EXPLORING THE MULTI-TOUCH INTERACTION DESIGN SPACE FOR 3D VIRTUAL OBJECTS TO SUPPORT PROCEDURAL TRAINING TASKS

by

SARAH HOLDERNESS
M.S. University of Central Florida, 2011

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2017

Major Professor: Joseph J. LaViola Jr.

ABSTRACT

Multi-touch interaction has the potential to be an important input method for realistic training in 3D environments. However, multi-touch interaction has not been explored much in 3D tasks, especially when trying to leverage realistic, real-world interaction paradigms. A systematic inquiry into what realistic gestures look like for 3D environments is required to understand how users translate real-world motions to multi-touch motions. Once those gestures are defined, it is important to see how we can leverage those gestures to enhance training tasks.

In order to explore the interaction design space for 3D virtual objects, we began by conducting our first study exploring user-defined gestures. From this work we identified a taxonomy and design guidelines for 3D multi-touch gestures and how perspective view plays a role in the chosen gesture. We also identified a desire to use pressure on capacitive touch screens. Since the best way to implement pressure still required some investigation, our second study evaluated two different pressure estimation techniques in two different scenarios.

Once we had a taxonomy of gestures we wanted to examine whether implementing these realistic multi-touch interactions in a training environment provided training benefits. Our third study compared multi-touch interaction to standard 2D mouse interaction and to actual physical training and found that multi-touch interaction performed better than 2D mouse and as well as physical training. This study showed us that multi-touch training using a realistic gesture set can perform as well as training on the actual apparatus.

One limitation of the first training study was that the user had constrained perspective to allow for us to focus on isolating the gestures. Since users can change their perspective in a real life training scenario and therefore gain spatial knowledge of components, we wanted to see if allowing users to alter their perspective helped or hindered training. Our final study compared training with Unconstrained multi-touch interaction, Constrained multi-touch interaction, or training on the actual physical apparatus. Results show that the Unconstrained multi-touch interaction and the Physical groups had significantly better performance scores than the Constrained multi-touch interaction group, with no significant difference between the Unconstrained multi-touch and Physical groups. Our results demonstrate that allowing users more freedom to manipulate objects as they would in the real world benefits training.

In addition to the research already performed, we propose several avenues for future research into the interaction design space for 3D virtual objects that we believe will be of value to researchers and designers of 3D multi-touch training environments.

This dissertation is dedicated to my husband Will, my daughter Rose, my mother Susan, and my father Roger, whose encouragement pushed me across the finish line.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Joseph J. LaViola Jr., for his continued guidance and support throughout my research and dissertation journey. I also wish to thank the esteemed members of my dissertation committee, Dr. Gita Sukthankar, Dr. Charles Hughes, and Dr. Martin Hachet for their valuable feedback and guidance. I am also grateful for the assistance of past and present members of Interactive Systems and User Experience Research Cluster of Excellence (ISUERCE), without whose support this work would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Multi-touch interfaces are now prolific with their use in monitors, laptops, tablets, and phones. Touchscreens initially became popular with the advent of the smart phone but the small display sizes limited the type of gestures that could be used. Users were limited to 1 or 2 finger gestures allowing for rotation, scaling, and translation (RST) of objects on the display. Now larger touch-screens are more affordable and more widespread such as hybrid laptops and larger displays like the Microsoft Perceptive Pixel. These larger displays make multi-touch gestures that incorporate multiple fingers or multiple hands possible but, for the most part, the main gestures used remained the same. These 1-2 finger RST gestures are ideal for operations in the same plane as the multi-touch display, however a definitive way of adapting gestures into the third dimension has not yet been determined.

Multi-touch interaction is currently used in 3D gaming, simulation, and training environments using the RST gestures but we believe the interaction could be improved to map more closely to a realistic experience. Ultimately, we envision 3D multi-touch interaction to be modeled more closely after interactions with the physical environment thus enhancing engagement or learning.

Statement of Research

A major challenge in 3D touch interaction is understanding how to translate interaction on a 2D surface to a 3D environment. For each touch point on the screen, we know its position and contact size and can estimate its orientation and pressure. There are interaction paradigms that map this 2D input to 3D interactions by building upon the standard multi-touch RST (Rotation, Scaling, Translation) interactions for operations such as pinching, zooming, scrolling. However, most of these paradigms are metaphors for interaction. Instead, we envision providing users the ability to mimic real world gestures and manipulate objects as they would in real life. We would also like users to be able to pick up, rotate and move objects in a way that's relevant to training environments. The first problem we would like to solve is how do we get closer to mimicking real world interactions. More specifically, how can we design realistic and intuitive multi-touch interactions for 3D virtual objects that mimic real world object interaction ?

The second question we would like to examine is whether implementing these realistic multi-touch interactions in training contexts provides procedural training benefits such as enhanced learning, reduced training times, and reduced costs. If realistic multi-touch interactions provide training benefits, we can leverage the prevalence of larger multi-touch displays to provide intuitive training experiences. VR and AR simulations are often examined as inexpensive alternatives to training on actual equipment however, multi-touch interaction might be more suitable for certain types of training that don't require complete immersion. In addition, multi-touch training environments could provide the same benefits without the risks of cybersickness symptoms such as eye strain, headache, dizziness or nausea that is common in VR and AR simulations [1].

To address these questions, we aim to create design guidelines for realistic and intuitive multitouch interaction for 3D virtual objects and we will determine whether implementing these interactions in multi-touch training environments benefits procedural training outcomes.

Thesis Statement

By using realistic multi-touch gestures to manipulate 3D objects, we can provide procedural training recall outcomes comparable to training in the real world.

Approach

To address these questions, we conducted two studies that resulted in design guidelines for realistic and intuitive multi-touch interaction for 3D virtual objects. After developing the guidelines we conducted two additional studies to measure whether or not implementing these interactions in multi-touch training environments benefits procedural training outcomes.

In the first study, we explore the best ways to use multi-touch interaction in 3D training environments [2]. We performed a gesture elicitation user study to define which gestures users preferred on large capacitive displays when interacting with 3D objects. We found that many users' first instinct is to use 1 or 2 finger gestures similar to RST gestures used on a phone. However, when prompted to do so, users tried to mimic physical, real-world actions such as using more fingers and applied pressure to signify more force or to move an object into the screen. These results signified that multi-touch gestures could mimic real world actions and be intuitive to users.

In our second user study we focused on defining the best ways to estimate pressure on capactivate displays to enhance physical gestures [3]. Today's most prevalent touchscreens use capacitive sensing which can report the location and surface area of a touch, but do not directly sense the pressure of the touch. Thus on capacitive screens, multi-touch pressure can only be estimated based upon touch contact size. We explored how best to interpret pressure from contact size by developing multiple estimation techniques. We also examined which of our pressure estimation techniques were preferred among different scenarios.

In our third study we examined how multi-touch interaction affects the learning of a procedural assembly task [4]. Our goal was to evaluate how our gesture set impacts learning in a training environment and if gesture similarities to physical actions better prepare the user for performing the physical steps in the real world. Since training can be looked at in many different ways we are focusing on procedural learning, or learning a repeatable step-by-step process. The study evaluates the knowledge transfer acquired with multi-touch interaction technology compared to standard training methods. We compared multi-touch interaction to 2D mouse interaction and to actual physical training. We found that multi-touch interaction performed better than 2D mouse and performed as well as physical training.

In our final study, we examined how having the freedom to explore in a training environment affects learning. Our third study, like many multi-touch training systems and 3D training systems in general, had a snap to position functionality that didn't give the user the freedom to manipulate the objects in great detail or to look around and examine different parts of objects as they would in the real world. We evaluate the benefits of multi-touch training that has unconstrained object manipulation, where the user can pick up, rotate and examine parts. We compared this unconstrained multi-touch group to a group that has constrained object manipulation, in which the parts automatically align themselves and the user cannot make their own rotations. In addition, we compared both groups to a control group performing physical training with the real world apparatus. The selected experimental task consists of assembling a dog treat dispenser prototype built with a Raspberry Pi and other electrical components. Since the apparatus and parts are small they themselves would be rotated by ones hands to examine and rearrange before assembly. This is different than training on a large apparatus where the user might walk around the apparatus to examine and place parts. We have chosen to look at a small apparatus so that we can isolate object manipulation, instead of exploring both object manipulation and camera/viewpoint manipulation.

Contributions

In this dissertation we:

- Developed design guidelines for realistic multi-touch gestures for 3D training environments
- Studied how using realistic multi-touch gestures for 3D training environments affects procedural training

In order to realize these contributions, we:

- Performed a gesture elicitation user study to define which gestures users preferred on large capacitive displays when interacting with 3D objects. We found that when prompted to do so, users tried to mimic physical, real-world actions such as using more fingers and applied pressure to signify more force or to move an object into the screen.
- Developed two pressure estimation techniques to interpret pressure from contact size and evaluated them in a user study with two different tasks.
- Evaluated the knowledge transfer acquired by training with multi-touch interaction using realistic gestures, compared to standard training methods. We found that multi-touch interaction performed better than 2D mouse and performed as well as physical training.
- Evaluated the knowledge transfer acquired by training with unconstrained multi-touch object manipulation compared to constrained object manipulation. We found that multi-touch interaction with unconstrained object manipulation performed better than constrained object manipualtion, and performed as well as physical training.

Dissertation Outline

In Chapter 2, we discuss work related to different types of multi-touch interaction, multi-touch interaction in education and training, and procedural training in virtual environments. Next, we present our work on user-defined gestures for 3D objects in Chapter 3, which led to the development of a taxonomy and gesture design guidelines. Following that, in Chapter 4, we discuss the follow up study on pressure estimation techniques for 3D interaction on capacitive touch screens. Chapter 5 presents the first training study that evaluated the use of 3D multi-touch gestures compared to other training methods. Our second training study examining multi-touch interaction with constrained and unconstrained object manipulation, is presented in Chapter 6. Next, we discuss our findings and future work opportunities in Chapter 7. Finally, we summarize our findings in Chapter 8.

CHAPTER 2: RELATED WORK

The goal of this chapter is to understand the current state of two areas of research: multi-touch gestures and training applications. First we will begin with a brief overview of terminology. Multi-touch gestures will be defined as predefined motions on the surface of the device that interact with objects in a virtual environment. The term multi-touch interaction will represent the entire interaction; the physical gesture, the result of the interaction, and the context in which the interaction is performed.

The multi-touch gesture discussion will begin with a brief historical overview of multi-touch gestures and recognition. We will continue with an overview of current multi-touch techniques for 3D environments, gesture elicitation studies and their resulting taxonomies, and gestures above the surface. Then we will discuss work relevant to pressure estimation on touch screens. The training discussion will begin with education and training applications with multi-touch interaction. Then we will review relevant procedural training applications in the virtual and augmented reality domains. Finally, we will cover learning theories related to enactive learning and increased interface complexity.

Types of Multi-touch Interaction

Multi-touch Gestures and Recognition

Early gestural input research began with work by Coleman creating a text editor based on proofreaders' pencil markup [5], Rhyne constructing a spreadsheet application that combined gesture and handwriting [6], and Buxton et al. producing a musical score editor with gestures to enter notes [7]. Following this work, Rubine automated gesture recognition by creating a system to use gesture examples instead of hand coded recognition [8]. Since then there have been many advances in gesture recognition using template matching [9], feature-based [10], signal-based [11], and continuous approaches [12]. Increased accuracy in gesture recognition has allowed for multi-touch interfaces to gain traction in a variety of domains. However, Ingraham found in a comprehensive research review that researchers still do not have consensus on how to implement intuitive interactions and users do not agree on which gestures are intuitive for which interactions [13]. Her findings also indicate that the intuition of multi-touch interactions can be improved by considering factors such as direct manipulation, physics, feedback, previous knowledge, and physical motion.

3D Direct Manipulation and Physics Simulations

There has been much recent work on the manipulation of 3D objects on multi-touch surfaces. Direct manipulation is a widely explored strategy for this task, since the direct manipulation RST method has wide appeal in 2D contexts. Reisman et al. extends this common 2D paradigm to 3D by allowing direct manipulation of 3D objects with 3 or more touch points [14]. Hancock et al. also explores direct manipulation in [15] using one, two and three touch input techniques in shallow depth 3D. Physics based approaches have been applied by Wilson et al. [16] by creating solid proxy objects in the scene for each touch points. Physics based grasping behavior has also been explored by Wilson in a later work [17] where objects are manipulated by a stream of fluid particles. Cohé adapted the common mouse and keyboard transformation widgets to the tactile paradigm by creating a new 3D transformation widget tBox [18]. Cohé's work focuses on the direct manipulation of objects or widgets, whereas our work explores how users intuitively act on, and how users prefer to act on, 3D objects from a certain domain.

Metaphors for Manipulation

Gestures that act as metaphors for real world actions have been seen in previous work. Hancock et al. explored propagating behaviors to other objects in the scene by using metaphors, such as throwing a blanket object on top of another object to cover it in a texture [19]. Ruiz et al. elicit user-defined gestures for mobile interaction and find several themes similar to the ones we are exploring: actions that mimic normal use (such as putting a mobile phone to the ear for a motion gesture to answer a call), and real-world metaphors (such as placing a phone face down to hang up a phone call as you would have with a rotary phone) [20]. Kray et al. also discover user-defined gestures that act as metaphors for connecting mobile devices, displays and tabletops, such as starting with two phones near each other and then pulling away to disconnect them [21]. Kurdyukova et al. investigate gestures for data transfer between iPads and other devices, finding that both experienced and inexperienced users rely on real-life metaphors when thinking of well-matching gestures [22].

Gestures Above the Surface

The availability of low cost tracking solutions such as the Microsoft Kinect and Leap Motion have allowed combining above the surface interaction with on the surface interaction. Marquardt et al. called the touch surface combined with the area above the surface the continuous interaction space [23]. They also presented a variety of interaction categories that exploit the space between these modalities such as extended continuous gestures to avoid occlusion, raycasting gestures for extended reach, lifting gestures to reveal objects and to adjust scale, stacking objects, and 6 degrees of freedom (DOF) manipulation. Wilson et al. proposed several metaphors to interact with different surface displays while capturing full body posture [24]. For example, after performing multi-touch interactions on a virtual object on the tabletop, the user may transfer the object to another display by simultaneously touching the object and the destination display. Or the user may pick up the object by sweeping it into their hand, see it sitting in their hand as they walk over to an interactive wall display, and drop the object onto the wall by touching it with their other hand. Mockup Builder is an on-and-above-the-surface interaction technique based on asymmetric bimanual interaction for creating and editing 3D models in a stereoscopic environment [25]. A user evaluation comparing Mockup Builder to Sketchup demonstrated promising results for this type of alternative modeling interface.

Low cost tracking solutions have also made estimating hand posture possible for surface interactions. The Extended Multi-touch approach presented by Murugappan et al. demonstrated the accuracy of hand posture tracking and user differentiation with the Microsoft Kinect [26]. Song et al. applied this type of hand posture augmented multi-touch to exploratory visualization applications [27].

Above the surface interaction has also been explored to make direct interaction with 3D stereoscopic objects possible. Toucheo implements a semi-transparent mirror to reflect a stereoscopic image over a touch screen [28]. The user can then interact with the stereoscopic image via 9 DOF widgets on the touchscreen. Triangle cursor allows direct multitouch interaction to specify a 3D position and yaw rotation above the interaction surface [29]. Both Toucheo and Triangle Cursor were designed to avoid occlusions and disturbing the stereoscopic perception.

User-Defined Gestures

To create a user-defined, intuitive gesture set, Wobbrock et al. performed a study that elicited natural gestures from naive users [30]. Participants were presented with tasks to perform by showing the effect and asking the user to perform a gesture that would cause that effect. The users were asked to perform the gesture one-handed and then two-handed. It was determined that the number of fingers used for gestures was arbitrary for the same task and that users preferred one-handed to two-handed gestures. Because the study primarily focused on desktop operations and tasks, the final gesture set was heavily influenced by WIMP paradigms and yielded mainly metaphorical or symbolic gestures. They have since evaluated their user-defined gesture set against a gesture set created by designers and shown that the user-specified gesture set is easier for users to master [31].

There have since been many studies eliciting user-defined gestures based on Wobbrock's experimental design. Cohé and Hachet conducted a user study to examine how users perform rotations, scaling, and translations on a 3D cube [32]. Our first study is similar to Cohé's except that we have added different objects and tasks to perform as well as two trials of the experiment. The first trial is similar to Cohé's in that it just asks the user to perform the gesture they think is appropriate for the given task. The second trial of the first study is different than Cohé's in that we ask the users to perform gestures as if they were manipulating the object in the real world. We believe that this is an important addition given our focus on multi-touch gestures for training and simulation applications in 3D environments. In addition to Wobbrock and Cohé's work, Ruiz et al. performed a study to elicit user-defined motion gestures for mobile interaction [20], Micire et al. studied user gestures for robot control and command in a 3D virtual environment [33], and Mauney et al. analyzed data from 9 different countries to determine cultural similarities and differences in user-defined gestures for touchscreen user interfaces [34].

Classifications and Taxonomies

Wobbrock et al. also presented a taxonomy of surface gestures based on user behavior [30]. Based on a collection of gestures from participants, their taxonomy classifies gestures into four dimensions: form, nature, binding, and flow. Cohé and Hachet and Ruiz et al. also adapted Wobbrock's original taxonomy to classify their specific gesture domains [32, 20]. We build upon the taxonomies created by Wobbrock and Cohé from the results of our first study in Chapter 3.

Pressure as Multi-touch Input

Non-Capacitive Sensing Pressure Estimation

Since any use of touch inherently uses pressure, there have been many investigations into incorporating pressure information into surfaces by using malleable materials, such as with liquid displacement sensing [35]. Early work investigating pressure as computer input began with Herot and Weinz in 1978 [36], followed by Buxton concluding in 1985 that pressure control without feedback (i.e a button click) can be difficult but it is a promising research area [37]. Since then there have been several investigations into using pressure sensors as input [38], what pressure force levels are comfortable for users [39], and how many levels are distinguishable [40]. Brewster et al. aimed to use pressure to improve input performance when entering mixed-case text. Their experiment used a mobile device with a resistive touch screen which measures pressure [41]. Their results demonstrated that pressure input can outperform a standard shift-key keyboard design for mobile text entry.

There have also been investigations into input devices with pressure sensors. Graspzoom attached a Force Sensitive Resistor to the backside of a mobile phone to allow single-handed input for bidirectional controls like zooming in and out [42]. Instead of a pinch to zoom gesture, which often requires two hands (one for holding the phone and one for gesturing), their input model uses tiny thumb gestures using pressure on the back of the phone. Pressure Widgets explore the design space of using the continuous pressure sensing capabilities of styluses to operate multi-state widgets [40]. Cechanowicz et al. investigated the use of a uni-pressure and dual-pressure augmented mouse. They found users can comfortably control up to 64 modes with a dual-pressure augmented mouse [43].

Since most of today's widely used multi-touch devices, whether mobile or desktop, use a capacitive sensor matrix, we chose to focus on the feedback available from capacitive devices. Capacitive devices only report the contact size based upon pixel coverage and are not capable of sensing pressure forces. Liquid displacement sensing would allow for more exact pressure sensing, and even vision-based systems can do better than capacitive by using the contact point's brightness [44]. However, even if high accuracy pressure sensing components were made widely available, it would make some actions very difficult as pressure increases the touch's friction on the surface. Recently, Apple began incorporating their force sensor and Force Touch gestures into mobile phones and trackpads, but the gestures are used mainly for simple desktop selection operations, not for applying force during movement [45].

Capacitive Sensing Pressure Simulation

We use the term pressure "simulation", since there are no pressure sensors on capacitive touch screens therefore all techniques explored are pressure estimations, instead of true sensing. There have been some pressure estimation techniques targeted at capacitive mobile devices that try to capture actual pressure by proxy techniques. For example, Pseudo-pressure is a pressure estimation method which assumes increases in pressure create both jitter in contact locations as well as increased touch duration [46]. This technique was used as a way to reject text entry suggestions on mobile phones. However, Pseudo-pressure's jitter would not be reliable on a large screen, especially when applying pressure during another translation or rotation gesture. The time duration would also not be applicable for the 3D tasks we evaluated. Vibpress utilized a mobile devices built-in microphone to detect five different pressure levels by mapping sound amplitudes to different pressure levels [47]. Forcetap analyzed acceleration data along the z-axis to differentiate between a strong tap and a gentle tap on touch screens [48]. All of these methods are limited for our purposes because they focus on mobile applications and aren't applicable for applying pressure during translation or rotation.

As a way of simulating pressure on capacitive devices, there have been several developments that take advantage of changes in contact size, as we explore in our second study. As explained in [44], contact size can be altered by either pressure or finger-tip angle. However, applying more or less pressure on a rigid surface will only slightly change the contact size thus finger-tip angle generates larger contact size deltas. An example of a multi-touch technique that takes advantage of contact size is Sim-Press which simulates clicking by mapping the changes in the finger's contact area to changes in pressure [49]. Similarly, Fat Thumb is a mobile technique for one handed zooming that uses increases in the thumb's contact size to trigger different zoom levels [44]. Another technique that uses the finger-tip angle in multi-touch interaction is Microrolls. Microrolls doesn't imitate pressure as Fat Thumb and SimPress do, but instead interprets small rocking movements of the thumb to trigger gestures without requiring any tangential movement [50]. Similar to Microrolls, Thumbrock [51] and Shear Force [52] interpret different actions or forces based upon the change in orientation of the finger.

In our second study we build upon this existing work in the following ways: First, similar to FatThumb and SimPress, in both of our estimation techniques we interpret relative changes in contact size as a way to estimate pressure. However, both of their approaches require calibration to determine passing a threshold point. We also examine an alternative technique: comparative pressure estimation, which knows nothing of the user's calibration contact size only their initial touch's contact size. Second, FatThumb and SimPress look at contact size as a threshold to activate either clicking or zooming modes, whereas we map pressure continuously to depth position. Third, these studies focus on mobile and/or GUI settings in 2D tasks; we focus on contact size changes during multi-touch interaction within 3D virtual environments. Finally, we also apply the two pressure estimation techniques to different 3D tasks that vary in the dexterity required by the users (fine versus gross motor skills).

Multi-touch Interaction in Education and Training

There has been much work in interactive tabletops in education for encouraging inquiry-based learning, experiential learning, or collaborative learning. Schneider describes four tabletop learning environments for science education and how each one positively affects inquiry-based learning [53]. The outcomes of how these systems can positively affected inquiry-based learning were: 1) rich visualizations may increase the way people collaborate, 2) rich interactions have the potential to foster engagement and exploration of a problem space, and 3) the main benefit of technology-enhanced learning environments may in fact not be learning, but preparation for future learning. Shaer et al. presented G-nome Surfer 2.0, a tabletop interface for fostering inquiry-based learning of genomics [54]. Their findings indicate that G-nome Surfer improves students' performance, reduces workload, and increases enjoyment. They also compared G-nome surfer on a tabletop to a multi-mouse implementation in a study and found that the tabletop condition resulted in four educational benefits: 1) increasing physical participation, 2) encouraging reflection, 3) fostering effective collaboration, and 4) facilitating more intuitive interaction.

The Flow of Electrons [55] provides an augmented workspace for learning physical computing experientially. By allowing users to place electronics components on the surface and experiment with wiring and the outcomes, it allows users to make mistakes without fear of breaking anything. It bridges the gap between digital information and actual hardware components. Similarly, Piper and Hollan compared the affordances of presenting educational material on a tabletop display with presenting the same material using traditional paper handouts [56]. The affordances of study materials on the tabletop versus paper allowed for more playfulness and experimentation since the notes or drawings created were not permanent and were easily edited. The ability for users to easily and immediately experiment and practice without the fear of making permanent mistakes is something that can benefit training applications as well, since users may be intimidated by expensive equipment or dangerous medical procedures.

Multi-touch interfaces have not been explored as much in the context of training. OrMiS [57], a tabletop interface for simulation-based training, where military officers use a map-based tool to carry out strategic maneuvers and combat, enabling large-scale training exercises without the cost of field deployment. OrMiS is designed to replace traditional PC-based simulation tools while improving ease of learning and better facilitating collaborative work. OrMis focuses on 2D map-based strategic tasks whereas we focus on training on a physical apparatus in a 3D environment using realistic gestures. For medical training SimMed implements a 3D simulation in an augmented tabletop environment to teach medical procedural skills for diagnosis and treatment [58]. They observed high levels of immersion and positive social aspects, as well as observations that suggest a significant learning effect. Their training was very open-ended in order to explore the diagnostic techniques trainees chose, whereas we are focused on training a specific step-by-step procedure. There are no other studies, that we are aware of, which evaluate the efficiency and effectiveness of procedural training using multi-touch interactive surfaces.

VR and AR Procedural Learning

Although there hasn't been much work with multi-touch systems in the areas of maintenance and medical training, there has been much work with Haptic, Virtual Reality (VR) and Augmented Reality (AR) systems. Haptic systems and VR technologies allow reproducing the conditions in surgery, yet without risks for the patient, and offer new opportunities for training. Gosselin et al. present a novel training platform offering high fidelity haptic interactions and show how this approach was applied in the context of maxillofacial surgery [59]. The first results demonstrate its efficiency in qualifying expertise and training people. PerioSim is a haptic simulator that has been developed as an aid for the sensorimotor skill acquisition in dentistry and is being evaluated in a classroom setting [60]. Kang et al. investigated several repetitive training schedules to determine which was the most effective in training on a robotic VR simulator, which simulates the da Vinci surgical system [61]. They found that daily 1-hour practice sessions performed for 4 consecutive days resulted in the best final score, continuous score improvement, and effective training while minimizing fatigue versus other training schedules.

There have been many studies evaluating the effectiveness of VR and AR in training compared to traditional methods, reporting in some cases that these methods perform as well as traditional hands-on methods. For instance, Ganier et al. compared tank maintenance procedure performance after conventional training with a real tank and those trained in a virtual environment. Both training groups were also compared to a no-training control group who carried out the procedure using only job instructions [62]. No differences were found for the testing task completion times for the test groups, but significant differences were found between both test groups and the control group. Indicating that procedural knowledge learned in a virtual environment is transferred to real-world performance. Another study evaluated different Virtual Reality interaction technologies in learning an industrial maintenance task [63]. The four different interaction technologies were basic mouse, 2D mocap, 3D mocap, and a haptic device. They found negligible impact on the learning of assembly task when the focus is on transfer of procedural knowledge rather than the transfer of sensorimotor skills. In addition, users that trained with mouse and 2D mocap took significantly less time training.

Henderson and Feiner evaluate an augmented reality (AR) user interface designed to assist users in the psychomotor phase of procedural tasks [64]. Their user study showed participants completed the psychomotor aspects of the assembly task significantly faster and with greater accuracy than when using assistance on a stationary LCD screen. Another study examined VR and AR techniques compared with classic training techniques. The VR and AR methods showed no significant differences over control methods. In addition, they found AR training for industrial maintenance and assembly (IMA) tasks can reduce the number of unsolved errors [62]. One of the main differences with these AR and VR studies is they they make use of 3D spatial interfaces [65] while we focus on 2D multi-touch gestures. We aim to expand upon these results in virtual procedural training by exploring whether multi-touch interaction, which leverages realistic gestures, can perform as well as training on the real equipment.

