguities.

In this stage, ambient lighting is increased and the trainer's skeleton is virtually aligned to the trainee on each rendered frame. Trainees are instructed to match their reflection with the trainer's green skeleton, as in Figure 6.1. Errors in positioning are shown as red circles overlaid on the user's joints, with the circle radius proportional to the joint error. If the largest error is found to be in the z-axis (depth), then a call-out window appears that shows the trainee a side-view of their body position superimposed on the trainer movement. Using the error to determine when to show the feedback is a form of bandwidth feedback (Sherwood, 1988; Winstein & Schmidt, 2004) which has been shown to improve learning. The author's original video is also displayed on the screen for additional visual reference. The video is automatically cropped based on the location of the trainer's skeleton within the video to preserve on-screen space.

6.5.3.2. Movement guide

The movement guide (Figure 6.7) provides a skeleton similar to that of the posture guide, but it moves in real-time and does not pause on the keyframes. In addition, the trainer's skeleton is aligned relative to the starting location of the trainee on each repetition – forcing the trainee to move with the skeleton to maintain alignment.

The movement guide also provides 'cue ribbons' – 3D trajectories that help the user with timing by visualizing the upcoming movements. The ribbons display the trajectory of the hands and feet 300ms ahead of the current frame, with the ribbon becoming more opaque the farther it is in the future. To reduce visual complexity, the ribbons are only displayed if the joint is going to surpass a velocity threshold of 75 cm/s within the subsequent 300 ms. If smoothness is specified as an important parameter, the ribbons are extended to movement 500m/s in the future, to allow trainees to better prepare their upcoming trajectories.

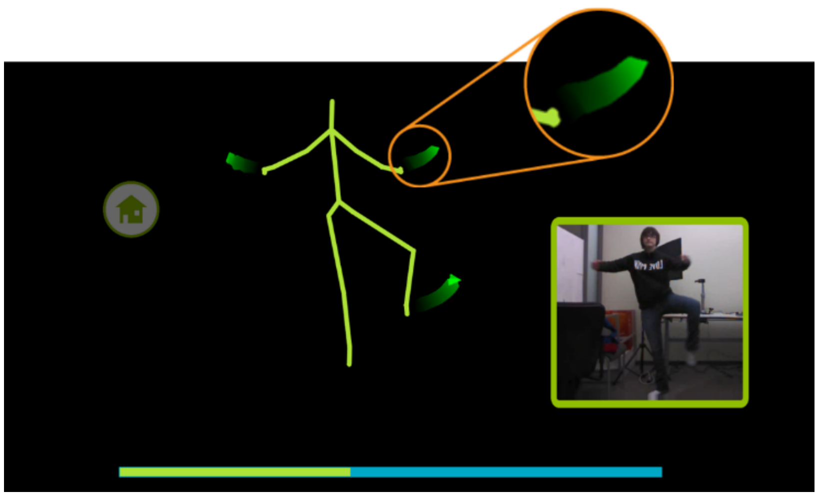

Figure 6.7. The movement guide encourages the trainee to match the trainer's movements at full speed, using green ribbons (inset) to cue upcoming movements.

6.5.3.3. Mirror guide

The mirror guide encourages the trainee to focus on their reflection, and does not provide any visual cues to guide the movements. A black screen is projected onto the mirror, and the ambient lighting is increased to enable the trainee to see a clear reflection of themself in the mirror. Audio cues are present to help with timing. This type of guide is similar to a ballet class, where the student practices in front of the mirror with the instructor counting the beats.

6.5.3.4. On your own

The 'On Your Own' guide mimics a real-world performance scenario where the trainee relies only on what they have learned. The ambient lighting is off, and a white image is projected onto the mirror, preventing the trainee from seeing any reflection. The only audio cue provided is the word 'and', used to indicate the start of the movement.

6.5.4. Post-Stage Feedback

Trainees can view their performance using the Post-Stage Feedback screen (Figure 6.8). This screen is displayed after each of the training stages, except the Demonstration. This screen summarizes each of the previous repetitions and presents the aggregate data for each keyframe. Trainees can navigate between keyframes to see an average score for that keyframe, as well as their body pose and a summary video for that frame. During Post-Stage Feedback, ambient lighting is kept at a moderate level, and the projected background is grey. This enables the trainee to see the on-screen content, while allowing them to see their reflection and use reflection-based buttons.

Error in posture is represented by two skeletons: the average skeleton of the trainee for all repetitions (blue), and the trainer skeleton (green). The same circles used in the posture guide are used for feedback to indicate relative joint error. While viewing the feedback, trainees can rotate the 3D view of the skeletons by walking left or right, allowing them to see errors in all dimensions.

A summary video is also provided for each keyframe, allowing the trainee to quickly assess their movements and compare them to the trainer. The trainer video is a static image taken from the recorded video, while the trainee video is an animated sequence of images, each taken from one repetition of the movement, displayed for 0.5s each. By animating the images, trainees can easily see the variations in their movement from frame to frame.

A group of 5 contextual buttons are displayed around the user, allowing them to re-perform the previous stage, advance to the next stage, navigate between keyframes, or view a video annotation if one has been associated with the current keyframe.

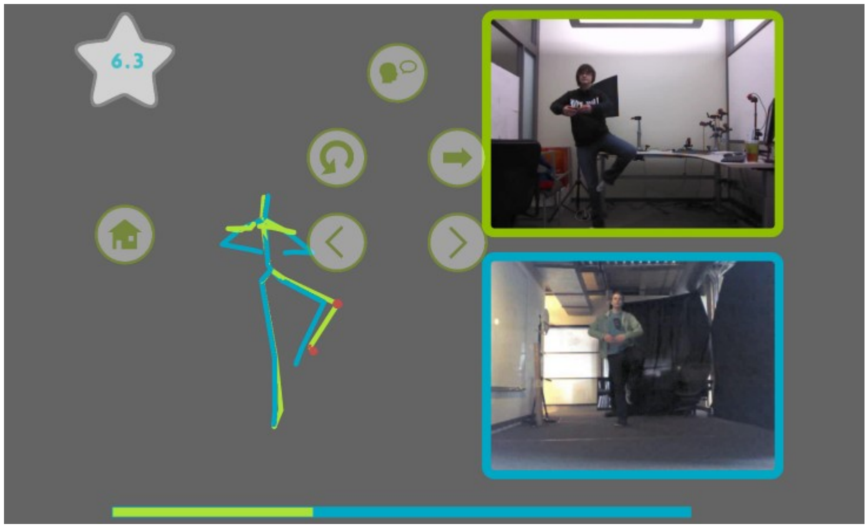

Figure 6.8. The Post-Stage Feedback screen, showing trainees their overall score, average position (skeleton), and video for each keyframe.

6.6. Evaluation

A controlled study was conducted to compare YouMove to traditional video-based instruction methods. Effectiveness of movement instruction was compared using the results of retention scores after training with each system. Subjects also provided qualitative feedback on the YouMove system, and the use of a mirror as an interactive display.

