Comparison of Cross-Device Interaction Techniques Between Mobile Devices and Large Displays

Eric V. Ruder

Bjarke M. Lauridsen

AALBORG UNIVERSITY
STUDENT REPORT

AALBORG UNIVERSITY
STUDENT REPORT

Aalborg University,
Department of Computer Science
Selma Lagerlöfsvej 300
Phone 99 40 72 28
http://www.cs.aau.dk

Title of Report:
Comparison of Cross-Device Interaction Techniques Between Mobile Devices and Large Displays

Project period:
IS10, Spring Semester 2016
1st February 2016
to 6th June 2016

Group:
IS105F16

Participants:
Bjarke M. Lauridsen
Eric V. Ruder

Supervisor:
Jeni Paay

Circulation: 4
Number of pages: 69
Appendices: A-E + 1 DVD

Abstract:

Research into cross-device interaction techniques has been gradually increasing through out the last couple of years. One of the focus points of all this research is the interaction between large displays and mobile devices. In this paper, we contribute to this body of knowledge by providing an empirical comparative study of 8 different cross-device interaction techniques: *Push* and *Pull* versions of *Grab*, *Swipe*, *Throw* and *Tilt*. The goal of these techniques was to investigate the different aspects of exchanging data between mobile devices and large displays. We found that *Push* and *Pull* versions of *Swipe* are both the most efficient and accurate techniques, closely followed by the *Push* and *Pull* versions of *Throw*. What these four techniques have in common is the ability to keep the cursor still while performing the technique. This suggest that a large contributing factor to a successful cross device interaction technique is the ability to hold the cursor still while performing the interaction technique.

TABLE OF CONTENTS

2016
AALBORG UNIVERSITY

Preface

This report is the product of our 10th semester work as software engineers with a special in Human Computer Interaction, at the Department of Computer Science at Aalborg University.

The theme of our project is cross-device interaction between mobile devices and large displays.

In this report, we present a research paper based on a comparative study between 8 different cross-device interaction techniques.

A DVD containing the source code of the developed systems is supplied together with this report. The DVD also contains a digital copy of this report as well as appendices with comments, pictures and data analysis results. Additionally, it contains all logged numbers used for the quantitative analysis, enclosed as raw experiment logs and as a SPSS data sheet.

We would like to thank Jeni Paay, Associate Professor at Aalborg University, for providing ongoing feedback and guidance throughout the project

Introduction

Our initial motivation for this and last semester's project was our interest in Cross-Device Interaction, Natural User Interfaces (NUI), and public spaces. One previous masters project that inspired us was a collaborative touch based wall, where users could come up and create music together by touching the wall at different places[1].

We started out by narrowing the subject and coming up with the current theme: cross-device interaction techniques between mobile devices and large displays. We then decided to work on a project were the main idea was to compare different cross-device interaction techniques that would be used to exchange data between mobile devices and large displays.

This is a relatively new field of study and as such, provides us ample opportunity to contribute information to its pool of knowledge. We are excited to be part of this exploration of cross-device, natural user interaction techniques.

The main product of this semester, the paper delivered in the report, presents a research experiment were we explore and compare four different cross-device interaction techniques for pushing data to a large display and four cross-device interaction techniques for pulling data from a large display. These techniques are all techniques that have been used before in research and prototypes, so that we could study something that could build on current understanding of these existing techniques. These techniques are measured in terms of their efficiency and accuracy, in comparison to each other. This is done by creating an experiment in which users are asked to perform each technique and hit targets that are displayed on a large display. We then present and discuss the given results.

[1] Damgaard, Madsen and Sørensen. 2011. *Experiencing music together: Control and Identity* http://goo.gl/yp9J5t

PAPER:

Comparison of Push and Pull Techniques for Cross-Device Interaction Between Mobile Devices and Large Displays

TOWARD CROSS-DEVICE NATURAL USER INTERACTIONS WITH LARGE DISPLAYS

ABSTRACT

RESEARCH INTO CROSS-DEVICE INTERACTION TECHNIQUES HAS BEEN GRADUALLY INCREASING THROUGH OUT THE LAST COUPLE OF YEARS. ONE OF THE FOCUS POINTS OF ALL THIS RESEARCH IS THE INTERACTION BETWEEN LARGE DISPLAYS AND MOBILE DEVICES. IN THIS PAPER, WE CONTRIBUTE TO THIS BODY OF KNOWLEDGE BY PROVIDING AN EMPIRICAL COMPARATIVE STUDY OF 8 DIFFERENT CROSS-DEVICE INTERACTION TECHNIQUES: *Push* AND *Pull* VERSIONS OF *Grab*, *Swipe*, *Throw* AND *Tilt*. THE GOAL OF THESE TECHNIQUES WAS TO INVESTIGATE THE DIFFERENT ASPECTS OF EXCHANGING DATA BETWEEN MOBILE DEVICES AND LARGE DISPLAYS. WE FOUND THAT *Push* AND *Pull* VERSIONS OF *Swipe* ARE BOTH THE MOST EFFICIENT AND ACCURATE TECHNIQUES, CLOSELY FOLLOWED BY THE *Push* AND *Pull* VERSIONS OF *Throw*. WHAT THESE FOUR TECHNIQUES HAVE IN COMMON IS THE ABILITY TO KEEP THE CURSOR STILL WHILE PERFORMING THE TECHNIQUE. THIS SUGGEST THAT A LARGE CONTRIBUTING FACTOR TO A SUCCESSFUL CROSS DEVICE INTERACTION TECHNIQUE IS THE ABILITY TO HOLD THE CURSOR STILL WHILE PERFORMING THE INTERACTION TECHNIQUE.