Learning with Increased Interface Complexity

Enactive learning interfaces, where knowledge constructed by an agent through its sensorimotor interactions with its environment, have been shown to be more effective than passive or vicarious learning methods [66]. With enactive learning in mind, we aim to leverage multi-touch interaction for realistic procedural training such as for industrial assembly tasks or medical procedures.

There has been some work in HCI that focuses on whether ”harder” interfaces, or interfaces that require greater effort from the user, benefit learning spatial tasks. Cockburn et al. found this to be the case in graphical user interfaces [67]. They state that if an interface design’s objective is to train users to interact with interfaces that depend on spatial properties, designers should explicitly increase the mental effort of interaction. From another perspective, the game ”Game Over!” was specifically designed to be inaccessible, in that it can be played by no one, in order to teach accessibility guidelines [68]. In our fourth study, by having unconstrained interaction where users are free to examine parts and by forcing users to rotate components until they are properly aligned before installing, we are inherently making the interaction more difficult. Our hypothesis in studies 3 and 4 are that by forcing this increased level of interaction the training more closely mimics the real world and will therefore benefit the learning of the real world task.

Conversely, Cockburn et al. also found that spatial memory performance deteriorated in both physical and virtual systems as their freedom to locate items in the third dimension increased [69]. Betella et al. also found that guided navigation resulted in better spatial memory versus free navigation in a 3D Mixed Reality Navigation task, citing equivalent exposure time and a larger screen sizes as possible reasons [70]. Our fourth study experiment also focuses on comparing how increasing the level of freedom in interaction affects memory, but we focus on procedural memory over spatial memory specifically. Richards and Taylor evaluated a 2D simulation versus a 3D virtual world for teaching the Marginal Value Theorem [71]. They found that due to potential cognitive overload and distractors in the virtual world, it appears that the two-dimensional NetLogo model delivered better learning outcomes. Based upon outcomes such as these, Stuerlinger et al. emphasize the value of constraints for 3D user interfaces for reducing cognitive load and present a set of guidelines for constraints [72]. From these works we understand that it is possible that by increasing the level of multitouch interaction in our training interface we may be introducing cognitive load which may distract from learning. However, we believe that the combined similarity of the interaction required in the virtual environment compared to the real world task environment benefits learning.

CHAPTER 3: USER-DEFINED MULTI-TOUCH GESTURES FOR 3D OBJECTS

Introduction

In this chapter, we begin to explore what gestures users choose when asked to interact with different 3D objects on a multi-touch surface. As a result, we also examine how familiarity with popular, metaphorical multi-touch gestures (e.g., a swipe gesture to unlock) translates to interacting with 3D objects that afford physical actions (e.g., a knob affords turning). We will use Wobbrock et al.'s definition of a metaphorical gesture as a gesture that acts on, with, or like something else, and the definition of a physical gesture as a gesture that should ostensibly have the same effect on a table with physical objects [30]. In this study we specifically focus on interactions with 3D objects that have rotational, tightening, or switching components on mechanisms that might be found in a mechanical equipment operation and training simulations. We believe this research is a necessary precursor for exploring how physical multi-touch gestures on objects can translate to learning the physical operations encountered in executing mechanical and maintenance tasks.

To begin our exploration, we performed a user study to elicit user-defined gestures for manipulating 3D objects on multi-touch surfaces using a study design established by Wobbrock et al [30]. The results indicate that users had a bias towards previously learned metaphorical multi-touch gestures when first asked to perform a gesture on a 3D physical object. We were also able to show that with instruction to interact with the object as they would in the real world, users would switch from a metaphorical gesture to a physical one. Our user study led to the following research contributions:

- A classification procedure to categorize gestures and to determine the nature of a gesture (i.e., whether it is metaphorical or physical in nature, or a combination of both).

- A user defined gesture set for multi-touch gestures applied to 3D objects.
- User preferences with regards to metaphorical versus physical gestures.
- A comparison to Cohé et al.'s work that elicits user gestures for RST operations on a 3D cube [32].

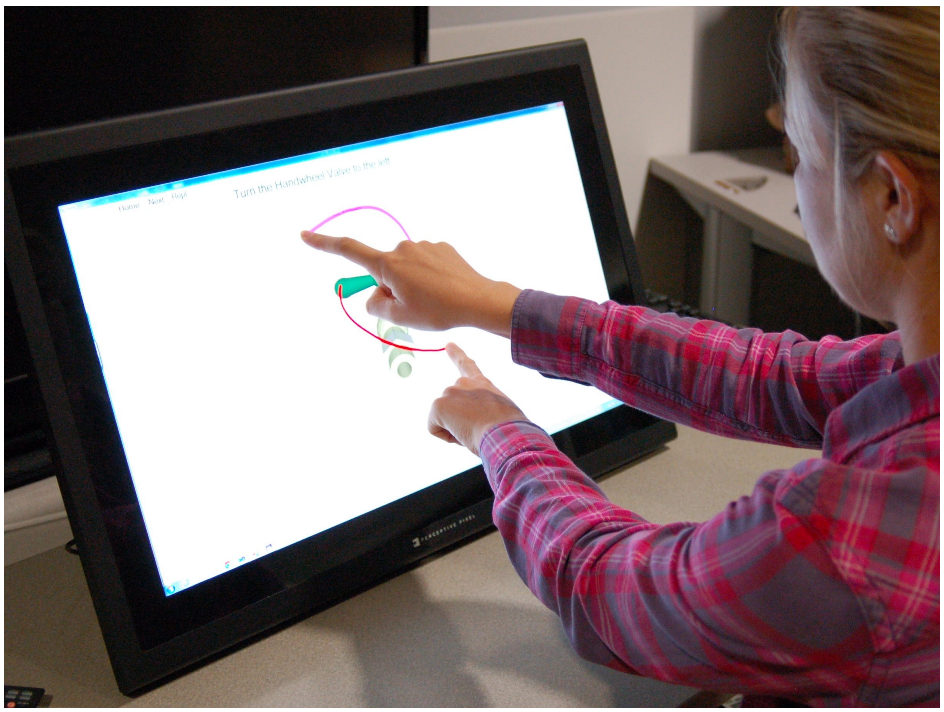


Figure 3.1: The Gesture Collection Apparatus and a user performing a two handed gesture.

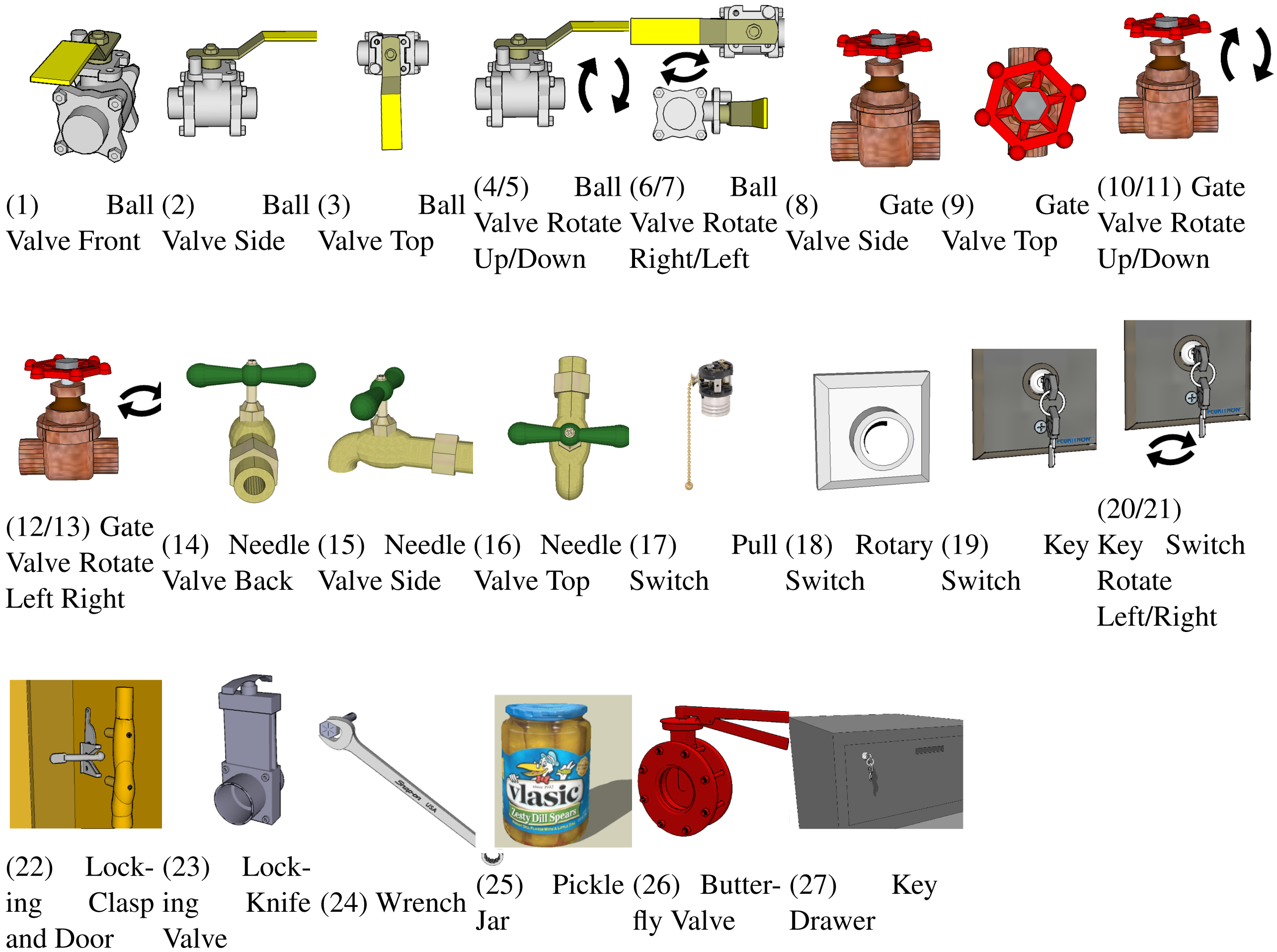


Figure 3.2: The 27 referents presented to participants in trials 1 and 2 of the user study.

User Study

Overview

The goal of our user-centered experimental design was to let users express their ideal gestures on objects that afford manipulation. The complex objects explored are valves, switches, tools, doors, and buttons that might be seen in mechanical equipment operation and training. Static images of these 3D objects were used, with minimal feedback given in the form of strokes drawn to the screen of their gesture path, in order to remove bias from expected reactions to gestures. Since all of our users have owned multi-touch devices there would be inherent bias in what users instinctively chose as their gestures. Thus, we decided to go through two trials of the experiment. In the first trial, we asked the users to perform the task on each object using whatever gesture they felt was appropriate. In the second trial, we asked the users to perform gestures that they would use if this object was a physical object in the real world.

Pilot Study

A Pilot study was conducted with 20 participants aged 19 to 26, with 13 males and 8 females. Both a 27" Perceptive Pixel LCD multi-touch display (PP display) and a Samsung Galaxy Tablet 10 (Galaxy Tab) were used. Although there are large differences in size and form factor between the Galaxy Tab and the PP display, there was no significant difference between devices in the gesture sets or number of fingers and hands used. Thus, the Galaxy Tab was omitted from this user study.

After the pilot study, we discovered gesture primitives (shown in Table 3.2) that applied to different objects in the current domain. We wanted to see if these gesture primitives applied to more complex interactions that required possibly two hands or combinations of gestures as well as navigational tasks. Thus after the pilot we added new referents to the experiment that required compound operations (referents 22, 23, 25, 26, and 27 shown in Figure 3.2). We also added new navigational referents which required the user to change the perspective of the object by rotating the viewpoint (referents 4, 5, 6, 7, 10, 11, 12, 13, 20, and 21). The navigational referents are different than the other referents that require activating or turning on, in that they lend themselves to metaphorical gestures. Due to this, we will leave these referents out when comparing the nature results of the two experiment trials.

Participants and Apparatus

There were 14 paid participants aged 18 to 29, all male. Two participants were left-handed and the remaining were right-handed. All participants owned a multi-touch device such as a phone, tablet or track-pad. Although all users were experienced with using multi-touch gestures on their devices, none had implemented a multi-touch application before. The experiment was conducted using a 27” Perceptive Pixel LCD multi-touch display (PP display). We developed an application (in C#) that displayed a static image of each referent next to an animated image of the task to perform (i.e., a valve opening) with a written description of the task (i.e. open this valve) as well. The application saved the users’ gestures to a database. The data saved for each contact point were the timestamp, size, pressure, touch id, and coordinates which allowed for animated playback. In addition, each user was recorded on video.

Procedure

Participants were asked to go through two trials of the experiment. For each trial the user was shown 27 referents (see Figure 3.2) accompanied by a task description on the screen (for example ”Turn the gate valve to the left.”). For the first trial, the participants were told to use whatever gestures they thought would be appropriate for accomplishing each task. For the second trial, the participants were told to use gestures as if each referent was a physical object in the real world. Immediately after each gesture was completed, participants rated their gesture on goodness and ease of use on a 7-point Likert scale. After the 14 participants had completed the experiment with 27 referents and 2 trials, a total of 756 gestures were collected and then classified.

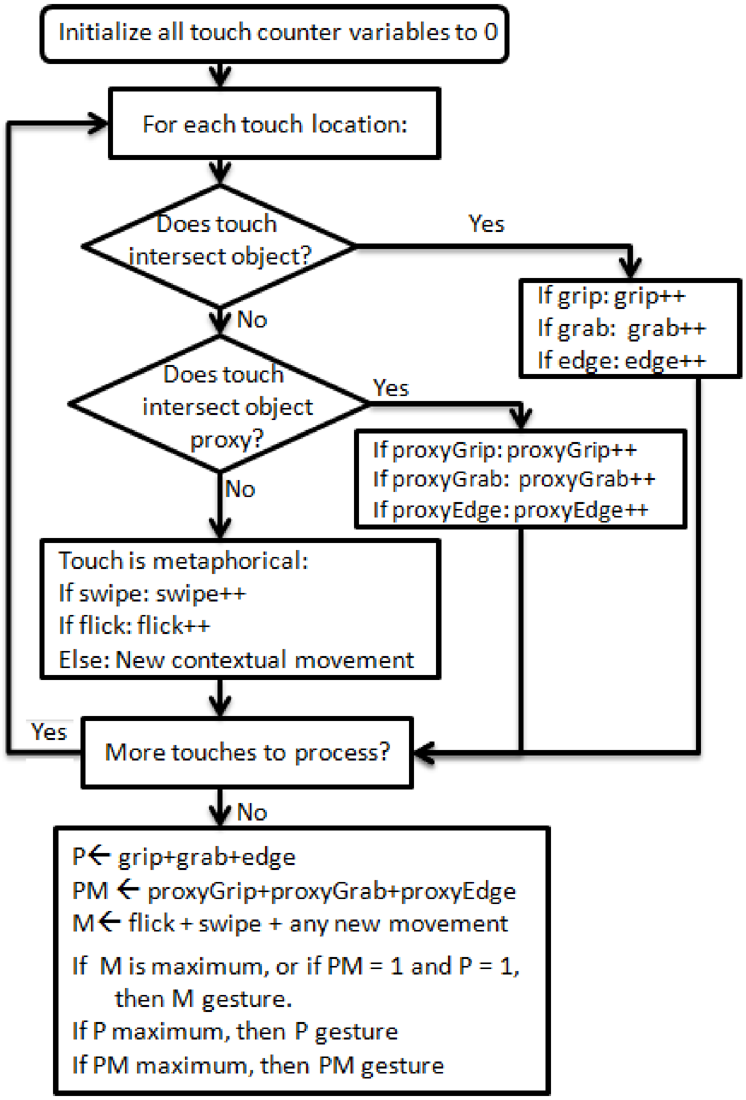


Figure 3.3: The nature classification process. The physical and proxy sub-categories are defined in Figure 3.4. The abbreviations P, PM, and M were used for Physical, Physically Metaphorical, and Metaphorical gestures respectively.

Classification Method

A systematic classification process was necessary to examine whether users interacted with the objects in a metaphorical or physical way, or a combination of both. We define physical gestures in the same way as Wobbrock does, gestures that should ostensibly have the same effect on a table containing those physical objects. However we distinguish ourselves somewhat from their definition by having the requirements that the gesture must use two or more fingers and majority of touches must make contact with the object. We make this distinction since if any of our objects was manipulated in the real world those requirements would be necessary. Metaphorical gestures are then any gesture that uses 1 finger if they are also acting like the physical motion, or any other gesture that is not representative of the physical motion but is a metaphor in another way (such as a line in the direction the object should move). Proxy gestures are gestures that act like the physical motion but do not make contact with the object. The reasons we make these distinctions is firstly to prepare for implementing a physics simulation driven by projected contact with the objects, and secondly to mimic real world interactions for the purposes of training.

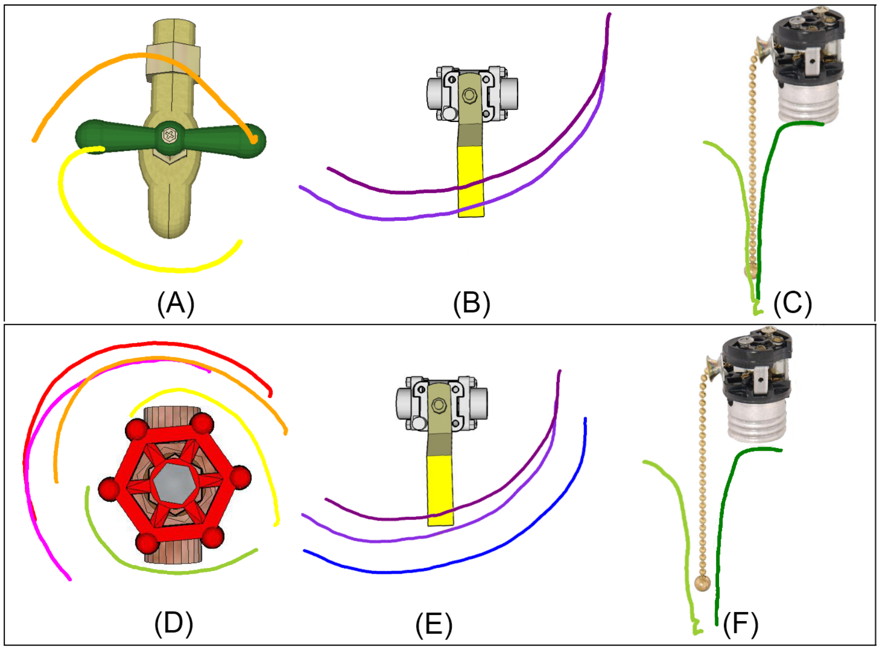


Figure 3.4: (A)-(C) demonstrate the Physical sub-categories from left to right: grip, edge, and grab. The gestures also correspond to 2 point finger turn, 2 finger curved line, and 2 finger pinch and pull. (D)-(F) demonstrate the Proxy sub-categories from left to right: proxy grip, proxy edge, and proxy grab. The gestures used are 5 point finger turn, 3 finger curved line, and 2 finger pinch and pull.

The classification process (Figure 3.3) begins by examining the initial location and path of each touch and determining if it intersected the object. If the touch intersected the object and followed the path in which was needed to apply force to the object, then it was classified as one of the physical gestures, either grip, grab, or edge (shown in Figure 3.4). A touch is classified as grip if its initial location intersects the object, edge if its initial location begins in empty space and then intersects the object, and grab if a pinching motion is done before the remaining gesture. Otherwise, if the touch did not intersect the object but the motion still followed the force path, it was labeled a proxy grip, proxy grab, or proxy edge gesture, which we define as physically based metaphorical gestures. These gestures mimic the physical motion done in real life, but since they do not make direct contact with the object they are metaphorical in nature.

Table 3.1: Sub classifications of gestures, as described in section 3.

Metaphorical		Physical	Metaphorical Physical
Hold (H)	Threshold Grip (TP)	Grip (P)	Proxy Grip (XP)
Swipe (S)	Threshold Grab (TB)	Grab (B)	Proxy Grab (XB)
Drag (D)	Threshold Edge (TE)	Edge (E)	Proxy Edge (XE)
Turn (N)			

Gestures that did not fall into the physical (P) or physically metaphorical (PM) categories, fell into the metaphorical (M) category. We originally established a threshold for number of fingers and pressure to classify metaphorical gestures. Since real world motions could not complete the tasks in our experiment with only 1 finger, the finger threshold was 1. We left out the pressure threshold since a suitable cross platform, cross gesture number was not feasible. The sub-categories for P, PM and M are shown in Table 3.1. After each touch location is classified, the sum of all of the P, PM, and M touches are compared to determine if the gesture is overall a P, PM, or M gesture. It is important to note that the abstract and symbolic gesture categories (presented in Wobbrock et al.'s taxonomy [30]) were left out of this classification method since no gestures in those categories were observed during our experiments.

Table 3.2: Observed gestures, as well as the observed range of fingers and hands used.

Gesture	Abbrev.	# Fingers	# Hands
Dot	D	1	1
Straight Line	SL	1-3	1
Curved Line	CL	1-5	1
Semicircle	SC	1-3	1
(Almost) Full Circle	AFC/FC	1-2	1
Spiral/ Multiple Circles	SP/MC	1	1
Finger Turn	FT	2-5	1
Hold and Turn	HT	2-5	1
Pinch	PI	2	1
Pinch and Pull	PP	2-3	1
Hold and Drag	HD	2	2
Both Hands, Opposing Directions	BHOD	2	2

Experiment Results

User-Defined Gesture Set

After all of the participants had run through the gesture collection experiment, each user's gestures were classified according to the overall observed gestures shown in Table 3.2, the number of fingers and hands used, and the category (see Figure 3.4). Confirming Wobbrock et al.'s findings, our observations indicate that the number of fingers used is arbitrary in the interpreted gesture. However, the number of fingers is still important in interpreting the nature of the gesture indicated in Table 3.2. For instance, the number of fingers is still important as it can be indicative of more or less force or control. The gesture used by the majority for both trials for each referent is shown in Table 3.3. Although the interpreted gesture did not change for some objects, the physicality of the gesture increased. For example, for referent 1, Ball Valve Front, the average number of fingers used increased from 1 (SD = 0) in trial 1 to 2.14 (SD = 1.04) in trial 2. For many objects the gesture changed from a metaphorical gesture to a physical gesture. For example, referent 5, Gate Valve Side, had a straight line Gesture in trial 1 with average number of fingers = 1.3 (SD=0.57), which is a metaphorical gesture that does not apply to the real world. In trial 2 referent 5 had a finger turn gesture with average number of fingers = 2.45 (SD=1.28) which is a physical gesture one would do in the real world.

The gesture set found in the pilot remained the same, even though we added more complex objects and navigational tasks. We found it interesting that users did not come up with new gesture primitives to accomplish these new tasks that required compound operations (specifically referents, 22, 23, 26, and 27). Instead, we found users used a combination of gesture primitives already defined to accomplish these tasks.

For each referent, the groups of gestures were used to calculate an agreement score A that specifies the degree of participant agreement in the gestures they selected. This method was replicated from Wobbrock et al.'s prior work, see Equation 3.1.

$$A = \frac{\sum\limits_{r \in R} \sum\limits_{P_i \subseteq P_r} \left(\left| \frac{P_i}{P_r} \right| \right)^2}{|R|} \tag{3.1}$$

The agreement scores indicate that there was more user consensus in objects that had a face on view to the area to be manipulated, such as referent 9, Gate Valve Top, in Table 3.3. For example, in trial 2 referent 8, Gate Valve Front, has an obscured view where A = 0.34 , whereas for referent 9, Gate Valve Top, A = 0.42. In addition, objects that were overall difficult to manipulate had a low Agreement score regardless of the angle (e.g., in trial 2 referents 20/21, Key Switch Rotate, $A = 0.23$).

Table 3.3: The data collected for the 27 referents is shown.

			Trial 1					Trial 2				
	Referent	Gesture	Class	A	M%	PM%	P%	Class	A	M%	PM%	P%
1	Ball Valve Front	CL	TP	0.63	64.3	14.3	21.4	P	1.06	14.3	7.1	78.6
2	Ball Valve Side	CL	TP	0.53	50.0	21.4	28.6	E	0.80	14.3	21.4	64.3
3	Ball Valve Top	CL	TP	0.64	57.1	0	42.9	P	0.89	14.3	28.6	57.1
4/5	Ball Valve Rotate Up	SL	S	0.45	92.9	0	7.1	S	0.41	92.9	0	7.1
6/7	Ball Valve Rotate Right	SL	S	0.49	100	0	0	N	0.29	92.9	0	7.1
8	Gate Valve Side	FT	TP	0.34	28.6	21.4	50.0	XP	0.54	7.1	42.9	50.0
9	Gate Valve Top	FT	E	0.42	35.7	7.1	57.1	P	0.65	14.3	28.6	57.1
10/11	Gate Valve Rotate Up	SL	S	0.30	92.9	7.1	0	S	0.26	92.9	0	7.1
12/13	Gate Valve Rotate Right	SL(FT)	N	0.36	100	0	0	N	0.24	92.9	0	7.1
14	Needle Valve Back	FT	P	0.45	35.7	7.1	57.1	P	0.75	7.1	14.3	78.6
15	Needle Valve Side	FT	P	0.50	28.6	7.1	64.3	P	0.74	0	35.7	64.3
16	Needle Valve Top	FT	P	0.55	21.4	7.1	71.4	E	0.64	7.1	7.1	85.7
17	Pull Switch	SL	TP	0.88	57.1	35.7	7.14	XP	0.98	21.4	57.1	21.4
18	Rotary Switch	FT	TP	0.44	42.9	14.3	42.9	P	0.77	7.1	35.7	57.1
19	Key Switch	FT	XP	0.34	28.6	50	21.4	XP	0.53	0	50	50
20/21	Key Switch Rotate Left	SL(FT)	S	0.27	92.9	0	7.1	S	0.23	92.9	0	7.1
22	Locking Clasp	CL	TP	0.44	92.9	0	7.1	P	0.28	42.9	7.1	50
	/ Door	SL			42.9	0	57.1			7.1	7.1	85.7
23	Locking Knife Valve	FT	P	0.63	21.44	0,0	78.6	P	0.63	0	28.6	71.4
		SL			28.6	0	71.4			0	28.6	71.4
24	Wrench	CL	TP	0.63	50	7.1	42.9	E	0.66	0	0	100
25	Pickle Jar	FT	XP	0.33	42.9	28.6	21.4	XP	0.39	14.3	42.9	35.7
26	Butterfly Valve	PI	B	0.36	21.4	0,0	78.6	B	0.63	14.3	7.1	78.6
		CL			35.7	0	64.3			7.1	0	92.9
27	Key and Drawer	FT	TP	0.31	50	21.4	28.6	P	0.54	14.3	57.1	28.6
		SL			28.6	7.1	64.3			7.1	14.3	78.6

Nature Dimensions

The results in Table 3.3 show the data collected for the 27 referents. For each referent the majority gesture chosen in trial 1 and trial 2 are shown in the Gesture column. Since most referents had the same Gesture value for trials 1 and 2, only one is shown and if there was a difference in trial 2 it is in parentheses. The gesture sub-classification is shown in the Class column. The Agreement scores (A), the percentage of Metaphorical (M), Physically based Metaphorical (MP), and Physical (P) gestures used, are also shown respectively. The results in Table 3.3 show, as expected, that the percentage of M gestures were higher during first pass of the experiment, and the percentage of MP or P gestures were higher during the second pass of the experiment. In trial 1 there were 41.5% observed metaphorical (M) gestures, 12.2% physically metaphorical (PM) gestures and 46.3% physical (P) gestures, and in trial 2 there were 10.2% M gestures, 24.8% PM gestures and 65.0% P gestures which shows an increase in physical gestures. However, the results do not show a majority of metaphorical gestures for the first pass of the experiment as we expected. Which indicates that some of the referents (such as those that required compound operations) lent themselves to more physical gestures without the users being explicitly told to do so. Viewing angle and the awkwardness of the object cause the percentage of P to be lower than expected in trial 2 as well, since if there is not a realistic way of physically interacting with the screen the user chose an MP or M gesture.