6.6.1. Participants

Eight participants (2 female) between 21 and 51 years old, (M = 30.1 years) were recruited from within the organization, but external to the group. No participants had prior knowledge of, or experience with the system or study, or a dance background.

6.6.2. Study Movements

We recorded four movements for the study – two ballet movements and two abstract movements. The ballet movements (variations on the Tendu and Developpe) were easier to conceptualize and only required a moderate amount of movement. The abstract movements were more difficult to perform, as they were a series of postures with no clear structure and required substantial movement. All movements used four keyframes.

6.6.3. Conditions

The two conditions for the study were *YouMove* and *Video*. The YouMove condition consisted of the YouMove system as previously described. None of the movements contained multimedia annotations or had global movement parameters specified. The *Video* condition consisted of a video projected on the mirror screen at the same size as the Demonstration guide (102 x 76 cm), with the area on the screen around the video projected white to prevent the participants from seeing their reflection.

6.6.4. Design

The study was conducted as a two factor repeated-measures design, with each participant learning two movements using YouMove (one abstract followed by one ballet), and two by using just the demonstration video. The condition order and movement pair to condition mapping was fully counterbalanced across the 8 participants.

6.6.5. Procedure

Each participant was introduced to the YouMove system using a simple tutorial movement. The experiment began once participants were comfortable with the system.

Each movement began with a pre-test phase, in which the participant watched a video of the movement twice, and then performed the movement five times during the 'On Your Own' guide. Next, the participant trained using either the full YouMove system, or just the Video. During training, the participant practiced 45 times, either by practicing along with the video, or by practicing with the various guides of the YouMove system. In the YouMove condition, users were free to navigate to any of the unlocked training stages.

Participants completed a retention test five minutes after completing the training. The test consisted of watching the demonstration video once to review the movement, then performing the movement five times during the 'On Your Own' guide. The Kinect recorded the participant's movements in all stages of the study.

A short questionnaire was given to participants after completing the tasks to elicit qualitative feedback. The study took approximately two hours to complete.

6.6.6. RESULTS

A repeated measures ANOVA with two independent variables: Condition (YouMove, Video), and Movement type (Ballet, Abstract) was performed. Performance (Figure 6.9) was measured by computing the RMSE between the space-and-time aligned user skeletons and the target skeleton, using the neck, hands, elbows, knees and feet. These joints were chosen because they are reliably tracked and capture the majority of the information in the movements. Similar analyses were conducted using only the joints specified by the content author, as well as using the joint that has maximal error (the measure used in computing the in-game score). All of these analyses produced equivalent statistical findings, and so we only present the RMSE analysis.

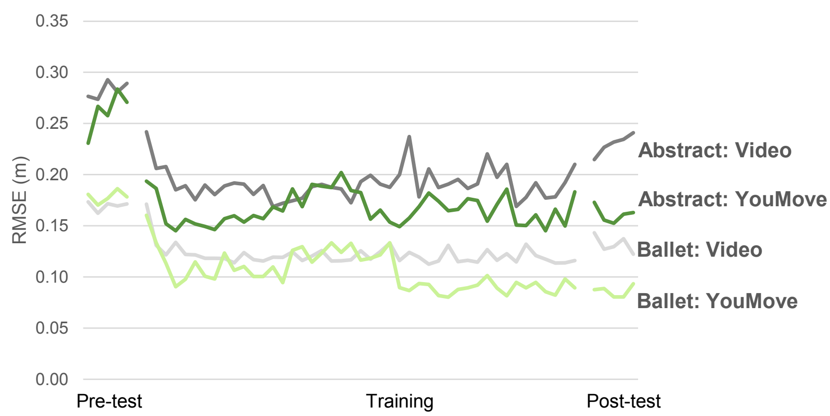

Figure 6.9. Performance on each trial, for each of the conditions, averaged over all 8 subjects.

The change from pre-test to post-test was significant ($F_{1,7} = 9.98$, $p = 0.02$), with YouMove scores improving by an average of 0.10m (44%), and the Video condition improving scores by 0.05m (20%), representing a medium-large effect size ($\eta^2 = 0.13$). The effect of movement type on the change was not significant ($F_{1,7} = 0.24$, $p > 0.6$), nor was the interaction between movement type and condition ($F_{1,7} = 0.23$, $p > 0.7$), indicating that YouMove's effectiveness is not dependent on movement difficulty.

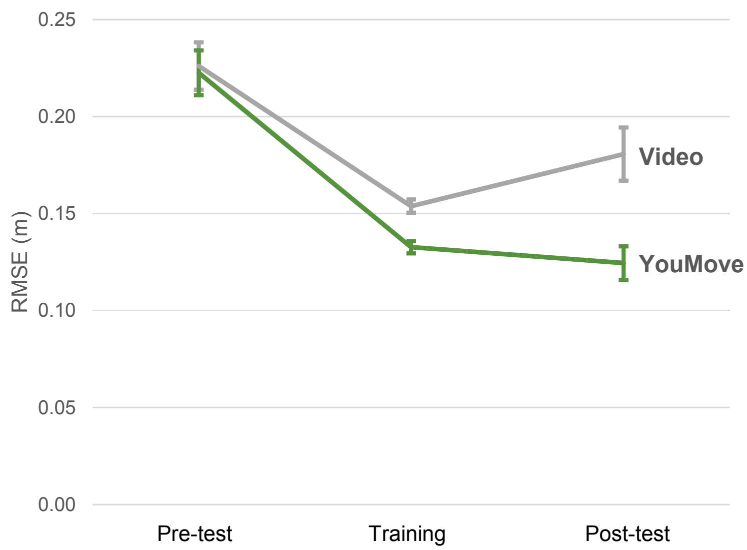

Figure 6.10. Pre-test, training, and post-test scores for the YouMove and Video conditions.

An ANOVA was also conducted on the post-test scores with factors of Condition and MovementType. The condition was significant ($F_{1,7}$ = 9.96, p = 0.02) with scores of 0.12m for YouMove and 0.18m for Video, an improvement of 33% (Figure 6.10). The movement type was also significant ($F_{1,7}$ = 114.2, p < 0.01), with scores of 0.10m for ballet movements and 0.19m for abstract movements, indicating that ballet movements were easier.

6.6.7. Stage Usage Analysis

For the *YouMove* condition, most participants performed the training guides in sequence, although some participants did choose to replay guides before continuing. After unlocking all stages, there was no clear preference, but users seemed less likely to return to the posture guide. This is likely due to the increased time the posture guide requires. The guide use by trial is visualized in Figure 6.11.

