AN ABSTRACT OF THE THESIS OF

Dale J. Cox for the degree of Master of Science in Computer Science presented on May 30, 2012.

Title: An Evaluation of Game Controllers and Tablets as Controllers for Interactive TV Applications.

Abstract approved:

____

Carlos Jensen

There is a growing interest in bringing online and streaming content to the television. Gaming platforms such as the PS3, Xbox 360 and Wii are at the center of this digital convergence; platforms for accessing new media services. This presents a number of interface challenges, as controllers designed for gaming have to be adapted to accessing online content. We conducted a user study examining the limitations and affordances of novel game controllers in an interactive TV (iTV) context and compared them to "second display" approaches using tablets. We looked at task completion times, accuracy and user satisfaction across a number of tasks and found that the Wiimote is most liked and performed best in almost all tasks. Participants found the Kinect difficult to use, which led to slow performance and high error rates. We discuss challenges and opportunities for the future convergence of game consoles and iTV. We also analyzed the usability of the interfaces themselves with respect to each device. Accuracy ratings and context of task type were used to determine ideal component attributes such as button size and spacing. Additional strategies like snapping cursor to buttons in the case of small targets were also suggested. Paying attention to the strengths and weaknesses of each input method, we put forth a set of design recommendations for future iTV interfaces that leverage novel input devices.

An Evaluation of Game Controllers and Tablets as Controllers for Interactive TV Applications

by
Dale J. Cox

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented May 30, 2012
Commencement June 2012

Master of Science thesis of Dale J. Cox presented on May 30, 2012

APPROVED:

Major Professor, representing Electrical and Computer Engineering

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Dale J. Cox, Author

ACKNOWLEDGEMENTS

During my time at Oregon State University, I was able to take many opportunities to push myself past what I thought my limits were. This would not have been possible without the help of many people. The following is a partial list of those people; there is nowhere near enough space to list them all.

I would like to express my sincerest gratitude to the OSU EECS faculty, staff and students who helped me in this process, especially my advisor Dr. Carlos Jensen. Dr. Jensen’s patience, helpfulness and experience were invaluable in guiding me through my graduate work.

I would also like to thank my committee members Dr. Mike Bailey, Dr. Ron Metoyer and Dr. Len Coop for dedicating their time to reviewing my work and attending my defense.

During the research and writing process, I learned much from my colleague Justin Wolford, who provided impressive technical ability, many laughs and good times on the racquetball court. Many thanks go out to Dedrie Beardsley, who helped me escape to sunny weather with her superior road warrior skills. She also dedicated large amounts of time and energy towards making our research study successful. I am very grateful to the Human Computer Interaction group as a whole, for their continual support, feedback and camaraderie.

Last but certainly not least, my friends and family who were always there with love and support during good and bad times, both personally and academically. My mom Cindy, brother Bryce and friends Corey & Kat, Robert & Rebecca, Brian, Luke and Karl kept me grounded through many challenging times.

CONTRIBUTION OF AUTHORS

Justin Wolford was primary author of the second manuscript and performed the statistical analysis for both manuscripts. For the first manuscript, he authored the quantitative results section. He also developed the drag and drop application used as the first task in the research study. Finally, Justin contributed to designing and conducting the research study.

Dedrie Beardsley helped design and conduct the research study, as well as analyzing the data collected during the experiment.

I was primary author of the first manuscript. I researched and authored the related work sections for both manuscripts, as well as performed qualitative analysis. I developed the applications to interface with the Microsoft Kinect and Nintendo Wii Remote used during the research study. I also contributed to designing and conducting the research study.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

1. Introduction

Interactive television has been in development for decades. It has a history of high expectations and underwhelming delivery that has resulted in many “false dawns” [1]. Broadcast television has remained largely unchanged with incremental interactive features being introduced, as in the case of On-Demand video, Digital Video Recorder (DVR) or an interactive channel guide. Chorianopoulos describes broadcast television as a produce-distribute-consume model compared to a create-share-control model interactive television [2]. The key concept being the user has a communication channel back to the service provider. This transfer of control has revolutionary potential in the world of television.

In recent years, broadband Internet has increased in popularity in the United States, just 9% in 2001 has jumped to 64% of US households subscribing to broadband service by 2009 [3]. Along with dramatically increased bandwidth, broadband also offers an always-on connection. These two attributes, created an infrastructure well suited for real-time or high bandwidth applications and services such as Netflix, Twitter and Facebook. These types of applications were not previously possible in the mostly dial-up world of the Internet.

Given the explosion of popularity for services like Netflix (21 million in 2010 [4]) and Hulu, combined with the increasing computing power of game consoles and televisions, a digital convergence began taking place in the entertainment world. Now televisions and game systems are connected to the Internet and rather than simply having access to content from a single cable provider, many Internet media services are now available. Interactive television attempts to integrate these media sources into a single point of entry. This diversity of services now available in our living rooms poses a set of interesting and challenging usability issues.

We performed a user study, which replicated a living room environment along with an iTV-like system. Our goal was to evaluate the performance of several non-traditional input devices as a navigation device. The devices used were the Microsoft Kinect, the Nintendo Wii Remote (Wiimote) and an Android tablet with 2 separate methods. The Kinect was developed for the Xbox 360 to allow free-hand spatial gesture recognition as an input method and was later adapted for general use. The Wiimote also uses spatial gestures but requires a controller to be held. The two tablet methods were a relative method, which was similar to a mouse touchpad and a mirror method, which “mirrored” the current television image allowing users to directly manipulate on a “second screen”. The Wiimote and relative input methods had out-of-the-box software available that satisfied our requirements. The Wiimote used an application called GlovePie to allow users to point the controller at the screen to position the cursor and use one of its physical buttons to click and drag. The relative method used an application called RemoteDroid to essentially turn the tablet into a large touchpad. For the Kinect and mirror methods, we were required to develop custom applications to interface with each device.

These applications were one of my major contributions to the project. The Kinect application went through several revisions. At first, I attempted to develop a single-handed solution using finger tracking based on an edge-following algorithm combined with convex hull detection. This method failed to produce a robust solution due to the poor resolution of the Kinect’s depth camera (640x480). The final version used a two handed approach which simply did hand tracking rather than having to follow fingers. The right hand was used to manipulate the cursor position, which followed the movement of the hand. The abstraction of an invisible vertical plane in front of the user was used to detect state changes for button clicking. When the left hand was moved quickly through the plane and back out, a click would be registered at the current location of the right hand. If the left hand were put through the plane, then lingered, a mouse-down would be registered signaling a drag operation. The right hand was then moved to the desired position and when the left hand was moved back out of the plane, a drop was registered at the current location of the right hand. This implementation worked fairly well in very controlled conditions. Any foreign objects in the sensor’s field of view raised the risk of false positives.

I created the mirror application by significantly modifying the open source project RemoteDroid, used for the relative method. A host application would run on the PC and a client application on the tablet. These two would talk via wireless network. The two major modifications were the introduction of an absolute coordinate system and a data stream, which continually updated the tablet display with the TV contents. The absolute coordinate system is similar to that used on modern smartphones and tablet interfaces. If a user pushed their finger at a position on the tablet, a click would be registered at that equivalent location on the TV screen. The host application would continually take a screen capture of the TV screen contents and send it to the tablet where it would be displayed.

We used these novel devices to perform a series of tasks representative of the input demands of modern interactive TV interfaces. The first task used a drag and drop application developed by Justin Wolford. A widget and target would appear in random locations. Users were required to drag the widget onto the target within 16 seconds. If a successful drop was recorded, the size of the target would shrink and the process was repeated. This task was an excellent measure of both speed and accuracy.

The final three tasks were navigation-based tasks to simulate a real-world environment. We chose to use the Xbox Media Center (XBMC) application because it represents a modern iTV-like interface. The first task required users to navigate to the weather settings screen and enter a new city. This may have been the most challenging task due to the text entry portion. The second was simply changing the selected weather city to the one entered in the previous task. Finally, participants would select a video and adjust a volume control slider which was the smallest target in all tasks.

We measured completion times and error rates, as well as taking pre and post experiment surveys. Qualitative data was also gathered during a post-experiment interview.

We performed a variety of statistical analysis techniques on the results data. Welch’s T test was used when comparing two quantities such as accuracy with respect to the outer edge of the screen versus the inner portion. When more than 2 quantities were examined, a one-way ANOVA test was used along with Tukey’s HSD for post-hoc analysis. An example of this is when looking for significant differences in device performance. When analyzing the percentage of text entry errors as displayed in figure 11, we transformed the raw data using the following equation: (1 - ((ideal clicks)/((ideal clicks)+(missed clicks))). One-way ANOVA and Tukey's HSD were then used on the results.

The first manuscript examines the performance of modern game controllers and tablets as input devices for an iTV-like interface. I performed an extensive literature review to find prior work in this emerging area, which I detail in the related work section. One of the goals was to evaluate devices other than a remote control or mouse and keyboard. In particular we looked at quantitative data from completion times and found interesting results when people try again after they miss the target in what we call “nudging”. Justin Wolford performed the statistical analysis for the results section. I performed a qualitative analysis of responses during interviews and surveys gathering device ratings and common statements.

The second manuscript attempts to describe a set of interface design guidelines based on device performance relative to target size and target screen position. I also performed the literature review for this manuscript and authored the related work section and much of the introduction. In particular, the research regarding “lean-back” viewing and PC vs TV considerations was very relevant to our topic. Justin Wolford also performed the statistical analysis in the results section for this manuscript. Recommendations such as minimum button size or introduction of “magnetic” button techniques were suggested for each device/interface combination.

Each manuscript looks at iTV interaction in a different way. The first manuscript focuses on devices with respect to the interface and the second focuses on the interface with respect to the devices. Both work toward a common goal of providing insight into the use of alternative input devices in a modern interactive television context.

2. First Manuscript: An Evaluation of Game Controllers and Tablets as Controllers for Interactive TV Applications

Dale Cox, Justin Wolford, Carlos Jensen, Dedrie Beardsley

School of EECS

Oregon State University

Corvallis, Oregon, 97331, USA

{ cox, cjensen } @ eecs.oregonstate.edu

{ wolfordj, beardsld } @ onid.orst.edu

Proceedings of the 2011 International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction (ICMI), Santa Monica, CA, USA

(Submission in Progress)

2.1 ABSTRACT

There is a growing interest in bringing online and streaming content to the television. Gaming platforms such as the PS3, Xbox 360 and Wii are at the center of this digital convergence; platforms for accessing new media services. This presents a number of interface challenges, as controllers designed for gaming have to be adapted to accessing online content. This paper presents a user study examining the limitations and affordances of novel game controllers in an interactive TV (iTV) context and compares them to "second display" approaches using tablets. We look at task completion times, accuracy and user satisfaction across a number of tasks and find that the Wiimote is most liked and performed best in almost all tasks. Participants found the Kinect difficult to use, which led to slow performance and high error rates. We discuss challenges and opportunities for the future convergence of game consoles and iTV.

2.2 INTRODUCTION

Interactive television promises to give viewers more flexibility and control over their viewing experience, while enriching it with a wealth of Internet accessible content and information. By giving the viewer a communication channel back to service providers, viewers can not just access new services, but also shape and control their viewing experience in ways that were not possible before. This has so far led to the emergence of services such as video on demand (Netflix, Hulu, etc.),the presentation of Internet content on TV screens (YouTube & Flickr channels on Apple TV, etc.), but could also allow for content ratings, interactive or contextual searching or social networking (see Boxee.tv).

For trivial tasks like navigating a simple movie rental UI, or controlling streaming video content, a traditional remote control is often sufficient. For instance, the Apple TV remote control is among the simplest available, with 7 buttons. However, as services become more complex and rich, requiring more or finer levels of control or interaction – such as navigating a non-trivial web page or GUI, carrying out drag and drop tasks, or more extensive text entry for search or socialization – more sophisticated input devices may be required. Some of these tasks overlap with those integral to the modern gaming experience. Video game consoles, with dedicated game controllers, due to their pervasiveness, connectivity and processing power are often at the center of this digital convergence of TV and Internet content. It is therefore important to examine how suitable current systems are for bridging this gap. The last generation of game consoles have each introduced some device capable of spatial gestures allowing the possibility for a natural user interface (NUI), and which could be especially helpful for navigating complex UIs. Microsoft released the Kinect for the Xbox 360 in 2010, a camera based system that tracks players' movement to allow for complex and natural interactions without holding any kind of controller. Nintendo released the Wii in 2006, which introduced the Wii Remote (nicknamed Wiimote), a wireless controller that tracks spatial movement through accelerometers and infrared sensors. Sony has a similar system for the Playstation 3.

These game consoles and their controllers have sparked the development of tools and solutions beyond those in traditional gaming. The motion-driven Nintendo Wiimote was the first to attract the attention of the hacker community outside the console market. The Bluetooth interface of the Nintendo Wiimote made it a simple and accessible device to "hack" and adopt for various uses. Soon after the Kinect was released, the open source community reverse engineered the device and released a driver package allowing others to develop systems that took advantage of its capabilities. Its USB interface makes it ideal for use with a PC. Today, several different open source SDKs exist, as well as an official Microsoft Kinect SDK for Windows.