Number of Fingers and Hands Used

As we assumed, most users used 1 or 2 fingers for the first trial and more than 2 in the second trial. The average number of fingers across all referents was 1.89 in trial 1, and 3.49 in trial 2. The average number of hands in trial 1 was 1.04 and in 1.07 in trial 2. The number of fingers used is significantly greater trial 1 versus trial 2 ($t_{12} = -5.99$, $p < 0.01$).

Gesture Rating

Similar to Wobbrock et al. ‘s previous work, after the participants completed each gesture they rated the goodness and ease of use on two Likert scales. The first Likert scale stated “The gesture I picked is a good match for [task here]” and the second stated “The gesture I picked is easy to perform.” Using Mann-Whitney tests, found that there were no significant differences in ratings between trials of the experiment. Again the ease of use ratings indicate that users did not perceive the physical gestures used in the second pass of the experiment to be more difficult to use than the metaphorical gestures used in the first pass of the experiment. This is interesting because symbolic or metaphorical gestures are considered simplifications and abstractions of real world actions, and therefore considered easier to perform.

There were, however, significant differences between referents. For example, referent 8 (Gate Valve Side) had an average goodness rating of 5.93 (sd = 0.10) and referent 9 (Gate Valve Top) had an average goodness rating of 6.34 (sd = 1.07), which are significantly different ($Z = -3.27$, $p < 0.05$). In addition, the ease of use ratings are significantly different as well ($Z = -2.14$, $p < 0.05$), where referent 8 again has a lower mean ($x = 5.51$, $sd = 1.43$) than referent 9 ($x = 5.99$, $sd = 1.14$). The camera angle (in the previous case) as well as the difficulty of manipulating the referent (referent 19 versus 18 for example) play a part in how the users perceived their gestures‘ goodness and ease.

Interviews

After the participants finished the 4 trials they answered 3 interview questions before the experiment was complete. The interview questions were: (1) “Did you notice that your gestures changed in the second pass of the experiment when I said to treat the objects as real, physical objects?”, (2) “What specifically changed about your gestures in the second pass?”, (3) “Which gestures did you prefer - the ones used in the first pass or those in the second pass, and why?” The majority of users answered yes (11/14) to the first question and more fingers and either pressure or parts of hand to the second question (11/14). One user stated that ”At first I interacted as though it was a phone app, and then I incorporated more fingers.” The other 3 participants were already using physical gestures and using more fingers and hands in the first pass of the experiment.

For the final question, the participants were evenly divided in their preferences, 6/14 (42.9%) indicated they preferred the first pass of the experiment mostly because they were easier to perform, and the physical gestures could be awkward. The other 6/14 participants indicated they preferred the second pass of the experiment. According to one user ”the gestures were similar to habits I use every day”, and another said ”it’s easier to do things as if you would in real life”. 2/12 (14.3%) said that their preference depends upon the referent and the viewing angle. Although it may seem that physical gestures would correspond to physical objects, we assumed the majority would prefer the metaphorical gestures from the first pass, since there is inherent bias from the use of multi-touch phones and tablets. Thus, we were surprised that user preferences were evenly divided. One user stood out from the others stating that he used the Mac track-pad and because of this he “prefers multiple fingers because it‘s a more intuitive experience to turn a knob the way you would in real life.” In addition, some users stated that it would depend on the object and the application. In particular, one user stated “It would be really cool to have a video game where you had to navigate the world by interacting with the picture like they were real objects. For simple interfaces, though, I would rather have simpler controls.”

Discussion

We classified each gesture into the categories within the Nature dimension (P, PM, or M) as well as the sub-categories shown in Table 3.1. All gestures were observed as either in the metaphorical category or in one of the physical or proxy sub-categories. The number of fingers played an important role in distinguishing between metaphorical and physical gestures. For instance, a curved line with one finger to open a ball valve (Figure 3.2-1) was considered a metaphorical gesture since in real life more force would be required, whereas a curved line with 2-5 fingers would be considered a physical gesture because it mimics the real life motion. This is not to say that more fingers always leads to a physical gesture. For instance, opening a gate valve (Figure 3.2-8) with a 3 finger straight line gesture would be considered a metaphorical gesture because that would not translate to real life whereas a multi-point finger turn gesture would.

In trial 1 there were 41.5% observed metaphorical (M) gestures, 12.2% physically metaphorical (PM) gestures and 46.3% physical (P) gestures, and in trial 2 there were 10.2% M gestures, 24.8% PM gestures and 65.0% P gestures which shows an increase in physical gestures. However, the results do not show a majority of metaphorical gestures for the first pass of the experiment as we expected. Which indicates that some of the referents (such as those that required compound operations) lent themselves to more physical gestures without the users being explicitly told to do so. Viewing angle and the awkwardness of the object cause the percentage of P to be lower in trial 2 as well, since if there is not a realistic way of physically interacting with the screen the user chose an MP or M gesture.

Our results contain both similarities and differences with Cohé's study [32]. Both studies determine how users intuitively manipulate 3D objects, where Cohé uses a 3D cube alone and we use more complex objects. Although the objects are different in our study the results are similar and complementary to Cohé's results upon further investigation. Cohé examines three parameters to categorize gestures: form (number of fingers), initial point location (IPL), and finger trajectory. With this information Cohé determines the location of the IPL on the cube (e.g., corner, edge, face, or external to the cube) and defines a relationship between the IPL and the transformation. For instance, for rotating a cube the most common choice was an IPL on a face parallel to the transformation axis (TA) with a trajectory along the transformation direction (TD). From these fine grained observations, Cohé found several overall characteristics of gestures for rotating an object:

- Curved - the trajectory is curved.
- Straight - the trajectory is straight.
- Grab - the user picks a point on the cube surface and then moves the object.
- Push - the user begins their gesture and the cube moves after it has been pushed, or when the finger trajectory intersects an edge orthogonal to the TA.

From these general characteristics emerged four gesture categories that encompass all rotation gestures: CurvedAndPush, StraightAndPush, StraightAndGrab, and CurvedandGrab. Similarly, Cohé came up with the following characteristics for translation gestures: Push, Without object referent, Grab-Lateral, Grab-Pull and Grab-Push. Straight and curved were omitted since there were no curved gestures observed for translation.

Our classification method also examines form, IPL, and finger trajectory to determine what the gesture shape is (listed in Table 3.2), then categorize each touch according to the process in Figure 3.3, and finally come up with an overall determination if the gesture was M, PM, or P in nature. Interestingly, we came up with similar categories for gesture classification. Our grip, edge, and proxy categories correspond to Cohé's grab, push, and without object referent categories respectively. We also added the category for a grab that represents a enclosing or pinching motion that someone would do to pinch or grab an object in the real world.

We believe our data from trial 1 would be similar to Cohé's for referents 4-7, 10-13, 20, and 21 since those referents are rotating the entire object's viewing angle. However, in trial 2 users were told to do gestures as they would in the real world, so the number of fingers used would have increased. In addition, to perform a rotation, users would mimic picking up an object and rotating it which doesn't fit into Cohé's categories for rotations. For the remaining referents for trial 1, the data would also be similar to Cohé's since turning a valve or switch on is rotating or translating a particular part of that component. Again in trial 2, these tasks would use more fingers and pressure and would not fit into Cohé's categories. It is interesting to note that Cohé defined all of the user's gestures as physical gestures. Whereas by our definition, all of the gestures performed by users would be metaphorical since they would not perform the desired action in the real world.

In our study users intuitively used gestures similar to Cohé's to manipulate real world objects, thus verifying Cohé's work. However, when asked to treat the referents as real world objects users used more physical gestures. In addition, when performing more physical gestures users consistently agreed they were no more difficult to perform than metaphorical gestures. This is ideal for future uses of physical gestures in simulation or training environments. Users also were evenly divided on their preference for metaphorical gestures or physical gestures. It is also interesting that the viewing angle affected the perceived difficulty of a gesture to perform and is something to keep in mind when designing these systems.

Conclusion

We have presented a user study that explores what gestures users choose to interact with 3D objects that have rotational, tightening, and switching components. We have also described a procedure for classifying gestures as metaphorical, physical, or physically metaphorical. Our results indicate that due to biases from previous multi-touch experience, the majority of participants intuitively try to interact with 3D objects using 1-2 finger gestures in a primarily metaphorical way. However, once prompted to use gestures as if manipulating physical, real-life objects, the users increased the number of fingers, hands and pressure used, and used more gestures that were physical in nature. The participants also found these physical gestures just as easy to perform as metaphorical gestures. Designers should take into consideration that users intuitively use the gestures they are most familiar with, so if they would like to elicit physical gestures there needs to be guidance in doing so. Other than intent, many other factors play a role in the gestures chosen, among them form factor and the perspective view of the referent. In addition to the gestures, there is also the intersection and interaction with the object to take into consideration when interpreting the gesture, as defined in the Physical and Proxy Nature sub-categories presented.

In this study we observed that many users increased pressure as a way to signify increasing force or depth with physical gestures. In the next chapter we explore two different techniques for estimating pressure on capacitive touchscreens within two different scenarios.

CHAPTER 4: MULTITOUCH PRESSURE SIMULATION IN 3D ENVIRONMENTS

Introduction

In Chapter 3, we found that many users increased pressure in order to make their gesture more closely mimic real world interactions. We observed that users increased pressure to convey applying force and depth movement. However, the most prevalent touch surfaces use capacitive sensing, which rely on the electrical properties of the human body to detect touch, and thus do not sense pressure. Therefore in order to implement pressure as input, we had several questions to answer. First, could we instead use changes in contact size (i.e., the surface area of the finger which comes into contact with the input surface) to estimate pressure? Second, since users contact sizes vary greatly, would calibration be beneficial? And thirdly, does the pressure estimation technique perform well for different types of tasks such as applying force and depth movement?

In order to answer these questions we conducted a 2 x 2 within-subjects experiment of 20 participants examining two different pressure simulation techniques (calibrated and comparative) with two different 3D tasks that require varying levels of sensorimotor skills (gross and fine). Specifically, we use variations in finger contact size (i.e., the surface area of the finger which comes into contact with the input surface) as a way to simulate pressure and translate it into meaningful 3D interactions. Since varying contact size is very similar to varying pressure and acts as a suitable metaphor, we use the term pressure "simulation". Since there are no pressure sensors on capacitive touch screens, we are instead interpreting changes in contact size by varying finger tilt angles, where larger contact size corresponds to heavier "pressure".

Our goal was to determine users perceptions (i.e., ease-of-use, gesture fit, and perceived efficiency) and actual performance (i.e., total completion time). To the best of our knowledge, no previous studies have examined using pressure or contact size for multi-touch object manipulation. In addition, no previous studies have compared calibrated or comparative pressure estimation techniques. Although we expected the calibrated pressure estimation technique to outperform the comparative technique, we found that our initial hypotheses were only partially supported. For the ball and hoops task (gross motor skill), the calibrated estimation technique was significantly better suited for the task. However, the opposite was true for the stove knob task (fine motor skill); the comparative estimation technique was significantly better than calibrated.

Figure 4.1: Our experiment apparatus included a 55-inch Perceptive Pixel display raised to standing height and tilted upwards by 30 degrees.

Study Design

Perceptions and Performance Metrics

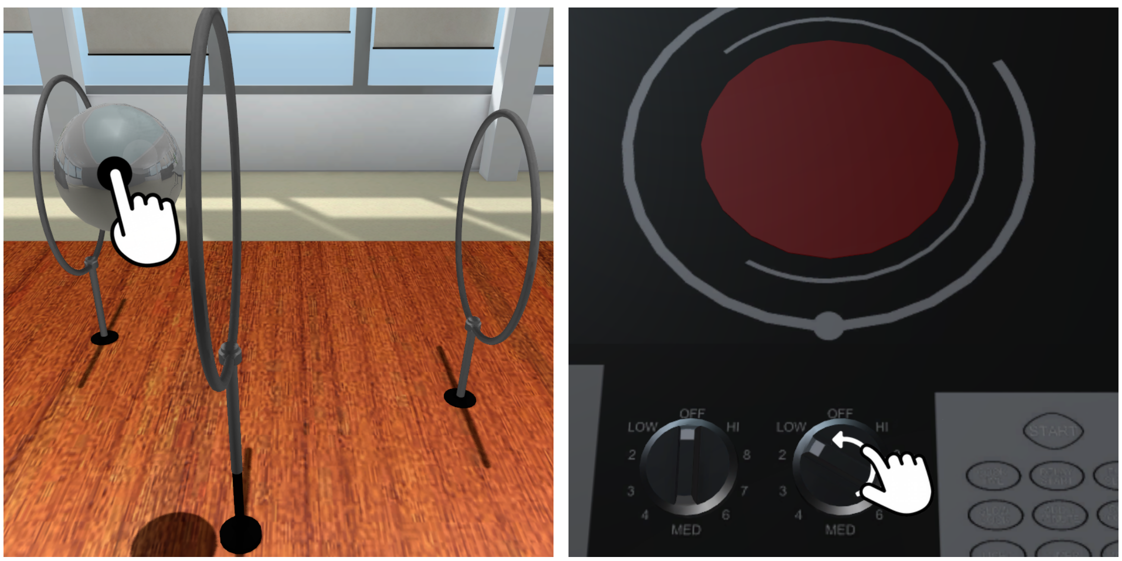
The dependent variables for all conditions within our experimental design included: 1) three perceived measures based on user ratings of ease-of-use, goodness of gesture fit, and perceived efficiency to complete the task, and 2) an objective measure of Task Completion Time (TCT). For the perceived measures, after each task participants were asked to answer the following questions using a 7-point Likert scale, based on previous work [30]:

- How easy was it to perform the gesture?
- The gesture I used was a good match for the task.
- I quickly completed the task.

TCT was measured as seconds to complete the task trial. The trial started when the user pressed a "Begin" button and automatically ended when the system detected the user had completed the task.

Motor Skills

The goal of 3D interaction is to map user input to mimic actual human motion in a natural, intuitive, and meaningful way. In order to do this, we need to have a deep understanding of the fundamentals of human motion. Sensorimotor skills involve the process of receiving sensory input and responding through motor (physical) output. Motor skills are coarsely separated into fine and gross categories; gross movements come from large muscle groups such as the shoulder or arm whereas fine motor skills are involved in smaller movements that occur in the wrists, hands and fingers [73, 74]. The use and assessment of both fine and gross motor skills are commonly used in rehabilitation [75] and for skilled training such as for surgeons [73], who often using haptics when applied to 3D virtual interactions [74]. Therefore, we wanted to ensure that our results for the different pressure estimation techniques could readily be generalized to 3D tasks that utilized both fine and gross motor skills. As such, we applied our pressure simulation techniques to two different tasks: (1) a gross motor skills task where users guided a ball through hoops arranged at different depths and (2) a fine motor skills task where users applied pressure while rotating a stove knob (shown in Figures 4.2a and 4.2b).



The gross motor skills task required guiding a ball through 3 hoops by translating while applying pressure for depth control.

The fine motor skills task required rotating a stovetop knob to a certain position by applying pressure while rotating.

Figure 4.2: The gross and fine motor skills tasks.

In both 3D tasks, pressure was being varied with the tilt of the fingertip to estimate pressure. The gross motor skill task required the gross motor skills of the shoulder to control x and y translational position. At the same time, the finger tilt variations controlled depth (z) position. Hard pressure increased the depth position away from the camera, light pressure decreased depth position towards the camera, and neutral pressure maintained depth position. In the ball and hoops task, pressure acts as an alternative to the pinch to zoom method used in Sticky Fingers for depth translation [19].

The fine motor skill task required pushing in a stovetop burner knob by surpassing a pressure threshold and maintaining that pressure while rotating the knob. It required fine motor skills of the thumb, index finger and wrist. We examine pressure to control depth while rotating in the stovetop burner task, where a pressure threshold has to be met in order to push the stovetop knob prior to rotation. Based on the inherent characteristics of gross motor skills, which involve larger movements, versus the finer tuned movements necessary for the fine motor skills task, our first hypothesis serves as a manipulation check of users' perceptions of the two different tasks:

H1 Users will perceive the gross motor task as significantly (a) easier to use, (b) better fit to the gesture, and (c) faster than the fine motor task.

Pressure Estimation Techniques

We explored two different pressure estimation techniques, calibrated pressure and comparative pressure, which both use the contact size of a touch point on a screen. The difference between methods is how they determine the neutral pressure value from which to compare corresponding increases and decreases in pressure. For both methods increases in contact size are interpreted as more simulated pressure within our defined metaphor. More pressure then corresponds to more depth movement into the environment since more force usually moves something away.

Calibrated pressure requires the user to calibrate their light, medium and heavy pressure contact sizes and then calculates an average neutral contact size. Our calibration exercise presented the user with a cube and asked them to press on it 5 times using their index finger (on the center and 4 corners) for light pressure, medium pressure, and then high pressure. Each participant was told that pressure was interpreted as contact size. It was then demonstrated that light pressure meant the tip of the finger, hard pressure meant the full pad of the finger, and medium pressure meant about halfway between the two. The neutral pressure was then calculated as the mean contact size of all of the collected contact sizes. Since capacitive screens start registering touches as they come into contact with the screen, the screen will register a few very light touches before the intended interaction begins. During our pilot study we determined that ignoring the first five events was appropriate to only record the intended interaction contact size.

For the calibrated method, the current pressure was calculated as the relative difference in area of the current touch's contact area from the saved neutral contact area:

$$((currentsize - neutralsize) - minimumsize)/neutralsize$$

For example, for a user that has minimum, neutral, and maximum values of 500, 1800, and 4100 their pressure range would be ((500- 1800)-500)/1800, ((4100-1800)-500)/1800 = (-1, 1). Since we take the average of all of their calibrated values the range is approximately (-1, 1). Thus, for increases in contact size, the pressure is positive, and for decreases in contact size the pressure is negative. The positive and negative pressure is useful for controlling bi-directional depth position, or z axis translation, where heavier than neutral pushes the object into the screen (away from the camera) and lighter than neutral pulls the object towards the user (towards the camera).

Comparative pressure was also calculated as the difference from neutral pressure, where the initial touch is assumed to be neutral and the minimum size is unknown:

$$((currentcontactsize - neutralsize))/neutralsize$$

For the same user, assume their initial touch is 500 (which also happens to be their minimum) then their range is (500-500)/500, (4100-500)/500 = (0, 7.2). This range not only eliminates negative pressure, but it skews positive pressure. In our application, this would allow the user to increase pressure at a faster rate and pass the pressure limit they would have had with the calibrated method. Then assume the same user has an initial touch of 4100 (which is their maximum) then their range is (-7.2, 0). Thus, if the user would like to move in a certain direction faster, skewing their neutral value in the opposite direction would be advantageous. However, if the user wants reliable bi-directional movement they would need to start with medium pressure.

Since the calibrated pressure can immediately classify a user's pressure level in the range from low to high instead of only being able to interpret relative increases and decreases as the comparative method does, the following hypotheses reflect our expectation that the calibrated method will be perceived as better by users and outperform the comparative method:

H2 The calibrated pressure estimation method will be perceived as significantly (a) easier to use, (b) better fit to the gesture, and (c) faster than comparative estimation technique.

H3 The Time to Complete (TCT) the tasks for the calibrated pressure estimation technique will be significantly faster than the comparative pressure estimation technique.

Methodology

User Study

We implemented a 2 x 2 repeated measures, within-subjects experimental design. The gross motor skills task required translation while varying pressure in order to guide a ball through hoops (shown in Figure 4.2a). The fine motor skills task required rotation while maintaining pressure to push in and turn a knob (shown in Figure 4.2b).

Participants and Apparatus

We recruited 20 participants (7 female, 13 male) ranging in age from 18 to 29 years (average: 20.9 years). Of the 20 participants, 13 owned a touch screen phone while the other 7 owned both a touch screen phone and tablet. All participants received $10 as compensation for their time. We conducted the experiment on a 55-inch Microsoft Perceptive Pixel display. The display was mounted on a stand so the bottom edge of the display was raised to 3.5 feet, approximately standing height, as shown in Figure 4.1. The display was tilted upwards by 30 degrees since tilted displays have shown to be for comfortable for users [76]. The apparatus also included a camcorder capturing the screen and the participant's arm and hand, and a table for the investigator to observe and take notes. Our application used Windows Touchinput events which return xContact and yContact properties in hundreths of a pixel in physical screen coordinates for both pressure estimation techniques [77]. We developed the 3D environment and user study application in the Unity3D game engine.

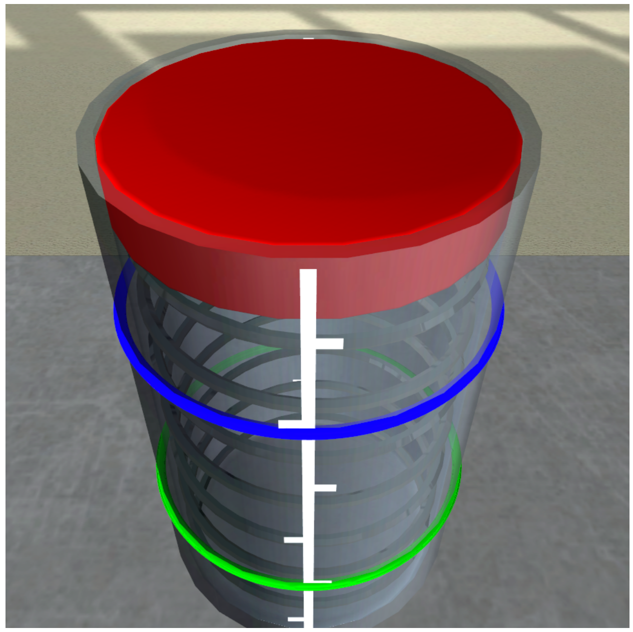


Figure 4.3: User had a practice session where they applied increasing and decreasing pressure levels on a 3D spring-loaded button.

Tasks and Procedure

Before participants began the experiment, the proctor explained to them what we were going to explore during experiment. They explained that we would be translating and rotating different objects on the screen, but to control depth position contact size would be used. Then they explained that contact size is similar to increasing and decreasing pressure, but is actually controlled by the tilt of your finger. The proctor demonstrated varying pressure levels while they were explained. Then, after each participant understood what we were measuring they began the study with the calibration session.

Next, users had a practice session where they applied increasing and decreasing pressure levels on a 3D spring-loaded button, shown in Figure 4.3. This allowed users to experience how finger tilt was interpreted as pressure. Following the calibration and practice sessions, participants were presented with the experimental tasks. The gross and fine motor skills tasks were each completed with both calibrated and comparative pressure estimation techniques, totaling four trials overall. To prevent ordering effects, half of the participants completed the gross motor task first and half completed the fine motor task first. The order of the estimation methods was also balanced within each task.

Before each trial, participants had a practice session to get comfortable with each combined task and estimation method. They were asked to practice the entire task at least twice or until they were comfortable performing the task. Since participants practiced the task multiple times, there were no repeated trials. Once the trial began, participants were instructed to complete the task as quickly as they could since each trial was timed. Following each trial, they were asked the three survey questions on ease-of-use, goodness of gesture fit, and speed. To test our hypotheses, we conducted repeated measures ANOVAs to assess both the main effects and interaction effects of our treatment conditions for all of the perceived dependent measures. An independent t-test was used to assess the differences in the actual TCTs since the two tasks were independent of one another and were not necessarily designed to take the equivalent time to complete.

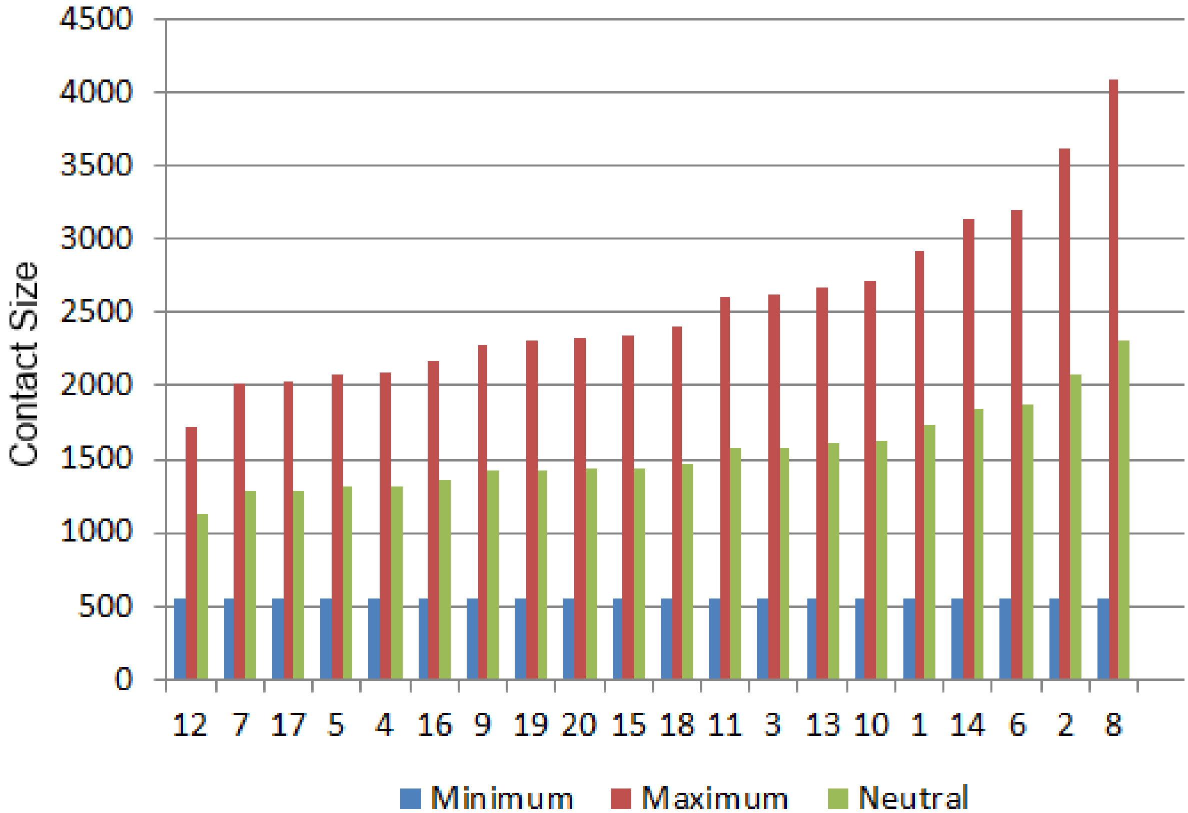


Figure 4.4: Results of each user's calibration session: the minimum and maximum contact sizes and the calculated calibrated neutral pressure values for all users, ranked from low to high in hundreths of a pixel.