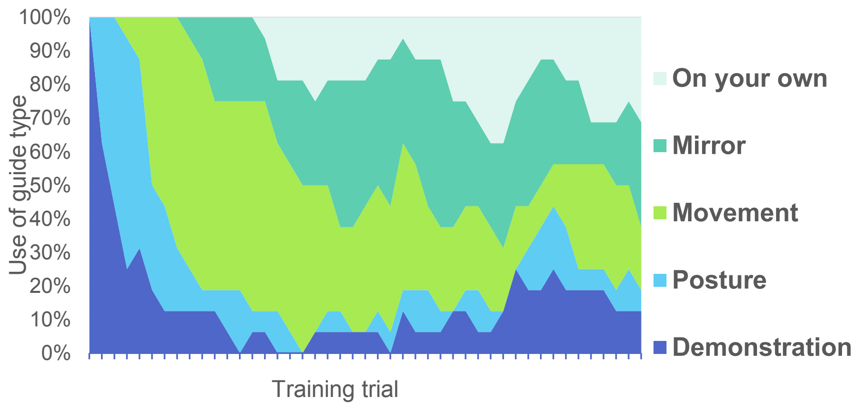

Figure 6.11. Frequency of use for each guide type for all participants during training with the YouMove system.

6.7. Discussion

Overall, we were pleased to see that when comparing pre-test and post-test results, learning increased by more than a factor of 2 (44% vs. 20%) with the YouMove system.

6.7.1. Analysis of learning

Across all conditions, participants appeared to learn the movement within the first few training trials (Figure 6.9) and then reached a plateau, with relatively constant performance. In the case of the video condition, this was likely because the trainee had truly plateaued, and learned all of the usable information from the video. With the YouMove condition, however, the gradual reduction of guidance meant that the trainee was continually learning the movement and making up for the lack of guidance with increased skill, resulting in consistently high performance.

The retention scores show improved learning with YouMove, as performance is maintained on the retention tests. In contrast, retention scores for video are worse than in training, evidenced by the upward slope in Figure 6.10.

While training with the YouMove system took longer (i.e., average of 20 minutes compared to 11 minutes with video), this cannot fully explain the results of learning, as the number of exposures to the movement remained constant. Additionally, performance on the video condition plateaued after approximately 8 repetitions, and unlike the YouMove condition, the stimuli did not change and so it is unlikely that any additional learning would occur if the video condition had been repeated for 20 minutes.

6.7.2. Mirror interaction

Dynamic lighting allowed the system to shift the attention of the user, and did not seem to distract participants. The dynamic lighting provided the intended effect and enabled the screen to be multiplexed, allowing the trainee to use both their reflection and the projected guide.

Some participants had difficulty lining up the reflection of their hand with the buttons for the reflection selection technique. To mitigate these problems, we have since added virtual cursors that appear if the system has been waiting more than five seconds for a selection to be made. These cursors provide novice users with the necessary feedback to correct their movement, yet do not interfere with the efficient reflection-based interaction.

6.7.3. Query by example

While the ability to search for similar movements was not tested in the user study, during informal use it proved to be fairly robust and useful. For novices, this feature will allow querying of large databases without having to know the name of the movement. This is especially useful in domains that have complex or cryptic names for their movements.

During training, the trainee's performance is recorded in the same format as the trainer's. This allows users to upload their movements for others to learn from, and enables them to review their prior performances at a later date.

6.7.4. Authoring

The authoring system proved to be easy to use. No formal study was conducted, but it was sent to an untrained user and they were able to record and annotate several movements with minimal effort. The movements were successfully transferred and used in the training system.

The authoring system benefits experts as it allows for content expansion. One minute of content creation generates approximately ten minutes of unique training content with stages that vary the method of presentation and provide feedback. This allows experts with limited time to produce a useful quantity of content with minimal time investment.

6.7.5. Training Stages

The Demonstration stage was well liked by users. This is likely due to familiarity with video as a teaching tool, and the ability for participants to easily understand the content. Some participants did, however, comment on the difficulty in judging limb placement and depth from video.

The Posture Guide allowed users to correct the movement, and many errors in positioning were corrected using the posture guide and depth callout. Though similar in nature to the posture guide, the movement guide was one of the most preferred guides. It provides guidance at full-speed while still providing useful feedback.

The Mirror Guide and 'On Your Own' guide were clearly valuable. Many participants felt that they had learned the movement, but when the guides were taken away they struggled to remember the sequence.

The Post-Stage Feedback was also well received. Five participants explicitly mentioned that the ability to move back and forth to change the viewpoint of the skeleton was a useful feature. The utility of the feedback is evident in the fact that many participants spent a substantial amount of time on this stage. They were aware that the amount of exposures to the movement was kept constant, and that any time spent on the feedback screen would lengthen the experiment, yet many participants still examined each keyframe.

6.8. Future Work

The addition of social features and richer gamification could greatly help YouMove. One can imagine online yoga, dance or martial arts classes, with competition from online peer groups, but more work is needed to achieve this.

6.9. Summary

In this chapter, we demonstrated that the framework and concepts from two-dimensional gesture learning generalize to gestures that are more complex. We have adapted various forms of visual feedback to support memorability of each of the gestures, showing not only skeletal movements, but dynamic trajectory highlights. Additionally, the video cues presented help provide unique visual cues that help users remember the appropriate movement. To support learning of the procedural component, the system provides summary feedback after each stage. Additionally, the adaptive guide makes the system approachable and usable by novices but gradually progresses them to expert levels of performance. There are still many open questions regarding how to optimally train users to perform the complex movements, however, this system provides evidence that gesture training by interactive systems can be much more effective than traditional instruction if the various aspects of gesture learning are accounted for.

6.9.1. Relevance to Gesture Learning

This chapter primarily examines the learning of the procedural component of gesture production. As with the previous chapter, a small set of gestures was utilized so the users did not have to focus on which gesture to learn. The design of the feedback used in YouMove was tailored to specifically maximize learning. The adaptive, faded feedback, as well as the averaged and user-driven feedback allowed the participants to receive optimized information from the system to aid them in learning. The system also demonstrated how various types of feedback could be integrated into a single, coherent system.

6.9.2. Limitations

The evaluation captures the combined effects of many novel elements. More work is needed to analyze each component and its contribution to learning. Long-term learning and retention are also important areas to explore.

One limitation of the system is related to the skeletal tracking of the Kinect. In particular, the Kinect has difficulty tracking movements that cause large amounts of occlusion. One possible improvement would be to leverage multiple Kinects (Butler et al., 2012). Improvements in sensing technology will open new domains of training. Playing musical instruments, surgery, many sports, and other motor-skill based domains could benefit from such a system.

Another limitation is the large mirror required for the training system. While the presented implementation uses a half-silvered mirror as a display, the software could also run as a traditional video-based augmented reality system, as in (Velloso, Bulling, & Gellerson, 2013). This would be more accessible to users, but does not provide the real-time feedback that the mirror does. It would be interesting to better understand any learning difference between a mirror and video based system on various devices (large screen, small screen, etc.).

Chapter 7

Thesis Conclusions

7.1. Summary

This thesis presented several contributions in the area of gestural learning within HCI. We contributed a framework which describes the design of gestural learning interfaces as being driven by a user's pre-existing knowledge, their support for learning the declarative mapping between gesture and action, and their support for learning the procedural component that results in accurate gesture execution.

With respect to the questions posed in Chapter 1, we now have the following answers:

Gesture learning is necessary in nearly any system involving gestures, and it cannot be completely avoided by allowing for user-defined gestures. We showed that user-defined gestures are not always self-consistent for a given high-level task, indicating a need for some amount of learning even within user-defined scenarios.