2016

AALBORG UNIVERSITY

Should I Swipe or Should I Throw?: Comparison of Cross-Device Interaction Techniques Between Mobile Devices and Large Displays

Bjarke M. Lauridsen
Aalborg University
blauri10@student.aau.dk

Eric V. Ruder
Aalborg University
eruder10@student.aau.dk

ABSTRACT

Research into cross-device interaction techniques has been gradually increasing through out the last couple of years. One of the focus points of all this research is the interaction between large displays and mobile devices. In this paper, we contribute to this body of knowledge by providing an empirical comparative study of 8 different cross-device interaction techniques: *Push* and *Pull* versions of *Grab*, *Swipe*, *Throw* and *Tilt*. The goal of these techniques was to investigate the different aspects of exchanging data between mobile devices and large displays. We found that *Push* and *Pull* versions of *Swipe* are both the most efficient and accurate techniques, closely followed by the *Push* and *Pull* versions of *Throw*. What these four techniques have in common is the ability to keep the cursor still while performing the technique. This suggest that a large contributing factor to a successful cross device interaction technique is the ability to hold the cursor still while performing the interaction technique.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation: Miscellaneous

Author Keywords

Interaction Techniques; Cross-Device Interaction; Natural User Interaction; Kinect; Mid-air Gestures; Smartphones; Large Displays; Data Transfer

1. INTRODUCTION

In line with technological advances, digital displays, for both domestic and public use, are available in increasingly larger sizes than just a few years ago. It has been shown that interacting with systems that use these large displays benefit from using mid-air pointing from a distance, making it possible to use hand gestures to navigate around the screen [10]. Mid-air interactions are being increasingly used for different applications, e.g., in gaming or virtual and augmented reality. Inexpensive consumer depth cameras, like the popular Microsoft Kinect, allow players to control video games using mid-air gestures. The Kinect has also been popular with researchers who have used them to explore new interfaces for applications and new human computer interaction possibilities, e.g., mid-air pointing or bodily movement to communicate with the system.

If we introduce a smartphone into this mid-air pointing and large display interaction, we enable the transfer of information to and from the display, e.g., text, videos, images and other media, between the smartphone and the system. But how do we decide which interaction techniques people should use with their mobile devices to make the transfer happen? Different kinds of mid-air techniques have been studied in different situations in the literature but we have not found any existing empirical research that compares alternative mid-air techniques using smartphones for two-way interaction with large displays. An understanding of how different techniques compare with each other in terms of effectiveness, efficiency and accuracy could help interaction designers make decisions about which interactions to implement in their systems. Interactions to support users pushing or pulling information between the system and their mobile devices could be designed based on which of these attributes is most important in a particular application context, for example, on a digital public notice board accuracy could be more important than efficiency. Another important aspect for interaction designers could be the "naturalness" or learnability of alternative techniques, for example, in a walk up and use scenario for a large digital display located in a public park.

To contribute to current knowledge on these issues, we have conducted an empirical study that compared and analyzed the data collected from an experiment on 8 different interaction techniques, that is, 4 techniques for push (from smartphone to display) and 4 techniques for pull (from display to smartphone). By comparing these techniques on parameters of hit rate, time taken, and distance from target, with two studies in a controlled environment, we are able to demonstrate which techniques are more precise than others and which are faster to complete.

In this paper we report on our findings from both studies, showing that a swiping technique is the most effective and accurate for both pushing and pulling data between a large display and a smart phone, compared to the other 6 techniques. We also show that throwing techniques closely follow in regards to an efficient and accurate exchange of data between the display and phone.

2. RELATED WORK

In this section we present related work which has been done in the area of techniques for transferring information and data between mobile devices and displays. We also present developments within the topics of pointing in mid-air and controlling a cursor on a display at a distance.

2.1 Large displays & Mid-air pointing

The interaction techniques for large displays most commonly used are touch and mid-air pointing. For pointing in mid-air there are different technologies such as Microsoft Kinect and Microsofts's new mixed reality glasses, named HoloLens. The Kinect sensor uses motion sensing technology and allows the user to control a game or an application with their body by pointing and gesturing. With HoloLens, the controlling interface is hand gestures combining the physical 3D world with the virtual or augmented reality made possible with the HoloLens. Using mid-air gestures can also make the physical space we move around in combine more seamlessly with what we see on a large display.

In 1984, Bolt [2] demonstrated a system utilizing voice recognition and gesture input for creating, moving, deleting, and manipulating objects on a large screen. The system combines the two technologies and creates an interface capable of receiving voice commands i.e. *"Create a blue square"* and coupling it with a pointing gesture and the word *"there"*.

Using a large display and mid-air pointing, Markussen et al. [13] explored an interaction concept called *Off-Limits* in which the user is able to interact with a large display outside the boundaries of the screen. The results show that, for off-screen interaction, touch is slower than using mid-air techniques but participants who used mid-air would undershoot their targets. This problem resulted in a model that corrects for undershooting thus creating a better mapping between where participants would like to point and where they actually point. The study showed that *Off-Limits* outperformed the naive implementation of an off-screen technique by being faster and requiring fewer interactions.

Jacobsen et al. [10] explore two different interaction interfaces for large displays, namely touch and mid-air gestures. With two experiments they aim to find out when users choose one interface over the other. The first experiment aims to compare touch and mid-air gestures and the results showed a high error rate for both, while the target selection time for mid-air was 40% more than for touch. Participants were given questions on subjective satisfaction and the results showed that 12 of 19 preferred touch and 7 of 19 preferred mid-air. In a second experiment users were free to choose which interface to use and during the experiment physical movements were required to simulate circumstances where it would be necessary to move away from the screen. The results revealed that in 42% of the trials made, participants chose to use mid-air gestures and that for medium to large target sizes where users were asked to move to and from the display, mid-air was used more often than touch. With 7 out of 10 participants preferring mid-air gestures for the second experiment, the cost of moving back and forth to use the touch interface would seem to make mid-air pointing preferable.