In addition, tablets continue to evolve and increase in popularity. Some proposed game controllers are exploring the use of such touch interfaces (most notably the upcoming Wii U). Tablets are also being considered as companions to both gaming devices and iTV services (often referred to as second screen navigation). According to a recent Pew study [7], tablet ownership nearly doubled over just the 2011 holiday season. There are a number of applications currently available that allow a tablet to serve as an input device for another computer such as IntoNow (http://www.intonow.com) that detects which show or movie you're watching and provides additional media content and social networking capabilities. These applications allow the tablet to function like a touch screen display or to track pad found on most laptops and would serve as a suitable baseline.

This paper explores the challenges and opportunities of using game-related control technologies to control interactive television applications through the study of a hypothetical, but representative set of navigation, selection, and control tasks for a iTV application. We examine learnability and ease of use, as well as accuracy and error rates.

The rest of this paper is laid out as follows. First we discuss related work looking at the control of interactive television applications as well as gaming systems. We then discuss our user experiment and the design considerations we took into account. Finally, we present our findings, and a discussion of future work.

2.3 Related Work

Much innovation has taken place in the design of new interaction techniques and devices for gaming devices. The research community is still catching up with the necessary evaluation of the potential, effectiveness and usability of these novel game input devices, both within their targeted use domain as well as in other environments like PC or interactive television.

Looking to the modern interactive TV interface, we see that it combines many types of user interaction. The areas we chose to focus on cover the core functionality of pointing, navigation, and text entry. Pointing is perhaps the more novel and difficult task with todays' hardware, but is a prerequisite for many of the more sophisticated types of applications and use cases. Individually, each of these topics has an established body of research, but in the context of a media center or iTV there is very little. We also present the recent adoption rate history of streaming services, which are at the center of modern interactive television systems.

2.3.1 Growing Popularity of Streaming Services

With the spread of broadband internet access throughout North America, high bandwidth services like high definition on-demand streaming video, previously limited in either quality or duration, have become commonplace. Between 2001 and 2009, broadband Internet use increased seven-fold, covering from 9% to 64% of American households [21]. A 2011 Nielsen study found that from 2008 to 2011 there was a 22% increase in the number of users watching video on the Internet, and an 80% increase in the average viewing time [19].

One of the key players in the Internet-based video-on-demand area has been Netflix. Netflix debuted as a DVD-by-mail service in 1997, and has since introduced and popularized a broadly available Internet streaming service. By the end of 2011, Netflix had over 21 million paying streaming subscribers [13]. In a Fall 2010 report by Sandvine, Netflix was shown to account for 20.6% of all downstream prime-time Internet traffic in North America [17]. Just 7 months later, Netflix users were consuming 29.7% of all downstream prime-time Internet traffic in North America [17].

Other online video services such as Hulu, Amazon Instant Video and YouTube (though the latter still mostly offers shorter clips, it has branched into feature content delivery as well) have also grown in popularity. Internationally, over 4 billion videos are viewed on YouTube each day [12]. Hulu just passed 1.5 million paying subscribers of its paid Plus service [18].

In part this success is driven by the growth of systems that help these users bridge the gap between the computer and the TV experience. This includes a plethora of streaming devices like the Roku and Apple TV, a new generation of connected TV's and DVD/Blue-ray players, and last but not least game consoles. Each of the three leading game consoles have added mechanisms for viewing streaming Netflix content on their devices. Services like Netflix and Hulu that began as a PC experiences, can now be accessed from a number of different devices and platforms. This has made enabled these services to go from a niche technophile market to appealing to the average consumer. In a 2011 Nielsen study, 50% of all Netflix users were found to watch Netflix content through a gaming console [16]. In the same study, Nielsen found 162 million Americans own a game console. This means that these platforms are natural ways to deliver these experiences. The need to manage users' media viewing experience has led to the development of media center applications like the Xbox Media Center (XBMC) and Windows Media Center. Internet-enabled set-top devices like Boxee and AppleTV have also appeared allowing easy streaming video viewing from a normal TV. These allow users to consolidate their media consumption, as well as manage their local library. Due to the interactive and highly customizable experience allowed by these services, the need for robust input methods will continue to gain importance.

2.3.2 Pointing and Navigation Using Novel UI Devices

Over the last few years there has been a growing trend to develop and evaluate what are being referred to as Natural User Interfaces (NUI's). These interfaces extend the basic direct manipulation paradigm by allowing users to interact with the computer with motions more closely resembling those we'd use in real life. Among the leading platforms for such interfaces we find game consoles. These techniques could help bridge the complexity gap between the new interactive TV applications and the interactions afforded by conventional remote controls. Because of space limitations we will only review some of the most directly applicable research to our study.

Starting with camera and motion based techniques, Cheng and Takatsuka [2] introduced dTouch, a finger pointing technique for large displays that uses an off-the-shelf webcam. Using the concept of a "virtual touchscreen", dTouch enables users to manipulate onscreen objects in an absolute coordinate system. They performed a user study comparing dTouch to a method using the EyeToy camera, used on the PlayStation console. Results indicated the two methods were comparable with users preferring dTouch.

Lee [3] described a cursor technique using the Wiimote that enabled finger-tracking through the use of reflective tags taped to the fingers of users. Rather than holding the Wiimote in the hand, they used the IR camera built into the Wiimote with an IR LED array to allow almost bare-hand operation. Lin et al. [4] demonstrated a technique similar to Lee's, but using a second Wiimote for additional functionality.

Using more traditional controllers, Natapov et al. [5] performed a comparative study evaluating the Wiimote and traditional gamepad for pointing and selecting tasks. Although the error rate was higher, 14 out of 15 participants said they preferred the Wiimote in a home entertainment environment. They found that the Wiimote had a 75% performance increase over the traditional gamepad when comparing speed and accuracy.

Finally, turning to smart phones and tablets, McCallum et al. [10] developed a hybrid system called ARC-Pad, which combined absolute and relative positioning techniques for use with large displays. A smart phone screen was used like a touchpad. ARC-Pad was compared against a traditional touchpad style interface, which employed cursor acceleration. ARC-Pad performed slightly better (166ms faster) than the relative in completion time. The results suggested as pixel distance increased beyond what was studied, ARC-Pad performance would change minimally while the relative touchpad would continue to worsen.

2.3.3 Text Entry With Keyboard Alternatives

Over the last two decades, the need for text entry without a traditional mechanical keyboard has increased. With the introduction of PDAs and smart phones, text entry presents a challenge due to a limited input area. Most interactive TV systems attempt to minimize the necessity of text entry through the use of various widgets and interface choices. Though the need may be reduced, it is difficult to completely do away with text entry for applications such as search or social media.

This has led TV manufacturers like Samsung to market 2-sided remote controls; one side having normal remote control functions and the reverse a full keyboard, or Sony to merge a PlayStation controller and a full keyboard in their Google TV products. While such solutions may provide speed advantages, they lead to cumbersome and intimidating user experiences. We examined alternatives to keyboard text entry, focusing on touchscreens, game controllers and freehand gesture techniques.

The Graffiti pen-based gesture alphabet was made popular by Palm in the late 90s. It allowed users to quickly input text using a proprietary alphabet. MacKenzie and Zhang [9] conducted a study analyzing the learnability and accuracy of Graffiti. Participants were given practice time using a reference chart showing the gesture alphabet. After practice they repeated the entire alphabet 5 times without having a reference available and again 1 week later. The results showed a nearly 97% character accuracy rate after 5 minutes of practice.

Tao, et al. [11] adapted the Graffiti alphabet to a freehand gesture-based text entry system called AirStroke. A user study was performed comparing two AirStroke implementations, one with word completion and one without. Participants completed 20 sessions each, over a period of two weeks in which error-rates and speed were recorded. Airstroke with word completion averaged 11 wpm while no word completion was at 6.5 wpm. The error rate with word completion averaged 6.6% compared to 11.8% without. Some participants initially reported arm fatigue, which lessened as their proficiency increased.

Several techniques have been developed enabling text input using a traditional gamepad. Költringer et al. [15] designed and evaluated TwoStick, a novel text entry system using both analog joysticks on an Xbox 360 controller. TwoStick was compared to a traditional selection keyboard. Initially, users typed slower and had a higher error rate using TwoStick, but after 15 sessions TwoStick averaged 14.87 wpm while the selection keyboard had a mean of 12.9 wpm. Wilson and Agrawala [14] also created a dual joystick QWERTY method, which showed modest improvement upon the traditional single stick selection keyboard.

Shoemaker et al. [8] compared 3 techniques for mid-air text input. A circle keyboard, QWERTY keyboard and cube keyboard all used a Wiimote as an input method. The QWERTY method performed best in accuracy and performance; this method is similar to our Wiimote text entry task. A questionnaire taken after the study revealed users preferred the QWERTY method overall.

Castellucci and MacKenzie [1] presented an alternative to an on-screen keyboard using the Wiimote called UniGest. UniGest is a technique that takes advantage of the motion-sensing capabilities of the controller to capture movement and rotation. A gesture alphabet is proposed which maps the gestures to character input. Their results predict an upper-bound of 27.9 wpm using the UniGest technique.

2.4 Methodology

This section describes a user study designed to measure the effectiveness of video game and tablet input methods in a iTV context. We used 4 input methods: the Microsoft Kinect, Nintendo Wiimote and 2 methods using an Android tablet; a condition where subjects had to scroll (relative coordinate condition), and one using an absolute coordinate space (mirror condition). The idea was that in the relative condition subjects would have to scroll around like when using a mouse pad, and in the absolute coordinate condition, the whole TV image would be represented on tablet at once. Participants completed pre and post-experiment questionnaires and also a post-experiment interview. All sessions were recorded using a video camera and screen capturing software.

All participants were recruited in pairs from a college campus and surrounding community. There were a total of 62 participants, 33 male and 29 female. All but 4 were right-handed. Their ages ranged from 18 to 57 years old with a mean of 24.5. Prior to the experiment, participants completed a questionnaire gathering demographic data and media viewing frequency. Each pair of participants was assigned 2 devices to use, and all device pair permutations were assigned randomly. The system ran on a PC hooked up to a 55" HDTV, set up in an environment designed to look and feel like a living room (see room layout in Figure 1). The subjects sat in the two center seats, while the experimenter sat off to the side with a good view of the subjects. The table in the middle was positioned far enough away that subjects could not use it to hold items while performing their task. There was a small table (15x15cm surface area) between the two chairs, large enough to hold a drink or a plate, but not both at the same time.

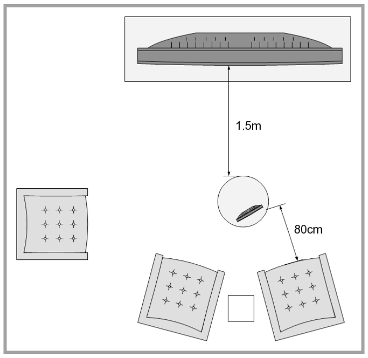

Figure 1: Room configuration used during the experiment.

A Nintendo wireless sensor bar was used in combination with a standard Nintendo Wii Remote for relevant conditions. A Microsoft Kinect sensor was used for the Kinect tasks, and a 10.1" tablet running Android 2.3 was used for the tablet tasks. A windows application called GlovePIE was used to control the cursor using the Wiimote. A GlovePIE script enabled the IR camera in the Wiimote to control the mouse cursor and the ‘A’ button to control the left mouse click.

A custom application was created to allow the Kinect to control the mouse cursor and left button. The application was written in C# using the OpenNI framework. To move the cursor, participants moved their right hand, which positioned the cursor similar to a traditional mouse. To initiate a drag, participants moved their left hand forward to cross the threshold of a virtual plane 30-40cm in front of them. This action is equivalent to a left mouse down event. To initiate a drop, participants would simply pull their arm back and break the plane in the opposite direction. This action is equivalent to a left mouse up event. To initiate a left click, participants move their left hand quickly through the plane and back out in one fluid motion.

The software used in the relative tablet condition was an open source Android application called RemoteDroid. This application turns the entire tablet into one large touchpad similar to what is found on most laptop computers (see Figure 2). This is application was paired with an application that runs on the host computer.

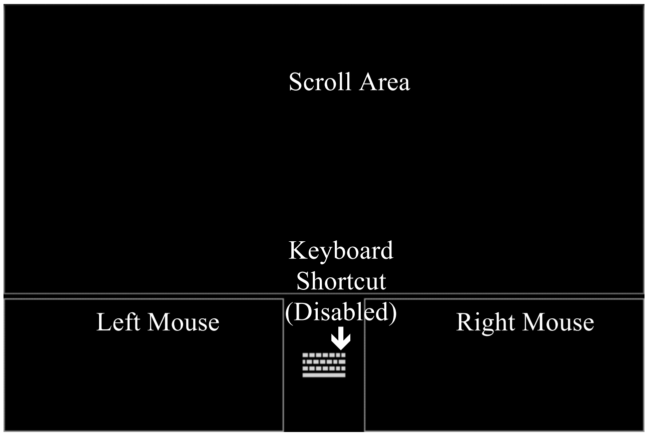

Figure 2: The configuration of the relative tablet app.

The mirrored condition used a modified version of the RemoteDroid application. It continually updates the tablet display with a screenshot of the TV. Instead of using a relative coordinate system where the cursor movement corresponds to relative changes in cursor position, an absolute coordinate system is used. By using an absolute coordinate system, a user can click on any location of the mirrored display and have it mapped to the equivalent location on the PC display.