Gestures do, in fact, have an advantage over traditional input methods, and this advantage comes from the visual system's ability to encode the gesture as a shape, rather than the motor system's encoding of the gesture. We analyzed what factors are at play when learning gestures and compared the effects of the visual and motor components. We found a strong effect of visual feedback in the recognition of learned gestures.

When training the procedural component of gestures, we should leverage as much knowledge as possible from motor learning and avoid designs that emphasize guidance over learning. Additionally, retention and transfer tests should be utilized so that learning can be accurately assessed. We analyzed various approaches to providing visual feedback in gesture training systems using a retention and transfer paradigm. We also developed an adaptive approach that provides ease of use for novices while still providing learning benefits.

Lastly, we showed that our findings from two-dimensional gestures, as well as other findings from motor learning generalize two more complex, full-body movements. We integrated a variety of features that support gesture learning into a single coherent system designed to provide maximal learning benefit.

7.2. CONTRIBUTIONS

Chapter 3 presented two studies to understand the benefits and limitations of user-defined gestures. Both studies used a gesture password creation paradigm to provide a simple, easily manipulated test bed on which to test hypotheses. The first study examined the effect of high-level instruction on the types of gesture passwords generated by participants and found significant effects on the complexity and structure of the gesture when the instruction was manipulated. The results support related work which analyzed user-defined gestures when varying the high-level task. The second study, inspired by work in embodied cognition, examined the effect of the orientation of a device on the structure of gesture that was created. This study revealed that the orientation of the device affected the types of gestures created, even within the same user. This suggests that user-defined gestures may not always be consistent even when the task remains the same, which indicates some need for gesture learning even within gestural interfaces that support user-defined gestures.

In Chapter 4, we presented two studies that examined the cognitive benefits of learning sequences using gestural input compared to learning those sequences using traditional pointing input. The studies focused on the declarative component of gesture learning, and used a two-alternative forced-choice recognition task to assess the degree to which the sequences were learned. The first study utilized a more ecologically valid paradigm, rendering strokes between each dot in the sequence as well as requiring participants to use a continuous stroking motion. Under these conditions, gestural input resulted in the sequences being learned more efficiently. In the second study, the visual feedback remained fixed between the two conditions so the only modification was the movements required by the participant. Under these conditions, the advantage that gestural input demonstrated in the prior study disappeared. These results point to the visual, rather than the motor component being the dominant factor when learning the declarative component of gestures.

In Chapter 5, we examined the role of visual feedback in learning the procedural component of the movement. Using a training and retention paradigm common to motor learning, we assessed the impact of guidance during gesture learning. We found that the type of visual feedback used during training dramatically influences how well the gestures are learned. Having visual feedback that participants could trace directly on top of resulted in worse performance than feedback that was scaled and offset from the desired gesture. We also developed and assessed an adaptive guide that preserves the ease-of-use of the traceable guides but gradually removes the guidance, which allows for greater learning. We find that the retention tests provide a very different view of learning than what has previously been used to evaluate gesture learning.

Lastly, Chapter 6 presented a novel system, YouMove, which extends our understanding of interactive gesture training to complex, 3D gestures. The system uses an interactive augmented-reality mirror to overlay feedback directly over-top of users' reflections. This provides a simple and natural metaphor for understanding the guidance provided by the system. It also leverages staged feedback, with each stage gradually reducing the amount of feedback provided to encourage learning. A retention test demonstrated enhanced learning with the YouMove system compared to a traditional video-based approach.

7.3. FUTURE WORK

7.3.1. UNDERSTANDING GESTURE LEARNING

There are many open questions within the scope of gesture learning and the framework presented in Chapter 1. At a high level, one of the foremost questions is to identify the relative importance of the procedural and declarative components with respect to their impact on user performance and usability. Within this, there are likely differences in user experience where novices may prefer systems which emphasize learning of the gesture vocabulary (declarative component) and provide heavy guidance for the procedural components, whereas users with more expertise may prefer the opposite.

Current gesture guides, including those presented in this thesis, do not take full advantage of the full body of literature from the motor learning domain to maximize training of the procedural component. There are dozens of concrete guidelines that are ready to be implemented and integrated into current gesture training systems. Integrating these principles into usable, practical systems remains a large challenge.

With respect to the declarative component, non-visual modalities for gesture learning may still prove beneficial. Prior work (Andersen and Zhai, 2010) examined the sonification of gestures, but found little benefit in terms of performance. They did not, however, examine learning effects. In a related domain, Grossman et al. (2007) examined the learning of shortcut keys, and found that audio was a very successful method in instructing users on the pairing between command and shortcut. Haptic modalities may also prove useful in rendering gestures to the user, perhaps leveraging benefits of dual coding (Paivio, 1973).

7.3.2. Applications

Emerging applications will provide a plethora of new challenges for gesture learning. Wearable interfaces are becoming increasingly more prevalent, and implantable interfaces are on the horizon (Holz, 2012). These new modes of interaction do not have high input and output bandwidth and they are to be used within highly dynamic environments. As such, future gesture learning approaches for these systems must address new issues such as skill transfer between different contexts (so users can accurately perform gestures in novel configurations), and new modalities for presenting feedback such as haptic or audio rendering.

Related to this, truly ubiquitous computing is poised to become a reality in the near future with sensors and intelligence embedded in our lights, appliances, etc. These intelligent environments currently lack a clear input space and obvious affordances which make them optimal candidates for gestural interaction. With these interfaces, it will be likely that some combination of user-defined and standardized gestures will be implemented to provide baseline functionality. However, to unlock the full potential of an intelligent environment, the user should be able to control many elements in rapid succession by phrasing actions together and configuring new functionality. For these more expressive scenarios, we envision offline configurations used to configure the desired functionality while training the user in the gestures required to operate the functionality. Engaging platforms will need to be developed that provide efficient tutoring, and these platforms will need to leverage fundamental knowledge about how gestures are learned and adapt them to each unique scenario.

Future work on the training of complex, 3D gestures will largely involve the development of new methods to provide real-time guidance and feedback within a user interface. The work presented in Chapter 6 required users to undergo offline training before being able to accurately reproduce the movement. However, this is not ideal and would not be suitable for interactive systems. However, these trainers for complex movement will have the greatest applicability going forward. They have applications across sports, dance, medicine, manufacturing, etc. Current approaches to virtual training are often bespoke, with limited applicability beyond their domain. Not only does this limit their applicability, but it requires each designer to understand fundamental principles of movement learning, which is beyond the scope of most software developers. Thus, a platform which is able to abstract as much expert knowledge as possible has great utility that spans many application domains.

Bibliography

Albaret, J.M. & Thon, B. (1998). Differential Effects of Task Complexity on Contextual Interference in a Drawing Task. *Acta Psychologia*, *100(1-2),* 9-24.