Bragdon et al. [4] created a system called Code Space to support developer meetings. The system uses cross-device interaction techniques for people to interactively participate and contribute to meetings by using hand gestures to point at and manipulate the content on the shared screen. They also use their handhelds and laptops to push and pull content to and from the shared screen. Techniques for manipulating objects include mid-air finger pointing and also mid-air phone pointing to move objects on the screen. Techniques for sharing objects include pointing with the phone and swiping to push and pull objects to and from the shared display. Another technique is transient sharing from e.g. a handheld device, which is performed by holding the device's screen perpendicular with the floor to share an object. A pilot evaluation was conducted and feedback from the participants indicated an overall positive attitude towards the system with comments such as *"this is awesome", "cool", "this is Minority Report stuff, I love it", "everyone can participate"*. Also, participants generally felt that the interactions were socially acceptable to perform within their team of fellow developers.

2.2 Target acquisition using hands

In the literature, research on different approaches to pointing and controlling, e.g, virtual pointers on a screen using hands and fingers, has been done by Mayer et al. [15] and Vogel et al. [22] among others. Mayer et al. present a study on 3 techniques for absolute distant pointing without visual feedback to see how precise participants were with the three techniques and if the precision could be improved. They found that for the most precise technique (*index finger ray casting*) the average error was 61.3 cm before applying a correction model. Using a correction model on the same technique they found that the average error could be reduced by 37.3% for both sitting and standing at different distances from the display. This means that pointing techniques without visual feedback can benefit greatly from correction models. Vogel et al. experimented with three pointing techniques to acquire targets on a very large display with high resolution and their findings show that a *RayCasting* technique (extends a ray from the index finger) is significantly faster than two other techniques, *Relative* and *RayToRelative* (a combination of *RayCasting* and *Relative*). In contrast, error rates are far lower for the two relative techniques and the largest difference between *RayCasting* and the relative techniques are for medium to small target sizes indicating that *RayCasting* is less accurate for smaller targets.

Hespanhol et al. [7] proposes a set of five mid-air gestures to perform selections and rearrange items on a large display using only hands. Each of their gestures are described with a scenario in which a given gesture is commonly used, e.g., a *push* gesture for pressing a button or a *grab* gesture with the scenario of grabbing a physical object which can then also be moved around. Results showed that *dwelling* and *grabbing* were the two fastest gestures for selecting and rearranging respectively and they were also the two gestures with the lowest amount of failures for both tasks.

2.3 Target acquisition using handheld devices

Boring et al. [3] experimented with using mobile phones to control a cursor on large displays with three different interaction techniques to move the cursor. The three techniques were *Scroll* (using buttons on the phone to move pointer), *Tilt* (tilting the phone to move pointer), and *Move* (moving the phone in a direction to move the pointer in the same direction). In the experiment each technique was tested and participants had to acquire a number of targets with long or short distances between them and with three different target sizes to hit. Results showed that for both larger target sizes and smaller distances the selection times were lower while the fastest technique for both distance and size were *Move*. For error rates, the results show that short distances and larger target sizes reduce the error rate while the technique with lowest error rate was *Scroll*. Questionnaires showed that for general comfort there was no trend towards any single technique but *Scroll* was significantly "easier to use" than *Tilt*, which participants seemed frustrated using.

A study by Nancel et al. [17] focuses on using handheld devices and high precision pointing techniques for acquiring targets on a large wall sized display. Their implementation uses the handheld device for controlling the pointer on the large display and a small area of the handheld device is used for relative pointing. One technique uses two fingers for coarse pointing and one finger for precision pointing. Another technique uses a head-based coarse pointing technique making it possible for the participants to roughly get the pointer close to the target using their head. They found that continuous head pointing is faster and more successful than their other techniques. A comparison showed that their technique performed as good as some state-of-the-art techniques such as LaserGyro[22] and SmoothPoint[6]. They showed that it was in fact possible to maintain precision and ample screen real estate on the handheld device.

Rashid et al. [18] explore two different techniques for interacting with and acquiring targets on a large display using a handheld device. The first technique is *Proximal Selection (PS)* which pulls a selected, or zoomed in, area of the large display onto the phone and the user is then able to select the correct target. The second technique is *Distal Selection (DS)* where the user points at the large display, zooms in on the selected area, and finally selects the desired target on the large display. In their experiment they found that, for complex tasks and with regards to time, *PS* outperforms *DS* but for simpler tasks *DS* was approximately 0.1 seconds faster but the effect was not significant. The error rate for the techniques showed that *DS* had fewer missed clicks for both small and large targets and that *DS* had a significantly lower error rate than *PS* only for small targets.

2.4 Data transfer & Interaction using handheld devices

Techniques for interacting with large displays using handheld devices are numerous and includes smartphones, gyro mice, game controllers (both with and without gyroscopes and accelerometers), and devices fitted with lasers. Mid-Air pointing in the beginning of the 21st century used laser pointers to interact with and select objects on large displays from a distance. Myers et al. [16] experimented with laser pointers for target acquisition, measured time, accuracy, and how good participants were to dwell on a target using the different lasers. They tested four devices (2 laser pointer, 1 Palm PC fitted with a laser, and 1 toy gun fitted with a laser) and different ways to hold them. The results showed that holding the Palm PC with one and two hands were the most stable but the Palm PC was also the one users found most cumbersome and heavy. They also did an experiment comparing 4 ways of selecting objects on a large display. The technique with the fastest selection time and the lowest error rate was touching directly on the SmartBoard used in the experiment and the laser pointer was slowest and had the second highest error rate. Based on the results, a suggestion was made to explore combining laser pointers with other techniques and use the laser to make a coarse grained selection and other techniques as the fine grained selection.