Subjects were trained on the devices they were going to use and given time to practice on a screen that allowed dragging and dropping an object and clicking a button. When they felt comfortable with the device they began the drag-and-drop task.

The only actions allowed in the drag and drop task were dragging and dropping a widget into a target box (see Figure 3). If the widget was dropped fully within the boundary of a target presented at a random point on the screen, then a hit was recorded, and the subject would be presented with a new, slightly smaller target. A miss was recorded if the user missed the widget when attempting to select it, or if they released the widget outside of the target box. They were able to keep trying until they ran out of time for the trial. If the user was unable to place the widget in the target within 16 seconds, the box and widget were moved to random locations on the screen and the target box got bigger. If the user hit the target, then both the widget and target were randomly moved and the target shrank, with the minimum size for the target being 3 pixels wider and taller than the widget.

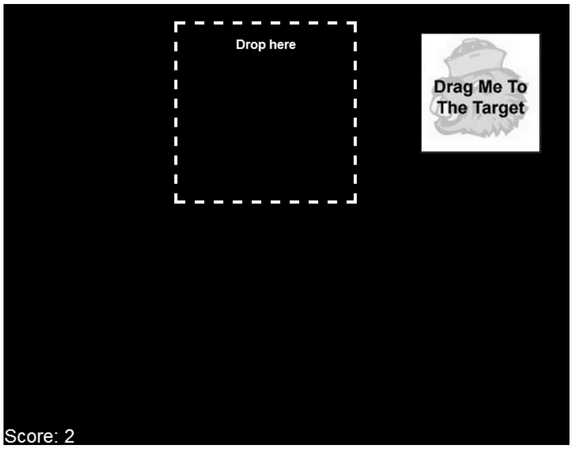

Figure 3: Sample screen for drag and drop task.

After completing the drag-and-drop accuracy task, we asked subjects to complete a number of navigation tasks on an iTV environment, simulated using the popular XBMC media center.

The navigation tasks included 3 different activities in XBMC. The first required the subject to navigate from the main menu to the weather settings screen and change the city name. This may have been the most challenging task due to the text entry requirement. Next, subjects would go to the weather screen and change the city currently displayed. This was difficult at times because it required clicking on a very small button. After changing the city, the subject would navigate to the movie selection screen and select a movie using a scrollbar. Finally, after the movie started, a slider was used to adjust movie volume. In all navigation tasks, an error was recorded if a subject clicked on a non-interactive item or if they clicked on the wrong UI widget.

We measured users’ time to complete tasks and their error rates. Additionally we collected qualitative data in a post experiment interview and survey. Finally we used screen capture software and a video camera to record subjects. Subjects performed the experiment in pairs, taking turns with each device (subject A would try device 1, then subject B would use the same device. Next Subject A would try device 2, and then Subject B would do the same). Pairs were randomly assigned two input methods. Because of previous research showing the importance of studying the effectiveness of UI techniques under similar manual loads [20], and the oft-informal nature of TV viewing, we decided to give each subject a slice of pizza and a drink to hold and consume during the course of the experimental tasks. Subjects were not allowed to place the food items on the floor or on the larger central table, but had to balance them on their seat or lap.

After both subjects completed all tasks using the first device assigned to them, they were introduced to the second device, and the process started anew, from the training period onward. After both subjects completed both conditions, they were asked to complete a short survey asking them about learnability, ease of use and practicality of the devices they had been assigned. All questions were on a 5-point Likert scale, 1 meaning strongly agree and 5 meaning strongly disagree.

Finally, they were interviewed to get a deeper understanding of their experience. We were interested in their satisfaction with the various input devices. This included the ease with which the subjects could use the device along with their enjoyment of using the device. Additionally we asked about their comfort level using the device in a social context where others were observing them.

2.5 Results

2.5.1 Drag and Drop Task

The main task we used to measure the efficiency of a UI technique for manipulation was the timed drag and drop task, as it combined selection, movement, as well as accuracy. The more drops a subject managed within the time allotted, the more accurate their manipulation of the widgets on the screen. The highest mean number of targets hit was with the mirror tablet, where subjects hit an average of 14.09 targets (see Figure 4). Subjects using the Wiimote and relative tablet scored 12.97 and 11.90 hits respectively. Those using the Kinect averaged a score of 7.37 hits. The Kinect did significantly worse than all other devices (One-way ANOVA $F(3,19)=54.5$, $P<0.001$

with Tukey's HSD for Post Hoc analysis). The relative tablet also did significantly worse than the Wiimote and mirror tablet ($P<0.05$). There was no significant difference between the Wiimote and mirror devices.

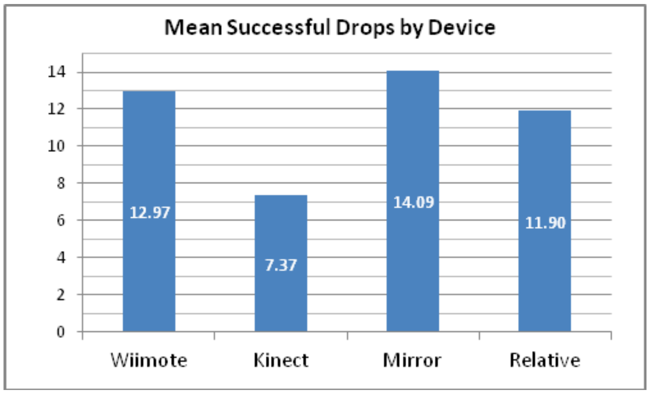

Figure 4: More drops show that the user was quicker and more accurate than users with fewer successful drops.

There is of course a direct relation between accuracy and speed in this task. The quicker the manipulation, the more likely you are to be able to complete the task, and even try multiple times in case of failure. Therefore an inaccurate but very quick technique could lead to misleading results. To investigate this we decided to look at the average target size for the last 5 targets subjects successfully hit. This allowed us to give subjects some additional practice time, and allowed subjects' performance to plateau. The results are shown in figure 5.

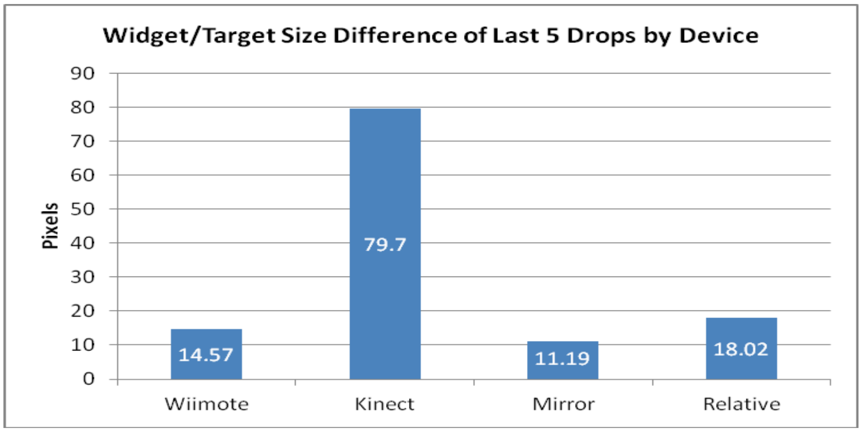

Figure 5: Average final margin of error in pixels by condition

Users were much less accurate with the Kinect than with any other device. Users of the mirror, relative and Wiimote conditions averaged a 13 to 19-pixel difference between the widget dimensions and the target dimensions. With the widget being 152px square, this meant a margin of error of less than ±9-12% of the widget size, or ±1-2% of the total screen real estate (1920x1080). For the Kinect the margin of error was ±45% of the widget, and ±6% of the total screen real estate. There were significant differences (One-way ANOVA $F(3,120)=13.77$, $P<0.001$ with Tukey's HSD for Post Hoc analysis), the Kinect was significantly different from the other devices ($P<0.001$). Other differences were not significant.

During the experiment and in the post experiment interview, several subjects mentioned that the sensitivity for the relative tablet was low and that they would have to slide their finger across the device more than once to get the cursor to traverse from one side of the screen to the other. Users needed to swipe 3 times to go from one side of the screen to the other. No enhancements such as cursor acceleration were implemented; this could potentially improve performance of the relative condition. This may in part explain why the relative tablet scored worse than the mirror and Wiimote. However, because speed and accuracy are often traded off against each other, it is not a given that acceleration would lead to better results. This is something that should be investigated more in-depth. Users were not able to move the cursor rapidly enough to hit the same number of targets.

The issues with using the Kinect were more pronounced and deep-rooted. Subjects were observed having a difficult time both beginning a drag (selecting and dragging the target) and dropping the widget into the target (widget would often be dropped prematurely and unintentionally). A less common but also real problem was that in order to establish a difference between a click event and a drag event users had to press forward and hold for 0.5 seconds before beginning the drag. It was common to see users attempt to drag before the drag event had been registered. They were told about this in the training but as the user began the trial and were trying to rush through the task, they would often not pause long enough. Some visual indicator to let them know that the event had been registered could have made a difference.

The more fundamental problem with the Kinect condition was the 2-handed operations. Subjects usually had little trouble placing a cursor over a target using one hand, though fatigue was mentioned as a concern in some trials. However, the action of bringing or removing the second hand from the camera plane often caused subjects to inadvertently rotate their bodies to retain balance, even while seated. This of course would make their targeting hand move, resulting in a missed target. This same phenomenon was observed time and again across tasks and subjects. The only effective remedy we saw was for subjects to plant their elbows in the seat, and use this to counter the natural body rotation action. Though effective, this led to a very restrictive seating position.

2.5.2 Nudging

There is a tradeoff between speed and accuracy, and with a sufficiently fast UI, users can home in on the target effectively. We referred to this behavior as “nudging”. In our experiment, this turned out to be a relatively common strategy; if a subject failed to hit the target on the first try, they would rethink their strategy (a longer pause) and then pick up the widget and home in through a series of rapid follow-up moves. This was especially common with the smaller targets, where the margin for error was low. The majority of times subject were able to hit the target in one or two attempts. However, some times it took longer.

Figure 6 shows how subjects using the Wiimote employed this strategy. Wiimote users were among the most successful and accurate, and the technique allowed for quick and easy nudging, or homing in on the target. As we can see, after a longer rethink following an initial miss, subjects engaged in a lot of rapid moves aimed at trying to hit the target. Subjects in this condition were still among the most accurate and successful. We see a very similar behavior among subjects using the mirror tablet application, though there is less of a long-tail (see Figure 7).

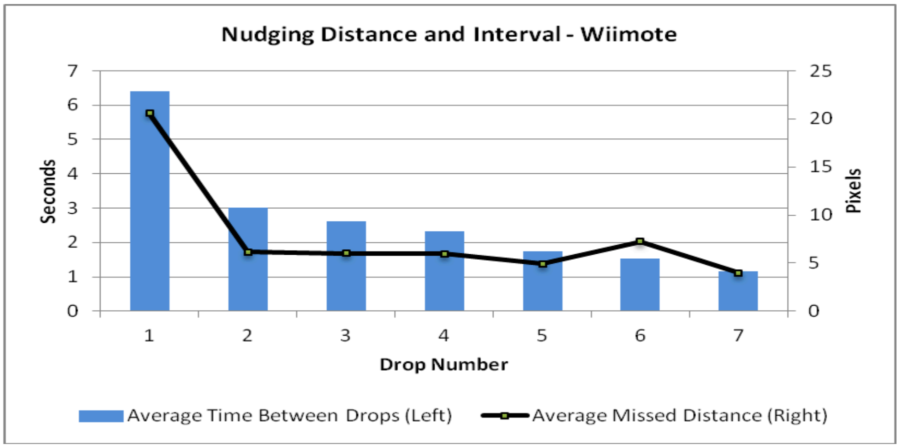

Figure 6: Nudging interval (left axis, blue columns, in seconds) and distance (black line, right axis, in pixels) over number of tries to hit one target – Wiimote.

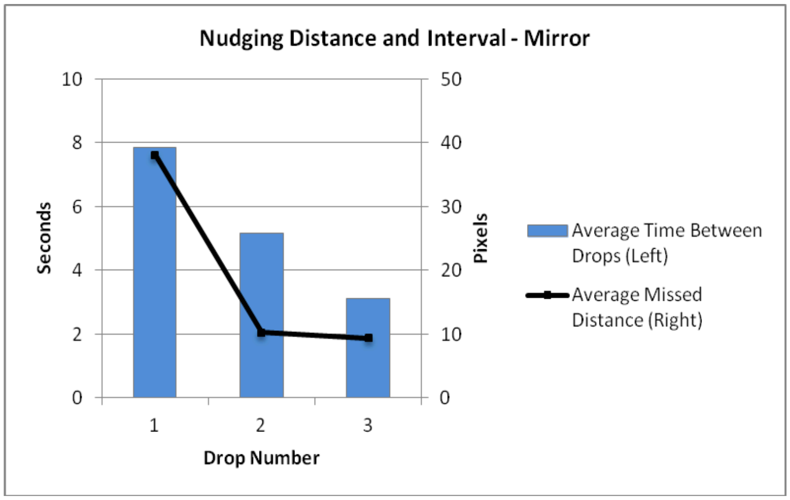

Figure 7: Nudging interval (left axis, blue columns, in seconds) and distance (black line, right axis, in pixels) over number of tries to hit one target – Mirror Tablet.