Results

Calibrated Neutral Pressure

We found a wide spread of calibration values from our participants, demonstrating the utility in calibration. The minimum pressure values, maximum pressure values and the calculated calibrated neutral pressure values for all users is shown in Figure 4.4, ranked from low to high. Interestingly, each user had the same minimum pressure value indicating a limitation in the sensor for exact sensing. The overall average neutral pressure is 1555.94 (stdev = 293.00). We can see from these values that there is a large difference in pressure sizes from the minimum calculated neutral value (1130.45) and the maximum calculated neutral value (2316.269), the maximum value is almost 2 times as large. In addition, the maximum contact size for the user with the smallest overall contact sizes (1716.39) is less than the calculated neutral value for the user with the largest overall contact sizes (2316.27). The user with smallest overall values would have a hard time using the system calibrated for the user with the largest values or even the median values.

Table 4.1: The averages and (standard deviations) of our dependent measures: ease-of-use, gesture fit, perceived efficiency, and task completion time (TCT).

	Gross Calibrated	Gross Comparative	Fine Calibrated	Fine Comparative
Easiness	6.30 (0.86)	5.30 (1.17)	4.20 (1.85)	5.55 (1.39)
Goodness	6.45 (1.09)	5.95 (1.36)	5.30 (1.78)	6.00 (1.08)
Speed	6.30 (0.80)	5.15 (1.31)	5.15 (1.31)	5.85 (0.99)
TCT	34.27 (23.00)	50.61 (35.39)	29.35 (27.53)	16.09 (13.80)

Hypotheses Testing Results

The averages and standard deviations of our dependent measures are summarized in Table 4.1. In the sections below, we will present the results of our hypotheses testing. Then, we will further interpret our results by examining the interaction effects between our test conditions.

Main Effects of Motor Skills

We found partial support for our Hypothesis 1; the gross motor task was perceived as significantly easier ($F_{1,19}$ = 10.97, p <0.005) and faster ($F_{1,19}$ = 4.54, p <0.05) than the fine motor task. However, the main effect of perceived goodness of gesture fit was not significant ($F_{1,19}$ = 3.34, p = 0.084). This result suggests that users perceived the gesture motions as equally well-suited for their respective tasks.

Main Effects of Pressure Estimation Techniques

Our Hypothesis 2 was also partially supported; users perceive the calibrated estimation technique as significantly faster ($F_{1,19}$ = 4.54, p <0.05) than the comparative estimation technique. However, it was not perceived as significantly easier ($F_{1,19}$ = 0.53, p = 0.487) or better suited to the gesture ($F_{1,19}$ = 0.22, p = .645) than the comparative pressure estimation method. Additionally, we had to reject Hypotheses 3 because the TCT for the tasks using the calibrated estimation technique was significantly slower ($t_{20} = 1.93, p < .05$) than the comparative estimation technique for the fine motor task. Table 4.2 summarizes our results based on our initial hypotheses.

Table 4.2: A summary of results based upon our initial hypotheses.

Hypotheses	Result
H1a: Ease-of-use, Gross >Fine Motor Task	ACCEPT
H1b: Goodness-of-fit, Gross >Fine Motor Task	REJECT
H1c: Perceived Speed, Gross >Fine Motor Task	ACCEPT
H2a: Ease-of-use, Calibrated >Comparative	REJECT
H2b: Goodness-of-fit, Calibrated >Comparative	REJECT
H2c: Perceived Speed, Calibrated >Comparative	ACCEPT
H3: TCT >Comparative	REJECT

Interaction Effects

Because we found only partial support for our first two hypotheses and had to reject our third, we continued our analysis to interpret the possible interaction effects between our test conditions and better understand the totality of our results. We found that our all of our repeated measures ANOVAs were dominated by significant interaction effects for the perceived measures: easiness ($F_{1,19}$ = 16.85, p <0.001), goodness ($F_{1,19}$ = 6.42, p <0.05), and speed ($F_{1,19}$ = 12.93, p <0.002). We also found a significant interaction effect in TCTs.

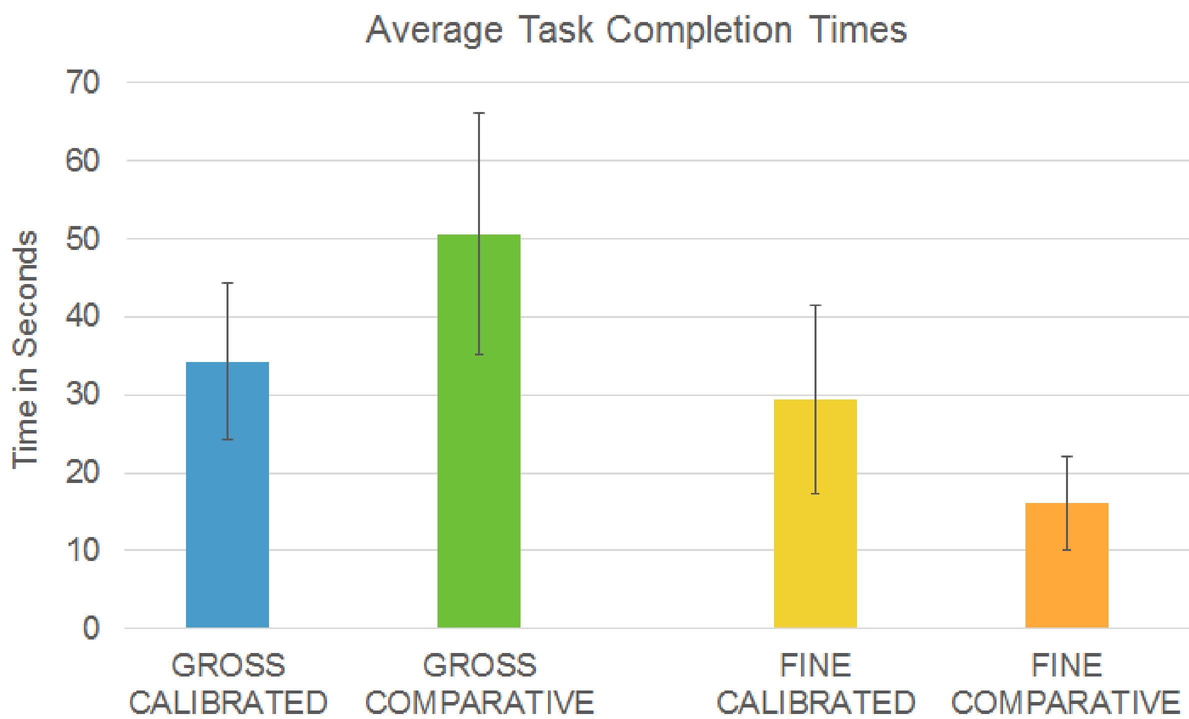


Figure 4.5: The average task completion times (TCT) for each trial of the experiment using either calibrated or comparative pressure in either the gross or fine motor tasks.

As illustrated in the interaction graphs in Figures 4.6a-4.6d, the classic X pattern between our test conditions demonstrates the strong interaction effects between the two pressure estimation techniques and the gross versus fine motor tasks. Across all of the perceived measures, we saw the same patterns. For the gross motor task, users perceived the calibrated estimation technique as significantly easier-to-use, better-fit to the gesture, and faster than the comparative estimation technique. The opposite was true for the fine motor task; the comparative estimation technique outperformed the calibrated. This pattern was also consistent (though inverse since lower TCT is considered better) for the objective measure of TCT between the conditions.

The interaction graphs also show a relatively flatter line for the comparative estimation technique across the perceived measures for two tasks, suggesting that the interaction effect was more pronounced for the calibrated estimation technique. In other words, the perceived variance in performance between gross and fine motor tasks was greater for calibrated estimation technique than the comparative. For TCT, the comparative estimation technique had the largest variance time to complete between gross and fine motor tasks, while the calibrated was relatively stable across tasks.

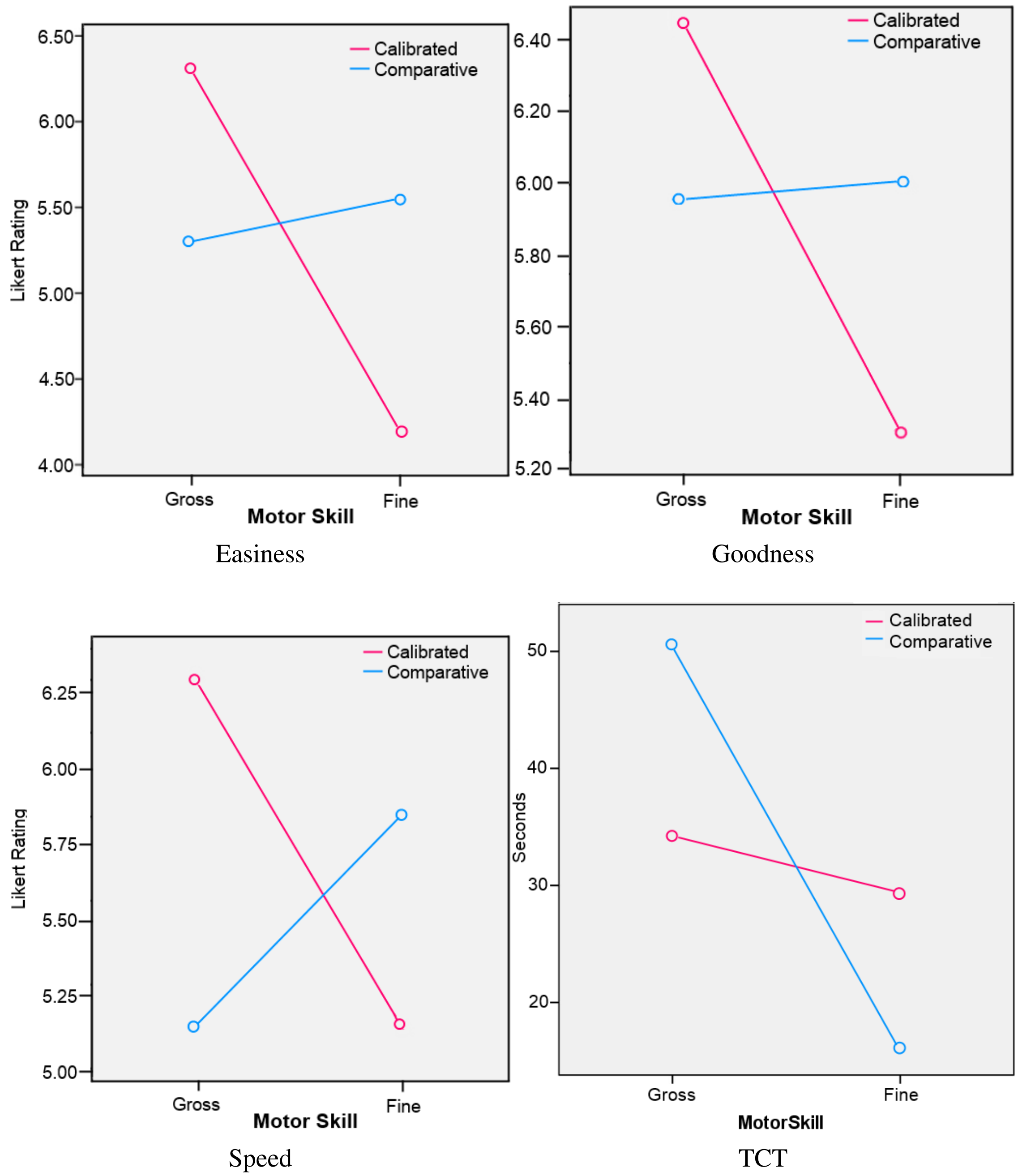


Figure 4.6: The interaction graphs for the two pressure estimation techniques and the gross versus fine motor tasks

Discussion

Interpreting Our Results

While we found mixed support for our initial hypotheses, the results from our study were even more insightful than if we had achieved the outcomes we had originally set forth in our hypotheses. Indeed, we found that *different* pressure estimation techniques perform *significantly* better for different 3D tasks based on the type of motor skill users have to perform. Because of the unanticipated results, we were challenged to reflect on our user study and the lessons we learned from our users. Here, we present some of those insights to help explain our results.

As anticipated, the gross motor task benefited from the calibrated pressure estimation technique for a number of reasons. First, bi-directional control required a predictable neutral position, which users were able to achieve through calibration. Second, the finger's orientation was roughly the same during both the calibration session and the gross motor task. Users were able to leverage a wider range in contact size to complement the larger degrees of freedom encompassed in the arm and shoulder motions of the gross motor task. In contrast, the comparative pressure took the first touch as the neutral position. If the user started with either a light or hard touch, this set their neutral point as either light or hard, which limited them to not being able to go any lighter or harder during the ball and hoops task. This was a limitation of the comparative pressure in a bi-directional task, since it would make it impossible to move the ball towards the camera in the gross motor task. Thus, if the user wanted to bring the ball towards the camera, they would have to let go of the screen and initiate their interaction again with a harder touch and then transition to light pressure. However, in a unidirectional task where a pressure threshold needs to be met, as in the fine motor task, it would make it easier to apply positive pressure past this threshold if the initial touch was very light.

Yet, we discovered the unexpected result that comparative pressure estimation was significantly better than calibrated pressure estimation for the fine motor task. We believe that the fine motor task benefited from comparative pressure for a number of reasons. From previous pilot testing, we determined the pressure threshold before the knob would depress to be positive 0.5 pressure units. Some users' fingers did not flex in a way for them to be able to surpass their calibrated threshold without putting their fingers into an unnatural or uncomfortable position. Also the orientation of the user's fingers changed as they were rotating which made it difficult to maintain pressure past the threshold value during rotation. Whereas with the comparative pressure (as discussed in Section Pressure Estimation Techniques) the user can reach a higher pressure value simply by starting with a very light pressure. Users learned that starting with light pressure for the comparative method was effective during the practice session. Users were then able to understand how to make the comparative method work for them during the actual task trial. If the users were unable to get the neutral position right for the rotation task on their first try, the comparative estimation technique allowed them to readjust each time they touched the screen. We believe that because the task required the use of fine motor skills, the ability to readjust with trial and error was an invaluable benefit to users.

Implications for Design

There are five important design implications that come out of this work. First, different estimation techniques are more optimal for different tasks based on whether the task is unidirectional or bidirectional. Calibration is more necessary for bidirectional tasks, whereas the comparative method performs well for unidirectional tasks. It is more important to bidirectional movement to determine where the neutral, low and high pressure values are, making calibration more necessary.

The second implication is that the user needs to know more about the implementation of the pressure estimation when calibration is not used. Unidirectional movement can get by better without calibration, but only if the user understands that applied pressure is interpreted as increases in contact size and that applying steady pressure outright will not work. Thirdly, if calibration or pressure estimation customization is not possible in a complex task that requires both unidirectional and bidirectional movement, then it would be best to use the comparative estimation technique. The comparative pressure estimation technique performed more consistently across both tasks, though sub-optimally for the gross motor skill task.

Finally, improvements to the calibration process could be made to make it applicable to more tasks. The same large pressure variation range available to a user when calibrating with their finger straight up and down, is not available in tasks that (1) use the thumb, (2) have a small area, and (3) require rotation. In addition, as a user's touch moves away from the center of the screen the finger pad's orientation is going to change slightly, affecting the contact size. Ideally, calibration should obtain finger pad representations at the outer 4 corners of the screen. Then if the calibration method was also cognizant of the user's position relative to the screen, it could make assumptions about the finger pad's orientation.

Conclusion

We evaluated the two pressure estimation techniques, calibrated and comparative, and their applications to the different motor skill tasks in a 2x2 within-subjects experiment. Although we expected the calibrated pressure estimation technique to outperform the comparative technique, we found that our initial hypotheses were only partially supported. Instead, we uncovered an insightful and unanticipated finding: different pressure estimation techniques are significantly better for different tasks based on the type of motor skills being performed.

We have now defined design guidelines for physical gestures and how to implement pressure as part of these physical gestures. In the next chapter we explore whether using these gestures in training applications increases knowledge transfer.

CHAPTER 5: MULTITOUCH INTERACTION AND PROCEDURAL TRAINING

Introduction

Once we had developed gesture design guidelines for realistic 3D multi-touch interaction in Chapters 3 and 4, we were ready to evaluate the training benefits of a training system that provides realistic affordances through these types of gestures. Specifically, we address the following research question: How does realistic multi-touch interaction affect the learning of a procedural assembly task? To address this question, this chapter presents a study evaluating the knowledge transfer acquired with multi-touch interaction technology compared to standard training methods. We compared multi-touch interaction to standard 2D mouse interaction and to actual physical training.

Figure 5.1: The motorized bicycle model in the virtual environment with the tools and parts on the table or in the tool bar for easy access.

Our study focuses on learning a repair and assembly procedure and the transfer of knowledge to performing the task on the real world apparatus. We chose a mechanical repair task since they are complex tasks which involve the knowledge of specific procedures, the location of parts, the interactions between parts, and the use of tools. In order to focus on the installation gestures themselves, we limited the object manipulation and constrained the viewpoint. We demonstrate realistic physical gestures and interaction techniques that aid in learning by creating realistic constraints that would be applicable to a variety of procedural tasks, such as gravity, using two hands, and applying pressure.

Figure 5.2: The physical motorized bicycle apparatus (5.2a) with the accompanying tools and parts (5.2b).

Training Systems and Task

The selected experimental task consisted of assembling and testing part of a 2-stroke bicycle-mounted engine (shown in Figure 5.2). The apparatus and task were selected based upon a number of constraints: (1) the task should use tools, parts and gestures that would be applicable to industrial, defense, or medical tasks, (2) the apparatus should be able to fit into our lab, (3) the task should be safe for all participants to perform, (4) the task should be significantly complex and non-obvious that a layman could not complete it without prior training. These constraints led us to repairing a 2-stroke bicycle engine where participants could assemble, test and adjust the clutch and other parts.

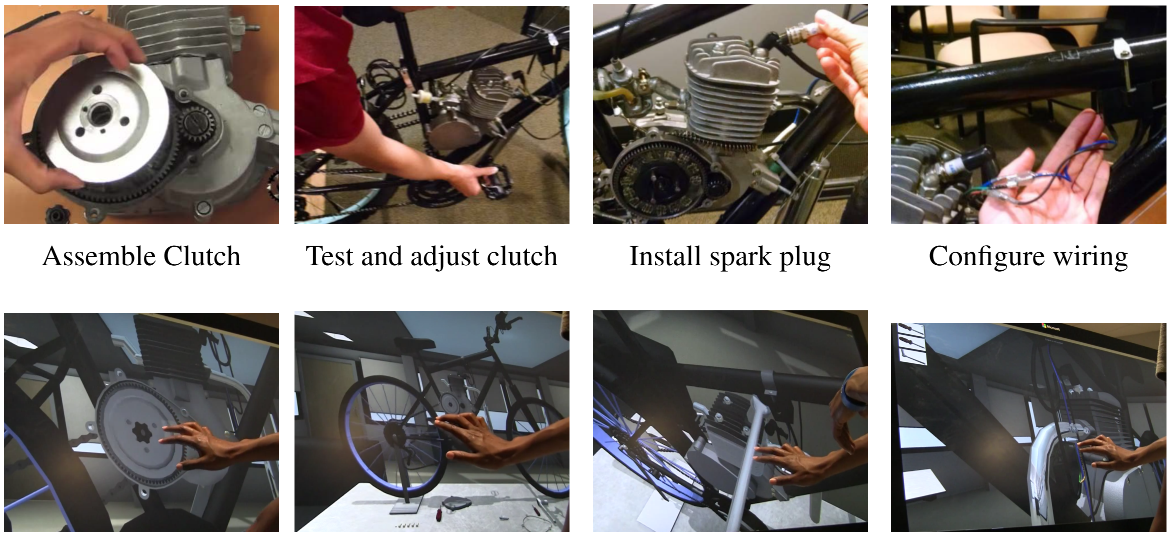


Figure 5.3: The four different sub-tasks (5.3a - 5.3d) shown on the physical bicycle apparatus (top), and in the virtual environment (bottom).

The overall task consisted of 22 steps grouped into 4 sub-tasks (shown in Figure 5.3): (1) clutch assembly, (2) clutch functionality testing and adjustments, (3) spark plug installation, and (4) wire configuration. Some examples of the operations participants had to perform are: tightening the clutch wheel nut to adjust the clutch plate, tightening screws with a screwdriver, installing the spark plug with a socket wrench, and plugging in wires. The task also required verifying the correct clutch plate tightness, where the engine needed to be disengaged with the clutch in and engaged with the clutch out. If either of these conditions are not true the clutch plate should be looser or tighter respectively. In total, participants had to manipulate roughly 25 parts, and use three different types of tools (shown in Figure 5.2b).

Motorized Bicycle

The physical apparatus was a motorized bicycle mounted on a small stand to allow pedaling the bike freely in place for testing purposes. The participants that trained on the physical bicycle had the advantage of experiencing the affordances provided by each of the tools and parts involved in the task. For instance, understanding the correct tightness of the clutch plate is an important step in the process. This can be experienced by tightening the nut about 80% of the way and making sure the clutch plate still has about 1/8" of movement. As described above, correct clutch plate tightness can be verified by pedaling the bike with both the clutch in and the clutch out. The user can verify the engine is disengaged visually by looking at the gears and observing the relative easiness of pedaling the bike. With the engine engaged, the user can feel that the bike is more difficult to pedal and hear the engine making a chugging sound. The ability to examine the parts and bike apparatus from any viewpoint or zoom level is also an advantage the physical model provides. As described above, the physical bicycle provides feedback in the form of opposing forces and aural and visual cues.

Virtual Model and Multi-touch Interaction

The virtual environment contained a detailed, working model of the bicycle, engine, components and tools involved (shown in Figure 5.1). A tool bar was also provided so that if a user was zoomed into a certain area of the bike they wouldn't have to zoom out to select a part from the table. Users could select the tool or part needed from the tool bar and it would hover next to the button. Each corresponding step in the task was replicated in the virtual model and required interaction. We ensured both the multi-touch and mouse interaction required the same number of steps to complete the task.

Previous work on user-defined multi-touch gestures for 3D objects influenced our multi-touch interaction model [2]. Based on this work, all objects, except for small screws, required two or more fingers to manipulate in order to make the interaction more realistic. The gestures incorporated were *double tap*, *translate*, *rotate*, *pressure*, *grasp*, *grip* and *hold*. *Double tap* zoomed in and out of the 5 main viewpoints of the bike. *Translate* was used to move parts and tools around. *Rotate* was used on the clutch plate to align it, and on the screwdrivers. *Pressure* was used to apply pressure when installing the spring and to attach the socket wrench to the spark plug. Pressure information was interpreted based off touch point size provided by the display device. *Grasp* was used on the clutch lever; multiple fingers had to push down on the lever while the thumb stayed on the handle grip. *Grip* was used on the socket wrench handle to grip it and rotate it around. *Hold* was used in 2-handed gestures where gravity or other forces would take over. For instance, if the user let go of the clutch plate before securing the clutch wheel nut with the other hand the plate would detach due to the force of the spring and fall to the table.

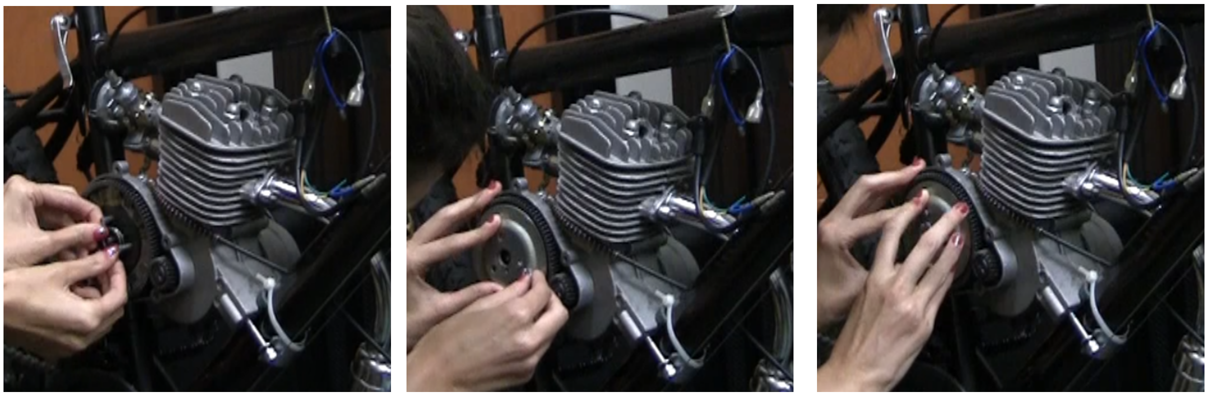
Where possible we aimed to replicate the affordances provided by the real model in the virtual model and multi-touch gestures. For example, the steps in the real task and the corresponding steps replicated in the virtual model follows and are shown in Figure 5.4. In the real model installing the clutch plate requires: (1) pressing and rotating the spring over the center bolt until it stays, (2) attaching the clutch plate by rotating it so it aligns with the three pegs, and then (3) holding and pressing the clutch plate over the spring while screwing on the clutch wheel nut. The corresponding multi-touch interaction required was identical, where the gestures required were: (1) translate and pressure rotate the spring, (2) translate, rotate and hold the clutch plate, (3) 2-handed interaction where one hand holds the clutch plate while the other hand translates and rotates the clutch wheel nut.

In the above example, even though the main gesture motions were identical, some affordances were not replicated due to the constraints of the interface. For instance, in the physical model you have to rotate and press on the spring until it clicks onto the center bolt, but sometimes this is tricky and can be hard understand how it feels when the spring attaches. Whereas in the multi-touch model, the user only needs to press and rotate 180 degrees and it will automatically connect. On the other hand, rotating and aligning the plate was more difficult in the multi-touch model since the user is required to exactly match up the plate with the three pegs and then press. In the physical model the same precision is not required. For step (3), the physical model's spring pushed back against the clutch plate reminding users to apply pressure to keep it in place. In the multi-touch environment, users had no such feedback and may forget that pressure is required to prevent the plate from detaching until the nut is secure.