One of the earlier examples of transferring data using handheld devices is presented by Rekimoto [19]. The system presented is called *Pick-and-Drop* and revolves around a handheld display and a pen capable of picking up objects on one device and transferring it to another device. The "Pick-and-Drop" metaphor is closely related to real world objects where, for example, a piece of paper is picked up from one table and placed on another table. Another implementation uses wall-sized displays as a common workplace for participants and the interaction between participants' PDAs and the wall-sized display would use *Pick-and-Drop*.

Approaches to transferring data by gesturing with handheld device have been documented in the literature and amongst them are throw and tilt gestures using handheld devices for interacting with large displays. Dachselt et al. [5] and Boring et al. [3] describe how a tilt technique with a handheld phone can be used to control a pointer on a remote display. In addition, Dachselt et al. describe a throwing gesture for transferring data (e.g. from a phone) to and from a large display and the application proposed also uses the concept of transferring an entire user interface between a phone and a display using the throw gesture. The idea behind transferring the entire interface is to allow seamless interaction between large display and phone and subsequently improve usability (phone to display) or mobility (display to phone).

Seifert et al. [21] presents *PointerPhone* that explores techniques for pointing at a remote screen with a mobile phone and also interacting with the screen at a distance using the phone. *PointerPhone* was qualitatively evaluated by asking participants to perform a set tasks. Observations and feedback from the evaluation sessions resulted in a set of design implications, one of which provides recommendations for *One- and two-handed interactions*. They found that when a task requires precise pointing users would hold the phone with two hands and when the task did not require precision, users would just use one hand. Zadow [23] is exploring cross-device data transfer, specifically bidirectional interaction techniques for transferring data between mobile phones and display walls, but they report no specific findings.

3. INTERACTION TECHNIQUES

In this section we illustrate and describe the interaction techniques we implemented in order to exchange data between mobile phones and large displays. This is done so that we can compare them to one another empirically.

There were several criteria behind the choice of these techniques. Table 1 shows the set of criteria that we based our choice of techniques on.

Attributes	Description
Number of hands	There must be both one-handed and two-handed techniques.
Previously used	To avoid designing and testing a set of novel techniques, we had the criterion that all techniques have been used by others in research studies.
Complexity	The techniques must differ in their complexity and therefore we included techniques with different amount of steps.
Activation method	The way each technique is activated must be different from each other.
Natural feel	There must be a natural and intuitive feel to the techniques in some way.

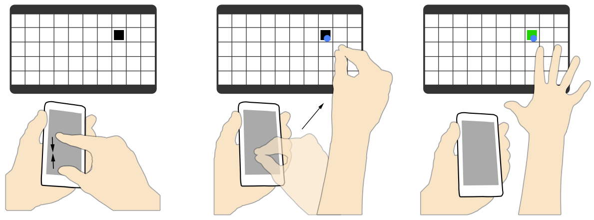
Table 1. Criteria for selection

The techniques chosen were found in the literature, some with minor modifications. We then created a symmetrical version of each technique so they would have both a *Push* and *Pull* version. *Push* means that the user is pushing data from the mobile to the large display, and *Pull* means the user will pull data from the screen onto the mobile device.

Eight techniques were used in the experiment: *Grab*, *Swipe*, *Throw* and *Tilt*, each with a *Push* and *Pull* variant.

Grab

The *Grab* technique is based on a grabbing gesture presented by Hespanhol et al. [7] as one of five proposed gestures. A related technique is described by Markussen et al. [14]. They present a mid-air word-gesture keyboard named *"Vulture"* that uses a pinch gesture (touching index finger to thumb) to give the user control of the pointer which can then be used to select letters. A variation of the *Grab* technique (fig. 1) is used in *Memory Stones* [9] by Ikemasu et al. as part of a system for exchanging information between different devices. Benko and Wilson [1] used the *Grab* technique in a system where the user interacts with visualizations inside a dome. *Grab* is a combination of the grabbing gesture and the pointing technique used by Scheible et al. [20]. This technique was chosen because we wanted to simulate the feeling of picking up an object of interest and placing it in a desired location. *Grab* is a complex technique, requiring a series of steps as well as using both hands to complete the interaction. The *Push* version of this technique is completed as follows: the user first grabs an object of interest from the telephone by pinching it with his fingers (fig. 1a), closing his hand, and metaphorically putting the object in his hands. The user then raises his closed hand and uses it as a pointer on the screen to indicate where he wants to place the object (fig. 1b). The final step is to open the hand and release the object onto the large display, where he was aiming (fig. 1c). The *Pull* version is a bit different. The user first places his open hand, used as a pointer, over the position of the object of interest on the large display (fig. 1c). The user then closes his hand over the object (fig. 1b) and finally places it on his phone by touching the screen with his closed hand (fig. 1b).

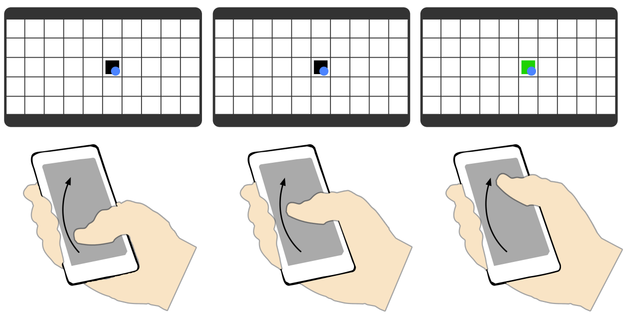


(a) (b) (c)