This strategy however seemed to be less successful, even counterproductive in the other two conditions (see Figure 8 and Figure 9). In the case of the Kinect condition, accuracy was an enormous issue, and though subjects were more successful with repeated tries, they did not home in on the target, but rather hit random new points. In the case of the relative tablet application, the nudging strategy appears to be counterproductive. Subjects would after the second try engage in very rapid moves that rather than take them closer to the target would distance them more. To us this is an important distinction between these two groups of techniques. Our subjects naturally gravitated to this strategy, and therefore it should be supported.

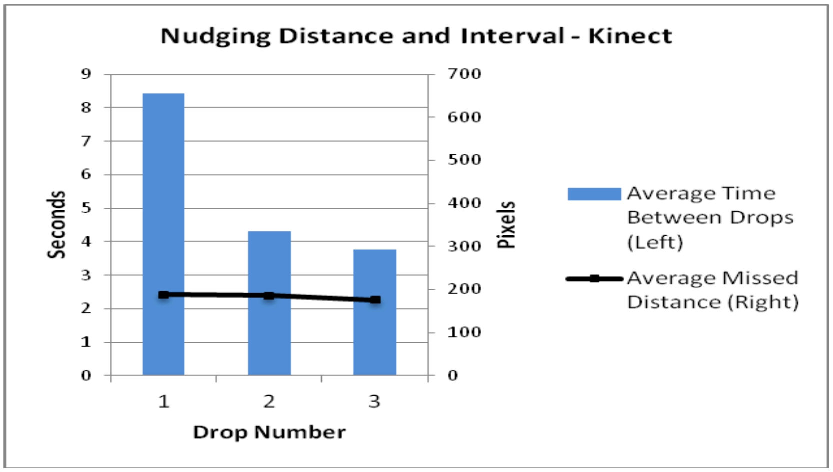

Figure 8: Nudging interval (left axis, blue columns, in seconds) and distance (black line, right axis, in pixels) over number of tries to hit one target – Kinect.

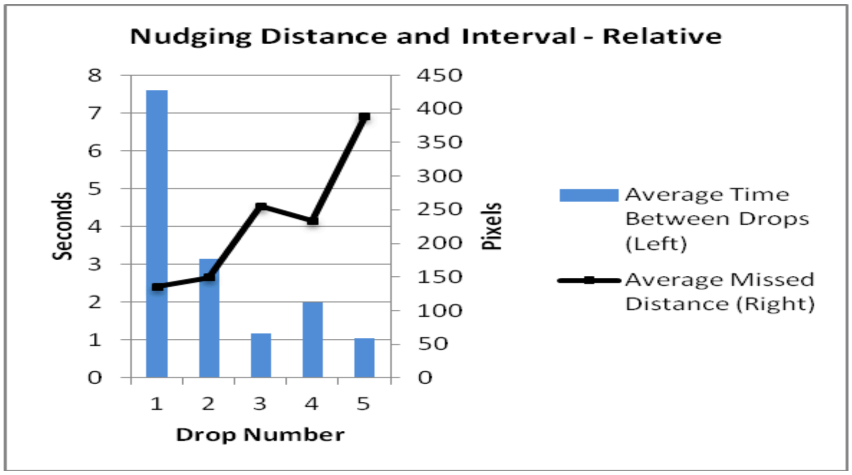

Figure 9: Nudging interval (left axis, blue columns, in seconds) and distance (black line, right axis, in pixels) over number of tries to hit one target – Relative Tablet.

2.5.3 Text Entry

An important task for iTV applications is text entry, as it allows more rapid customization, search, etc. We chose to examine two factors, the number of clicks that landed off of the intended target (key) and the time it took users to input the text string (a 9 character string).

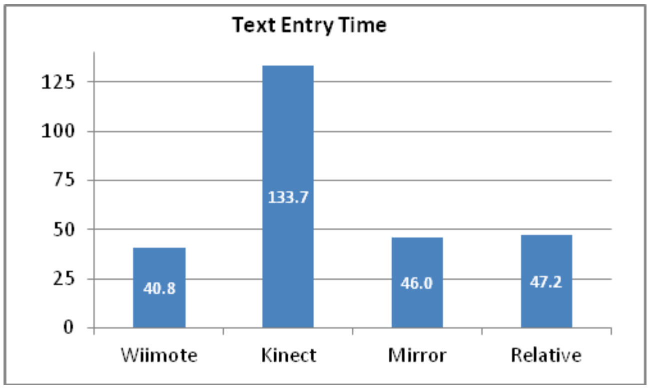

Figure 10: Average text entry time in seconds

As we see in Figure 10, text entry was significantly slower with the Kinect when compared to the other devices (One-way ANOVA F(3,115)=19.53 P<0.001 with Tukey's HSD for Post Hoc analysis). The Kinect was significantly slower than all other devices (P<0.001). There were no other significant differences.

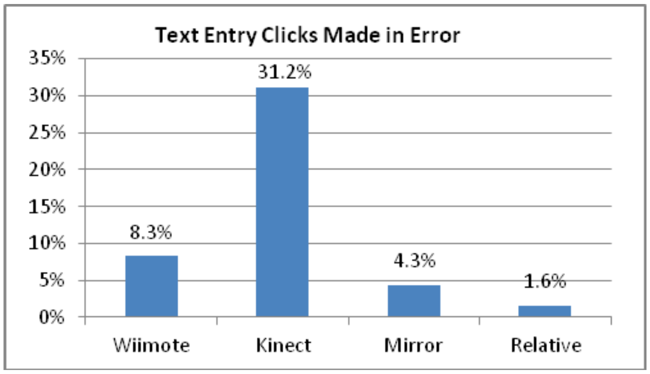

Figure 11: Text entry mistakes using virtual keyboard

Again, because a lack of accuracy can lead to slower task completion, we chose to look at how many mistakes subject made. A mistake in this case could be a subject hitting the wrong letter, or trying to click on something other than a letter on the on-screen virtual keyboard. As in just about every task in our experiment, the Kinect fared most poorly with 31.2% of clicks missing their target. This is significantly worse than the other three devices (One-way ANOVA F(3,116)=19.14 P<0.01 with Tukey's HSD for Post Hoc analysis).

Of the remaining devices the Wiimote fared the worst with an error rate of 8.3%, though this did not affect completion time. The mirror app came next, with an error rate of 4.3%, likely caused by the small size of the keys when shown on the tablet. The error rate for the relative app was a surprisingly low 1.6%. The difference between these devices was not significant.

2.5.3 Effects of Prior Experience

Prior experience can have a significant impact on performance, especially when dealing with the novel. Subjects were asked to rate their prior experience with devices similar to those used in our study. A linear regression was used to compare the number of successful drops in the drag and drop task and the speed of text entry versus their prior experience (see Table 1).

Table 1: Slope of linear models. Steeper slopes indicate stronger experience effect.

	Hits vs. Experience	Text Entry vs. Experience
Wiimote	0.61	0.33
Kinect	0.51	-5.42
Relative	0.39	-0.79
Mirror	0.25	-7.85

We see that the prior experience played the largest role in the Wiimote case. Despite being seen as universally easy to use, subjects were able to effectively leverage prior experience to improve further. The same was the case in the Kinect condition, though here, novices really suffered, and even experts performed marginally. There may have been a floor effect here as fewer subjects had experience with Kinect compared to other devices, and those that did have experience ended to have less experience than with the Kinect than with other devices.

More surprising, there was only a relatively mild learning effect for the two tablet solutions. While experience did help, it seems that these two techniques were so universally well-known and intuitive that all subject were able to use them effectively regardless of experience level.

2.6 Discussion

In our post experiment interviews we focused on understanding the limitations and advantages of the different approaches. One thing we stumbled on were issues related to the sensitivity, or lack thereof for some of our conditions. 76% of users or the relative tablet mentioned that sensitivity was an issue (it took too long to scroll from one side of the display to the other). 43% of Wiimote subjects complained about the device being too sensitive, reacting to slight hand tremors.

Despite the negative results, 13% of Kinect subjects commented positively about its usefulness. This was obviously surprising, but shows that people like the concept of this technique, if not the implementation. We also found that 40% of those who used the mirror tablet and 47% of those who used the Wiimote commented positively about these. Surprisingly only 13% of the relative tablet users commented positively about it despite the high performance.

When asked if the input method could be learned quickly, the Wiimote won out, and it was also rated as the least awkward to use. Subjects liked the simplicity of the Wiimote, both in interface navigation and physically. One participant said, "It was simple. Just point and click. You just aim at it and it’s right there." One of the most common answers about what people liked about the Wiimote input method was that it had only 1 button. People also liked the familiarity with holding the Wiimote, that it felt like a remote control and had a physical button.By far, the least liked device was the Kinect. The most frequent negative comments had to do with physical fatigue and issues with sensor range and sensitivity. Having to hold their right arm up to position the cursor and left arm for click control resulted in almost all Kinect users complaining of arm fatigue. Finding a one-handed method for controlling the system could result in a significant improvement, as indicated by 60% of Kinect users. 60% also complained about the sensor range or sensitivity. To limit interference, the sensor was placed 80cm away from the subject. As a consequence, subjects felt they were unable to move their hands as far to the left and right as they would like. Although most comments focused on why the Kinect was not effective, several participants liked how it did not require them to hold a physical device. One participant talked about how nice it would be to have no remotes and control everything with gestures. Most subjects however indicated that they would be embarrassed to use this technique in front of friends and family.

The two tablet techniques achieved roughly the same ratings, which were generally good. 30% of mirror application users commented positively on being able to directly manipulate the interface. One person said "I really liked being able to click on exactly what I can see on the screen." Others disagreed, saying how they prefer to only have one screen to interact with. One participant commented "Occasionally I found myself not knowing which screen to look at."

A side-effect of our implementation of the mirror app was noticeable "lag" between the TV and the tablet images of between 0.25 and 0.5 seconds. This delay was often mentioned as an annoyance. Likewise, nearly everyone who used the relative tablet app disliked its low sensitivity and the lack of acceleration techniques. With appropriate tweaking, both of these techniques would have likely scored higher on both likability and effectiveness.

2.7 Future Work

This was meant as an exploratory study examining the relative merits of a number of gaming-related UI methods, and their usefulness in an iTV setting. Looking forward, there are several improvements worth exploring based both on user feedback and our findings. As mentioned in the results and discussion sections, implementing motion smoothing for the Wiimote, acceleration for the relative tablet app, and reducing the lag for the mirror app are natural next steps. We believe all of these could drastically improve the user experience.

In implementing smoothing for the Wiimote, a slight delay will be introduced. Pavlovych and Stuerzlinger [22] studied the relationship between jitter and latency and their results could help inform an appropriate balance.

A more tricky problem was the noisiness and occasional false positive for hands for the Kinect. We were unable to use the Kinect API’s native skeleton tracking because only the upper half of the users' body was visible to the sensor. Instead, we used a hand tracking method and filtered based on depth field data. Even with these precautions, a knee or other object could register as a hand. This led to a very frustrating user experience. In future revisions, we would look for a more robust tracking solution. We did not use the official Kinect SDK as it was not available in time.

When asked what they would change about the Kinect method, several people said they would prefer a one-handed solution. We think this would substantially decrease the physical fatigue and provide a more intuitive experience. Due to the Kinect's 640x480 resolution depth camera, robust finger tracking was difficult. Perhaps with a different library or algorithm, a more feasible approach could be found. One option might be the work of Oikonomidis et al. [6], who have demonstrated complex finger articulation, though not in real-time.

2.8 Conclusions

We chose not to investigate the use of voice commands in our experiment, in part because it would be difficult to filter noise and could be socially awkward. As the introduction of the Siri system on the iPhone, and the flurry of interest this has caused, these assumptions and prejudices may need to be revisited in future work.

With the growing availability of broadband Internet access, highly extensible game consoles, and the increasing popularity of social and streaming online entertainment services, their convergence is presenting a number of new challenges for HCI researchers. To the best of the authors' knowledge, there has been very little research on the adoption and use of novel game controller technology in a media center or iTV context. We hope our research will serve as a base for future work in this area.

Looking at the results we see that devices designed for gaming have the potential to be effective input devices for a typical iTV interface. We also see that some devices are better suited to this task than others. The Wiimote was effective, well liked, and very easy to learn. At the same time, it offered ample room for improvement as users gain experience. On the other hand, it is potentially limiting UI-wise, as all information has to be displayed on the primary display. The tablet systems were both well liked and effective as well, though they potentially offer more flexibility and exploration, albeit at a much higher hardware cost.

The Kinect, though appealing to many subjects due to its novelty and the promise of device-free interaction, proved to be too unreliable and cumbersome to use for any extended period of time. While it may be refined with better hardware and algorithms, its suitability and desirability for a social lean-back viewing experience may be limited. Fear of ridicule as much as physical fatigue and the problem of interference from others' movement are serious problems that need to be overcome.

2.9 REFERENCES

[1] S.J. Castellucci and I.S. MacKenzie. "Unigest: text entry using three degrees of motion" In extended abstracts of the 2008 Conf on Human factors in computing systems; CHI '08. ACM, New York, NY, USA, 3549-3554.