Andersen, T.H. & Zhai, S. (2010). "Writing with Music": Exploring the Use of Auditory Feedback in Gesture Interfaces. *Transactions of Applied Perception*, *7(13)*, 17.

Anderson, F., Annett, M. & Bischof, W.F. (2012). Tabletops in Motion: the Kinetics and Kinematics of Interactive Surface Physical Therapy. In the *Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems*, 2351-2356.

Anderson, F., & Bischof, WF. (2013). Learning and Performance with Gesture Guides. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 1109-1118.

Anderson, G., Nasiopoulos, E., Foulsham, T., Chapman, C., & Kingstone, A. (2012). Hide and Seek: the Ultimate Mind Game. *Journal of Vision*, *12(9),* 733.

Annett, M., & Bischof, W. F. (2013). Your left hand can do it too!: investigating intermanual, symmetric gesture transfer on touchscreens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1119-1128.

Appert, C. and Bau, O. (2010). Scale Detection for a Priori Gesture Recognition. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 879-882.

Appert C. & Zhai, S. (2009). Using Strokes as Command Shortcuts: Cognitive Benefits and Toolkit Support. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2289-2298.

Autodesk. (2014). Maya. http://www.autodesk.com/products/autodesk-maya/overview [Computer Software].

Baddeley, A.D. & Dale, H.C.A, (1966). The Effect of Semantic Similarity of Retroactive Interference in Long and Short-Term Memory. *Journal of Verbal Learning and Verbal Behavior*, *5(5)*, 417-420.

Baddeley, A. & Hitch, G. (1974). Working Memory. The *Psychology of Learning and Motivation*, *8(1)*, 47-68

Baddeley, A. (1994). The Magical Number Seven: Still Magic After All These Years? *Psychological Review*, *101(2)*, 353-356.

Bailly, G., Lecolinet, E., & Nigay, L. (2008). Flower Menus: A New Type of Marking Menu with Large Menu Breadth, Within Groups and Efficient Expert Mode Memorization. In the *Proceedings of Advanced Visual Interfaces,* 15-22.

Bau O. & Mackay W. (2008). Octopocus: A Dynamic Guide for Learning Gesture-Based Command Sets. In the *Proceedings of the ACM Symposium on User Interfaces and Software Technology,* 37-46.

Baudel, T. & Beaudouin-Lafon, M. (1993). Charade: Remote Control of Objects Using Free-Hand Gestures. *Communications of the ACM*, *36(7),* 28-35.

Bederson, B.B. (2004). Interfaces for Staying in the Flow. *Ubiquity, 5(27)*, 1.

Bennett, M., Mccarthy, K., O'Modhrain, S., & Smyth, B. (2011). Simpleflow: Enhancing Gestural Interaction with Gesture Prediction, Abbreviation and Autocompletion. In *Proceedings of INTERACT*, 591-608.

Berry, D.C. & Broadbent, D.E. (1988). Interactive Tasks and the Implicit-Explicit Distinction. *British Journal of Psychology, 79(2)*, 251-272.

Bonneau, J. (2012). The Science of Guessing: Analyzing An Anonymized Corpus of 70 Million Passwords. In the *Proceedings of Security and Privacy*, 538-552.

Bonneau, J., Preibusch, S. and Anderson, R. (2012). A Birthday Present Every Eleven Wallets? The Security of Customer-Chosen Banking Pins. In the *Proceedings of Financial Cryptography*, 25-40.

Bower, G.H. and Winzenz, D. (1970). Comparison of Associative Learning Strategies. *Psychonomic Science*, *20(2),* 119-120.

Bower, G.H. (1970). Organizational Factors in Memory. *Cognitive Psychology, 1(1)*, 18-46.

Bragdon, A., Uguray, A., & Wigdor, D. Anagnostopoulos, S., Zeleznik, R., & Feman, R. (2010). Gesture Play: Motivating Online Gesture Learning with Fun, Positive Reinforcement and Physical Metaphors. In the *Proceedings of Interactive Tabletops and Surfaces*, 39-48.

Bragdon, A., Zeleznik, R., Williamson, B., Miller, T., & Laviola J.J. (2008). Gesturebar: Improving the Approachability of Gesture-Based Interfaces. In the *Proceedings of the ACM Symposium on User Interfaces and Software Technology,* 2269-2278.

Brandl, P., Forlines, C, Wigdor, D., Haller, M., & Shen, C. (2008). Combining and Measuring the Benefits of Bimanual Pen and Direct-Touch Interaction on Horizontal Interfaces. In the *Proceedings of Advanced Visual Interfaces*, 154-161.

Brydges, R., Carnahan, H., Backstein D., & Dubrowski, A. (2007). Application of Motor Learning Principles to Complex Surgical Tasks: Searching for the Optimal Practice Schedule. *Journal of Motor Behavior*, *39(1)*, 40-48.

Butler, D. A., Izadi, S., Hilliges, O., Molyneaux, D., Hodges, S., & Kim, D. (2012). Shake'n'sense: Reducing Interference for Overlapping Structured Light Depth Cameras. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 1933-1936.

Buxton, W. Chunking and Phrasing and the Design of Human-Computer Dialogues. (1986). In the *Proceedings of IFIP World Computer Congress*, 475-480.

Cao, X. and Zhai, S. (2007). Modeling Human Performance of Pen Stroke Gestures. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1495-1504.

Castellucci, S.J. & Mackenzie, I.S. (2008). Graffiti vs. Unistrokes: An Empirical Comparison*. In the Proceedings SIGCHI Conference on Human Factors in Computing Systems*, 305-308.

Chapman, C.S., Gallivan, J.P., Wood, D.W., Milne, J.L., Culham, J.C., & Goodale, M.A. (2010). Reaching for the Unknown: Multiple Target Encoding and Real-Time Decision-Making in a Rapid Reach Task. *Cognition*, *116(2)*, 168-176.

Chase, W.G. and Simon, H.A., (1973). Perception in Chess. *Cognitive Psychology, 4(1)*, 55-81.

Chatty, S. and Lecoanet, P. (1996). Pen Computing for Air Traffic Control. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 87-94.

Clarke, N., Furnell, S., Rodwell, P. & Reynolds, P. (2012). Acceptance of Subscriber Authentication Methods for Mobile Telephony Devices. *Computers and Security, 21(3)*, 220-228.

Cockburn, A., Kristensson, P. O., Alexander, J., & Zhai, S. (2007). Hard lessons: effort-inducing interfaces benefit spatial learning. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1571-1580).

Cohen, N., Eichenbaum, H., Deacedo, B.S., & Corkin, S. (1985). Different Memory Systems Underlying Acquisition of Procedural and Declarative Knowledge. *Annals of the New York Academy of Sciences*, *444*, 54-71.

Cohen, A.D, & Aphek, E. (1980). Retention of Second-Language Vocabulary Overtime: Investigating the Role of Mnemonic Associations. *System*, *8(3),* 221-235.

Craik, F.I.M and Lockhart, R.S. (1972). Levels of Processing: A Framework for Memory Research. *Journal of Verbal Learning and Verbal Behaviour*, *11(6),* 671-684.