Figure 1. *Push Grab* technique

Swipe

The *Swipe* technique (fig. 2) was utilized by Bragdon et. al in *Code Space* [4]. They developed a system that would support developer meetings with the help of smart phones and the Kinect. Bragdon et al. describe the technique as *"cross-device interaction with touch and air pointing"* and the swiping motion is described as *"flicking up on the touch screen"*. Seifert et al. [21] also describe *Swipe* for both the *Push* and *Pull* directions with respect to data exchange between a large display and a mobile phone. This technique was chosen because it has a very simplistic design, with a very low level of complexity since it requires very few steps to activate. It is also a one handed technique and requires very little effort from the user to use. The *Push* and *Pull* version of this technique are very similar. First the user points at the desired location with the phone using a stretched arm (fig. 2a) and then swipes his finger on the screen (fig. 2c). The direction he swipes depends on whether the user wants to *Push* or *Pull* information. If he swipes away from himself, he pushes information to the screen. If he swipes towards himself, he is pulling information from the screen onto the device.



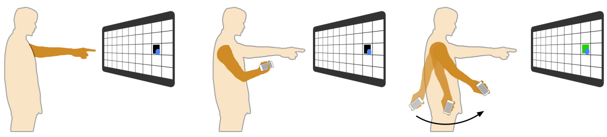
(a) (b) (c)

Figure 2. *Push Swipe* technique

Throw

The *Throw* technique (fig. 3) is a combination of two techniques. The first is a pointing technique used by Walter et al.

in *Cuenesics* [24] were the user uses his hands to control a cursor on the screen. The second is a technique used by Yatani et al. [25] in *Toss-it* to throw information between handheld devices and by Scheible et al. in *MobiToss* [20] were the system is used to submit information onto a large public display. This technique was chosen because of its natural and playful design, as well as mirroring the idea of throwing something, like a ball, somewhere or to someone. *Throw* is a two handed technique, as well as having a larger number of steps to take in order to activate it. The *Throw* technique is performed as follows: First the user points at the targeted position with a finger (fig. 3a). Then, if the user is pushing data from the phone, he has to select the data to be pushed (fig. 3b). The user then performs a swinging motion with the hand which is holding the phone. If he is pushing, then he swings towards the screen(fig. 3c), if he is pulling he swings away from it.

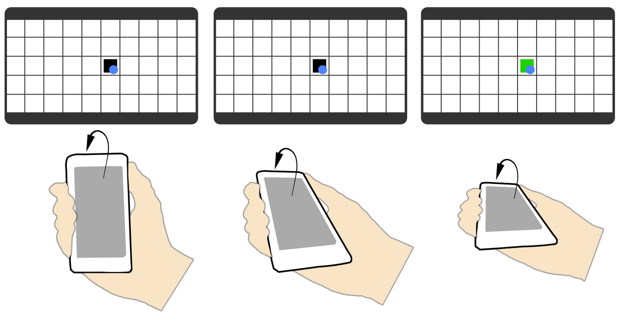


(a) (b) (c)

Figure 3. *Push Throw* technique

Tilt

The *Tilt* technique (fig. 4) is used by Lucero et al. in *MobiComics* [12]. They created a system in which users would transfer objects from a large display onto their mobile devices. The *Tilt* technique was chosen because it is a one handed, low complexity technique with few steps needed to activate it. Just like the *Swipe* technique it is easy, straight forward and feels natural to use. The *Tilt* technique is performed as follows: The user first points at the target location with the phone (fig. 4a). If the user is performing a *Push* technique, he tilts the phone away from himself(fig. 4b, fig. 4c). If he is performing a *Pull* technique, he does the opposite and tilts the phone towards himself.



(a) (b) (c)

Figure 4. *Push Tilt* technique

All techniques that were chosen have been used in other research systems to facilitate the interaction between mobile devices and large public displays. They differ in the way they are activated as well as the number of hands that are used to perform the technique. For our two handed techniques, we have *Throw* and *Grab*, and for one handed techniques we have *Tilt* and *Swipe*. Each technique also has different ways to activate it. The *Throw* and *Tilt* techniques require the user to move the phone, whereas the *Swipe* and *Grab* techniques require the user to touch and manipulate the screen of the device in order to activate them.

All the techniques used in this experiment require some combination of mid-air pointing, touch gestures and phone movements to perform. These were all implemented using the Microsoft Kinect V2 as well as the accelerometer and touch sensors on the phone. The Kinect utilizes its depth camera in order to give information about a users location in physical space, allowing us to track the position of the users hands and building the techniques around that. The touch sensor on the phone was used in order to recognize touch and swipe gestures. The accelerometer was used to detect significant movement on the phone and use that for the *Tilt* and *Throw* techniques. A detailed implementation of the system can be found in [11].

4. EXPERIMENTS

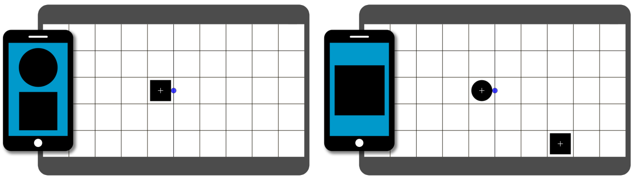
In order to compare different techniques to each other we conducted an experiment in which each participant would perform the techniques in a controlled laboratory environment. There are two studies to this experiment because after we finished the first study we realized that before we could say anything about accuracy in pixels we would have to refine the experimental design to require people to aim at a precise point and tell the participants to be as precise as they could. Not having told our first 51 participants to aim and be precise we repeated the study, adding a precise point to aim at for each target. The first study will be referred to as *Target-Study*, and the second will be referred to as *Accuracy-Study*.

4.1 Experimental Task

The interaction techniques were implemented in a test application were the main goal was to correctly put shapes onto the large display from the smart phone or pull them away from the large display and put them on the phone. The shapes would represent data, and two shapes(square and circle) were used in order to simulate the effect of choice of data.