We emphasized the model fidelity and interaction details even further to articulate any sensorimotor skills that wouldn't be found in the user manual alone. As for interaction details, the number of rotations required to rotate the clutch wheel nut, the screws with the screwdrivers, and the spark plug with socket wrench were the same for the actual bicycle apparatus and the virtual model. As for model fidelity, in order to get the clutch plate to the correct tightness, participants were directed to tighten the clutch wheel nut approximately 80% of the way and then perform the clutch testing procedure to verify it was working correctly. In the virtual model, 80% tightness also mapped to 1/8" of clutch plate movement so users could press on the clutch plate and see how much it depressed. We also indicated that it was harder to pedal the bicycle with the engine engaged by requiring 2 fingers and pressure to pedal when engaged and 1 finger versus no pressure threshold when disengaged.

Virtual Model and Mouse Interaction

The mouse interaction was intended as a control training group, providing only basic click and drag interaction. We implemented a simplified mouse condition over an optimized mouse condition to demonstrate that it's not only the number of steps the user completes, but the level of interaction that affects learning. We ensured both the multi-touch and mouse interaction required the same number of steps. With the mouse the user would drag the part or tool to the desired position and then when close enough it would snap to place. To perform rotations the user would click again and the part or tool would animate as it rotated into place. If a mistake was made, the user could click to uninstall and then move the part away. Since rotations only required one click to animate they could be completed much quicker than in the multi-touch interface where a user would have to perform multiple rotation gestures to install the part completely.



Pressing and rotating the spring over the center bolt until it clicks into place

Pressing and aligning the clutch plate over the spring and three pegs

Tightening the clutch wheel nut over the clutch plate to the appropriate tightness

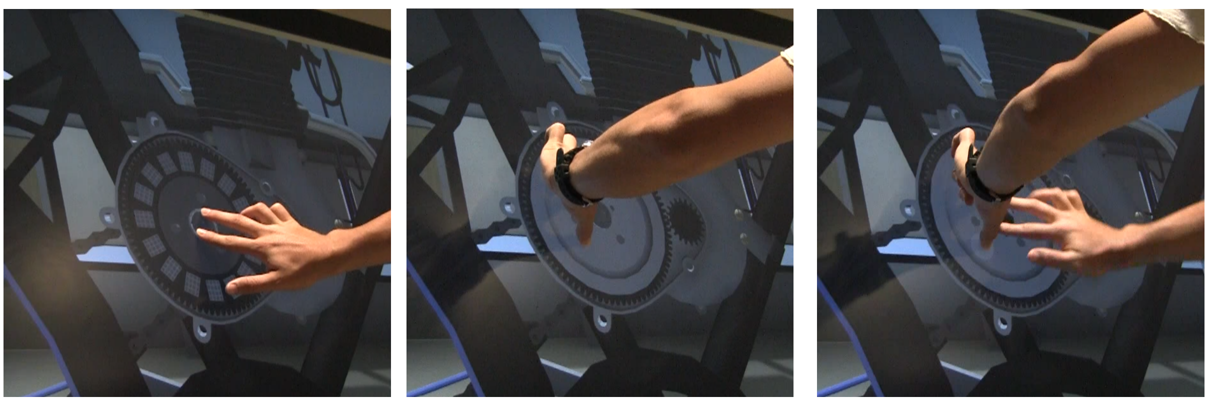


Figure 5.4: The three steps in assembling the clutch (5.4a - 5.4c) shown on the actual bicycle apparatus (top), and in the virtual environment (bottom).

User Study

We conducted a user study to explore the effectiveness of gesture-based multi-touch training. The study examined three different methods of training how to repair a 2-stroke bicycle engine. The three methods were using the actual equipment, using a virtual model with gesture based interaction, and using a virtual model with a mouse. All groups trained with the same paper manual and returned three days later for a post-training test. During the post-test participants were asked to perform the task on the actual equipment as quickly as possible from memory and were graded on correct steps and timed. Our hypotheses were:

H1 The participants trained using the gesture interface will score as well on the post-test as the participants trained with the actual equipment.

H2 The participants trained using the gesture interface will score better on the post-test than the participants trained using the mouse interface.

H3 Due to discrepancies between manipulating the actual components and the virtual model, the participants trained using the gesture interface will perform the post-test slower than the participants trained with the actual equipment.

H4 Since the gesture participants may have maintained some muscle memory of the task, participants trained using the gesture interface will perform the post-test faster than the participants trained using the mouse interface.

H5 The training time for the mouse participants will be faster than the bicycle and multi-touch groups since they don't have to go through the longer physical gestures.

Participants and Apparatus

There were 36 participants (13 female, 23 male) aged 18 to 37 ($\bar{x}$ = 23.6,σ = 4.58) randomly distributed into the three groups. The participants were recruited from a University setting from a wide variety of majors including Speech Pathology, Biology, Marketing, Business, Information Technology, and more. Participants rated their mechanical experience on a Likert scale from 1 to 7, ($\bar{x}$ = 3.06, σ = 2.04), and none of the participants had worked on a similar engine previously.

Training was either conducted on the computer via a 55-inch Microsoft Perceptive Pixel multi-touch screen, or on the actual bicycle and engine. Post-tests were performed on the physical bicycle.

Study Design

The experiment follows a between-subjects design with 36 participants randomly divided into 3 experimental groups (each participant was assigned to a group in alternating order). Two groups trained in a virtual environment, one using multi-touch gestures or the other using mouse input. The third group was trained on the actual bicycle.

Participants were timed during training and testing and scored on the test. Participants were also asked to rate the following statements on a Likert scale from 1 (strongly disagree) to 7 (strongly agree), both post-training and post-testing:

Post-Training questions:

- Q1 I feel prepared to complete the repair task after completing the training
- Q2 I thoroughly understand the concepts that I learned during the training

Post-Testing questions:

- Q1 I was sufficiently prepared to complete the repair task upon arrival today
- Q2 I thoroughly understood the concepts that I learned during the training

Procedure

Testing occurred exactly 3 days after training to avoid short-term memory effects. For training, all groups followed the same protocol. First, participants were presented with the training apparatus (whether it be the computer or bicycle) and the paper manual. Second, they were familiarized either with the tools or the interaction model. Second, participants were asked to follow along with the paper manual as they conducted the task. They were told they could complete the training as many times as they wanted, so long as they could learn the task and describe the process back to the proctor correctly.

Table 5.1: Steps in the procedure that participants were scored on during the post-test.

	Steps		Steps
1	Clutch lever	11	Testing - clutch down to adjust
2	Clutch button	12	Flathead screw
3	Spring	13	Flathead screwdriver
4	Clutch plate	14	Gasket
5	Clutch wheel nut	15	Gear case cover
6	Begin testing	16	5 screws
7	Testing - pedal bicycle	17	Phillips Screwdriver
8	Testing - clutch down, engine disengaged	18	Spark plug
9	Testing - clutch up, engine engaged	19	Socket wrench
10	Testing - diagnose correct adjustment	20	CDI cap
		21	Connected wiring
		22	Wiring correct

Three days later, participants returned to perform the task on the real bicycle engine. They were asked to perform the task correctly from memory without the user manual. They were scored on the steps shown in Table 5.1. One point was given for each step completed correctly and performed in the correct order, for a total of 22 points. If they skipped a step then later remembered and backtracked to complete it, they received a 1/2 point.

Results

Training

Training Time

The participants could go through the training as many times as they wanted to learn the task and describe the process back to the proctor, the average total training time by group is shown in Figure 5.5. We found that all of the participants in the Bike group only went through the process once, whereas all of the participants in the Multi-touch and Mouse groups went through the training twice. The bike trainees would have to wait on the proctor to dismantle the training model which may have discouraged them from going through it twice. Once the virtual users had learned the interface, they were able to quickly go through the training a second time. The virtual medium affords quickly repeating the training scenario as compared to the physical training apparatus.

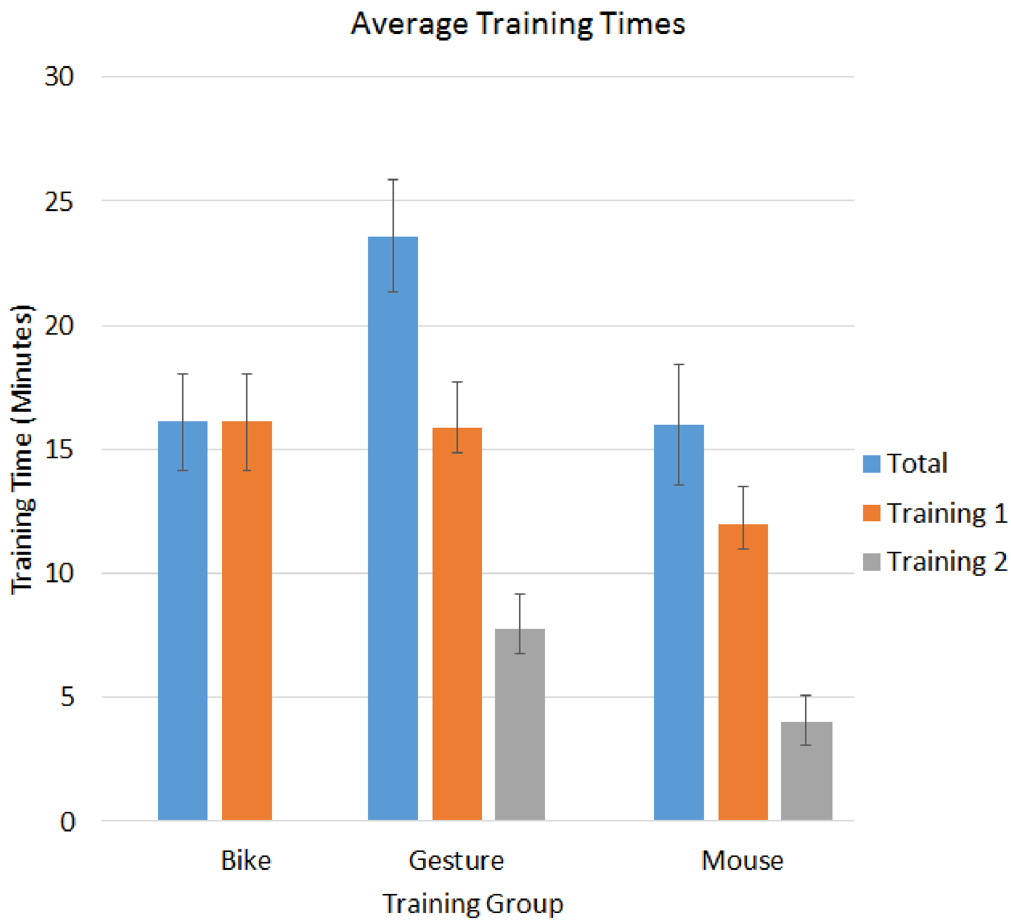


Figure 5.5: Average training times by training group. Participants could go through the training multiple times, however the participants in the physical bike group all opted to only perform the training once.

An independent samples t-test did indicate the multi-touch group spent significantly more time than the bike group ($t_{22} = -2.42, p<0.025$) and the mouse group ($t_{22} = -2.96, p<0.007$) on the first training session. However, the bike and mouse groups training times were not significantly different ($t_{22} = 0.623, p = 0.540$). This result demonstrates that multi-touch participants had to perform detailed, realistic interactions and may have also spent extra time adjusting to new gestures.

Questionnaire

We performed a Kruskal-Wallis test (results are shown in Figure 5.6) which showed that there were no significant differences between the different groups for their responses to each of the 4 Likert questions (presented in the Participants and Apparatus section). These results indicate that participants from all groups felt that they were prepared after the training, and still felt that they were prepared after performing the actual post-test.

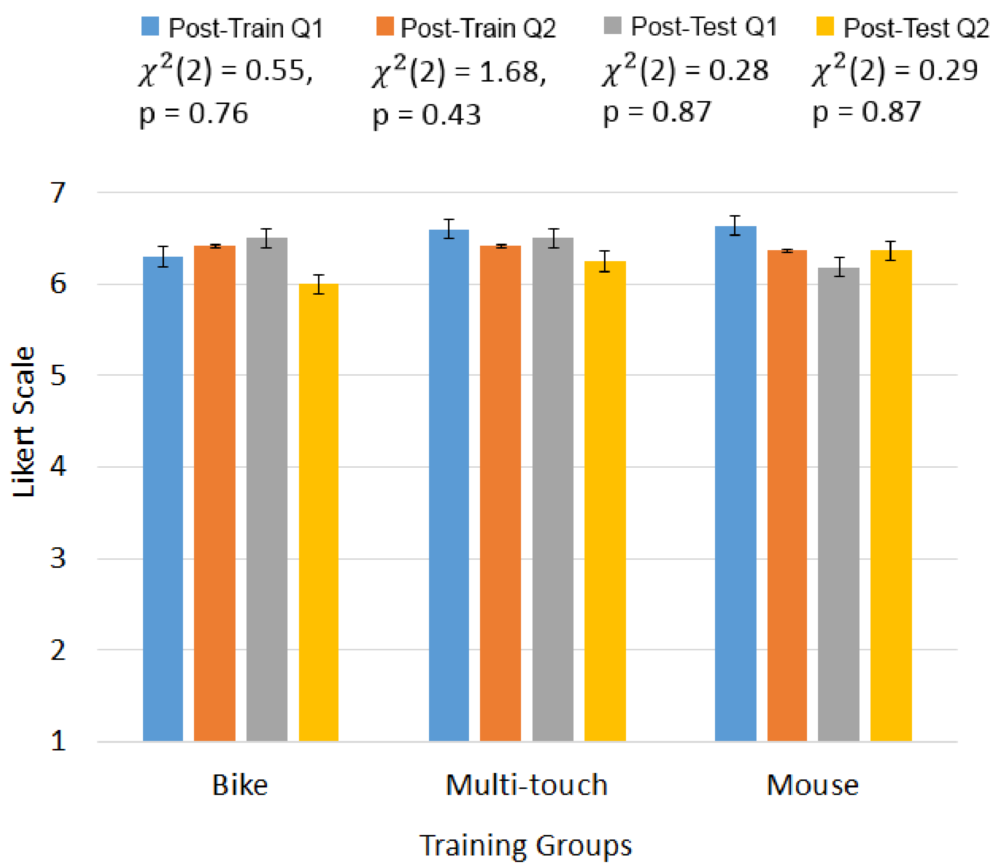


Figure 5.6: Average user Likert ratings on the preparedness and understanding questions asked after training and after testing.

Testing

Testing Scores

When the participants returned for testing, they were asked to perform the task on the real bicycle and engine from memory. They were scored on number of steps completed correctly and timed. The average scores by group are shown in Figure 5.7. The scores were normally distributed for the Bike, Gesture, and Mouse groups as assessed by Shapiro-Wilk's test ($p > .05$). A one-way between-subjects analysis of variance (ANOVA) was run to compare the effect of the training method on the testing score. The main effect that training method had on the dependent variable, testing score, was found to be significant ($F_{2,33}$ = 5.36, $p < 0.010$). Independent samples t-tests indicated the gesture group scored significantly better than the mouse group $(t_{22} = 2.50, p<0.025)$ and the bike group also scored significantly better than the mouse group $(t_{22} = 2.58, p<0.020)$. However, the bike and gesture groups scores were not significantly different $(t_{22} = 0.240, p = 0.812)$. This result demonstrates that participants that trained with Multi-touch performed as well as those trained on the actual apparatus and better than those trained on the Mouse.

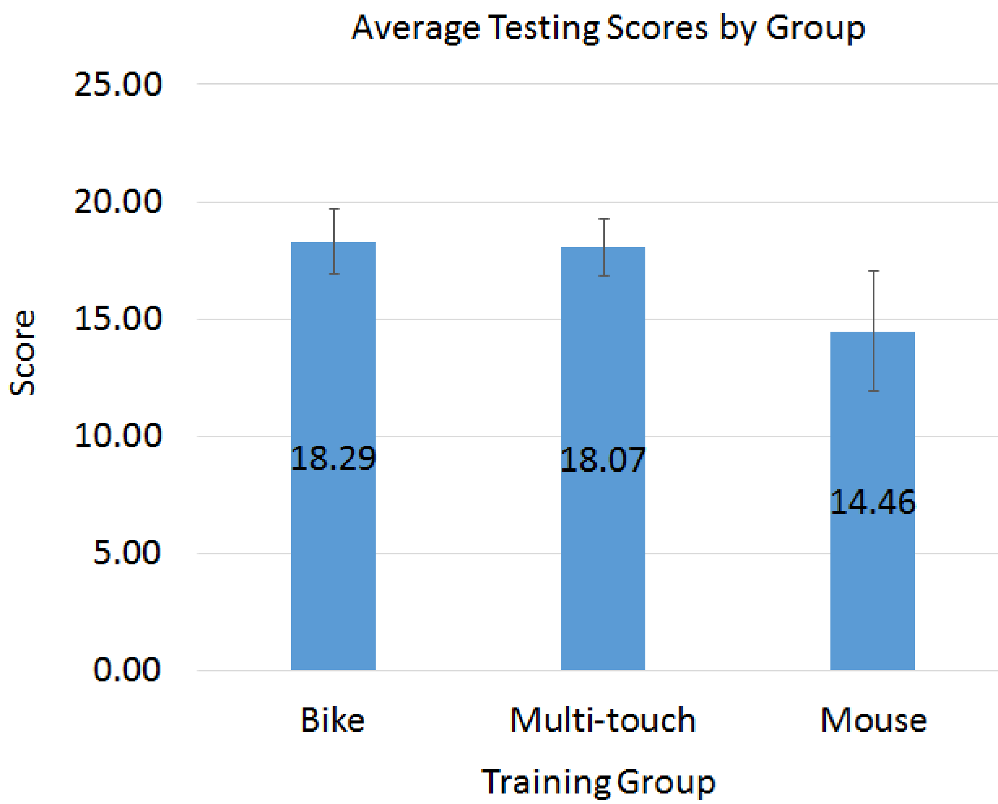


Figure 5.7: Average testing scores by training group.

Testing Errors

Figure 5.8 shows each step of the testing process and how many users got the step correct, separated by testing groups. The errors occurred when a user forgot or skipped a step. If a user remembered to do a step, there was never a case where the user didn't know how to operate the tool or part in that step, but instead users completely forgot the process altogether.

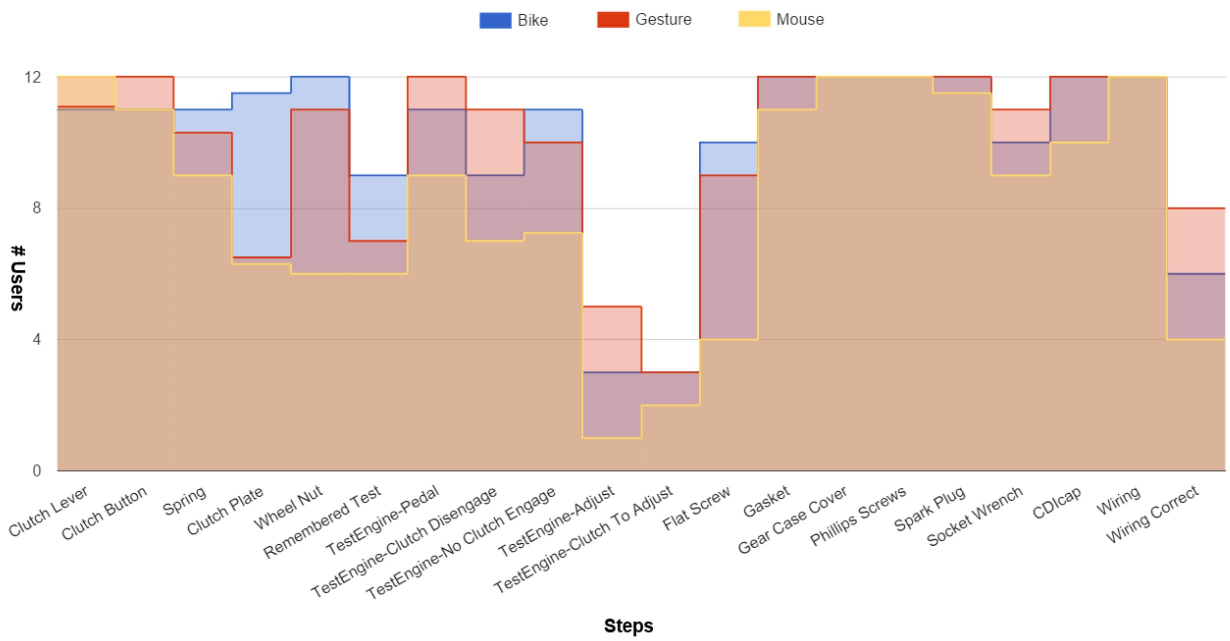


Figure 5.8: The number of users that completed each testing step correctly for each training group; the bike group in blue, the gesture group in red, and the mouse group in yellow. This graph also demonstrates on which steps users made the most errors by showing where the deficits are. For example, for the flat screw step only 4 of the mouse users performed the step correctly, whereas 9 gesture users and 10 bike users performed it correctly.

Testing Scores

The testing times were also measured (shown in 5.9) and found to be normally distributed for the Bike, Gesture, and Mouse groups, as assessed by Shapiro-Wilk's test ($p > .05$). A one-way between-subjects analysis of variance (ANOVA) was run to compare the effect of the training method on the testing score. The main effect that training method had on the dependent variable, testing score, was not significant ($F_{2,33}$ = 1.64, p = 0.209).

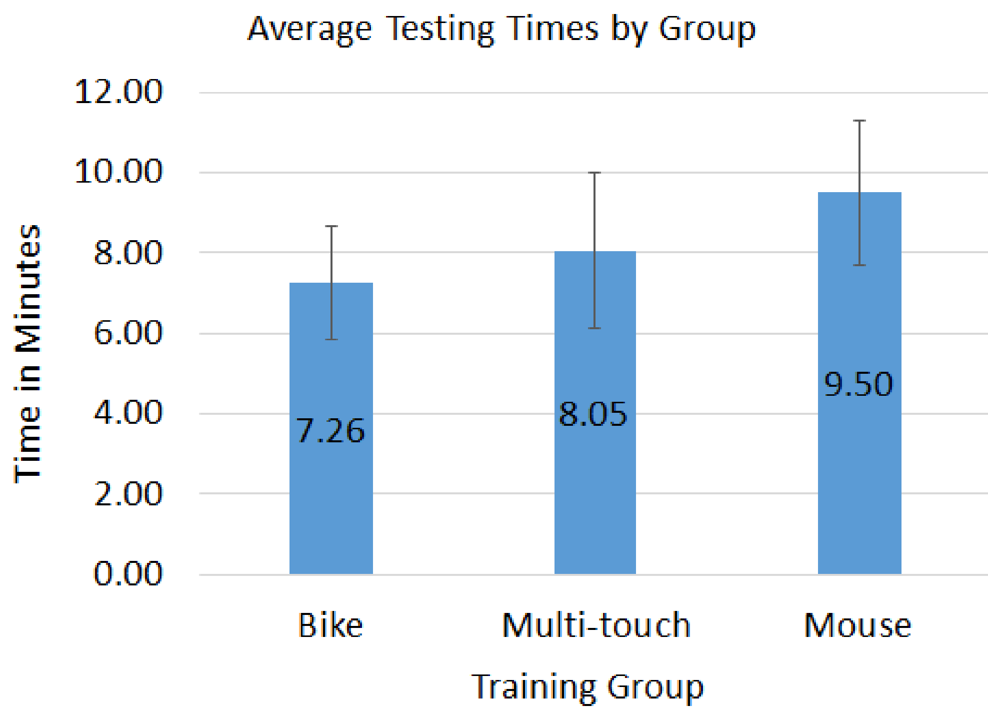


Figure 5.9: The average testing times by training group.

Discussion

The performance results (shown in Figure 5.7) demonstrate that participants trained with multi-touch interface scored as well as those trained on the actual apparatus and better than those trained on the mouse interface, confirming Hypotheses H1 and H2. However, the post-test times were not statistically significant and do not confirm Hypotheses H3 and H4. This could be due to a few reasons. It's possible incorrect steps could still lead to the same testing time as a test that was completed without errors. In addition, the inexperience of some participants using tools, causing them to be more cautious than others, may have lead to higher variance within groups. We believe that a few factors led to the multi-touch group scoring as well as the bike group. First, the affordances of the multi-touch model were closely mapped to the affordances of the real model, allowing multi-touch users to perform nearly the same physical gestures in the simulation as they would in real life. Secondly, strict constraints used in the multi-touch training (discussed in the Virtual Model and Multi-touch Interaction section) demanded more concentration allowing better memory of the steps as well as the physical motions required. Thirdly, users felt that repeating the training would be quick and easy since they could simply restart the environment to start again with the dismantled model versus undoing each step performed on the physical model.

Hypothesis H5 was partially confirmed, the Mouse group trained faster than the Multi-touch group but not the Bike group. We believe the gesture group had longer training times since there was a gesture learning curve and mistakes were made more often during training due some strict constraints. It could be argued that the multi-touch group performed better than the mouse group since they spent more time training. However, the majority of this time was spent learning, adjusting and executing gestures, not in additional training repetitions. Since the number of steps were identical and the virtual model was the same between the mouse and multi-touch groups, we believe the better multi-touch performance was due to the similarities of the multi-touch model affordances and physical gestures with the actual bicycle model and physical movements. Although the multi-touch training was not as efficient, time-wise, as the mouse training it was more effective.

The multi-touch interface provides the ability for users to manipulate tools and parts and perform realistic physical gestures as they would in a real-world procedure. Although our gesture set was somewhat small (7 gestures), it is still possible to achieve realistic actions; the gestures become more realistic when tied to physics forces and constraints. It is important for the physical motions and constraints within the procedural steps to be replicated where possible. For instance, consider a task in real life where one hand is required to hold a part in place while the other attaches it. The corresponding multi-touch interaction should also require a 2-handed gesture instead of objects snapping into to place or ignoring gravity. In some instances the stricter constraints in the multi-touch interaction may have led to more mistakes by users in training, but may have forced them to learn the steps better since they had to use more precise gestures.

For instance, a more complex 2-handed gesture was used for steps 4 and 5 (from Table 5.1) which required holding the clutch plate while tightening with the clutch wheel nut. As shown in Figure 5.8, more multi-touch users remembered the wheel nut step versus mouse users, potentially due to this increased amount of interaction. On the other hand it is interesting that approximately the same number of mouse and multi-touch users missed the clutch plate step. Some virtual users said this was because the shininess and/or shape of the plate in real life was not reflected in the virtual model and was hard to recognize.

In order to maintain the feeling of holding and manipulating the parts directly, or direct manipulation, we chose the viewpoints in the virtual environment carefully. A perceptual disconnect can occur when an object moves away from a user's touch and the object is no longer underneath the user's hand. This is particularly apparent when translating objects along multiple axes.

On a multi-touch interface, the user gets force feedback from the surface, however they don't get full 3DOF positional force nor any rotational force feedback. We chose to simulate applying force with pressure and use of more fingers. Pressure and the number of fingers required are good indicators of force but don't translate exactly to force in real life. For instance, attaching the spring to the bolt in the physical model requires pressing hard while rotating. In the multi-touch model users need to rotate 180 degrees and meet a pressure threshold. During the post-test, the multi-touch users expressed confusion as to why the spring wasn't attaching as easily as in the training.