[2] K. Cheng and M. Takatsuka. "Initial evaluation of a bare-hand interaction technique for large displays using a webcam" In Proc. of the 1st ACM SIGCHI symposium on Engineering interactive computing systems (EICS '09). ACM, New York, NY, USA, 291-296.

[3] J. C. Lee. Hacking the Nintendo Wii Remote. IEEE Pervasive Computing, 7(3):39–45, July 2008.

[4] J. Lin, H. Nishino, T. Kagawa, and K. Utsumiya. "Free hand interface for controlling applications based on Wii remote IR sensor." In Proc. of the 9th ACM SIGGRAPH Conf. on Virtual-Reality Continuum and its Applications in Industry (VRCAI '10). ACM, New York, NY, USA, 139-142.

[5] D. Natapov, S.J. Castellucci, and I.S. MacKenzie. "ISO 9241-9 evaluation of video game controllers." In Proc. of Graphics Interface 2009 (GI '09). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 223-230.

[6] I. Oikonomidis, N. Kyriazis, and A. Argyros. "Efficient model-based 3d tracking of hand articulations using kinect." Procs. of BMVC, Dundee, UK, pages 1–11, 2011.

[7] L. Rainie. “Tablet and E-book reader Ownership Nearly Double Over the Holiday Gift-Giving Period” Pew Internet. 2012.

[8] G. Shoemaker, L. Findlater, J.Q. Dawson, and K.S. Booth. “Mid-air text input techniques for very large wall displays” In Proc. of Graphics Interface 2009 (GI '09). pp. 231-238.

[9] I.S. MacKenzie and S.X. Zhang. “The immediate usability of graffiti.” In Proc. of the conf. on Graphics interface '97, Toronto, Ont., Canada, Canada, 129-137.

[10] D.C. McCallum and P. Irani. “ARC-Pad: absolute+relative cursor positioning for large displays with a mobile touchscreen.” In Proc. of the 22nd annual ACM symposium on User interface software and technology (UIST '09). ACM, New York, NY, USA, 153-156.

[11] T. Ni, D. Bowman, and C. North. “AirStroke: bringing unistroke text entry to freehand gesture interfaces.” In Proc. of the 2011 annual conference on Human factors in computing systems (CHI '11). ACM, New York, NY, USA, 2473-2476.

[12] YouTube, 2012. YouTube Press Statistics: http://www.youtube.com/t/press_statistics

[13] Netflix, “Netflix Q4 2011 Letter to Shareholders.” http://goo.gl/4vgP1

[14] A. D. Wilson and M. Agrawala. “Text entry using a dual joystick game controller.” In Proc. of the SIGCHI conf. on Human Factors in computing systems (CHI '06), ACM, New York, NY, USA, 475-478.

[15] T. Költringer, P. Isokoski, and T. Grechenig. “TwoStick: writing with a game controller.” In Proc. of Graphics Interface 2007 (GI '07). ACM, New York, NY, USA, 103-110.

[16] Nielsen. “State of the Media: Consumer Usage Report,” 2011. http://goo.gl/PzAzc

[17] Sandvine, 2011. “Sandvine. Global Internet Phenomena Spotlight: Netflix Rising.” http://goo.gl/dbkPz

[18] Hulu, 2012. “Hulu Financial Results:” http://blog.hulu.com/2012/01/12/2011-2012-and-beyond/

[19] Nielsen, 2011. “Nielsen: The Cross-Platform Report.” http://goo.gl/JxEpn

[20] A. Oulasvirta and J. Bergstrom-Lehtovirta. “Ease of juggling: studying the effects of manual multitasking.” In Proc. of the 2011 annual conf. on Human factors in computing systems (CHI '11). ACM, New York, NY, USA, 3103-3112.

[21] NTIA, 2010. “Exploring the Digital Nation: Home Broadband Adoption in the United States.” http://goo.gl/ErsnS

[22] A. Pavlovych & W. Stuerzlinger. “The tradeoff between spatial jitter and latency in pointing tasks.” In Proc. of the 1st ACM SIGCHI symposium on Engineering interactive computing systems (EICS '09). ACM, New York, NY, USA, 187-196.

3. Second Manuscript: Using Game Controllers in Interactive Television: Guidelines for Menus and UI Elements

Justin Wolford, Dale Cox, Carlos Jensen, Dedrie Beardsley

School of EECS

Oregon State University

Corvallis, Oregon, 97331, USA

{ wolfordj, beardsld } @ onid.orst.edu

{ cox, cjensen } @ eecs.oregonstate.edu

3.1 ABSTRACT

Due to the growing popularity of new interactive media and devices, users are faced with increasingly complicated user interfaces, as these systems seek to span multiple uses such as movie watching and game play, web browsing, or email. A simple solution is to default to the standard PC experience; a mouse and a keyboard. However, this can be a very intimidating approach to non-technical users, and is the antithesis of the "lean back" experience associated with TV viewing. The alternative is to try and leverage novel UI techniques, such as those pioneered by game controllers and tablets to bridge the gap. This paper looks at the results of a user study in which subjects used new input devices to navigate a media center application. The devices included the Nintendo Wiimote, the Microsoft Kinect, and two tablet apps. This paper presents design guidelines for these devices in an interactive TV context.

3.2 INTRODUCTION

Interface design guidelines are necessary to create a consistent and intuitive user experience. While general design guidelines exist (see Nielsen's design heuristics for instance [25]), custom interface guidelines are often produced for devices, tasks or a combination of both. For example, Apple maintains the iOS Human Interface Design Guidelines for their iPod, iPhone and iPad [26]. Google likewise maintains interface design recommendations for Android tablets and handsets [27]. These guidelines, though overlapping, address the unique characteristics of the devices they are aimed at and their affordances, as well as aim to create an intuitive and unified user experience. These guidelines are especially important when designing for devices where interaction is somehow limited, as they embody the accumulated knowledge about UI pitfalls and problems.

Designing for Interactive Television (iTV) presents an interesting set of challenges. In some ways, iTV has borrowed much of its technology from the Internet-infused PC and associated domains, where the user is inherently involved and in control, but must operate in a living room environment, where the user inherently expects to lean back and be entertained with little or no interaction from their end.

As more and more TV viewers migrate away from standard broadcast TV and start using more interactive and rich systems, there is a growing need for interfaces to support these more complex viewing habits. These interfaces may need to merge different media outlets, accommodate web browsing and incorporate social media all at once, while at the same time allowing for a relaxed and social experience. Current interfaces for these services are often optimized for the standard mouse and keyboard input method, something which research has shown people find cumbersome in a living room environment [14, 15]. Our goal was therefore to examine how novel UI techniques hold up in an iTV setting, and document key design guidelines and considerations for using these.

iTV is a somewhat ambiguous term. For our purposes we will define iTV as a television viewing experience that extends beyond simply switching channels to the highly interactive. This could mean additional content from Internet source to supplement TV programming, as well as services and applications to provide services such as video-on-demand (VOD) or digital-video-recording (DVR), social networking, search functionality, and interactive information displays and games. Cesar and Chorianopoulos proposed the concept of create-share-control for iTV versus produce-distribute-consume for traditional broadcast television [16]. This comparison clearly shows the difference between a passive and active experience. Content delivery has perhaps been the largest barrier to changing the status quo of broadcast television. Services like Netflix, Hulu and VOD have revolutionized the world of media distribution. Although interactive TV has been evolving for decades, only within the last several years has the Internet infrastructure supplied the potential for iTV to grow. Broadband Internet subscribers jumped from 9% to 64% of US households between 2001 and 2009 [11]. Previously, the majority of Americans had dial-up access, which typically had an access speed of 56Kbps, where it is common for broadband speeds to reach 10Mbps.

iTV services like GoogleTV and AppleTV integrate Netflix and Hulu as native applications into an application framework. While only loosely integrated, these services are accessible through similar interfaces. One of the most widely used Internet streaming providers, Netflix, had over 21 million subscribers by the end of 2011 [7]. Netflix consumed 20.6% of all downstream primetime bandwidth in North America according to a 2010 report by Sandvine [9]. At its core, VOD and Internet video services provide iTV users a much richer experience than broadcast alone. Along with this additional control, comes an inherent need for a more capable interface that enable search, managing queues and users, and other functions.

Many devices today have integrated Internet access, including TVs, game consoles and Blu-ray players. Internet connectivity on game consoles specifically enables a wide variety of services from online gaming to streaming video. In a 2011 Nielsen study, it was found that 50% of Netflix users accessed content through a game console [8]. These latest generation of game consoles also bring with them exciting new opportunities on the interaction front, all sporting novel interface devices (also referred to as Natural User Interfaces (NUI's)). These allow users to interact with these devices through motion and gestures, in a potentially more expressive and intuitive way.

Most iTV systems have used a traditional remote control. Though some systems add custom shortcut buttons for accessing functions such as VOD, this does not address the difficulty of navigating an extensible, dynamic environment. Using a remote is extremely limiting when a system is trying to integrate services like VOD, DVR, web browsing and social networking. More recently, Sony and Samsung have experimented with 2-sided remotes, or remotes, or remotes including a full QWERTY keyboard. While these solve some problems, they introduce others.

For our input devices we used the Microsoft Kinect, the Nintendo Wii and an Andriod tablet running two different pointing solutions. The first version, named the mirror tablet, mirrored the TV display and allowed users to directly click on the screen to interact with UI elements. The second version, which we called the relative tablet, was an enlarged touchpad like one would find on a laptop. Users dragged their finger on the tablet to move a cursor on the screen, and users could tap the display or use a virtual button in the corner to click or select.

3.3 Related Work

3.3.1 iTV Considerations

It is easy to imagine an average iTV user being highly engaged with the content and willing to interact with various social and entertainment channels. In reality, one often faces a more complex situation. From a national survey of TV viewers, Lee et al. found the majority of viewers fit into one of four groups based on attentiveness: sole attention, split attention, peripheral activity and background noise [20]. Respondents would move between modes depending on time of day or content, and on average had their TV on for 5.4 hours per day, but saw themselves as actually watching for only 3.5 [20]. Two-thirds of the time people are multitasking, performing other activities such as eating, reading, talking or chores [23]. With such a varying degree of attentiveness, requiring a forgiving interface with flexible interaction from a variety of distances and angles becomes important.

Many of the services driving iTV adoption originated on the PC. This includes Netflix streaming, Hulu and social networks like Facebook and Twitter. Despite this overlap, the device environment and use cases are very different. Broadcast television viewing is generally considered a lean-back activity where users are not required to actively engage in the process other than changing a channel or volume. iTV incorporates several lean-forward technologies requiring a more participatory experience.

Pemberton et al. described several major differences between the usability of the PC and an iTV, and that established HCI evaluation techniques developed for PCs may need modifications in an iTV context [17]. The physical environment is different as viewers watch television from a distance, resulting in much smaller viewing angles [18]. There is often a conflict between watching a broadcast stream and using interactive components – users may be trying to juggle multiple information channels and dealing with the increased cognitive load may be need to be reflected in the design and allocation of screen “real-estate” [17]. Another difference is the real-time nature of television. If a user is watching a live broadcast such as a sporting event and wants to access additional content, the interface response must be fluid and react immediately as to not interrupt the viewing experience [17].

To address the challenge of changing attention levels and the increasingly social needs of iTV systems, Geerts et al. [24] developed a set of sociability heuristics for “social TV”. Among these, were guidelines to support a diverse set of input methods and provide both synchronous and asynchronous communications services. By giving users multiple input options such as voice and a remote control, these services would be flexible enough to meet to their needs based on the situation rather than vice-versa. In a multi-person home or other collocated environment, viewers found it especially important to have the option of real-time chat services, as well as functions similar to email or a multi-user channel guide with recommendations. These guidelines signal a shifting focus from pushing content (broadcast) to user-centric systems allowing personalization and content creation.

In a study centered on trying to avoid the need to adapt to existing third-party or cross-domain interfaces and devices, Bernhaupt et al. [15] developed three pairs of interface and remote control combinations, then performed a comparative study. By designing the interface and remote in the same development cycle, they were able to include only the needed functionality on the remote instead of having to adapt to a third-party design. They found a 6-button remote along with a simple, but efficient interface led to the least amount of button pushes and highest satisfaction rating.

With the increasing popularity of DVRs and VOD, users are increasingly in control of when to watch content. A study of DVR early adopters found that all but one household switched to almost exclusively watching pre-recorded content [19]. In the same study, they found that browsing archives of queued recordings supplanted channel surfing. The input demands for this activity adds an extra layer of complexity. This is an example of the growing complexity of the modern interactive TV experience.

3.3.2 Pointing and Navigation with Tablets and Game Controllers

Pointing and navigation are crucial to the modern iTV experience. In the age of apps and web-based content, the variety of interfaces encountered is immense. Regardless of complexity, being able to intuitively manipulate cursor position and interact with UI elements are key requirements. Natural user interfaces (NUI) like the Kinect or Wiimote, have found an active research community, investigating their affordances and limitations. A popular pointing strategy with NUIs is to use the subjects' hand(s) to directly manipulate cursor position. We have found several studies evaluating this technology and present them in this section. Several devices were used, but the Wiimote specifically offers multiple approaches due to its unique selection of sensors. A finger tracking method was demonstrated by Lee [2] using a Wiimote. He used the IR camera built into the controller in combination with an IR LED array and reflective tags taped to the users' fingers. A similar method was created by Lin et. al [3] which used a second Wiimote controller to enable additional control possibilities. Suggested applications included browsing pictures, reading e-books or controlling presentations.

dTouch is a free-hand pointing technique developed by Cheng and Takatsuka[1] for large screens taking advantage of off-the-shelf webcams. The concept of a "virtual touchscreen" was used to manipulate on-screen objects in an absolute coordinate system. A comparative study comparing dTouch to a method using the Playstation console's EyeToy camera. The two methods had similar performance results, but users preferred dTouch.