A grid system was implemented, were each target could be located in a particular grid cell. The grid had two different sizes. One was a large cell system, which had 5×10 cells, and each cell measured 122 pixels (14.6 cm) on each side. The other was a small cell system, with 10×20 cells, each cell measuring 61 pixels (7.3 cm) on each side.

Shapes would appear in the cells, and the sides of the square shapes and the diameter of the circle shapes are both 80% of the cell width. These shapes would appear according to a predetermined series of locations, so that we could ensure that each target had an equal distribution of distances between them. For the user, the sequence would appear as random. These shapes would represent targets which the user had to hit. A blue circle was used, as a cursor, to represent were the user is pointing on the screen. This is where the user would “hit” whenever he performed an attempt with a given technique.



(a) *Push* (b) *Pull*

Figure 5. The screens on both the large display and the phone for the *Accuracy-Study*.

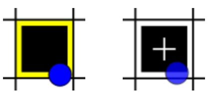
During the *Push* phase of the experiment, users would be presented with one shape on the screen and two on the smartphone, as seen in Figure 5a. Here, the screen would tell the users what the correct shape was, and the user had to perform the technique by selecting the correct shape on the phone. During the *Pull* phase of the experiment, users would be presented with two shapes on the large screen, and one shape on the smart phone. Here, the phone would be telling users what the correct shape was and users would have to pull the correct shape from the large display. In the experiment, we conducted two separate studies, using the same implementation of techniques and experimental method.

Study One: Hit rate and time taken

During the *Target-Study*, the circle used as a cursor was a solid blue color. There was also a bright yellow highlight in whichever cell the cursor was currently over. This can be seen in Figure 6a. This was for providing feedback to the user about whether or not he was hitting the correct cell.

Study Two: Precise distance from target

During the *Accuracy-Study*, we added a white cross to each target to provide a precise point for the users to aim at. The circle cursor was also made opaque so that users could better see the cross when the cursor was over it, as well as removing the highlight so that users would not feel that hitting any part of the cell was acceptable. This can be seen in Figure 6b. In study two, we explicitly asked users to be as precise as possible while performing each technique.

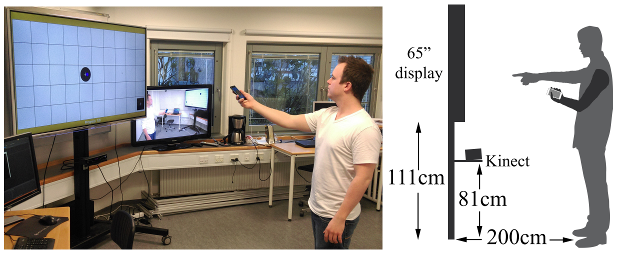


(a) (b)

Figure 6. (a) The targets in the first part of the experiment. (b) The targets in the second part of the experiment.

4.2 Experimental Setup

The experiment was conducted in laboratory setting where we setup a large 65” screen (1920×1080 pixels) and a smaller 42” screen (1024×768 pixels) as seen in Figure 7a. A Microsoft Kinect v2 was mounted below the large display (81 cm above the floor) and we marked the floor with a cross (200 cm from the Kinect) where participants were instructed to stand. We chose 200 cm from the display because this is an optimal operating distance for the Kinect. The height of the Kinect with regards to the floor was chosen through physical adjustment to get the optimal position for a person who is 180 cm tall (based on an estimation of average user height for Danish participants). The phone we used in this experiment was a Samsung Galaxy S2 (4.3” screen). The setup is illustrated in Figure 7b.



(a) (b)

Figure 7. (a) the setup in the usability lab. (b) illustrates the setup and distance between participant and screen, floor and Kinect, and floor and the bottom of the large display.

4.3 Experimental Design

A within-group design was used for the experiment and each participant used all 8 techniques once during the experiment. We have 2 different target sizes (large, small) and 8 techniques (push = 4, pull = 4) as the independent variables. For each technique there are a total of 18 targets and 3 practice targets at the beginning of each technique. Practice targets allow the participants to get familiar with the technique before we start collecting data for a technique. The order in which the participants are presented with a technique is randomized to minimize the learning effect of each technique. For both *Target-Study* and *Accuracy-Study* 51% of the participants started with *Push* techniques and 49% started with *Pull* techniques. For *Target-Study* 27% of the participants started the test with *Grab*, 21% with *Swipe*, 25% with *Throw*, and 27% with *Tilt*. Similarly, for *Accuracy-Study* 22% of the participants started the test with *Grab*, 27% with *Swipe*, 24% with *Throw*, and 27% with *Tilt*. For *Target-Study* the total amount of attempts collected were 2 target sizes × 8 techniques × 9 repetitions × 51 participants = 7344 attempts. For *Accuracy-Study* the total is 2 target sizes × 8 techniques × 9 repetitions × 33 participants = 4752 attempts.

Participants

For the *Target-Study*, 51 people took part. They ranged in age from 21 to 52 (M: 27.98) and were between 1.56m and 1.98m (M: 1.79m). 15.7% of participants were women, and 84.3% were male. All of the participants owned smartphones and had owned one for 2-12 years (M:5.9).

For the *Accuracy-Study*, 33 people took part. They were between 20 and 55 years old (M: 23.18) and were between 1.56m and 2m tall (M: 1.77m). 30.3% of the participants were female, while 69.7% were male. They had all owned smartphones for between 1 and 9 years (M: 5.5).