For force feedback, the user could also be required to apply an opposing force in the form of more fingers and more pressure. For instance, the physical model's spring pushed back against the clutch plate reminding users to apply pressure to keep it in place. In the multi-touch environment, users had no such feedback and may forget that pressure is required to prevent the plate from detaching until the nut is secure. By requiring the user, in this instance, to apply pressure as an opposing force, they may learn the existence of force feedback in that action. Pressure could be improved upon by training users on how pressure is measured on a touch screen, calibrating pressure on a per user basis, or by providing visual and aural cues. Since forces cannot be felt by the user, visual and aural cues must compensate for that lost sense of force.

The realistic multi-touch gestures were tightly linked with physics in the virtual environment, procedural constraints, simulating forces, and carefully chosen viewpoints in order to create a convincing simulation of the physical actions within the maintenance procedure. Keeping in mind these considerations, multi-touch can provide a realistic and effective training environment for procedural tasks.

Conclusion

In this chapter we presented the use of realistic multi-touch interaction in a 3D training environment as a way to enhance learning of sensorimotor skills a well as procedural knowledge. We conducted a between subjects experiment with 36 participants distributed into 3 groups in order to evaluate the effectiveness of multi-touch training. One group used multi-touch interaction in the 3D training environment, the second used basic mouse-based interaction, and the third trained on the real equipment. A post-training test carried out 3 days later evaluated performance in conducting the real task from memory. Results show that the multi-touch interaction and the real task groups had significantly better performance scores than the mouse interaction group, with no significant difference between multi-touch and real task groups. We demonstrated that multi-touch interaction trains participants on the task as well as training on the actual equipment, suggesting multi-touch interaction is a worthwhile training tool for procedural knowledge that requires sensorimotor skills.

CHAPTER 6: CONSTRAINED AND UNCONSTRAINED MULTITOUCH INTERACTION IN PROCEDURAL TRAINING

Introduction

In chapter 5 we evaluated the training benefits of a multi-touch training system that used realistic physical multi-touch gestures in a virtual environment to provide realistic affordances [4]. While the previous study required realistic gestures to install and test parts during training, it required minimal object manipulation. We hypothesize limiting object manipulation by the user in this way takes away the benefit of focusing on the spatial properties of objects and negatively affects training. Inversely, it's also possible that limiting object manipulation may have benefits, such as reducing cognitive load and other distractors as discovered by Cockburn [67].

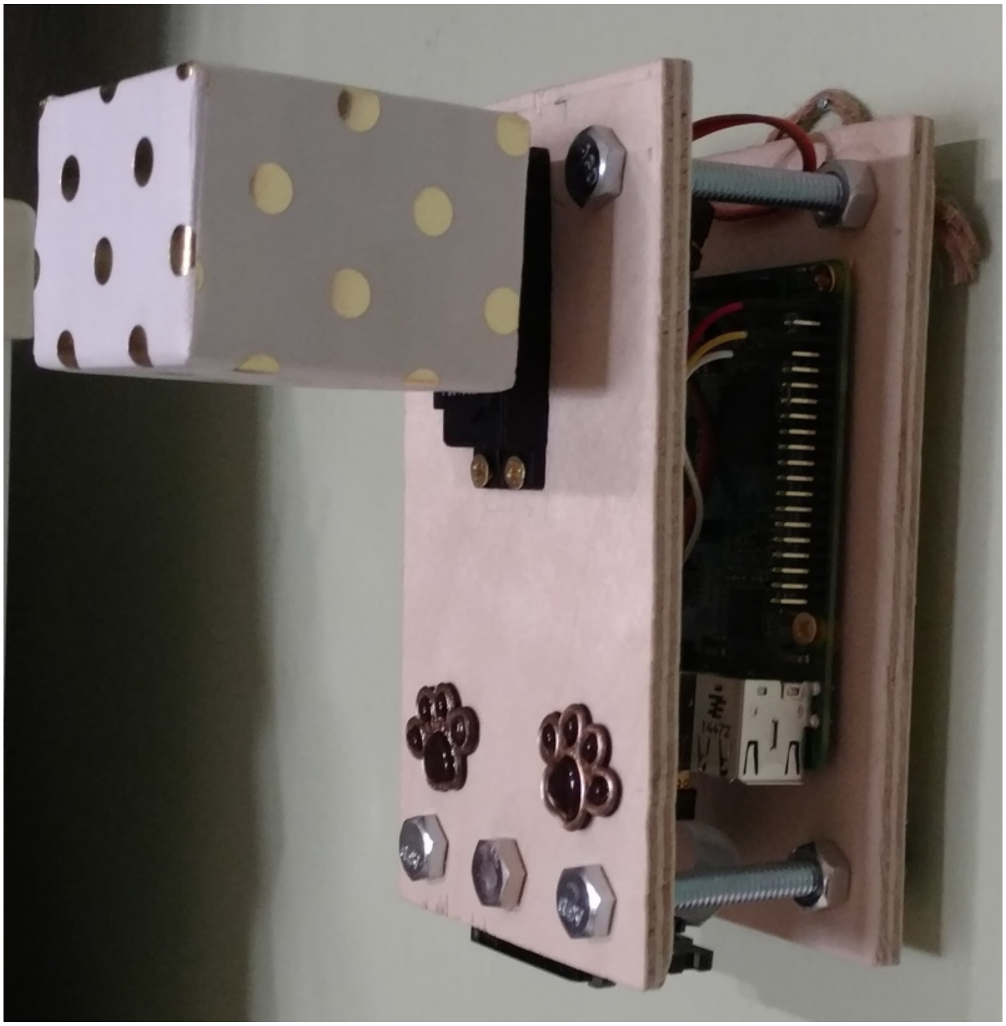


Figure 6.1: The dog treat dispenser prototype built with a raspberry pi, servo, and camera that users were trained to create.

We aim to evaluate the benefits of multi-touch training that has unconstrained object manipulation, where the user can pick up, rotate and examine parts. We compare this unconstrained multi-touch group to a group that has constrained object manipulation, where the parts automatically align themselves and the user cannot make their own rotations. In addition, we will compare both groups to a control group performing physical training with the actual apparatus. The selected experimental task consists of assembling a dog treat dispenser prototype built with a Raspberry Pi and other electrical components (shown in Figure 6.2).

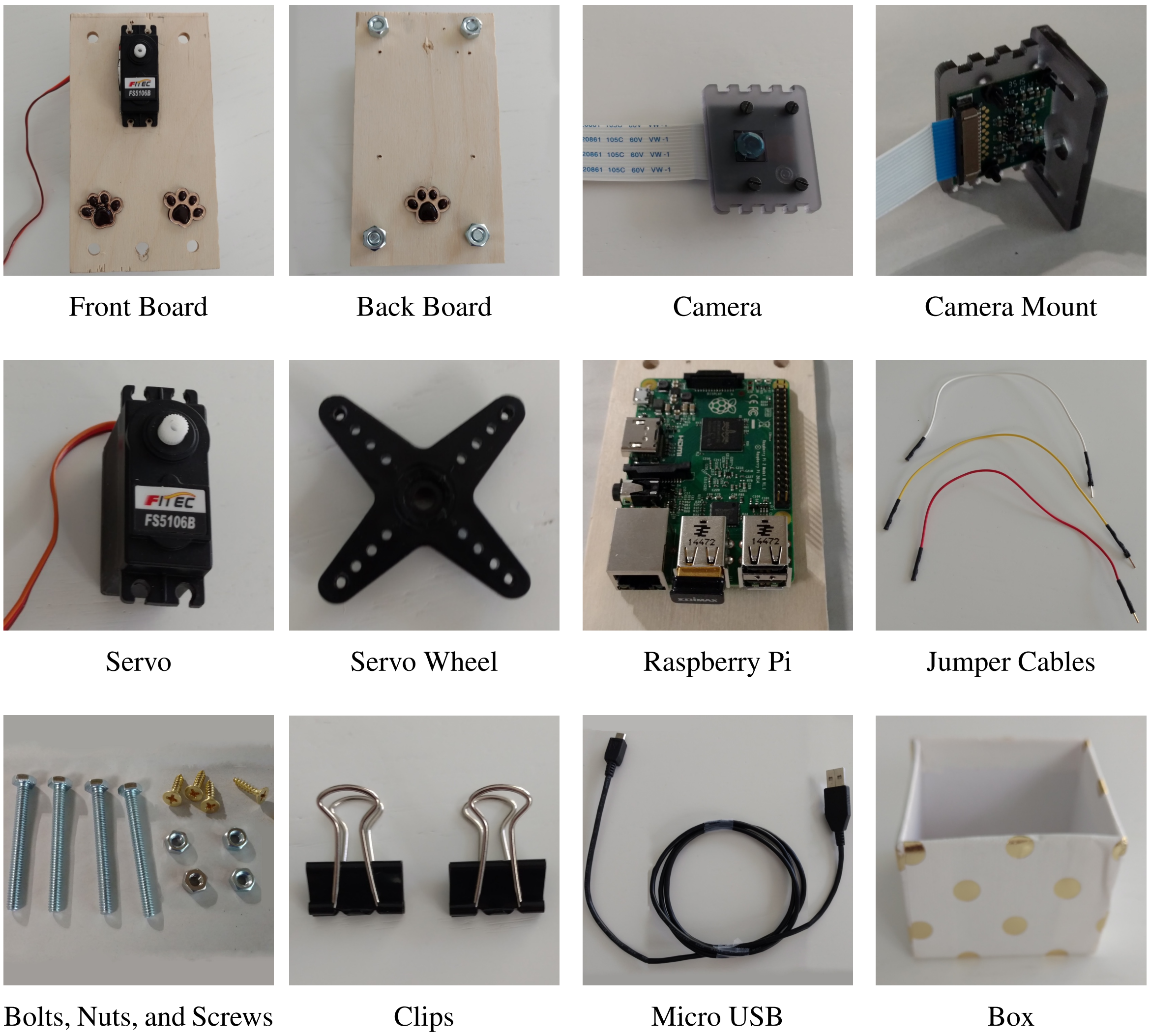


Figure 6.2: The parts used in assembling the prototype.

Training Systems and Task

The selected experimental task consisted of assembling a dog treat dispenser prototype built from a Raspberry Pi and other electrical components (shown in Figure 6.1). Since we wanted to focus on object manipulation and how that affects training, we selected the apparatus and task based upon a few requirements. First, the task should be confined to a small area like a table top so that the user could be seated and easily manipulate the parts in front of them to complete the task. Otherwise, if the task required larger immobile parts like an engine, the user would have to navigate or walk around in the environment as well as manipulate objects. Second, the task should use tools, parts and gestures that would be applicable to industrial, defense, or medical tasks so that the outcomes are relevant for training applications in these domains. Third, the task should be significantly complex and non-obvious that a novice could not complete it without prior training. These requirements led us to assembling a small electronic prototype where participants could assemble and then test the final product.

The overall task consisted of 13 general steps, where each step could require multiple repetitions of the same step (shown in Table 6.1). Some examples of the operations participants had to perform are: mounting parts with bolts, nuts or screws and connecting parts with different wires or cables. In total, participants had to manipulate 30 parts using their hands or a screwdriver (shown in Figure 6.2).

Physical Apparatus

The physical apparatus consisted of a Raspberry Pi and electronic components required to assemble the dog treat dispenser prototype arranged on a small table. The participants that trained on the physical apparatus had the advantage of experiencing the affordances provided by each of the tools and parts involved in the task. The ability to examine the parts from any viewpoint or zoom level is also an advantage the physical model provides. In addition, the physical apparatus provides feedback in the form of opposing forces and aural and visual cues.

Table 6.1: Steps in the procedure that were described in the user manual and that participants were scored on during the post-test.

	Steps
1	Insert the Servo
2	Attach Servo with 4 screws
3	Connect 4 bolts to front plate with 4 nuts
4	Connect camera to camera mount
5	Mount camera with short bolt and nut
6	Secure Raspberry Pi to back board with 4 screws
7	Connect the servo to the Raspberry Pi with 3 jumper cables
8	Connect camera ribbon to Raspberry Pi
9	Screw nuts 1/3 way down each front board bolt
10	Align back plate on top of front board bolts
11	Secure back plate with 4 nuts
12	Connect box to servo arm with 2 clips
13	Plug in micro usb cable

Virtual Apparatus

The virtual environment contained a detailed, working model of the Raspberry Pi, components, and tools involved (shown in Figure 6.3). Each corresponding step in the task was replicated in the virtual model and required interaction. We ensured both the Constrained and Unconstrained multi-touch interaction required the same number of steps to complete the task. In both virtual environments animations were used to show a part being installed. For example, in the Constrained group if the user dragged the servo to the correct hole to insert it into the front board, it would rotate and drop into the hole so they could visualize how to insert it. For the Unconstrained group, they would have had to rotate the servo first so it would only animate dropping into the hole.

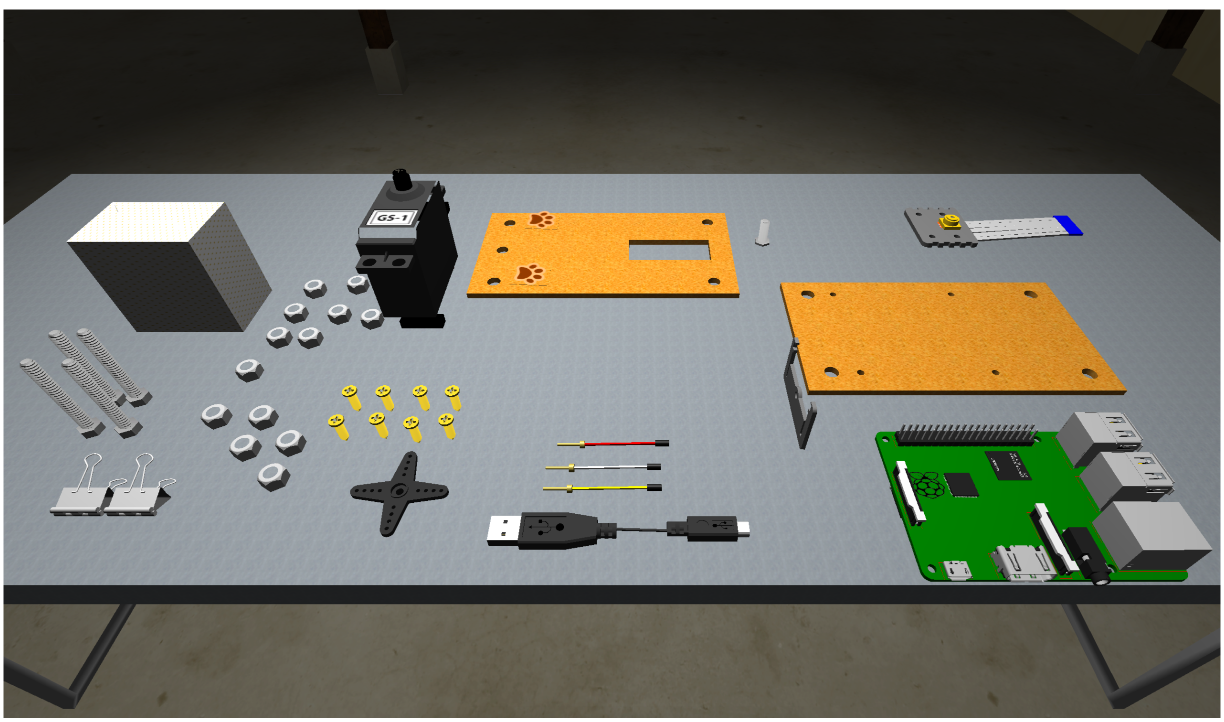


Figure 6.3: The virtual environment with the parts and tools on the table required to assemble the Raspberry Pi prototype.

Unconstrained Multi-Touch Interaction

In the unconstrained virtual environment we wanted the user to feel like they were actually picking up, examining and attaching components. In order to do this we wanted gravity to be used when dragging parts on the table, but when objects were ”picked up” we wanted gravity off so that users could feel like they were holding and examining the objects. Previous work on user-defined multi-touch gestures for 3D objects by [2] and [32] influenced our multi-touch interaction model for object manipulation. Buchanan et al. observed grasping and pinching gestures for grabbing or picking up objects which lead to gestures 6.4a and 6.4b [2]. Once the objects were selected, or ”picked up” the object moved to a fixed spot in the center of the screen closer to the camera and gravity was turned off. When the object was de-selected the part would then go back to its previous position. Cohe et al. observed both dragging perpendicular to the axis of rotation and orbit gestures for rotating objects around a specific axis which lead to 6.4d-6.4f [32].

In order for participants to become familiar with this gesture set we created a practice task which included only a board with 4 holes, a bolt and a Raspberry Pi. The participants had to rotate the bolt to the correct orientation to insert it into the board. Then they had to rotate the Raspberry Pi to match the orientation shown in an image in the paper user manual. The Unconstrained users went through this practice task only once and then begain the training task. They were also given a card showing the different gestures shown in Figure 6.4 in case they needed a reminder.

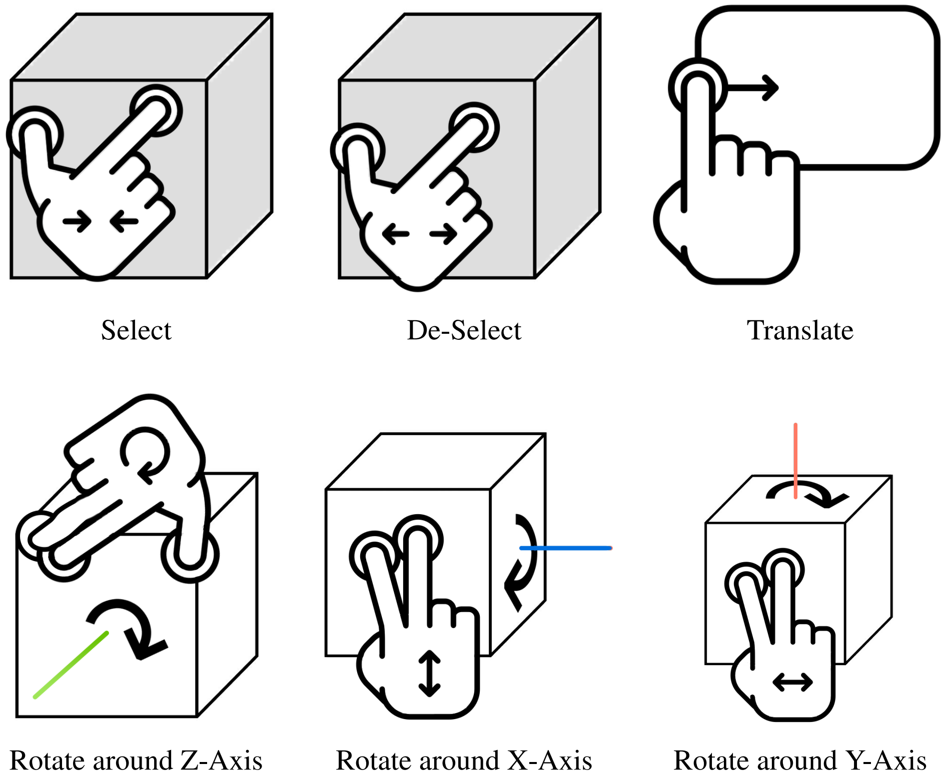


Figure 6.4: The gestures used in the Unconstrained group to manipulate the parts during assembly in the virtual environment. Once a part was selected with gesture 6.4a it would hover as if picked up from the table. Then the user could manipulate it with gestures 6.4c-6.4f. Gesture 6.4b would then release the object back to the table. The Constrained group could only use gesture 6.4c.

Constrained Multi-Touch Interaction

The Constrained multi-touch interaction didn't have gravity turned on and didn't have the ability for users to "pick up" or rotate objects. They could only drag objects to a target where they wanted them to be installed. The part would then orient itself to the correct orientation. If the following step required a part to be oriented differently to install the next part it would orient itself automatically. For example, the front board would flip over after the servo was installed to allow easy access to the servo wires.

User Study

We conducted a user study to explore the effectiveness of Unconstrained multi-touch interaction during training. The study examined three different methods of training how to assemble a Raspberry Pi dog treat dispenser prototype. The three methods were using the actual equipment, using a virtual model with Unconstrained multi-touch interaction, and using a virtual model with Constrained multi-touch interaction. All groups trained with the same paper manual and returned two days later for a post-training test. During the post-test participants were asked to perform the task on the actual equipment as quickly as possible from memory and were graded on correct steps and timed. Our hypotheses were:

H1 The participants trained using the Unconstrained multi-touch interface will score as well on the post-test as the participants trained with the actual equipment.

H2 The participants trained using the Unconstrained multi-touch interface will score better on the post-test than the participants trained using the Constrained multi-touch.

H3 Due to discrepancies between manipulating the actual components and the virtual model, the participants trained using the Unconstrained multi-touch interface will perform the post-test slower than the participants trained with the actual equipment.

H4 Since the Unconstrained multi-touch participants may have learned more about the parts and procedure, participants trained using the Unconstrained multi-touch interface will perform the post-test faster than the participants trained using the Constrained multi-touch interface.

H5 The training time for the Constrained multi-touch participants will be faster than the physical and Unconstrained multi-touch groups since they don't have to precisely align components.

Participants and Apparatus

There were 39 participants (8 female, 31 male) aged 18 to 34 ($\bar{x}$ = 21.8,σ = 3.65) randomly distributed into the three groups. The participants were recruited from a university setting from a variety of majors, including engineering and non-engineering majors. Sixteen of the participants had used a Raspberry Pi before. Training was either conducted on the computer via a 55-inch Microsoft Perceptive Pixel multi-touch screen, or on the actual Raspberry Pi prototype. Post-tests were performed on the actual Raspberry Pi prototype.

Study Design

The experiment follows a between-subjects design with 39 participants randomly divided into 3 experimental groups (each participant was assigned to a group in alternating order). Two groups trained in a virtual environment, one using Unconstrained multi-touch interaction or the other using Constrained multi-touch interaction. The third group was trained on the actual Raspberry Pi prototype.

Participants were timed during training and testing and scored on the test. Participants were also asked to rate the following statements on a Likert scale from 1 (strongly disagree) to 7 (strongly agree), both post-training and post-testing:

Post-Training questions:

Q1 I feel prepared to complete the assembly task after completing the training.

Q2 I thoroughly understand the procedure that I learned during the training.

Post-Testing questions:

Q3 I was sufficiently prepared to complete the assembly task upon arrival today.

Q4 I thoroughly understood the procedure that I learned during the training.

Procedure

Testing occurred exactly 2 days after training to avoid short-term memory effects. For training, all groups followed the same protocol. First, participants were presented with the training apparatus (whether it be the computer or Raspberry Pi and parts) and the paper manual. Second, they were familiarized either with the parts and tools or the interaction model. Finally, participants were asked to follow along with the paper manual as they conducted the training task. Participants then returned 2 days later to perform the post-test, which was assembling the actual prototype from memory without the manual.

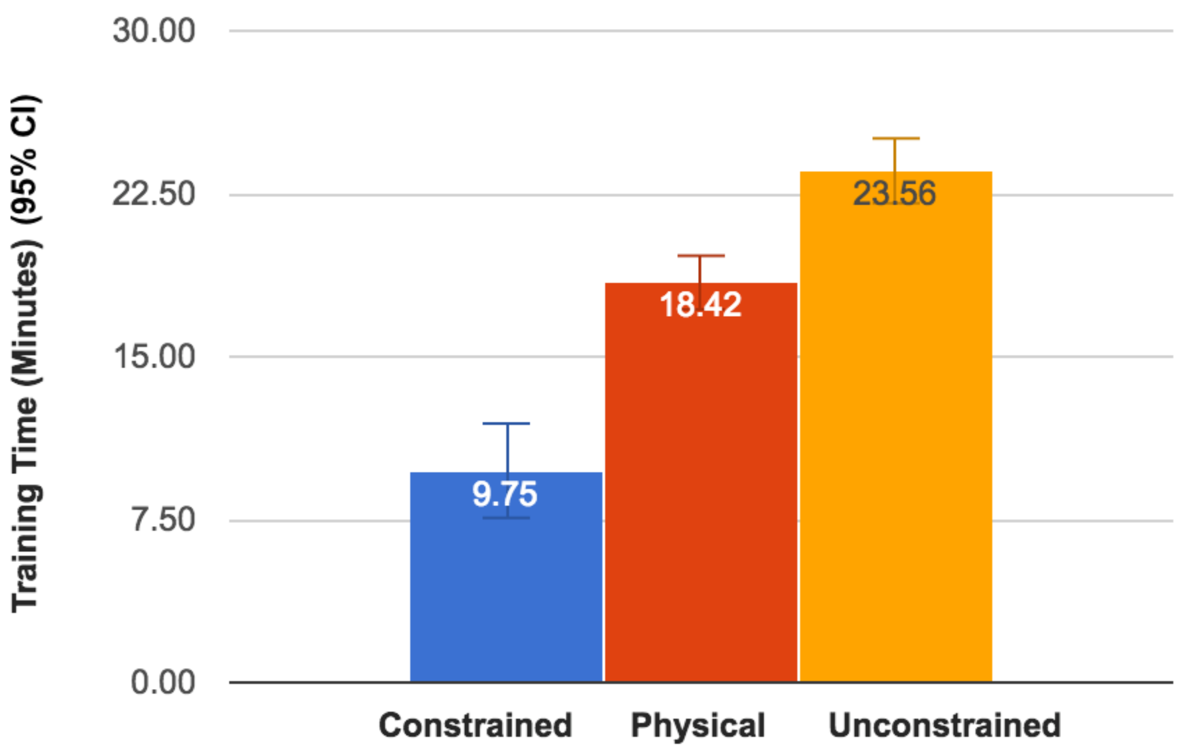


Figure 6.5: The average training times by group.

Two days later, participants returned to perform the task on the real Raspberry Pi prototype. They were asked to perform the task correctly from memory without the user manual. They were scored on the steps shown in Table 6.1, where each step could actually count as multiple steps. For example, when inserting the servo we also checked if the servo was inserted in the correct side of the board and in the correct orientation. One point was given for each step completed correctly and performed in the correct order, for a total of 33 points. We allowed them to do the steps out of order during the test as long as it resulted in the correct outcome. Many steps required the previous step to be completed, so going out of order would ultimately result in longer time since users would have to undo something in order to proceed.