Natapov et al. [4] evaluated the Wiimote and a traditional gamepad with no spatial gesture capability for performing both pointing and selection tasks. Despite having an error rate around 30% higher than the gamepad, the Wiimote achieved a performance rating 75% better than the traditional controller when looking at speed and accuracy. 14 out of 15 participants commented how they would prefer the Wiimote method.

The last area we'll look at is pointing using a smart phone or tablet. Two of our conditions use a tablet using either a relative or absolute positioning system. McCallum et al. [6] created the ARC-Pad, a combination of the two techniques targeted at large screens controlled from a smart phone. This design was unique because it did not require an explicit context switch between absolute and relative coordinates. They performed a user study comparing ARC-Pad to a traditional touchpad with cursor acceleration for a series of cursor movements. The results showed ARC-Pad reduced screen touches by half and the task completion time stays nearly constant when screen size scales up while the touchpad steadily increased. This fits with Fitts' law [30].

3.3.3 Beyond Direct Manipulation

With the mainstream introduction of NUIs such as the Microsoft Kinect or Nintendo Wiimote, the traditional direct manipulation paradigm is shifting. The possibility of using direct spatial gestures allows users to interact with objects with little or no physical assistance. This novel interaction technique, comes at a potential cost. Developing a robust intuitive gesture parsing or computer vision algorithms is non-trivial. For a user, there is the additional cognitive load introduced by a physical multitasking and adapting to a new system.

In the case of the Wiimote, the subtle shake of the users can be picked up and translated to a shaky cursor. A common solution to this kind of issue is to average over previous cursor positions in order to smooth movement. This has the side effect of introducing a slight delay. Pavlovych and Stuerzlinger [12] conducted a study investigating the effect of jitter and latency on performance using a Wiimote. One of their goals was to investigate the tradeoff between latency and jitter. For latency levels up to 58ms, there is no significant difference. They found that the appropriateness of filtering is context specific. When working with smaller targets, aggressively filtering of jitter will causes subjects to overshoot their targets. In this case, lowering the amount of jitter that is filtered out can actually decrease the error rate.

Mandryk and Lough [28] set out to examine the link between the intended use of a system and completion time of the task. The four tasks examined were: target, dual target, flick, and dock (drag). All tasks began with the same initial motion. Their results showed the velocity was significantly higher just prior to target acquisition in target and dual target compared to flick and dock.

There is a growing area of research around so-called “second screen” applications. These are supplemental apps that supply context or interaction not usually available through the television. Viewers access content while the show is airing or afterward. Part of the rise in popularity of the second screen is due to the substantial growth in tablet sales. Tablet sales nearly doubled over the 2011 holiday season [5] and a 2011 Nielsen report showed that 45% of tablet owners looked up information related to the show they were watching [8].

Basapur et al. [21] performed a field trial of a second screen prototype enhancing broadcast programming. The app provided a parallel feed that supplied content such as IMDB trivia, social networking and related multimedia. Results showed users felt "empowered" and the parallel feed let them pay more attention to the show because they didn't have to look up information on their PCs. Another interesting finding was that some participants who were used to using TV to "wind down" at the end of the day found the dual interaction too active of an experience.

3.4 Methodology

We conducted a user study focused on studying the usability of an iTV interface combined with NUI input devices. A total of 4 input conditions were used by participants to carry out a series of tasks representative of typical use cases. The 4 input conditions were the Wiimote, Kinect, relative tablet and mirror tablet. In total, we had 62 participants, 33 male and 29 female. Their average age was 24.5 years old, with the youngest being 18 and the oldest 57. 58 were right-handed. All participants were recruited in pairs from a college campus and nearby community.

3.4.1 Configuration

We attempted to simulate a living room environment by setting up a room with 2 armchairs, a center table and a small end table (see figure 12). The participants were asked to relax in the armchairs and the experimenter sat in an armchair of to the side. The display used was a 55” 1080P HDTV connected to a PC. The small end table was positioned between the chairs for subjects to place a plate or drink on but it was not large enough to hold both. Each person was given a plate with a slice of pizza and a beverage. The goal was to induce common multi-tasking, as past research by Oulasvirta et al. [22] showed the importance of considering these effects on interface evaluation. The act of multitasking also offers a more realistic reproduction of a living room environment.

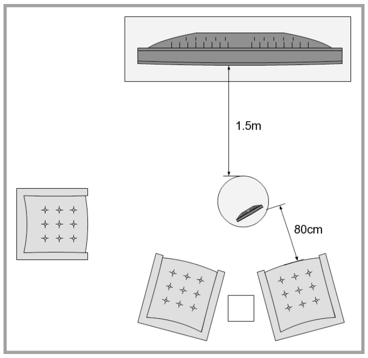

Figure 12: Room configuration used during the experiment.

The Kinect sensor was placed on the center table pointed at one participant to try and limit its field of vision and possible interference form the other participant. To take advantage of the Wiimote’s IR capabilities we used a Nintendo wireless sensor bar placed in front of the television. To control the Wiimote, an application called GlovePIE was used to run a script enabling the Wiimote to control the mouse cursor. The ‘A’ button was used to simulate a left click.

For both tablet conditions, a 10.1” tablet running Android 2.3 was used. The relative condition used an unmodified copy of the open source Android application “RemoteDroid”. The application has a client (on the tablet) and a server (on the PC) that communicate over WiFi. It basically turns a tablet screen into a large touchpad, behaving like a touchpads found in most laptops. The mirror condition used a customized version of the same app. The tablet screen was continually updated with an image captured from the active PC screen. The relative coordinate system was replaced by an absolute coordinate system (2:1 downscaling), so that when a user tapped a location on the tablet, a click would be registered on the PC at the matching x/y coordinates.

The software to control the Kinect was a custom application written in C# using the OpenNI framework. It controls the actions of the cursor and left clicking through spatial gestures. The cursor tracks the movement of the users' right hand and adjusts the cursor accordingly. To emulate clicking and drag and drop, we employed the abstraction of a virtual plane in front of the user. To click, the user holds their right hand on the target and move their left hand forward and back through the plane in one fluid motion. This registers a click. To drag and drop, the click gesture is performed, but the user keeps their left hand extended. The right hand is then moved to the desired drop position, and the icon is released when the user pulls their left hand back through the plane.

3.4.2 Procedure

All participants completed a pre-experiment questionnaire for demographic data, device use history and media use patterns. All sessions were recorded using both a video camera, and screen recording software. Each participant pair was assigned two devices/conditions, with device permutations being assigned randomly. Both subjects would use each device, so each session generated a total of 4 trials. Prior to starting the set of tasks, subjects were trained in how to use each device. After training, they were allowed to practice for as long as needed.

Participants took turns using each device and performed 4 tasks during each trial; each designed to simulate the input demands of an iTV system. Completion times for each task were recorded, as were successful and missed clicks. For the purpose of analysis, we classified UI elements based on type, size, and placement. In terms of placement, we expected users to have trouble with edge/border conditions if at all. We therefore defined a 10% border around the edge of the screen (108 pixels top and bottom, and 192 pixels left and right). Each buttons' location on the screen was examined to look for performance differences based on button placement. Buttons in the center were marked as "inside", buttons entirely in the border area are referred to as "outside", and buttons that span the boundary were marked as being in both.

The first task was a drag and drop task, where a randomly placed, 150x150 pixel widget was to be dragged onto a target that would shrink in size with each successful drop. The target started out as 220x220 pixels and would appear at a random location. Each time the widget was successfully dropped within the bounds of the target, the target would shrink 10% and the both the widget and the target randomly moved. If a successful drop was not completed within 16 seconds, the target size would increase by 5%. This was done to test speed and accuracy for each device.

For the remaining 3 tasks, users were asked to perform a sequence of navigation tasks using the media center application XBMC (xbmc.org). XBMC is representative of a modern media center/iTV interface, and runs on most PCs today.

The first XBMC task was to navigate to the weather settings screen and change the city name. Users were required to navigate through 7 screen transitions and click on varying button sizes. This task required the most button presses, with the ideal case being 22 clicks (18 inside, 2 outside, 2 both). We chose to focus on were the letters of the on-screen keyboard and the "done" button were located (center, see figure 13). The letters and "done" button were 75x75 and 298x75 pixels respectively. Participants found this to be a challenging task due to the size of the targets.

portlan|

Figure 13: Text entry screen with buttons emphasized.

The next task was to navigate to the weather screen and change the city selected in the previous step. To accomplish this task, the users had to use two small arrow buttons to select the city displayed from a list. There were only 5 clicks in this task (1 inside, 4 outside). The arrow buttons were 49x33 pixels, and were located on the inside portion of the screen. Clicking the 92x45 "home" button in the far upper-left (outside) corner.

Finally, subjects were asked to navigate to the video selection screen, choose a movie, and adjust a volume slider. The total click count was 11 for this task (5 inside, 4 outside, 2 both). The volume slider was in the top portion of the screen and measured 9x35 pixels, which is the smallest target in the experiment (see figure 14). The audio settings and "X" button were 60x60 and 85x31 pixels respectively.

Volume -23.4 dB
Volume amplification 0.0 dB
Audio offset 0.000s
Audio stream AC3 5.1 (1/1)
Output stereo to all speakers
Audio output Analog
Enable subtitles
Subtitle offset 0.000s
Subtitle None
Browse for subtitle...
Set as default for all movies

Figure 14: Audio settings screen with buttons emphasized.

After one subject finished all tasks, their partner would use the same device to do the same tasks. The process was then repeated for the second device. Afterwards subjects completed a survey about learnability, ease of use, and the real-world usability of each device using a 5-point Likert scale. Finally, participants were interviewed in order to get a more thorough understanding of their experiences.

3.5 Results

3.5.1 Accuracy and Location

We primarily measured the accuracy users were able to achieve with each device using the drag and drop task. Figure 15 shows the target size for the last 5 successful drops users made in the drag and drop task. We focused on the last 5 hits as this allowed subjects time to achieve and plateau at their optimal accuracy. The mirror application had the highest average level of accuracy , followed closely by the Wiimote and the relative app. The Kinect performed notably worse.

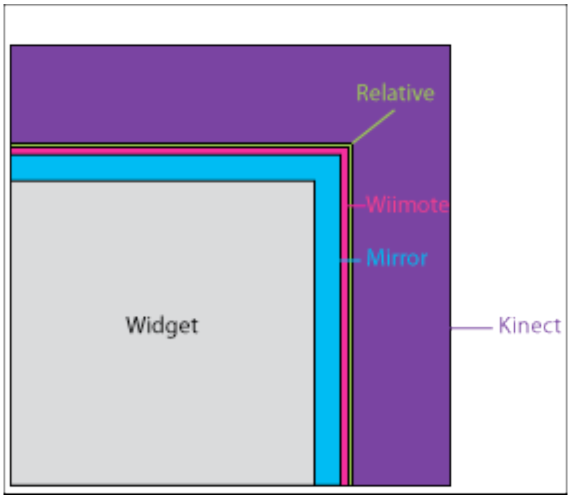

Figure 15: The average size of the last 5 hits relative to the size of the widget. The mirror had the highest accuracy.

3.5.1.1 Kinect

For the Kinect the average size of the target for the last 5 successful drops was 68 pixels larger than the size of the widget. This is a good measure of the upper limit of accuracy achievable by Kinect users using our two handed setup such as ours. We observed that users had a harder time hitting a target accurately than they did releasing the drag accurately, meaning that most subjects ended up wasting time trying to pick up targets.

Part of the problem is grounded in the two-handed design we chose for the Kinect condition. When brining the second hand forward to select a target, most subjects naturally rotated their torso to maintain balance. This would in most cases cause their targeting hand to move off their intended target.

This lack of accuracy was again observed in the text entry task, which involved clicking on small buttons (75x75 pixels), close to the observed accuracy threshold. In this task users missed their intended target an average of 5.5 over an optimal 8-click task. In other words, 39% of the observed clicks were made in error. It took Kinect users the longest to complete this task taking on average 134 seconds.

Users were significantly more successful at hitting the “Done” button, which was significantly wider than the text buttons (298x74 pixels), with only 18% of clicks were in error. This was corroborated by the video selection task, where users selected a large video thumbnails, where user only missed an average of 13% of the time. This indicates that button size is a critical design consideration for user interfaces design for the Kinect. Overall Kinect users missed small targets 45% of the time, and large targets 30% of the time (One-way ANOVA $F(2,817)=4.171$ $P<0.05$, Post-hoc analysis with Tukey’s HSD) . There was no significant difference involving medium buttons. However, these results are skewed upwards by buttons on the left side of the screen so buttons which were more central or right proved much easier to hit.