Cook, S. W., Mitchell, Z. & Goldin-Meadow, S. (2008) Gesturing makes learning last. *Cognition* 106(2). 1047 – 1058.

Datagenetics. (2012). PIN Analysis. Retrieved from http://www.datagenetics.com/blog/september32012/index.html.

Davis, D., Monrose, F., & Reiter, M.K. (2004). On User Choice in Graphical Password Schemes. In the *Proceedings of the USENIX Security Symposium,* 11-23.

Dearborn, K. and Ross, R. (2006). Dance Learning and the Mirror: Comparison Study of Dance Phrase Learning with and Without Mirrors. *Dance Education*, *6(4),* 109-115.

De Luca, A., Langheinrich, M., & Hussmann, H. (2010). Towards Understanding ATM Security: A Field Study of Real World ARM Use. In the *Proceedings of the Symposium on Usable Privacy and Security*, 16-25.

Doctorow, C. (2011). *Haunted Mansion Hitchhiking Ghosts Go Digital, Play High-Tech Pranks on Riders*. Retrieved from http://boingboing.net/2011/04/06/haunted-mansion-hitc.html.

Donovan, J.J. & Radosevic, D.J. (1999). A Meta-Analytic Review of the Distribution of Practice Effect: Now You See It, Now You Don't. *Journal of Applied Psychology, 84(5)*, 795-805.

Dowling, W.J. (1973). Rhythmic Groups and Subjective Chunks in Memory for Melodies. *Perception and Psychophysics*, *14(1),* 37-40.

Dudukovic, N.M. and Wagner, A.D. (2007). Goal-Dependent Modulation of Declarative Memory: Neural Correlates of Temporal Recent Decisions and Novelty Detection. *Neuropsychologia, 45(11)*, 2608-2620.

Ellenbogen, J.M., Hulbert, J.C., Stickgold, R., Dinges, D.F., & Thompson-Schill, S.L. (2006). Interfering with Theories of Sleep and Memory: Sleep, Declarative Memory, & Associative Interference. *Current Biology, 16(3)*, 1290-1294.

Ericsson, K.A., Krampe, R.T., & Tesch-Romer, C. (1993). The Role of Deliberate Practice in the Acquisition of Expert Performance. *Psychological Review, 100(3),* 363-406.

Fitts, P.M. and Posner, M.I. (1967). *Human Performance*. Brooks and Cole, Oxford, England.

Fleetwood, M., Byrne, M., Centgraf, P., Dudziak, K., Lin, B., & Mogilev, D. (2002). An Evaluation of Text-Entry in Palm OS- Graffiti and the Virtual Keyboard. In the *Proceedings of the Human Factors and Ergonomics Society, 617*-621.

Fothergill, S., Mentis, H., Kohli, P., & Nowozin, S. (2012). Instructing People for Training Gestural Interactive Systems. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 1737-1746.

Freeman, D., Benko, H., Ringel Morris, M., & Wigdor, D. (2009). Shadowguides: Visualizations for In-Situ Learning of Multi-Touch and Whole-Hand Gestures. In the *Proceedings of Interactive Tabletops and Surfaces*, 167-172.

Fronteir Developments, Kinect Disneyland Adventures, *Microsoft Studios* [Video Game].

Gobet, F., & Simon, H.A. (1998). Expert Chess Memory: Revisiting the Chunking Hypothesis. *Memory*, *6(3),* 225-255.

Goldin-Meadow, S., Cook, S. W., & Mitchell, Z. A. (2009). Gesturing gives children new ideas about math. *Psychological Science*, *20*(3), 267-272.

Gomita. (2012). Firegestures. Mozilla Add-Ons. Retrieved from http://addons.mozilla.org/en-us/firefox/addon/firegestures.

Grossman, T., Dragicevic, P., & Balakrishnan, R. (2007). Strategies for accelerating on-line learning of hotkeys. In *Proceedings of the SIGCHI conference on Human factors in computing systems.* 1591-1600.

Guimbretiere, F. & Winograd, T. (2000). Flowmenu: Combining Command, Text and Data Entry. In the *Proceedings of the ACM Symposium on User Interfaces and Software Technology*, 213-216.

Haist, F., Shimamura, A.P., & Squire, L.R. (1992). On the Relationship between Recall and Recognition Memory. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *18(4),* 691-702.

Helsen, W.F., Hodges, N.J., Van Winckel, J., & Starkes, J.L. (2000). The Roles of Talent, Physical Precocity and Practice in the Development of Soccer Expertise. *Journal of Sports Sciences*, *18(9),* 727-736.

Henry, F. M., & Rogers, D. E. (1960). Increased response latency for complicated movements and a "memory drum" theory of neuromotor response. *Research Quarterly. American Association for Health, Physical Education and Recreation*, *31*(3), 448-458.

Hinckley, K., Baudisch, P., & Ramos, G., (2005). Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in Scriboli. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 451-460.

Holz, C., Grossman, T., Fitzmaurice, G., & Agur, A. (2012). Implanted user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 503-512.

Iverson, J. M., & Goldin-Meadow, S. (2005). Gesture paves the way for language development. *Psychological science*, *16*(5), 367-371.

Kafka, P. (2012). Eric Schmidt Can't Wait for Self-Driving Cars, Won't Predict When Google Maps Come Back To Apple. Retrieved from http://allthingsd.com/20121010/live-from-new-york-walt-mossberg-kara-swisher-interview-eric-schmidt.

Kane, M.J., & Engle, R.W. (2000). Working-Memory Capacity, Proactive Interference, & Divided Attention: Limits of Long-Term Memory Retrieval. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *26(2),* 336-358.

Kita, S. (2000). How representational gestures help speaking. *Language and Gesture.*. 162 – 185.

Kray, C., Nesbitt, D., Dawson, J., & Rohs, M. (2010). User-Defined Gestures for Connecting Mobile Phones, Public Displays, & Tabletops. In the *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services*, 239-248.

Kristensson, P.O. & Denby, L.C. (2011). Continuous Recognition and Visualization of Pen Strokes and Touch-Screen Gestures. In the *Proceedings of the Eurographics Synposiu on Sketch-Based Interfaces and Modeling,* 95-102.

Kristensson, P.O. & Zhai, S. (2004). SHARK: A Large Vocabulary Shorthand Writing System for Pen-Based Computers. In the *Proceedings of the ACM Symposium on User Interfaces and Software Technology,* 2004, 43-52.

Kurtenbach, G.P., Sellen, A.J., & Buxton, W. (1993). An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Journal of Human Computer Interaction, 8(1)*, 1-23.

Kurtenbach, G.P., Moran, T.P., & Buxton, W. (1994). Contextual Animation of Gestural Commands. *Computer Graphics Forum, 13(5)*, 305-314.

Kurtenbach, G.P. & Buxton, W. (1993). The Limits of Expert Performance Using Hierarchic Marking Menus. In the *Proceedings of INTERACT*, 482-487

Kurtenbach, G.P. (1993). The *Design and Evaluation of Marking Menus*. (Doctoral Dissertation). University of Toronto, Toronto, Ontario, Canada.