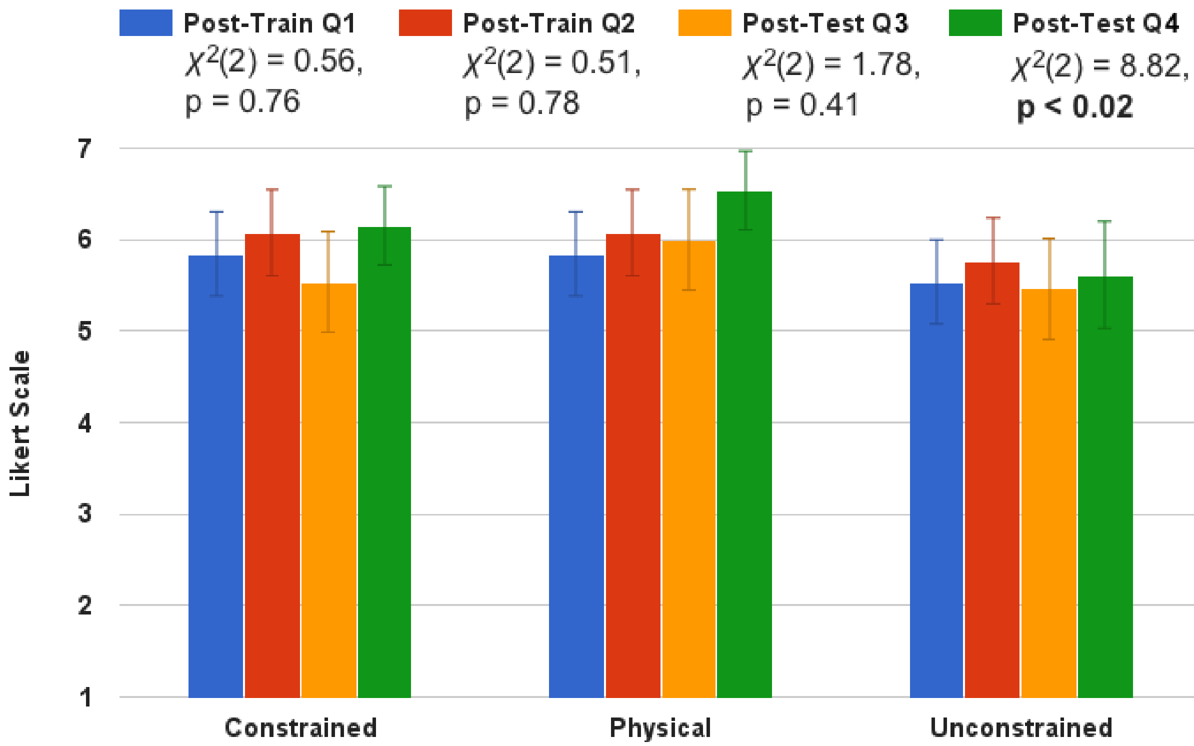


Figure 6.6: Average user Likert ratings on the preparedness and understanding questions asked after training and after testing.

Results

Training

Training Times

The participants from all three groups were asked to go through the training exercise once by following along with the same paper user manual. Participants were told before beginning that they would have to return 2 days later to complete the real-life task from memory without the user manual. The average training time by group is shown in Figure 6.5. We first ran a One Way ANOVA to determine that there were significant differences between groups ($F_{2,36}$ = 19.55, p <0.001). Then we performed an independent samples t-test to compare groups to each other. The Constrained group performed the training significantly faster than both the Physical group ($t_{24} = 4.58, p<0.001$) and the Unconstrained group ($t_{24} = 6.29, p<0.001$). However, the training times for the Physical and Unconstrained groups were not significantly different ($t_{24} = 2.01, p = 0.056$). These results were as expected and confirm Hypothesis H5; the training time for the Constrained participants will be faster than the Physical and Unconstrained groups since they didn't have to precisely align components. We also observed that many participants had to put in extra effort to rotate components in the Unconstrained group since there was a learning curve for the rotational gestures, which explains the slightly longer average training times compared to the Physical group.

Questionnaire

We performed a Kruskal-Wallis test (results are shown in Figure 6.6) which showed that there were no significant differences between the different groups for their responses to the first 3 of the 4 Likert questions (presented in the Participants and Apparatus section). These results indicate that participants from all groups felt that they were prepared after the training, and still felt that they were prepared after performing the actual post-test. However, for the last post-test question Q4 ("I thoroughly understood the procedure that I learned during the training"), the Kruskall-Wallis test showed there were significant differences between groups ($\chi^2(2) = 8.82, p<0.02$). A Mann-Whitney U test revealed there were only significant differences between the Physical and Unconstrained groups ($U = 30.5, p<0.003$), signifying in hindsight the Unconstrained group felt they didn't understand the procedure as well as those in the Physical group.

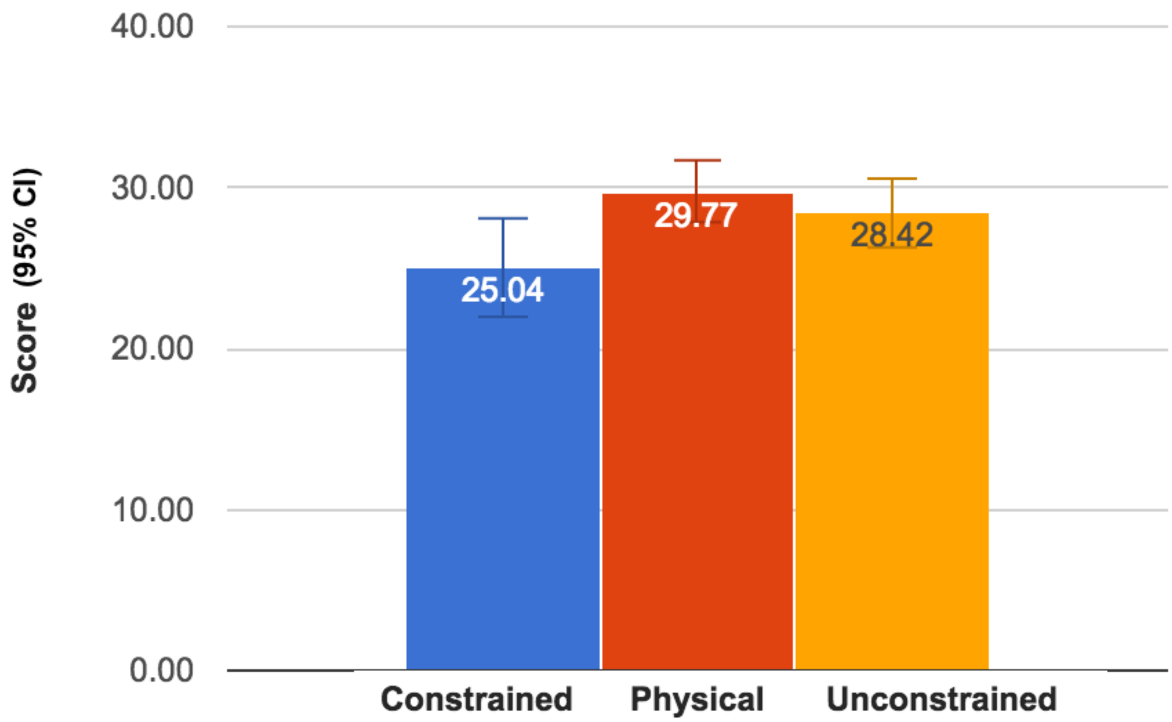


Figure 6.7: The average testing scores by group.

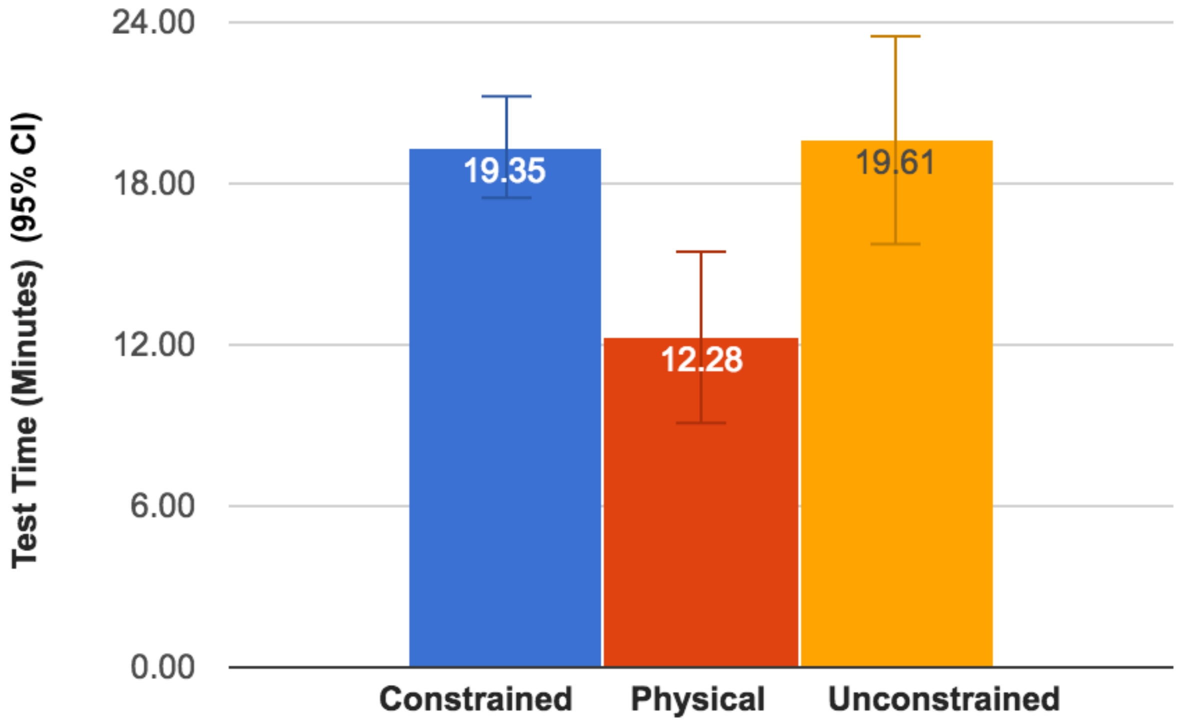


Figure 6.8: The average testing times by group.

Testing

Testing Scores

When the participants returned for testing, they were asked to perform the task on the Raspberry Pi prototype from memory. They were scored on number of steps completed correctly and timed. The average scores by group are shown in Figure 6.7. The scores were normally distributed for the Constrained, Unconstrained, and Physical groups as assessed by Shapiro-Wilk's test ($p > .05$). A one-way between-subjects analysis of variance (ANOVA) was run to compare the effect of the training method on the testing score. The main effect that training method had on the dependent variable, testing score, was found to be significant ($F_{2,36}$ = 8.60, p <0.001). Independent samples t-tests indicated the Unconstrained group scored significantly better than the Constrained group ($t_{24} = 2.62, p<0.02$) and the Physical group also scored significantly better than the Constrained group ($t_{24} = 3.87, p<0.001$). However, the Physical and Unconstrained group scores were not significantly different ($t_{24} = 1.36, p = 0.19$). This result demonstrates that participants that trained with Unconstrained interaction performed as well as those trained on the actual apparatus and better than those trained with Constrained interaction, which supports Hypotheses H1 and H2.

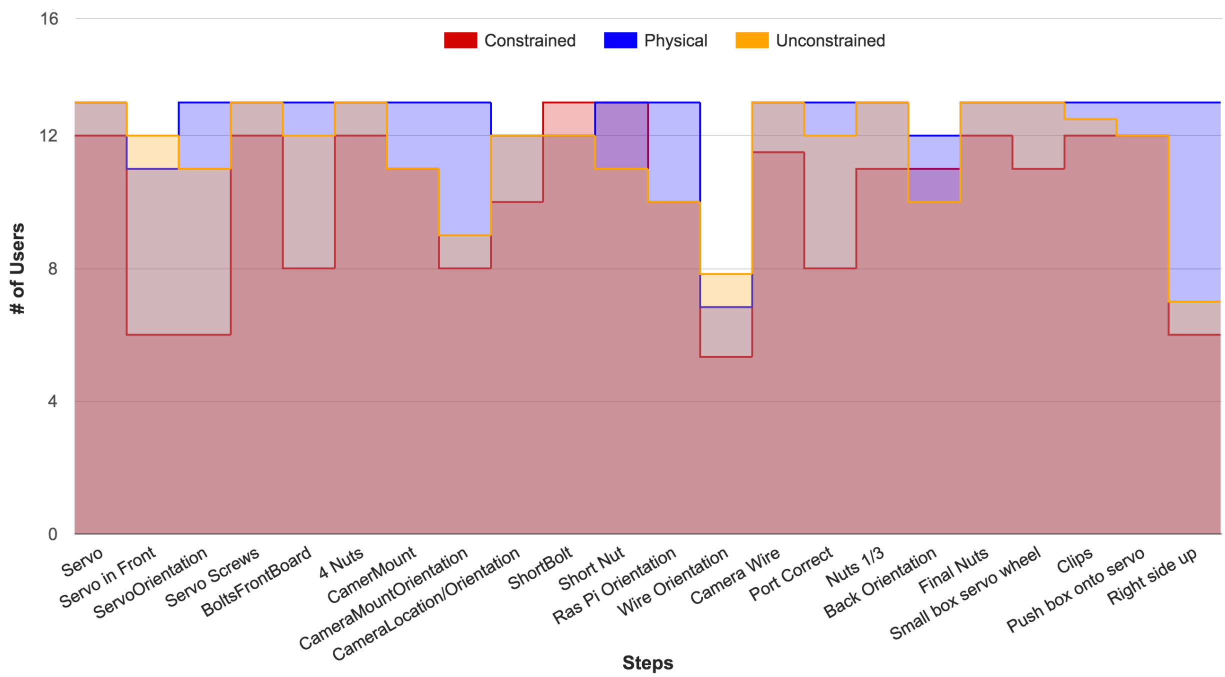


Figure 6.9: The number of users that completed each testing step correctly for each training group; the Physical group in blue, the Constrained group in red, and the Unconstrained group in yellow. Note that steps all users performed correctly are not shown and the wire orientation is shown as a fraction of correct wires/6.

Testing Errors

Figure 6.9 shows each step of the testing process and how many users got the step correct, separated by testing groups. We did not score based on steps that were performed out of order, but only on if a step was done incorrectly. The errors occurred when a user forgot or skipped a step completely or put the part on the wrong location or orientation. Figure 6.9 also demonstrates on which steps users made the most errors by showing where the deficits are. For example, the wire orientation as well as the servo orientation were the hardest for users to remember. The steps that all users remembered and performed correctly were omitted from the graph (installing the 4 bolts, Raspberry Pi, 4 Raspberry Pi screws, servo wires, back board, and micro USB cable).

Testing Times

The testing times were also measured (shown in Figure 6.8) and found to be normally distributed for the Constrained, Unconstrained, and Physical groups, as assessed by Shapiro-Wilk's test (p $>$.05). A one way between-subjects analysis of variance (ANOVA) was run to compare the effect of the training method on the testing score. The main effect that training method had on the dependent variable, testing time, was significant ($F_{2,36}$ = 7.77, p $<$0.002). An independent samples t-test revealed that there were not significant differences between the Constrained and Unconstrained groups ($t_{24} = 0.11, p = 0.92$), but the Physical group performed the post-test significantly faster than both the Unconstrained ($t_{24} = 3.73, p{<}0.001$) and Constrained ($t_{24} = 3.75, p{<}0.001$) groups. These results confirm Hypothesis H3 (Unconstrained test time $>$Physical test time), but not Hypothesis H4 (Unconstrained test time $<$Constrained test time).

Discussion

The performance results (shown in Figure 6.7) demonstrate that participants trained with Unconstrained multi-touch interaction scored as well as those trained on the actual apparatus and better than those trained with Constrained multi-touch interaction, confirming Hypotheses H1 and H2. We believe that a few factors led to the Unconstrained group scoring as well as the Physical group. First, the object manipulation affordances in the Unconstrained interaction model were closely mapped to the manipulation affordances of the real model. The user had the ability to pick up and examine parts, and was forced to put them into the correct orientation to be able to install them. We believe this requirement allowed users to become more familiar with the properties of each component. Secondly, if they could not install a part due to incorrect orientation they had to re-evaluate the instructions which required more focus and understanding of the current task. Thirdly, there was a learning curve associated with the gestures which also required more time and focus spent on the current task. We did consider the alternative outcome that this gesture learning curve might distract participants as found in [69] and [70]. However our results indicate that instead the increased difficulty of the interface enhanced learning.

The post-test times indicated that both Multi-touch groups had significantly slower task completion times in the post-test, which confirmed Hypotheses H3 and rejects H4. This could be due to a few reasons. It's possible incorrect and skipped steps could still lead to the same testing time as a test that was completed without errors. In addition, the inexperience of some participants using tools and electronic parts could have caused them to be more cautious than others, may have lead to higher variance within groups. The Physical group was probably more familiar with how the parts felt and fit together so they were able to move faster through the post-test. In addition, to ensure training conditions were similar across groups, each group only went through the training once. Had the virtual groups gone through the training multiple times they probably would have learned the procedure better and the task completion times may have been faster.

Hypothesis H5 was also confirmed, the Constrained group trained faster than the Unconstrained group but not the Physical group. We believe the Unconstrained group had longer training times since there was a gesture learning curve and mistakes were made more often during training due to some strict alignment requirements. It could be argued that the Unconstrained group performed better than the Constrained group since they spent more time training. However, this time was spent learning, adjusting and executing gestures, not in additional training repetitions. Since the number of steps were identical and the virtual model was the same between the Constrained and Unconstrained groups, we believe the better Unconstrained performance was due to the gained familiarity with the objects and the extra focus required to complete the training task. Although the Unconstrained training was not as efficient, time-wise, as the Constrained training it was more effective.

In some instances the stricter constraints in the Unconstrained multi-touch interaction may have led to more mistakes by users in training, but may have forced them to learn the steps better since they had to use more precise gestures. For instance, looking at Figure 9 we can see that although all users remembered to insert the servo and install the 4 bolts, the Constrained group made more errors on these steps by inserting the servo upside down or screwing it into the wrong side of the board. Many participants from the Constrained group also inserted the 4 bolts into the wrong board or the wrong way into the board.

The Questionnaire results for Q1-Q3 indicate that participants from all groups felt that they were prepared after the training, and still felt that they were prepared after performing the actual posttest. However, for the last post-test question Q4 (”I thoroughly understood the procedure that I learned during the training”), the Unconstrained group rated it significantly less than the Physical group. This signifies that in hindsight the Unconstrained group felt they didn’t understand the procedure as well as those in the Physical group. We believe this may be due to the gesture learning curve they experienced during the training. It's possible that since few users were extremely comfortable with the gestures, since they had to adjust rotations several times, they didn't feel that they had mastered the procedure as well. Even though the Unconstrained group did in fact score as well as the Physical group.

Limitations and Future Work

Since we wanted to evaluate how the freedom to manipulate objects affects learning a procedural task, we chose not to implement many installation gestures. For instance, we did require the Unconstrained group to screw in screws and nuts but we didn't require any more complex installation gestures as in Buchanan et al. [4]. In future work, we would like to look at the combination of object manipulation gestures and installation gestures. In addition, examining how increasing the training time with each method affects learning would be interesting.

An interesting outcome was that many users that trained in either of the multi-touch groups said that they enjoyed training in a virtual environment first since it was an easier way to train without worrying about breaking anything. It would be interesting to look at if training in virtual environments actually motivates students to be more interested in electronic, computer engineering, or electrical engineering. Since we had the participants use a paper training manual we would like to incorporate training instructions and guidance into the virtual environment. We would also like to do a similar study comparing multi-touch gestures to Virtual Reality, full body gestures, and training on physical systems using Augmented Reality potentially on a more complex procedural task.

Conclusion

In this chapter we presented the use of object manipulation gestures combined with alignment requirements in a 3D training environment as a way to enhance learning of a procedural task. We conducted a between subjects experiment with 39 participants distributed into 3 groups in order to evaluate the effectiveness of Unconstrained and Constrained multi-touch interaction during training. One group used Unconstrained multitouch interaction in the 3D training environment, which meant they could pick up and rotate objects and were also required to properly align components to install them. The second group used Constrained multi-touch interaction, where they could only drag and drop components to install them and components aligned to their positions automatically. And the third Physical group trained on the real apparatus. A post-training test carried out 2 days later evaluated performance in conducting the real task from memory. Results show that the Unconstrained multi-touch interaction and the Physical task groups had significantly better performance scores than the Constrained multi-touch interaction group, with no significant difference between the Unconstrained multi-touch and Physical task groups. We demonstrated that Unconstrained multi-touch interaction trains participants on the task as well as training on the actual equipment, suggesting multi-touch interaction is a worthwhile training tool for procedural knowledge that requires spatial knowledge of components.

CHAPTER 7: DISCUSSION AND FUTURE WORK

In this dissertation, we explored how to define realistic multi-touch gestures for 3D tasks and how these gestures affect procedural training. First a user-defined gesture study led us to design guidelines for realistic multi-touch gestures for 3D tasks in Chapter 3. Second we conducted a follow-up study on how pressure can be implemented on multi-touch displays in Chapter 4. We then implemented these gestures in two different training scenarios to see how they compared to traditional training methods in Chapters 5 and 6.

Existing bias, motivation and viewpoint play a role in the gesture chosen to represent a physical action. The results from Chapter 3 indicate that due to biases from previous multi-touch experience, the majority of participants intuitively try to interact with 3D objects using 1-2 finger gestures in a primarily metaphorical way. However, once prompted to use gestures as if manipulating physical, real-life objects, the users increased the number of fingers, hands and pressure used, and used more gestures that were physical in nature. The participants also found these physical gestures just as easy to perform as metaphorical gestures. Thus if interface designers want to elicit physical gestures there needs to be guidance in doing so. The form factor and the perspective view of the referent also played a role in the gesture chosen. If the viewpoint of the referent did not align with the multi-touch surface users were more likely to choose a metaphorical gesture since a physical gesture on the surface would not align with the referent.

Contact size can be used to simulate pressure for simulated force or for depth control on a capacitive touchscreen. Since pressure played such a large role in how users performed physical gestures, we examined how best to estimate pressure on a capacitive touchscreen. Capacitive touch screens don't sense pressure but can report contact size. Thus we studied two different pressure estimation techniques that use contact size, calibrated and comparative, on two different types of tasks, gross and fine. We found that the calibrated estimation technique performed better on the gross task, which was a bi-directional translation task. Whereas the comparative technique performed better on the fine task, which was a uni-directional rotation task. Since bi-directional control required a predictable neutral position, which users were able to achieve through calibration, the calibrated method performed better than the comparative method on the gross task. However the calibrated method did not perform as well for the fine, uni-directional rotation task. For the fine task, some users fingers did not flex in a way for them to be able to surpass their calibrated threshold without putting their fingers into an unnatural or uncomfortable position. Also the orientation of the users fingers changed as they were rotating which made it difficult to maintain pressure past the threshold value during rotation. Whereas with the comparative pressure technique the user can reach a higher pressure value simply by starting with a very light pressure. Users learned that starting with light pressure for the comparative method was effective during the practice session. If calibration or pressure estimation customization is not possible in a complex task that requires both gross and motor skills, then it would be best to use the comparative estimation technique. The comparative pressure estimation technique performed more consistently across both motor skills, though it performed sub-optimally for the gross motor skill. Thus we used comparative pressure in the first training study of Chapter 5.

Physical gestures used in a procedural training task on mechanical components can benefit training outcomes. In our first training study we compared users that trained with a multi-touch interface to those trained with a keyboard and mouse with a control group training on the real world physical apparatus. For the multi-touch interface we implemented the gestures with the design guidelines defined in Chapter 3 with the comparative pressure examined in Chapter 4. The performance results demonstrated that participants trained with the multitouch interface scored as well as those trained on the real world apparatus and better than those trained on the mouse interface.

We believe that a few factors led to the multi-touch group scoring as well as the real world group.

First, the affordances of the multi-touch model were closely mapped to the affordances of the real model, allowing multi-touch users to perform nearly the same physical gestures in the simulation as they would in real life. Secondly, strict constraints used in the multi-touch training demanded more concentration allowing better memory of the steps as well as the physical motions required. The multi-touch interface of Chapter 5 provides the ability for users to manipulate tools and parts and perform realistic physical gestures as they would in a real-world procedure.

Although our gesture set in Chapter 5 was somewhat small (7 gestures), it is still possible to achieve realistic actions; the gestures become more realistic when tied to physics forces and constraints. We believe it is important for the physical motions and constraints within the procedural steps to be replicated where possible. For instance, consider a task in real life where one hand is required to hold a part in place while the other attaches it. The corresponding multi-touch interaction should also require a 2-handed gesture instead of objects snapping into to place or ignoring gravity. In some instances the stricter constraints in the multitouch interaction may have led to more mistakes by users in training, but may have forced them to learn the steps better since they had to use more precise gestures.

Viewpoint plays an important role in how realistic a gesture feels. When designing the study of Chapter 5, we chose the viewpoints in the virtual environment carefully in order to maintain the feeling of holding and manipulating the parts directly, or direct manipulation. A perceptual disconnect can occur when an object moves away from a users touch and the object is no longer underneath the users hand. This is particularly apparent when translating objects along multiple axes.

Multi-touch interaction with unconstrained object manipulation results in better procedural training outcomes. In Chapter 6's study we found participants that trained with Unconstrained multi-touch interaction scored as well as those trained on the real world apparatus and better than those trained with Constrained multi-touch interaction. We believe that a few factors led to the Unconstrained group scoring as well as the Physical group. First, the object manipulation affordances in the Unconstrained interaction model were closely mapped to the manipulation affordances of the real model. The user had the ability to pick up and examine parts, and was forced to put them into the correct orientation to be able to install them. We believe this requirement allowed users to become more familiar with the properties of each component. Secondly, if a user could not install a part due to incorrect orientation they had to re-evaluate the instructions which required more focus and understanding of the current task. Thirdly, there was a learning curve associated with the gestures which also required more time and focus spent on the current task. We did consider the alternative outcome that this gesture learning curve might distract participants, however our results indicate that instead the increased difficulty of the interface enhanced learning.

As part of our experiments we looked at two different multitouch training environments: one where the focus was on the physical gestures with a constrained viewpoint (Chapter 5), and another where the focus was on object manipulation where the user could view objects from different angles (Chapter 6). Across both studies, participants with higher forms of interaction outperformed participants with less interaction. Users trained on constrained multi-touch outperformed those that only had access to mouse based interaction and users that trained on unconstrained multi-touch outperformed those who trained with constrained multi-touch. We believe this is for a few reasons: first the multi-touch interactions mimick the real world interactions which may provide muscle memory. Second, enactive learning, where actions have consequences, can provide better learning outcomes. Third, unconstrained object manipulation allow users more spatial awareness of objects. Finally, more difficult interfaces can require more time and focus.

Future Work

The continuing popularity of multi-touch devices makes it increasingly valuable to explore how training applications can best leverage realistic multi-touch interaction. We began by exploring design guidelines and evaluating multiple use cases in this work, but we believe there are many opportunities for this work to be continued and extended. This section offers suggestions on how to build upon this work further over the next few years.

Contact Size Calibration

The contact size calibration could be improved upon for our Calibrated pressure estimation method presented in Chapter 4. As a user's touch moves away from the center of the screen the finger pad's orientation is going to change slightly, affecting the contact size. Ideally, calibration should obtain finger pad representations at the outer edges of the screen, as well as the center. Then if the calibration method was also cognizant of the user's position relative to the screen, it could make assumptions about the finger pad's orientation.