Beyond button size, other tasks were also affected by the lack of accuracy for the Kinect. In particular the volume setting task where users had to set the volume using a small slider caused significant problems. In this task it took users an average of 4 tries to select the slider or get it within the specified range. We also found that placing items near the outside of the screen led to significantly worse performance than placing them in the center of the screen (Welch's t test $t(205.1)= -4.21$, $P<0.001$). Users missed inside targets 37% of the time, and outside targets 61% of the time. They are actually more likely to miss outer targets than to hit them. This seemed particularly bad for the left edge of the screen because the Kinect would confuse the left and right hand as they got close together to interact with these elements.

3.5.1.2 Wiimote

The Wiimote allowed for a high degree of accuracy. In the widget dropping task the average size of the target dimensions in the last 5 successful drops was only 19 pixels larger than the target. However, the Wiimote had an unusual number of attempted missed drops in the drag and drop task. When near the target, users would rapidly grab and release the widget trying to register a hit, “nudging” the widget onto the target. The Wiimote was the only device which exhibited this behavior. It is possible that the presence of a physical button allowed users to do this more easily.

In the text entry task users missed their targets 10% of the time. They completed this task in a mean time of 40.8 seconds which was the fastest of any devices. We observed very little frustration in this task. A few users would use two hands or find some other method of increasing stability, countering hand jitter. One user commented: “I even noticed when I was typing that it does jump a little bit. It really makes you have to hold your hand steady”. Our Wiimote implementation had very little smoothing. It would likely be possible to smooth input further, but this could cause a lag, something users in other conditions found problematic.

Another small button task involved hitting the home button in the upper left corner. After completing the weather task, users would miss in 27% of attempts at hitting the home button. This button is slightly smaller than the text entry keys and is in a peripheral part of the screen. The volume selection task had the highest accuracy demands, with users missing the scroll bar or the intended volume 63% of the time. While bad, this was a very marked improvement from the Kinect task. The aforementioned oversensitivity of the sensor, and the jitteriness of the Wiimote could have contributed to the accuracy problems.

There should be little need for buttons smaller than the home button and tasks requiring more accuracy than the volume selection task. In fact, in both of these tasks, there was ample screen real-estate that could have been used to make the task easier by enlarging the button slightly or making the slider bar longer to increase the target area size.

When users discussed the Wiimote one user commented that “I thought that the Wiimote was better for just general purpose use, like it’s fine for just navigating a menu. But I don’t think it’s as precise as the tablet for dragging the box, the picture, the really small box”. This seems to support that the Wiimote was fast and accurate enough, but there is a lower limit due to the jitter from the device and the users own hand tremmors. As long as buttons remain as large or larger than the home button (92x45 pixels), the Wiimote will work well.

The sensitivity was a concern among users despite overall high performance. Many users found that the Wiimote was too sensitive and that they would like to lower the sensitivity. One user explained: “If you could adjust the sensitivity, that would be a lot better”. An option to allow the user to set the smoothing may be useful. Another user suggested: “Maybe having it snap to certain things, that might have been kind of useful”. This idea of snapping to items that can be interacted with warrants further exploration and may be a good option when small UI elements are needed. With the Wiimote we were somewhat surprised to see that there was no significant difference between button sizes or button placement. Performance only decreased significantly when buttons got very small.

3.5.1.3 Mirror

Of all the devices we examined in this study, the mirror tablet was the most accurate in the widget drag and drop tasks. In this task the target in the last 5 successful hits averaged only 13.4 pixels larger than the widget. This shows that a very high level of accuracy is achievable with the tablet. When one user was asked what they found enjoyable during the experiment they responded “For very small movements it [the mirror app] was really precise”.

As users moved into other tasks, their performance seemed to decline slightly. In the text entry task mirror tablet users missed 4% of the time. With the mirror we found a significant difference between large and small buttons (One way ANOVA $F(2,834)=3.519$, $P<0.05$, Post-Hoc analysis using Tukey’s HSD). Users would miss small buttons on average 21% of the time and for large buttons an average of 7% of the time.

Surprisingly the mirror app was not the best at any of the navigation tasks. One of the limitations of our setup was a delay (lag) between the onscreen display and the tablet display. This delay was long enough that users sometimes thought they had missed a button and attempt to click a second time. This would register as a miss. When one user was asked if there was anything he explained: “I couldn’t really use just the tablet to navigate precisely… that just goes back to lag though.” Another user said: “It felt like I was in the right place but I wasn’t as precise as I thought I was”.

We also found that placement near the edge of the screen results in worse performance (Welch's t test t(215.1)= -4.084 P<0.001). Users missed inside targets 13% of the time and outer targets 33% of the time. This could in part be caused by the target being close to the edge of the tablet. This could cause the hardware to misread the target.

3.5.1.4 Relative

Users of the relative app averaged a target size 18.9 pixels larger than the widget for the drag and drop task. This shows that the relative tablet is slightly less accurate than the Wiimote and Mirror app. This is surprising because the relative app should have given users better resolution in the dragging to hit the smaller targets by offering a 1:1 correlation in terms of pixels. On the other hand, it took multiple swipes across the tablet to move from one edge of the screen to the other. Many users wanted it to be more sensitive, one commenting: “I didn’t really like the mouse sensitivity, I thought it could move faster”.

The rankings look somewhat different for the text entry task. Here the relative tablet ended up being the most accurate, with users only missing 2% of targets. However, the device ranked 3rd in terms of speed. This shows that there is a tradeoff in terms of speed and accuracy.

Finally the volume adjustment task, the relative app performed the best. One possible explanation is that the mechanism for clicking and releasing for the relative tablet is separate from the mechanism for pointing. Pressing a button the Wiimote could cause the device to move slightly, when using the mirror app lifting a finger from a drag event could move the pointer slightly, and any move with the Kinect could cause undesirable movement. With the relative tablet users could point to the desired location, remove their pointing finger, confirm that the location is correct then click without the possibility of moving the pointer. As one user put it “I think that was a big part that made it more intuitive and easy, because it was more predictable”.

With the relative app we find significant differences in terms of button sizes (One-way ANOVA $F(2,836)=6.28$, $P<0.01$, Post-hoc analysis with Tukey’s HSD), with users on average missing 12% of small targets and <1% of the time for large targets. We also found a significant difference between buttons at the center of the screen and those on the edges of the screen (Welch's t test $t(311.5)= -1.50$, $P<0.05$), missing inside targets 6% and outside targets 13% of the time.

3.5.2 Task Types

There were several kinds of tasks users had to perform; drag and drop, button selection, and scrolling were the main kinds of tasks. In this section we look at these tasks with respect to the different devices.

3.5.2.1 Kinect

A limiting factor in looking at task types for the Kinect was its poor accuracy. Because we used an off the system that was not specifically designed for devices like the Kinect, few of the UI elements were optimized for this device. It is therefore difficult to disentangle UI element size for type.

Tasks involving text entry will be difficult because of accuracy issues. To enter text it is necessary to show a full keyboard on the screen, which limits the maximum size of the virtual buttons. When interviewed, 30% of subjects pointed to text entry as being a difficult task, more so than on any other device. As an alternative, some form of voice command may offer a solution. Many users mentioned the voice capabilities of the Kinect as being a possible solution to many of the difficult tasks in the post experiment interview.

The interface navigation setup seemed to work reasonably well. Although there are other navigation styles for Kinect interfaces that often involved swiping to get from menu to menu such as the Xbox 360 dashboard, no users brought this up in the interviews whereas many users talked about alternatives for other tasks. This may indicate that a button based navigation system will work for the Kinect so long as the buttons are adequately sized and placed in locations that are not difficult to get to. One user commented “There are a couple of pretty small buttons in some of the settings. The weather settings were a little difficult.” and another user on corners “I can’t really imagine even using the bottom left corner, I just don’t see how that would work!”. These issues make it difficult to really look at other tasks.

However, in the drag and drop task, elements were fairly large and we observed that people generally had a little easier time dropping the widget into a small target than they did clicking on small targets in other tasks. So dragging and dropping may be a reasonable task expectation for the Kinect.

3.5.2.2 Wiimote

Of all the devices the Wiimote had the fewest users mention text entry as being difficult. This would indicate that even though the Wiimote is very dissimilar from a keyboard it did not pose a problem for users to use the on screen keyboard so this seems like a valid solution to text entry tasks. One user suggested that adding audio feedback may improve this task “Sound response, every time you have to move over a key, it makes a little click sound”.

Navigation also was easy for users of the Wiimote. The button based navigation system in XBMC posed no problem for users with one user saying “I thought that the Wiimote was better for just general purpose use” when comparing it to the mirrored tablet. Even the buttons located in the extremes of the screen did not seem to cause any problems for users.

The main task that seemed to prove difficult for users of the Wiimote was the scrolling task. One user, when asked if there was anything she wanted to do but couldn’t simply replied “The scroll”. This task required a high level of accuracy to get the volume within the specified range. At that point the sensitivity to hand shake would mean that the scroll bar was going in and out of the target area and it was a matter of getting lucky when letting go of the handle. This is similar to the behavior of users trying to get the widget into the smallest targets where they would quickly and repeatedly grab and release the widget hoping to land it within the bounds of the target. This seems to be a larger problem with users releasing in a small than it does for users selecting a small target as other small buttons seemed to pose less of a challenge.

3.5.2.3 Mirror

Positive comments about the text entry task were rare because none of the devices proved to be ideal. However, 13% of users of the mirror comment positively about the text entry task which was the most out of any of the devices. One user commented “I mean it was really easy to type, I could hold a soda in my hand and still be typing. I didn’t necessarily have to hold the tablet, I could have two things in my hand if I really wanted to”. In terms of time to complete the text entry task, the mirror tablet was the 2nd fastest. However it is difficult to know how much users times would have improved if the display lag had been reduced. Several users pointed out that it would make sense to have a keyboard that was native to the tablet for text entry instead of having to rely on the particular application to provide one.

The mirror tablet did not appear to be any more successful than the Wiimote and scored lower than the relative tablet in the volume slider task. This seems to be at odds with the results in the drag and drop task and would merit some further investigation. Several factors may have contributed. The lag may have made it difficult for users to judge when the slider was in the correct location. Their finger may have been obscuring the slider because the slider was too small on the screen. It may be that it is difficult to accurately click but easy to accurately release.

3.5.2.4 Relative

The relative tablet performed well on all the tasks. Despite its lower score in the drag and drop task users were much faster at adjusting the volume with it. Users averaged 15 seconds to complete the volume task with the relative tablet. This is 6.5 seconds faster than the Wiimote and 8 seconds faster than the mirror tablet. All of these devices were much faster than the Kinect which took 33.25 seconds on average for users to adjust the volume.

Users seemed to find it very natural. As one pointed out “The tablet seemed so much easier because there’s so much technology that we already have that’s exactly the same. Like scrolling on the laptop, it’s kind of similar” which is true of the relative tablet. It worked much like a larger laptop track pad and so it was well suited for these tasks which were designed for use with a typical computer interface.

3.5.3 Other Considerations

Beyond the button size and placement and task types, there were other issues that were more general but were specific to the various devices. Here we discuss some of the things that seemed to impact the user experience but were not part of the user interface but that need to be considered when designing a UI for these devices.

3.5.3.1 Kinect

Users expressed some concerns about using the Kinect as an input device because of the more active nature of having to gesture using your arms to accomplish tasks. These ranged from how one would hold a drink and use the device, to simply getting tired while using the Kinect. When asked if there was anything the user wanted to do but couldn’t one subject answered “Drink water” referring to the difficulty they were having trying to use the Kinect and drink the water that was given to them during the experiment. Another user said that “Your arms get tired” and “You have to really be focused on it. There’s no chatting with your friends. It’s like, ‘don’t talk to me while I’m trying to do this!’”. All of these comments point to the same idea that using the Kinect requires mental and physical work which can be tiring over long tasks.

Our text entry task where the user simply had to type in a city name took on average more than 2 minutes. During this time users often showed signs of getting agitated or physically tired. Long tasks, particularly more taxing long tasks should be avoided when designing for the Kinect.

3.5.3.2 Wiimote

Like the Kinect, the Wiimote requires significant physical motion to interact. So there are similar concerns as with the Kinect in terms of long taxing tasks. One when asked what they did not like about the Wiimote one user responded “When you are doing something for a long time, just holding your hand up there". While observing users we noticed that after using the device for a while users would sometimes try to use a second hand or prop their hand on the arm of the chair or their leg. This would sometime lead to confusion because the angle between the Wiimote and sensor bar would change and users would take a few moments to readjust to the new position.

3.5.3.3 Mirror

Given the difference between user performance in the drag and drop task and the tasks in the actual application this last issue of difficulty clicking with high accuracy seems like a good candidate as the cause for the discrepancy. Even though users were able to drag within 13 pixels of the target, they may well not have the same ability to directly click a target that is only 13 pixels square.

Another concern users had when using the mirror was smudges on the device. One user commented "Having the pizza and the tablet right there…I was worried about the greasy fingers and getting the device dirty" although there is not much that can be done from an interface perspective it is worth considering the capabilities of the tablet display. Another user commented "You couldn't see the screen sometimes on the tablet. You would have to have it at the right angle to see Weather, System, and Videos". In our study the UI was not bright which may have made it harder to see from some users. On the other hand, if the UI was brighter it may increase visibility to the point where it would become a distraction from the main screen. So the brightness of the UI certainly needs to be considered for both the main screen and the tablet.