Task & Procedure

The task and procedure were the same for both studies, with the additional instructions in *Accuracy-Study* to be as precise as possible. Before a participant starts the experiment, the general purpose of the study is explained to the participant and we inform them about what is going to happen. Then a demonstration video of a technique is shown on the large screen and after watching the video it plays in a loop on the small screen. The participant is then presented with the grid and one target after another will appear in the grid until the participant has attempted to hit all targets. After completing one technique the remaining techniques follow with the same procedure as the first until all 8 techniques have been performed. When a participant has done all the techniques we give them a short demographic questionnaire including age, height, gender, current phone, year of first smartphone, and if they have had prior experience with systems like the Nintendo Wii or the Microsoft Kinect. The average time for completing the experiment per participant was 25±5 minutes.

5. RESULTS

The following section presents the results that were gathered throughout the experiment. Here we aim to uncover what variables in our studies were of significance. Some data points needed to be removed from the experiments, in order to gain a clearer understanding of the experimental outcomes.

For the *Target-Study*, we started with a total of 7344 attempts. 176 were removed because of system errors, where the system wrongly activated a technique attempt even though the user did not intend to do so. Another 406 attempts were removed as outliers using the Outlier Labeling method described by Hoaglin and Iglewicz in Resistant Rules for Outlier Labeling [8]. This gave us a total of 6762 attempts for the first study.

For the *Accuracy-Study*, we started with a total of 4752 attempts. 111 attempts were removed due to system errors. Another 138 attempts were removed with the same Outlier Labeling method used above. This gave us a total of 4503 attempts for the second study.

Table 2 shows the final number of attempts each technique had in the two different studies.

	STUDY ONE					STUDY TWO			
	Grab	Swipe	Throw	Tilt		Grab	Swipe	Throw	Tilt
Push	830	835	814	862	Push	551	576	561	582
Pull	784	893	867	877	Pull	533	576	569	555

Table 2. Number of attempts for each technique in each study

5.1 STUDY ONE: Success rate

Here we present results relating to whether or not the user was successful in hit the target, which we will be referring to as effectiveness when discussing the results in the *Target-Study*.

To see whether or not each technique had an association with the success rate of each attempt, and was not just noise, we performed a Pearsons Chi-Square test. For the *Push* techniques, $\chi^2(3) = 121.950$, $p < 0.001$, and for the *Pull* techniques we got $\chi^2(3) = 438.473$, $p < 0.001$. This means that both *Push* and *Pull* techniques had an association with the success rate of each attempt. Table 3 shows the success rate for each of the techniques.

	Hit Success Means			
	Grab	Swipe	Throw	Tilt
Push	95.9%	96%	93.2%	83.3%
Pull	94%	97.5%	96.7%	71.8%

Table 3. Success rate for each technique in *Target-Study*

We then looked at each technique individually to see where the association may be. Our post-hoc tests show that the *Throw Push* technique did not have a significant association ($p = 0.55$) with the success rate of each attempt. All other techniques had a significant association ($p < 0.004$) with the success rate of each attempt.

5.2 STUDY ONE: Time taken

Here, we present results in regards to how long each user took in performing each technique in the *Target-Study*. When discussing these results, we will be referring to a technique's efficiency.

We performed a linear mixed effects model analysis on the data to see how time was affected by the different aspects of our experiment. We found that neither *Effectiveness* nor *Direction* had an effect on the time each user took per attempt. However, the *Target Size* ($F_{1,6695.228} = 91.634, p < 0.001$) and the *Technique* ($F_{1,6695.228} = 91.634, p < 0.001$) did have significant effects on the time taken. We performed a post-hoc LSD pairwise comparison to see how each technique differed from one another and found that all techniques were significantly different ($p < 0.001$) from each other.

We also found that there were other interactions between the variables that were affecting the time for each attempt differently. $Direction \times Technique$ ($F_{3,6694.657} = 52.272, p < 0.001$), $Effectiveness \times Technique$ ($F_{3,6696.169} = 5.227, p < 0.001$) and finally $Effectiveness \times Direction \times Technique$ ($F_{3,6696.038} = 10.235, p < 0.001$) all showed to be significant interactions. A post-hoc LSD pairwise comparison on *Direction* and *Technique* showed that for each technique the difference in time between *Push* and *Pull* were significant. We then did a post-hoc LSD pairwise comparison on *Effectiveness* for each *Technique* and *Direction* to see where that significance was and the only significantly different pair was between a successful and unsuccessful attempt of the *Grab Pull* technique ($p < 0.001$) meaning that an unsuccessful *Grab Pull* takes a significantly longer time to perform than a successful *Grab Pull*.

Table 4 and Figure 8 shows the mean time taken for each technique and direction.

	Efficiency Means			
	Grab	**Swipe**	**Throw**	**Tilt**
Push	5.05 (1.39)	3.9 (0.98)	4.68 (1.11)	5.06 (1.79)
Pull	5.74 (1.57)	3.67 (0.86)	4.12 (1.01)	4.55 (1.61)

Table 4. Efficiency means and (standard deviation) in seconds for each technique in *Target-Study*.

	Accuracy Means			
	Grab	**Swipe**	**Throw**	**Tilt**
Push	16.8 (10.5)	14.14 (9.32)	16.24 (10.29)	28.03 (19.7)
Pull	17.65 (10.92)	12.4 (8.59)	15.07 (10.25)	32.71 (21.49)

Table 5. Accuracy means and (standard deviation) in pixels for each technique in *Accuracy-Study*.

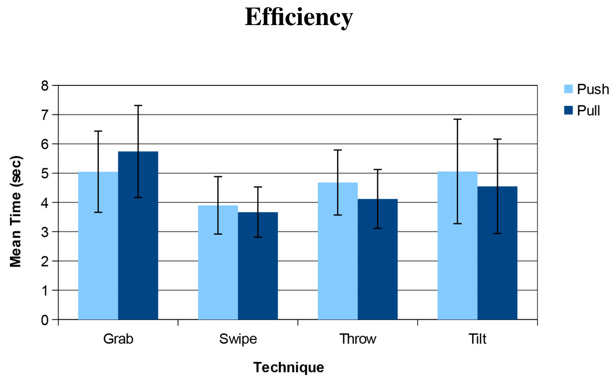


Figure 8. Mean and standard deviation (seconds) for each technique in regards to efficiency.