Learning Types

In our experiments we did not examine what types of learners (i.e. Visual, Auditory, Reading/Writing, Kinesthetic) or what types of users (i.e. Novice or Expert) benefit the most from this type of mulit-touch training interface. In addition, it would be valuable to examine which specific parts of the 3D interface correspond to specific learning goals. We would also like to evaluate realistic gestures for non-procedural learning. For instance instead of training to learn a procedure, users could train on how a component, such as the raspberry pi, works in general using different examples. Then users could be tested in a different context to see if they can extrapolate their training knowledge to a new task.

In-Situ Gesture Learning

One obstacle of both studies was understanding how to best teach the gestures required for the study. In each experiment, we had a practice scene where users could experiment with the gestures, however there was often still a steep learning curve during the training. The proctor usually had to explain the gestures verbally again during the training session when the user could not recall which gesture to use or was having trouble performing a gesture correctly. ShadowGuides and Octopocus have presented different visualizations for in-situ learning of multi-touch gestures and mouse gestures respectively [78, 79]. It would be interesting to see how these visualization can be implemented in a training environment. It would also be interesting to see how to distinguish between visualization cues for installation gestures and for direct manipulation gestures.

Above the Surface Interaction

Our research exclusively focused on interactions with the surface of the multitouch display however there is research evaluating above the surface interaction techniques by Marquardt et al [23]. They outlined techniques such as lifting gestures to reveal objects and to adjust scale, stacking objects, and 6 degrees of freedom (DOF) manipulation. It would be interesting to compare our techniques with Marquardt et al. 's for the same operations to see which users prefer in a training environment.

Navigation

We have evaluated two different training applications; one where installation gestures were used but object manipulation and viewpoint were constrained, and another where object manipulation was unconstrained. It would be interesting to look at a training environment that requires navigation, such as repairing a large apparatus, and see if the freedom to navigate the environment helps or hinders learning. We believe requiring gestures for installing parts, object manipulation, and navigation may cause the cognitive burden to be too high. It would be interesting to find the ideal balance between having too little and too much freedom to interact with the environment.

CHAPTER 8: CONCLUSION

With the prevalence of multi-touch interfaces and affordable large touch screens, multi-touch gestures that incorporate multiple fingers or multiple hands are possible. We envision realistic interactions to be used on these large multi-touch surfaces to mimic training in the real world. However, a major challenge in 3D touch interaction is understanding how to translate interaction on a 2D surface to a 3D environment. We would like users to be able to pick up, rotate and move objects in a way that's relevant to training environments. The first problem we aimed to solve is how do we get closer to mimicking real world interactions. The second question we examined is whether implementing these realistic multi-touch interactions in training contexts provides procedural training benefits.

In this dissertation, we conducted two studies that resulted in design guidelines for realistic and intuitive multi-touch interaction for 3D virtual objects. After developing the guidelines we conducted two additional studies to measure whether or not implementing these interactions in multi-touch training environments benefits procedural training outcomes.

In the first study, we explore the best ways to use multi-touch interaction in 3D training environments [2]. We performed a gesture elicitation user study to define which gestures users preferred on large capacitive displays when interacting with 3D objects. We found that many users' first instinct is to use 1 or 2 finger gestures similar to RST gestures used on a phone. However, when prompted to do so, users tried to mimic physical, real-world actions such as using more fingers and applied pressure to signify more force or to move an object into the screen. These results signified that multi-touch gestures could mimic real world actions and be intuitive to users.

In our second user study we focused on defining the best ways to estimate pressure on capactivate displays to enhance physical gestures [3]. Today's most prevalent touchscreens use capacitive sensing which can report the location and surface area of a touch, but do not directly sense the pressure of the touch. Thus on capacitive screens, multi-touch pressure can only be estimated based upon touch contact size. We explored how best to interpret pressure from contact size by developing multiple estimation techniques. We also examined which of our pressure estimation techniques were preferred amongst different scenarios. We found different pressure estimation techniques are significantly better for different tasks based on the type of motor skills being performed.

In our third study we examined how multi-touch interaction affects the learning of a procedural assembly task [4]. Our goal was to evaluate how our gesture set impacts learning in a training environment and if gesture similarities to physical actions better prepare the user for performing the physical steps in the real world. Since training can be looked at in many different ways we are focusing on procedural learning, or learning a repeatable step-by-step process. The study evaluates the knowledge transfer acquired with multi-touch interaction technology compared to standard training methods. We compared multi-touch interaction to 2D mouse interaction and to actual physical training. We found that multi-touch interaction performed better than 2D mouse and performed as well as physical training.

In our final study, we examined how having the freedom to explore in a training environment affects learning. Our third study, like many multi-touch training systems and 3D training systems in general, had a snap to position functionality that didn't give the user the freedom to manipulate the objects in great detail or to look around and examine different parts of objects as they would in the real world. We evaluate the benefits of multi-touch training that has unconstrained object manipulation, where the user can pick up, rotate and examine parts. We compared this unconstrained multi-touch group to a group that has constrained object manipulation, in which the parts automatically align themselves and the user cannot make their own rotations. In addition, we compared both groups to a control group performing physical training with the real world apparatus. The selected experimental task consists of assembling a dog treat dispenser prototype built with a

Raspberry Pi and other electrical components. We demonstrated that Unconstrained multi-touch interaction trains participants on the task as well as training on the actual equipment and better than Constrained interaction.

Multi-touch interactions for 3D environments could improve training outcomes if used effectively. We found that in order to create realistic gestures, the gestures should be designed to be physical in nature, meaning they should provide the same affordances as interactions with the same real world objects. Along the same vein, if interactions in the real world require effort or force, the replicated multi-touch interaction should require more fingers and pressure. In two user studies, we found that increasing the level of interaction with a training environement enhances learning of a procedural task. We believe there are still many opportunities for this work to be continued and extended. We hope that our work will serve as a guideline for future exploration into improving training with the usage of 3D multi-touch interaction.

APPENDIX : IRB APPROVAL LETTERS

University of Central Florida Institutional Review Board
Office of Research & Commercialization
12201 Research Parkway, Suite 501
Orlando, Florida 32826-3246
Telephone: 407-823-2901 or 407-882-2276
www.research.ucf.edu/compliance/irb.html

Approval of Human Research

From: **UCF Institutional Review Board #1**
FWA00000351, IRB00001138

To: **Sarah Buchanan**

Date: **August 27, 2012**

Dear Researcher:

On 8/27/2012 the IRB approved the following human participant research until 08/26/2013 inclusive:

Type of Review:	Submission Response for UCF Initial Review Submission Form Expedited Review Category # 7 *This approval includes a Waiver of Written Documentation of Consent*
Project Title:	Multi-Touch Gesture Collection
Investigator:	Sarah Buchanan
IRB Number:	SBE-12-08599
Funding Agency:	JHT, Inc.(JHT)
Grant Title:	
Research ID:	N/A

The Continuing Review Application must be submitted 30days prior to the expiration date for studies that were previously expedited, and 60 days prior to the expiration date for research that was previously reviewed at a convened meeting. Do not make changes to the study (i.e., protocol, methodology, consent form, personnel, site, etc.) before obtaining IRB approval. A Modification Form **cannot** be used to extend the approval period of a study. All forms may be completed and submitted online at https://iris.research.ucf.edu .

If continuing review approval is not granted before the expiration date of 08/26/2013, approval of this research expires on that date. When you have completed your research, please submit a Study Closure request in iRIS so that IRB records will be accurate.

Use of the approved, stamped consent document(s) is required. The new form supersedes all previous versions, which are now invalid for further use. Only approved investigators (or other approved key study personnel) may solicit consent for research participation. Participants or their representatives must receive a copy of the consent form(s).

In the conduct of this research, you are responsible to follow the requirements of the Investigator Manual.

On behalf of Sophia Dziegielewski, Ph.D., L.C.S.W., UCF IRB Chair, this letter is signed by:

IRB Coordinator

University of Central Florida Institutional Review Board
Office of Research & Commercialization
12201 Research Parkway, Suite 501
Orlando, Florida 32826-3246
Telephone: 407-823-2901 or 407-882-2276
www.research.ucf.edu/compliance/irb.html

Approval of Human Research

From: **UCF Institutional Review Board #1**
FWA00000351, IRB00001138

To: **Sarah Buchanan**

Date: **December 10, 2014**

Dear Researcher:

On 12/9/2014, the IRB approved the following human participant research until 12/08/2015 inclusive:

Type of Review:	IRB Continuing Review Application Form
Project Title:	Multi-touch Gesture Evaluation
Investigator:	Sarah Buchanan
IRB Number:	SBE-13-09836
Funding Agency:	JHT, Inc.(JHT)
Grant Title:	
Research ID:	16408195

The scientific merit of the research was considered during the IRB review. The Continuing Review Application must be submitted 30days prior to the expiration date for studies that were previously expedited, and 60 days prior to the expiration date for research that was previously reviewed at a convened meeting. Do not make changes to the study (i.e., protocol, methodology, consent form, personnel, site, etc.) before obtaining IRB approval. A Modification Form **cannot** be used to extend the approval period of a study. All forms may be completed and submitted online at https://iris.research.ucf.edu .

If continuing review approval is not granted before the expiration date of 12/08/2015, approval of this research expires on that date. When you have completed your research, please submit a Study Closure request in iRIS so that IRB records will be accurate.

Use of the approved, stamped consent document(s) is required. The new form supersedes all previous versions, which are now invalid for further use. Only approved investigators (or other approved key study personnel) may solicit consent for research participation. Participants or their representatives must receive a copy of the consent form(s).

All data, including signed consent forms if applicable, must be retained and secured per protocol for a minimum of five years (six if HIPAA applies) past the completion of this research. Any links to the identification of participants should be maintained and secured per protocol. Additional requirements may be imposed by your funding agency, your department, or other entities. Access to data is limited to authorized individuals listed as key study personnel.

In the conduct of this research, you are responsible to follow the requirements of the Investigator Manual.

On behalf of Sophia Dziegielewski, Ph.D., L.C.S.W., UCF IRB Chair, this letter is signed by:

Joanne Muratori

Signature applied by Joanne Muratori on 12/10/2014 05:02:06 PM EST

IRB manager

University of Central Florida Institutional Review Board
Office of Research & Commercialization
12201 Research Parkway, Suite 501
Orlando, Florida 32826-3246
Telephone: 407-823-2901 or 407-882-2276
www.research.ucf.edu/compliance/irb.html

Approval of Human Research

From: **UCF Institutional Review Board #1**
FWA00000351, IRB00001138

To: **Sarah Buchanan**

Date: **May 17, 2016**

Dear Researcher:

On 05/17/2016, the IRB approved the following human participant research until 05/16/2017 inclusive:

Type of Review:	UCF Initial Review Submission Form
Project Title:	Multi-touch Procedural Training
Investigator:	Sarah Buchanan
IRB Number:	SBE-16-12142
Funding Agency:	JHT, Inc.(JHT)
Grant Title:	
Research ID:	N/A

The scientific merit of the research was considered during the IRB review. The Continuing Review Application must be submitted 30days prior to the expiration date for studies that were previously expedited, and 60 days prior to the expiration date for research that was previously reviewed at a convened meeting. Do not make changes to the study (i.e., protocol, methodology, consent form, personnel, site, etc.) before obtaining IRB approval. A Modification Form **cannot** be used to extend the approval period of a study. All forms may be completed and submitted online at https://iris.research.ucf.edu .

If continuing review approval is not granted before the expiration date of 05/16/2017, approval of this research expires on that date. When you have completed your research, please submit a Study Closure request in iRIS so that IRB records will be accurate.

Use of the approved, stamped consent document(s) is required. The new form supersedes all previous versions, which are now invalid for further use. Only approved investigators (or other approved key study personnel) may solicit consent for research participation. Participants or their representatives must receive a copy of the consent form(s).

All data, including signed consent forms if applicable, must be retained and secured per protocol for a minimum of five years (six if HIPAA applies) past the completion of this research. Any links to the identification of participants should be maintained and secured per protocol. Additional requirements may be imposed by your funding agency, your department, or other entities. Access to data is limited to authorized individuals listed as key study personnel.

In the conduct of this research, you are responsible to follow the requirements of the Investigator Manual.

On behalf of Sophia Dziegielewski, Ph.D., L.C.S.W., UCF IRB Chair, this letter is signed by:

Joanne Muratori

Signature applied by Joanne Muratori on 05/17/2016 03:45:11 PM EDT

IRB Manager

LIST OF REFERENCES

[1] Joseph J LaViola Jr. A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin*, 32(1):47–56, 2000.

[2] Sarah Buchanan, Bourke Floyd, Will Holderness, and Joseph J LaViola. Towards user-defined multi-touch gestures for 3d objects. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*, pages 231–240. ACM, 2013.

[3] Sarah Holderness, Jared Bott, Pamela Wisniewski, and Joseph J LaViola. Exploring multi-touch contact size for z-axis movement in 3d environments. In *Proceedings of Graphics Interface 2017*. ACM, in press.

[4] Sarah Buchanan, Jared Bott, and Joseph J LaViola Jr. The influence of multi-touch interaction on procedural training. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*, pages 5–14. ACM, 2015.

[5] Michael L Coleman. Text editing on a graphic display device using hand-drawn proofreader's symbols. In *Pertinent Concepts in Computer Graphics, Proceedings of the Second University of Illinois Conference on Computer Graphics*, pages 283–290, 1969.

[6] J Rhyne. Dialogue management for gestural interfaces. *SIGGRAPH Comput. Graph.*, 21(2): 137–142, April 1987. ISSN 0097-8930.

[7] William Buxton, Richard Sniderman, William Reeves, Sanand Patel, and Ronald Baecker. The evolution of the sssp score editing tools. *Computer Music Journal*, pages 14–60, 1979.

[8] Dean Rubine. Specifying gestures by example. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '91, pages 329–337, New York, NY, USA, 1991. ACM. ISBN 0-89791-436-8.

[9] Lisa Anthony and Jacob O Wobbrock. $n-protractor: A fast and accurate multistroke recognizer. In *Proceedings of Graphics Interface 2012*, pages 117–120. Canadian Information Processing Society, 2012.

[10] Brandon Paulson and Tracy Hammond. Paleosketch: accurate primitive sketch recognition and beautification. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 1–10. ACM, 2008.

[11] Derek Anderson, Craig Bailey, and Marjorie Skubic. Hidden markov model symbol recognition for sketch-based interfaces. In *AAAI fall symposium*, pages 15–21, 2004.

[12] George Lucchese, Martin Field, Jimmy Ho, Ricardo Gutierrez-Osuna, and Tracy Hammond. Gesturecommander: Continuous touch-based gesture prediction. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 1925–1930, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1016-1.

[13] Amy Ingram, Xiaoyu Wang, and William Ribarsky. Towards the establishment of a framework for intuitive multi-touch interaction design. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 66–73. ACM, 2012.

[14] Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han. A screen-space formulation for 2d and 3d direct manipulation. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, UIST '09, pages 69–78, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-745-5.

[15] Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. Shallow-depth 3d interaction: design and evaluation of one-, two- and three-touch techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1147–1156, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9.

[16] Andrew D. Wilson, Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. Bringing physics to the surface. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 67–76, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-975-3.

[17] Andrew D. Wilson. Simulating grasping behavior on an imaging interactive surface. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 125–132, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2.

[18] Aurélie Cohé, Fabrice Dècle, and Martin Hachet. tbox: a 3d transformation widget designed for touch-screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 3005–3008, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9.

[19] Mark Hancock, Thomas ten Cate, and Sheelagh Carpendale. Sticky tools: full 6dof force-based interaction for multi-touch tables. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 133–140, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2.

[20] Jaime Ruiz, Yang Li, and Edward Lank. User-defined motion gestures for mobile interaction. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 197–206. ACM, 2011.

[21] Christian Kray, Daniel Nesbitt, John Dawson, and Michael Rohs. User-defined gestures for connecting mobile phones, public displays, and tabletops. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, pages 239–248. ACM, 2010.

[22] Ekaterina Kurdyukova, Matthias Redlin, and Elisabeth André. Studying user-defined ipad gestures for interaction in multi-display environment. In *Proceedings of the 2012 ACM in-*

ternational conference on Intelligent User Interfaces, IUI '12, pages 93–96, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1048-2.

[23] Nicolai Marquardt, Ricardo Jota, Saul Greenberg, and Joaquim A Jorge. The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *IFIP Conference on Human-Computer Interaction*, pages 461–476. Springer, 2011.

[24] Andrew D Wilson and Hrvoje Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 273–282. ACM, 2010.

[25] Bruno R De Araújo, Géry Casiez, Joaquim A Jorge, and Martin Hachet. Mockup builder: 3d modeling on and above the surface. *Computers & Graphics*, 37(3):165–178, 2013.

[26] Sundar Murugappan, Vinayak, Niklas Elmqvist, and Karthik Ramani. Extended multitouch: Recovering touch posture and differentiating users using a depth camera. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 487–496, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1580-7.

[27] Peng Song, Xiaoqi Yan, Wooi Boon Goh, Alex Qiang Chen, and Chi-Wing Fu. Hand-posture-augmented multitouch interactions for exploratory visualization. In *SIGGRAPH ASIA 2016 Technical Briefs*, page 27. ACM, 2016.

[28] Martin Hachet, Benoit Bossavit, Aurélie Cohé, and Jean-Baptiste de la Rivière. Toucheo: multitouch and stereo combined in a seamless workspace. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 587–592. ACM, 2011.

[29] Sven Strothoff, Dimitar Valkov, and Klaus Hinrichs. Triangle cursor: Interactions with objects above the tabletop. In *Proceedings of the ACM International Conference on Interactive*

Tabletops and Surfaces, ITS '11, pages 111–119, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0871-7.

[30] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-defined gestures for surface computing. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, New York, NY, USA, 2009. ACM.

[31] Meredith Ringel Morris, Jacob O Wobbrock, and Andrew D Wilson. Understanding users' preferences for surface gestures. In *Proceedings of graphics interface 2010*, pages 261–268. Canadian Information Processing Society, 2010.

[32] Aurélie Cohé and Martin Hachet. Understanding user gestures for manipulating 3d objects from touchscreen inputs. In *Proceedings of Graphics Interface 2012*, GI '12, pages 157–164, Toronto, Ont., Canada, Canada, 2012. Canadian Information Processing Society. ISBN 978-1-4503-1420-6.

[33] Mark Micire, Munjal Desai, Amanda Courtemanche, Katherine M. Tsui, and Holly A. Yanco. Analysis of natural gestures for controlling robot teams on multi-touch tabletop surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, New York, NY, USA, 2009. ACM.

[34] Dan Mauney, Jonathan Howarth, Andrew Wirtanen, and Miranda Capra. Cultural similarities and differences in user-defined gestures for touchscreen user interfaces. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 4015–4020, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-930-5.

[35] Otmar Hilliges, David Kim, and S. Izadi. Creating malleable interactive surfaces using liquid displacement sensing. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, pages 157–160, Oct 2008.

[36] Christopher F Herot and Guy Weinzapfel. One-point touch input of vector information for computer displays. *ACM SIGGRAPH Computer Graphics*, 12(3):210–216, 1978.

[37] William Buxton, Ralph Hill, and Peter Rowley. Issues and techniques in touch-sensitive tablet input. *ACM SIGGRAPH Computer Graphics*, 19(3):215–224, 1985.

[38] Mandayam A Srinivasan and Jyh-shing Chen. Human performance in controlling normal forces of contact with rigid objects. *ASME DYN SYST CONTROL DIV PUBL DSC, ASME, NEW YORK, NY,(USA), 1993,*, 49:119–125, 1993.

[39] Sachi Mizobuchi, Shinya Terasaki, Turo Keski-Jaskari, Jari Nousiainen, Matti Ryynanen, and Miika Silfverberg. Making an impression: force-controlled pen input for handheld devices. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1661–1664. ACM, 2005.

[40] Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. Pressure widgets. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 487–494. ACM, 2004.

[41] Stephen A Brewster and Michael Hughes. Pressure-based text entry for mobile devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, page 9. ACM, 2009.

[42] Takashi Miyaki and Jun Rekimoto. Graspzoom: zooming and scrolling control model for single-handed mobile interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, page 11. ACM, 2009.

[43] Jared Cechanowicz, Pourang Irani, and Sriram Subramanian. Augmenting the mouse with pressure sensitive input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1385–1394. ACM, 2007.

[44] Sebastian Boring, David Ledo, Xiang'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. The fat thumb: using the thumb's contact size for single-handed mobile interaction. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pages 39–48. ACM, 2012.

[45] Apple iOS. ios maps, 2015. URL `http://www.apple.com/ios/maps/`.

[46] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. Pseudo-pressure detection and its use in predictive text entry on touchscreens. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, pages 383–392. ACM, 2013.

[47] Sungjae Hwang, Andrea Bianchi, and Kwang-yun Wohn. Vibpress: Estimating pressure input using vibration absorption on mobile devices. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 31–34, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2273-7.

[48] Seongkook Heo and Geehyuk Lee. Forcetap: extending the input vocabulary of mobile touch screens by adding tap gestures. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 113–122. ACM, 2011.

[49] Hrvoje Benko, Andrew D Wilson, and Patrick Baudisch. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1263–1272. ACM, 2006.

[50] Anne Roudaut, Eric Lecolinet, and Yves Guiard. Microrolls: expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 927–936. ACM, 2009.

[51] David Bonnet, Caroline Appert, and Michel Beaudouin-Lafon. Extending the vocabulary of touch events with thumbrock. In *Proceedings of Graphics Interface 2013*, GI '13, pages 221–228, Toronto, Ont., Canada, Canada, 2013. Canadian Information Processing Society. ISBN 978-1-4822-1680-6.

[52] Seongkook Heo and Geehyuk Lee. Indirect shear force estimation for multi-point shear force operations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 281–284, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0.

[53] Bertrand Schneider. Designing tabletop activities for inquiry-based learning: Lessons from phylogenetics, neuroscience and logistics. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 289–294, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1209-7.

[54] Orit Shaer, Megan Strait, Consuelo Valdes, Taili Feng, Michael Lintz, and Heidi Wang. Enhancing genomic learning through tabletop interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2817–2826. ACM, 2011.

[55] Bettina Conradi, Verena Lerch, Martin Hommer, Robert Kowalski, Ioanna Vletsou, and Heinrich Hussmann. Flow of electrons: an augmented workspace for learning physical computing experientially. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 182–191. ACM, 2011.

[56] Anne Marie Piper and James D Hollan. Tabletop displays for small group study: affordances of paper and digital materials. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1227–1236. ACM, 2009.

[57] Christophe Bortolaso, Matthew Oskamp, T.C. Nicholas Graham, and Doug Brown. Ormis: A tabletop interface for simulation-based training. In *Proceedings of the 2013 ACM Inter-*

national Conference on Interactive Tabletops and Surfaces, ITS '13, pages 145–154, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3.

[58] Ulrich von Zadow, Sandra Buron, Tina Harms, Florian Behringer, Kai Sostmann, and Raimund Dachselt. Simmed: combining simulation and interactive tabletops for medical education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1469–1478. ACM, 2013.

[59] Florian Gosselin, Sylvain Bouchigny, Christine Mgard, Farid Taha, Pascal Delcampe, and Cdric dHauthuille. Haptic systems for training sensorimotor skills: A use case in surgery. *Robotics and Autonomous Systems*, 61(4):380 – 389, 2013. ISSN 0921-8890. Models and Technologies for Multi-modal Skill Training.

[60] M. Kolesnikov, M. Zefran, A.D. Steinberg, and P.G. Bashook. Periosim: Haptic virtual reality simulator for sensorimotor skill acquisition in dentistry. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 689–694, May 2009.

[61] Sung Gu Kang, Byung Ju Ryu, Kyung Sook Yang, Young Hwii Ko, Seok Cho, Seok Ho Kang, Vipul R. Patel, and Jun Cheon. An effective repetitive training schedule to achieve skill proficiency using a novel robotic virtual reality simulator. *Journal of Surgical Education*, 72 (3):369 – 376, 2015. ISSN 1931-7204.

[62] Franck Ganier, Charlotte Hoareau, and Jacques Tisseau. Evaluation of procedural learning transfer from a virtual environment to a real situation: a case study on tank maintenance training. *Ergonomics*, 57(6):828–843, 2014.

[63] Yaiza Vélaz, Jorge Rodríguez Arce, Teresa Gutiérrez, Alberto Lozano-Rodero, and Angel Suescun. The influence of interaction technology on the learning of assembly tasks using virtual reality. *Journal of Computing and Information Science in Engineering*, 14(4):041007, 2014.

[64] S.J. Henderson and Steven K. Feiner. Augmented reality in the psychomotor phase of a procedural task. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 191–200, Oct 2011.

[65] Doug A Bowman, Ernst Kruijff, Joseph J LaViola Jr, and Ivan Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley, 2004.

[66] DL Holton. Constructivism+ embodied cognition= enactivism: theoretical and practical implications for conceptual change. *Recuperado a partir de http://bit. ly/gVAOOu*, 2010.

[67] Andy Cockburn, Per Ola Kristensson, Jason Alexander, and Shumin Zhai. Hard lessons: effort-inducing interfaces benefit spatial learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1571–1580. ACM, 2007.

[68] Dimitris Grammenos. Game over: learning by dying. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1443–1452. ACM, 2008.

[69] Andy Cockburn and Bruce McKenzie. Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 203–210. ACM, 2002.

[70] Alberto Betella, Enrique Martínez Bueno, Ulysses Bernardet, and Paul FMJ Verschure. The effect of guided and free navigation on spatial memory in mixed reality. In *Proceedings of the Virtual Reality International Conference: Laval Virtual*, page 7. ACM, 2013.

[71] Deborah Richards and Meredith Taylor. A comparison of learning gains when using a 2d simulation tool versus a 3d virtual world: An experiment to find the right representation involving the marginal value theorem. *Computers & Education*, 86:157–171, 2015.

[72] Wolfgang Stuerzlinger and Chadwick A Wingrave. The value of constraints for 3d user interfaces. In *Virtual Realities*, pages 203–223. Springer, 2011.

[73] HH Kaufman, RL Wiegand, and RH Tunick. Teaching surgeons to operateprinciples of psychomotor skills training. *Acta neurochirurgica*, 87(1-2):1–7, 1987.

[74] Camille K Williams and Heather Carnahan. Motor learning perspectives on haptic training for the upper extremities. *Haptics, IEEE Transactions on*, 7(2):240–250, 2014.

[75] MC Cirstea and MF Levin. Improvement of arm movement patterns and endpoint control depends on type of feedback during practice in stroke survivors. *Neurorehabilitation and Neural Repair*, 2007.

[76] C. Muller-Tomfelde, A. Wessels, and C. Schremmer. Tilted tabletops: In between horizontal and vertical workspaces. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, pages 49–56, Oct 2008.

[77] Microsoft Windows Dev Center. MS Windows touchinput structure, 2015. URL `https://msdn.microsoft.com/en-us/library/windows/desktop/dd317334(v=vs.85).aspx`.

[78] Dustin Freeman, Hrvoje Benko, Meredith Ringel Morris, and Daniel Wigdor. Shadowguides: visualizations for in-situ learning of multi-touch and whole-hand gestures. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 165–172. ACM, 2009.

[79] Olivier Bau and Wendy E Mackay. Octopocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 37–46. ACM, 2008.