Lavery, J. J. (1962). Retention of simple motor skills as a function of type of knowledge of results. *Canadian Journal of Psychology*, 16(4), 300.

Lepinski, J., Grossman, T., & Fitzmaurice, G. (2010). The Design and Evaluation of Multi-Touch Marking Menus. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 2233-2242.

Levenshtein, V.I. (1966). Binary Codes Capable of Correcting Deletions, Insertions, & Reversals. *Soviet Physics Doklady, 10(8)*, 707–710.

Li, F.C., Dearman, D., & Truong, K.N. (2009). Virtual Shelves: Interactions with Orientation Aware Devices. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 125-128.

Li, W., Grossman, T., & Fitzmaurice, G. (2012). Gamicad: A Gamified Tutorial System for First Time Autocad Users. In the *Proceedings of the ACM Symposium on User Interfaces and Software Technology*, 103-112.

Li, Y. (2009). Beyond Pinch and Flick: Enriching Mobile Gesture Interaction. *Computer*, *42(12)*, 87-89.

Li, Y. (2010). Protractor: A Fast and Accurate Gesture Recognizer. In the *Proceedings* of the *SIGCHI Conference on Human Factors in Computing Systems*, 2169-2172.

Luck, S.J. & Vogel, E.K. (1997). The Capacity of Visual Working Memory for Features and Conjunctions. *Nature*, *390(6657)*, 279-280.

Mackenzie, I.S. & Soukoreff, R. W. (2002). Text Entry for Mobile Computing: Models and Methods, Theory and Practice. *Human-Computer Interaction*, *17(2),* 147-198.

Marsh, E.J., Edelman, G., & Bower, G.H. (2001). Demonstrations of a Generation Effect in Context Memory. *Memory and Cognition*, *29(6)*, 798-805.

Markman, A.B. and Brendl, C.M. (2005). Constraining Theories of Embodied Cognition. *Psychological Science*, *16(1)*, 6-10.

Microsoft. (2012). Sign in with A Picture Password. Retrieved from http://Windows.Microsoft.Com/En-US/Windows-8/Picture-Passwords.

Microsoft Research. (2012). Holoflector. Retrieved from http://Research.Microsoft.Com/Apps/Video /Dl.Aspx?Id=159487.

Morris, M.R., Wobbrock, J.O., & Wilson, A.D. (2010). Understanding Users' Preferences for Surface Gestures. In the *Proceedings of Graphics Interface*, 261-268.

Nacenta, M.A. Kamber, Y., Qiang, Y. & Kristensson, P.O. (2013). Memorability of pre-designed and user-defined gesture sets. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 1099-1108.

Nelson, D. L., Reed, V. S., & McEvoy, C. L. (1977). Learning to order pictures and words: A model of sensory and semantic encoding. *Journal of Experimental Psychology: human learning and memory*, *3*(5), 485.

Niu, Y. and Chen, H. (2012). Gesture Authentication with Touch Input for Mobile Devices. *Security and Privacy in Mobile Information and Communication Systems, 94(1), 1*3-24.

Norman, D. (2010). Natural User Interfaces Are Not Natural. *Interactions*, *17(3),* 6-10.

Ng, A., Lepinski, J., Wigdor, D., Sanders, S., & Dietz, P. (2012). Designing for Low-Latency Direct-Touch Input. In the *Proceedings of the ACM Symposium on User Interfaces and Software Technology,* 453-464.

Oldfield, R. C. (1971). The Assessment and Analysis of Handedness: the Edinburgh Inventory. *Neuropsychologia, 9(1),* 97-113.

Ouyang, T. and Li, Y. (2012). Bootstrapping Personal Gesture Shortcuts with the Wisdom of the Crowd and Handwriting Recognition. In the *Proceedings* of the *SIGCHI Conference on Human Factors in Computing Systems,* 2895-2904.

Paivio, A. & Kalman, C. (1973). Picture superiority in free recall: Imagery or dual coding? *Cognitive psychology* 5(2), 176-206.

Palmiter, S. & Elkerton, J. (1991). An Evaluation of Animated Demonstrations of Learning Computer-Based Tasks. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 257-263.

Panzer S., Krueger, M., Muehlbauer, T., & Shea, C.H. (2010). Asymmetric Effector Transfer of Complex Movement Sequences. *Human Movement Science*, *29(1),* 62-72.

Park, J., Shea, C.H., & Wright, D.L. (2000). Reduced Frequency Concurrent and Terminal Feedback: A Test of the Guidance Hypothesis. *Journal of Motor Behavior*, *32(3)*, 278-296.

Pashler, H., Roher, D., Cepeda, N.J., & Carpenter S.K. (2007). Enhancing Learning and Retarding Forgetting: Choices and Consequences, *Psychnomic Bulletin and Review*, *14(2),* 187-193.

Pongnumkul, S., Dontcheva, M., Li, W., Wang, J., Bourdev, L., Avidan, S., & Cohen, M.F. (2011). Pause-and-Play: Automatically Linking Screencast Video Tutorials with Applications. In the *Proceedings of the ACM Symposium on User Interfaces and Software Technology,* 135-144.

Proteau, L., Marteniuk, R.G., & Lévesque, L. (1992). A Sensorimotor Basis for Motor Learning: Evidence Indicating Specificity of Practice. *Experimental Psychology*, *44(3)*, 557-575.

Roediger, H.L. (1990). Implicit Memory: Retention without Remembering. *American Psychologist*, *45(9),* 1043-1056.

Rohrer, D., Taylor K., Pashler, H., Wixted, J.T., & Cepeda, N.J. (2005).The Effect of Overlearning on Long-Term Retention. *Applied Cognitive Psychology*, *19(3),* 361-374.

Rubine, D. Specifying Gestures by Example (1991). In the *Proceedings of SIGGRAPH*, 329-337.

Ruiz, J., Li, Y., & Lank, E. (2011). User-Defined Motion Gestures for Mobile Interaction. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 197-206.

Sainburg, R.L. and Wang, J. (20020. Interlimb Transfer of Visuomotor Rotations: Independence of Direction and Final Position Information. *Experimental Brain Research*, *145(4),* 437-447.

Salmoni, A.W., Schmidt, R.A., & Walter, C.B. (1984). Knowledge of Results and Motor Learning: A Review and Critical Reappraisal, *Psychological Bulletin*, *95(3),* 355-386.

Sato, Y., Saito, M., & Koike, H. (2001). Real-time input of 3D pose and gestures of a user's hand and its applications for HCI. In *Proceedings of IEEE Virtual Reality, 2001.* 79-86.

Savion-Lemieux, T. & Penhune, V. (2005). The Effects of Practice and Delay on Motor Skill Learning and Retention. *Experimental Brain Research*, *161(4),* 423-431.

Schmidt, R.A. (1991*). Frequent Augmented Feedback Can Degrade Learning: Evidence and Interpretations.* In J. Requin and G. E. Stelmach (Eds.), *Tutorials in Motor Neuroscience*, 59-75.