3.5.3.4 Relative

The main complaint from users of the relative tablet was the previously mentioned sensitivity. Many users wanted to be able to scroll the pointer from one edge of the screen to the other edge of the screen without having to perform a second drag. This could be solved easily by giving the users control over the speed of the pointer.

3.5.1 Results Summary

We compared four different devices in a stand alone drag and drop task and in a series of tasks that users would perform in a typical media center using the XBMC application. The drag and drop task showed that the Kinect was slower and required a much larger allowance for inaccuracy than the other three devices. Among the other devices the results were all fairly similar but we observed a tendency for users of the Wiimote to rapidly click and release attempting to nudge the widget into the target. Otherwise the Wiimote, mirror and relative tablets were all fairly similar in ther performance with the relative tablet possibly being somewhat slower due to its lower sensitivity.

Similar results were seen in the navigation tasks. Users of the Kinect took longer and were less accurate than users of the other three devices. The combination of the Kinect results in the navigation and drag and drop tasks show that a converting a standard desktop application to an application which would be reasonable to use with a Kinect may not be reasonable. Instead alternative UI interactions should be considered that do not rely on selecting targets via a point but instead might user gestures to scroll through options sequentially and then another gesture to select the option. There are several Xbox 360 applications which use this style of menu navigation system such as the games Dance Central or Kinect Adventures.

The Wiimote proved to be successful both in the drag and drop and the navigation tasks. It would appear that there would be little difficulty almost directly using a UI that was originally designed for the desktop with a Wiimote. The main issue users faced was physical tiredness when having to point at the screen for extended periods of time and jitter of the pointer due to the natural shakiness of users hands. Given these limitations, UI's should try to avoid long tasks and where possible replace them with shorter tasks. To help cope with the natural shaking of users hands UI's should incorporate some sort of smoothing algorithm, avoid small buttons or implement snapping, where the cursor would latch onto interactive elements as it approaches them. Beyond these suggestions, there seems to be little that needs to be changed to accommodate use by the Wiimote.

The mirror tablet was the most successful in the drag and drop task but proved to be less successful in the actual navigation tasks. The main issue users had with this device was the lag in the visual update on the tablet which may have been responsible for the poor navigation performance. Newer hardware or better software for mirroring the display could help mitigate this problem. Additionally other steps could be taken to keep the user more aware of the current state of the program. There could be more audio cues to let the user know when they have successfully or unsuccessfully interacted with an element. These would not be subject to the same delay if they were produced by the main display. Additionally anything that could be processed only would help the situation. So introducing some sort of soft keyboard that was only displayed and processed on the tablet could improve text entry. Users often compared the tablet to an iPhone or iPad so following similar designs as would be used for these devices would likely be a safe bet.

The relative tablet ranked 3rd in the drag and drop task but performed quite well in the navigation tasks. Its functionality is very similar to a track pad on a laptop so it is not surprising that it performed well in a program that was designed primarily for desktops and laptops. There is very little that users suggested in terms of improving the UI for this device with their main complaints being the sensitivity of the track pad. So offering users an option to adjust the sensitivity would be a reasonable solution. Otherwise, following similar design practices for those already employed to make UIs for laptops and desktops makes sense for the relative tablet. The only exception being that text entry should be kept to a minimum as with all devices tested, text entry was not a quick task.

3.6 Future Work

We have identified several areas where current devices seem to perform well and other areas where they seem to be lacking. This leaves many areas open to future work. For the Kinect other methods of interaction need to be considered. Particularly the use of voice commands in a social environment. Many users said they may feel awkward gesturing to command their TV. Would they feel more comfortable controlling their TV with voice commands in a social environment?

The mirror display was popular among users with their main complaint being lag. Reducing the lag between the two displays could provide large gains in user satisfaction. Furthermore looking for additional ways to augment the TV view with additional content on the mirror tablet could yield interesting new ways for users to interact with their media experience.

3.7 Conclusions

As interactive television becomes more commonplace there will be a need for interfaces that support devices better suited for television watching than the standard mouse and keyboard. Beyond this applications are getting more complex as users want to do more than just watch TV, they want to interact socially about their favorite programs or be able to check their email. To facilitate this new user interfaces will need to be designed. We hope to guide the future design of these interfaces through our research.

On one end, we find that users of the relative tablet which is very similar to a trackpad had little difficultly using the XBMC interface and that there were few changes that could be made to improve their experience. On the other end of the spectrum was the Kinect. Users found that it was interesting, but the XBMC interface was very poorly suited to users trying to interact with the Kinect. Interfaces for this device need to be rebuilt from the ground up to require less accuracy and rely more on larger gestures. Bringing in additional modes of input like voice recognition may be worth considering as well.

The Wiimote and the mirror tablet fall somewhere in the middle of this spectrum. Users liked these and generally performed well but with some UI changes these devices could perform much better. For the Wiimote designs need to take into consideration the inaccuracy arising from any shakiness in the users hand. So designs eschewing smaller elements when possible and considering techniques like snapping to interactive elements to improve the experience. There have been many techniques developed to aid in the acquisition of small targets using strategies like "magnetic" cursors or zooming in on the local area. We feel the "sticky" icons approach developed by Worden et al. [13] would not only increase Wiimote accuracy, but also reduce user frustration.

There are also changes that could be made when using a second display as a mirrored display. In this situation the visual lag between the two screens becomes obvious so steps need to be taken to minimize the problems this causes. Trying to offer audible cues to notify the user when an action takes place could help as could adding elements to the 2nd display that are not strictly mirrored so that the processing can take place on that display, like the addition of a soft keyboard run on the tablets software.

3.8 REFERENCES

[1] K. Cheng and M. Takatsuka. "Initial evaluation of a bare-hand interaction technique for large displays using a webcam" In Proc. of the 1st ACM SIGCHI symposium on Engineering interactive computing systems (EICS '09). ACM, New York, NY, USA, 291-296.

[2] J. C. Lee. Hacking the Nintendo Wii Remote. IEEE Pervasive Computing, 7(3):39–45, July 2008.

[3] J. Lin, H. Nishino, T. Kagawa, and K. Utsumiya. "Free hand interface for controlling applications based on Wii remote IR sensor." In Proc. of the 9th ACM SIGGRAPH Conf. on Virtual-Reality Continuum and its Applications in Industry (VRCAI '10). ACM, New York, NY, USA, 139-142.

[4] D. Natapov, S.J. Castellucci, and I.S. MacKenzie. "ISO 9241-9 evaluation of video game controllers." In Proc. of Graphics Interface 2009 (GI '09). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 223-230.

[5] L. Rainie. "Tablet and E-book reader Ownership Nearly Double Over the Holiday Gift-Giving Period" Pew Internet. 2012.

[6] D.C. McCallum and P. Irani. "ARC-Pad: absolute+relative cursor positioning for large displays with a mobile touchscreen." In Proc. of the 22nd annual ACM

symposium on User interface software and technology (UIST '09). ACM, New York, NY, USA, 153-156.

[7] Netflix, “Netflix Q4 2011 Letter to Shareholders.” http://goo.gl/4vgP1

[8] Nielsen. “State of the Media: Consumer Usage Report,” 2011. http://goo.gl/PzAzc

[9] Sandvine, 2011. “Sandvine. Global Internet Phenomena Spotlight: Netflix Rising.” http://goo.gl/dbkPz

[10] Nielsen, 2011. “Nielsen: The Cross-Platform Report.” http://goo.gl/JxEpn

[11] NTIA, 2010. “Exploring the Digital Nation: Home Broadband Adoption in the United States.” http://goo.gl/ErsnS

[12] A. Pavlovych & W. Stuerzlinger. “The tradeoff between spatial jitter and latency in pointing tasks.” In Proc. of the 1st ACM SIGCHI symposium on Engineering interactive computing systems (EICS '09). ACM, New York, NY, USA, 187-196

[13] Worden A, Walker N, Bharat K, Hudson S, 1997 "Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons", in Proceedings of CHI'97 Conference on Human Factors in Computing Systems. (Atlanta, Georgia, March 22-27) pp 266-271

[14] K. Chorianopoulos, G. Lekakos, and D. Spinellis. 2003. Intelligent user interfaces in the living room: usability design for personalized television applications. In Proceedings of the 8th international conference on Intelligent user interfaces (IUI '03). ACM, New York, NY, USA, 230-232.

[15] R. Bernhaupt, M. Obrist, A. Weiss, E. Beck, and M. Tscheligi. 2008. Trends in the living room and beyond: results from ethnographic studies using creative and playful probing. Comput. Entertain. 6, 1, Article 5 (May 2008), 23 pages.

[16] Cesar, P. and Chorianopoulos K. (2009). The Evolution of TV Systems, Content, and Users Toward Interactivity. Foundations and Trends in Human–Computer Interaction: 2(4): 279–374.

[17] L. Pemberton and R. Griffiths. Usability evaluation techniques for interactive television. In Proc. 10th HCI International Conference, June 2003.

[18] Deering, M. 1998. The Limits of Human Vision, in 2nd International Immersive Projection Technology Workshop.

[19] Brown, B., and Barkhuus, L. 2006. The television will be revolutionized: effects of PVRs and filesharing on television watching. In Proceedings of the Conference on Human Factors in Computing Systems (Montreal, Canada, 2006).

[20] B. Lee and R. S. Lee. 1995. How and why people watch TV: implications for the future of interactive television. Journal of Advertising Research. 35.6. p9.

[21] S. Basapur, G. Harboe, H. Mandalia, A. Novak, V. Vuong, and C. Metcalf. 2011. Field trial of a dual device user experience for iTV. In Proceedings of the 9th international interactive conference on Interactive television (EuroITV '11). ACM, New York, NY, USA, 127-136.

[22] A. Oulasvirta and J. Bergstrom-Lehtovirta. “Ease of juggling: studying the effects of manual multitasking.” In Proc. of the 2011 annual conf. on Human factors in computing systems (CHI '11). ACM, New York, NY, USA, 3103-3112.

[23] M. Clancey. 1994. “The Television Audience Examined.” Journal of Advertising Research 34, 4: special insert.

[24] D. Geerts, D. De Grooff. 2009. Supporting the social uses of television: sociability heuristics for social tv. In Proc. of the 27th international conf on Human factors in computing systems (CHI '09). ACM, New York, NY, USA, 595-604.

[25] Nielsen, J., and Molich, R. 1990. Heuristic evaluation of user interfaces. In Proc. CHI'90, ACM, 249-256.

[26] Apple iOS Human Interface Guidelines. http://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/MobileHIG.pdf

[27] Google Android User Interface Guidelines. http://developer.android.com/guide/practices/ui_guidelines/index.html

[28] R. L. Mandryk and C. Lough. 2011. The effects of intended use on target acquisition. In Proc. of the 2011 annual conf on Human factors in computing systems (CHI '11). ACM, New York, NY, USA, 1649-1652.

[29] I. Scott MacKenzie and W. Buxton. 1992. Extending Fitts' law to two-dimensional tasks. In Proc of the SIGCHI conf on Human factors in computing systems (CHI '92), Penny Bauersfeld, John Bennett, and Gene Lynch (Eds.). ACM, New York, NY, USA, 219-226.

[30] Fitts, P. M. 1954. The information capacity of the human motor system in controlling of the amplitude of movement. Journal of Experimental Psychology, 47, 381-391.

4. Conclusion

Interactive television has the potential to increase access and lower the barrier of use for many media services. Part of its success lies in the delivery of an intuitive user interface and input device. We feel our work shines some light on potential pitfalls when choosing an input device and designing a user interface in an iTV context.

The first transcript examined several devices and evaluated their usability with an iTV-like interface. We found the Wiimote generally performed best overall. In a post-study survey, participants also found the Wiimote the least awkward and easiest to learn. The most accurate based on the drag and drop task was the mirror condition. Despite the lag it suffered from, 30% of users commented positively about it. We feel the mirror tablet has potential to achieve better ratings and be more useful, perhaps in future work with an augmented "second screen".

The second transcript focused on developing a set of interface design guidelines by evaluating the performance of each device in the context of target size and target screen location. By designing an interface and input device in the same design cycle, both the device strengths can be taken advantage of and the weaknesses can be better contained. The Wiimote cursor for example suffered from a shaky cursor, a driver solution would be to implement smoothing to dampen the instability. An interface design that would complement this would be to stay away from using smaller buttons or if necessary, use a technique that snaps the cursor to the target. Also, when using a tablet, introducing audible cues to add context to the interact may provide a more intuitive user experience.

Interactive television research is still an evolving field, as is using NUI devices for general purpose navigation. We hope our results help others to find new directions in usability research.

Bibliography

[1] Cooper, W. 2005 Integrating Interactivity into Mainstream Broadcasting. NAB 2005 Broadcast Engineering Conference Proceedings. National Association of Broadcasters

[2] Cesar, P. and Chorianopoulos K. (2009). The Evolution of TV Systems, Content, and Users Toward Interactivity. Foundations and Trends in Human–Computer Interaction: 2(4): 279–374.

[3] NTIA, 2010. “Exploring the Digital Nation: Home Broadband Adoption in the United States.” http://goo.gl/ErsnS

[4] Netflix, “Netflix Q4 2011 Letter to Shareholders.” http://goo.gl/4vgP1