Schmidt, R.A. & Lee, T. (2011). *Motor Control and Learning: A Behavioral Emphasis, 5th Edition.* Human Kinetics Ltd.

Schmidt, R. A. & Wulf, G. (1997). Continuous Concurrent Feedback Degrades Skill Learning: Implications for Training and Simulation*. Human Factors*, *39(4),* 509-525.

Segen, J. (1998). Fast and Accurate 3D Gesture Recognition Interface. In the *Proceedings of Pattern Recognition*, 86-91.

Shea, J.B. and Morgan, R. (1979). Contextual Interference Effects on the Acquisition, Retention, & Transfer of a Motor Skill. *Journal of Experimental Psychology: Human Learning and Memory*, *5(2),* 179-187.

Sherwood, D.E. (1983) The Impulse Variability Model: Tests of Major Assumptions and Predictions, Doctoral dissertation, University of California, Los Angeles.

Sherwood, D. E. (1988). Effects of bandwidth knowledge of results on movement consistency. *Perceptual and Motor Skills*, 66, 535-542.

Slackmecka, N.J., & Graf, P. (1978). The Generation Effect: Delineation of a Phenomenon. *Journal of Experimental Psychology: Human Learning and Memory, 4(6),* 592-604.

Squire, L.R. (1992). Declarative and Nondeclarative Memory: Multiple Brain Systems Supporting Learning and Memory. *Journal of Cognitive Neuroscience*, *4(3),* 232-243.

Squire, L.R. and Zola, S.M. (1996). Structure and Function of Declarative and Nondeclarative Memory Systems. In the *Proceedings of the National Academy of Sciences*, *93(24),* 13515-13522.

Stickgold, R. (2005). Sleep-Dependent Memory Consolidation. *Nature*, *437(1)*, 1272-1278.

Sutherland, I. E. (1964). Sketch Pad A Man-Machine Graphical Communication System. In the *Proceedings of the Design Automatic Conference*, 6329-6346.

Stanekova, L. and Stanek, M. (2013). Analysis of Dictionary Methods for PIN Selection. *Computers and Security*, *39*, 289-298.

Tu, H. Ren, X., & Zhai, S. (2012). A Comparative Evaluation of Finger and Pen Stroke Gestures. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 1287-1296.

Tulving, E. and Markowitsch, H.J. (1998). Episodic and Declarative Memory: Role of the Hippocampus. *Hippocampus*, *8(3),* 198-204.

Van Merriënboer, J.J. & Sweller, J. (2005). Cognitive Load Theory and Complex Learning: Recent Developments & Future Directions. *Educational Psychology Review*, *17(2),* 147-177.

Velloso, E., Bulling, A., & Gellerson, H. Motionma. (2013). Modelling and Analysis by Demonstration. In The *Proceedings Of ACM Conference On Human Factors In Computing Systems,* 1309-1318.

Vogel, D. & Balakrishnan, R. (2004). Interactive Public Ambient Displays; Transitioning from Implicit To Explicit, Public To Personal, Interaction With Multiple Users. In The *Proceedings of the ACM Symposium on User Interfaces & Software Technology*, 137-146.

Von Zezschwitz, E., Dunphy, P., & De Luca, A. (2013). Patterns in the Wild: A Field Study of the Usability of Pattern and Pin-Based Authentication on Mobile Devices. In the *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services,* 261-270.

Vogel, D. and Balakrishnan, R. (2004). Interactive Public Ambient Displays: Transitioning from Implicit To Explicit, Public To Personal, Interaction with Multiple Users. In the *Proceedings of the ACM Symposium on User Interfaces and Software Technology,* 137-146.

Walker, N. (1986). Direct Retrieval from Elaborated Memory Traces. *Memory and Cognition*, *14(4),* 321-328.

Ward, P., Hodges, N.J., Williams, A.M., & Starkes, J.L. (2004). Deliberate Practice and Expert Performance. *Skill Acquisition in Sport: Research, Theory & Practice*, *231*, 434-473.

Weiss, R. and De Luca, A. (2008) PassShapes: utilizing stroke based authentication to increase password memorability. *Proceedings of the 5th Nordic conference on Human-computer interaction,* 383-392.

Wigdor, D., Benko, H., Pella, J., Lombardo, J., & Williams, S. (2011). Rock and Rails: Extending Multi-Touch Interactions with Shape Gestures To Enable Precise Spatial Manipulations. In *Proceedings SIGCHI Conference on Human Factors in Computing Systems*, 1581-1590.

Wilson, M. (2002). Six Views of Embodied Cognition. *Psychonomic Bulletin and Review*, *9(4),* 625-636.

Winstein, C.J. and Schmidt, R.A. (1990). Reduced Frequency of Knowledge of Results Enhances Motor Skill Learning. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *16(4)*, 677-691.

Wobbrock, J.O., Morris, M.R. and Wilson, A.D. (2009). User-Defined Gestures for Surface Computing. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1083-1092.

Wobbrock, J. O., Myers, B., & Kembel, J. (2003). Edgewrite: A Stylus-Based Text Entry Method Designed for High Accuracy and Stability of Motion. In *Proceedings of the ACM Symposium on User Interfaces and Software Technology,* 61-70.

Wobbrock, J.O., Wilson, A.D. and Li, Y. (2007). Gestures without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. In the *Proceedings* of the *ACM Symposium on User Interfaces and Software Technology,* 159-168.

Wolf, C.G., & Morrel-Samuels, P. (1987). The Use of Hand-Drawn Gestures for Text Editing. *International Journal of Man-Machine Studies*, *27(1),* 91-102.

Wulf, G. Raupach, M., & Pfeiffer, F. (2005). Self-Controlled Observational Practice Enhances Learning. *Research Quarterly For Exercise & Sport, 76(1)*, 107-111.

Wulf, G., & Schmidt, R. A. (1989). The learning of generalized motor programs: Reducing the relative frequency of knowledge of results enhances memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, I5, 748-757.

Wulf, G., & Shea, C. H. (2004). Understanding the role of augmented feedback. *Skill acquisition in sport: Research, theory and practice*, 121-144.

Wulf, G., Shea, C., & Lewthwaite, R. (2010). Motor skill learning and performance: a review of influential factors. *Medical Education*, 44(1), 75–84.

Wulf, G., Shea, C. H., & Matschiner, S. (1998). Frequent feedback enhances complex motor skill learning. *Journal of motor behavior*, 30(2), 180-192.

Yao, W. X., Fischman, M. G., & Wang, Y. T. (1994). Motor skill acquisition and retention as a function of average feedback, summary feedback, and performance variability. *Journal of motor behavior*, 26(3), 273-282.

Zeleznik, R. C, Bragdon, A., Liu, & Forsberg, A. (2008). Lineogrammer: Creating Diagrams by Drawing. In the *Proceedings of the ACM Symposium on User Interfaces and Software Technology*, 161-170.

Zhai, S. and Kristensson P.O. (2003). Modeling Human Performance of Pen Stroke Gestures. In the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 97-104.