5.3 STUDY TWO: Distance from target

Here we present the results in regards to how far away from the center of the target (in pixels) each user was when performing the technique in the *Accuracy-Study*. This is referred to as a techniques accuracy when we discuss the results.

We performed a linear mixed effects model analysis on the data to see if each technique had a significant effect on the accuracy of each attempt.

We found that *Technique* ($F_{3,4458.26} = 193.869, p < 0.001$) and *Target Size* ($F_{1,4462.203} = 100.016, p < 0.001$) had an effect on accuracy, but the *Direction* of each technique did not have an effect on accuracy. We then performed a post-hoc LSD pairwise comparison to see were the differences lay, and found that all techniques were significantly different from each other ($p < 0.003$ for all).

We also found that the *Direction* × *Technique* interaction had a significant effect on the accuracy ($F_{3,4457.354} = 8.882, p < 0.001$). We then performed another LSD pairwise comparison between *Technique* for each *Direction*. We found that the only pair that was not significantly different from each other was between the techniques *Grab Push* and *Throw Push* ($p = 0.508$). All others were significantly different from one another ($p < 0.004$). Lastly, we did a LSD pairwise comparison between *Direction* for each *Technique* and the only pair not significantly different from each other were *Grab Push* and *Grab Pull* ($p = 0.355$). The difference between *Pull* and *Push* for all other techniques were significant ($p < 0.044$).

Table 5 and Figure 9 shows the mean distance from the center for each technique and direction.

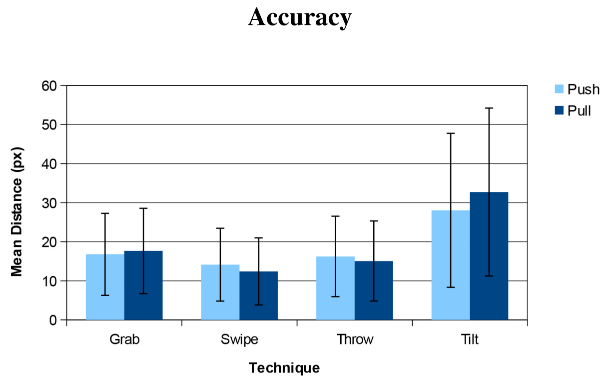


Figure 9. Mean and standard deviation (pixel) for each technique in regards to accuracy.

6. DISCUSSION

When discussing the results from our experiment, it is important to make sure that all terms used are clearly defined. When we talk about *Effectiveness*, we refer to a technique's success rate, or how often it successfully hit the target on the screen. When talking about *Efficiency*, we refer to how fast a technique is to perform. Finally, when we talk about *Accuracy*, we are talking about how precise the technique is, or how close to the center of the target the attempt for the technique was.

6.1 Effectiveness

Our results for effectiveness, or the success rate of each technique, show that there is in fact some association between each technique and whether or not an attempt was successful. Success is a very simple term and as such a great deal of information about the success of a technique is lost when the answer is a simple hit or miss. In order to better understand the association between a technique and a successful attempt, we can use accuracy as a more specific and precise definition of success. The closer an attempt is from the center of the target, the more successful it is. We will therefore use accuracy as more precise measure of a techniques success (see section 6.3).

6.2 Efficiency

Once we start looking at the results regarding the efficiency of each technique, some rather interesting things come to light.

We see that *Swipe* is the fastest technique by far. With a mean time per attempt of 3.67 seconds and 3.9 seconds for *Pull* and *Push* respectively, it outperforms the next fastest technique, *Throw*, with a mean time of 4.12 seconds and 4.68 seconds per attempt for *Pull* and *Push* respectively. One might have expected both one handed techniques to be the fastest, since they have very few steps that must be taken to activate these techniques, but *Throw* is faster than *Tilt*. This can be due to a fact that people felt very uncomfortable with the *Tilt* technique. When performing it, users would be very cautious because the cursor tended to move a large distance after each attempt. This would lead to people performing a very cautious and slow tilting movement with the phone, and thus not triggering the technique and having to perform it multiple times. Not surprisingly *Grab* is the slowest technique, since this is by far the most complex technique, with the most number of steps needed in order to perform it.

Another interesting thing to notice is that a success did not, in fact, have a significant effect on the time for each attempt. This makes sense when considering that the aiming part of each technique is shared between all techniques and as such should take the same amount of time. Once the user has acquired the target, he starts performing the technique, and this is where the difference in time comes from.

As such, it is not surprising to see that the target size does have a significant effect on time. The aiming process for the smaller targets takes longer, and affects all techniques similarly.

We also note that the direction of each attempt did not have a significant effect on the efficiency. This is because not all techniques are affected similarly by the direction. While *Swipe*, *Throw* and *Tilt* all take less time to perform while pulling, *Grab* has the opposite effect. This can be explained once we examine the interaction between direction, technique and success. Our pairwise comparisons show that the significant interaction between direction, technique and success comes the from difference between the successful and unsuccessful attempts for the *Grab Pull* technique. Here it shows that an unsuccessful attempt with the *Grab Pull* is significantly slower than a successful one. This can be explained by the implementation of the technique. Once the user closes his hand in an attempt to grab the shape on the screen, he is no longer allowed to retry, even if he missed. This is counter-intuitive to reality, where if someone was to miss a object he was grabbing, he would simply open his hand and retry. This confused our participants. Whenever they missed, as they would try to open their hand in order to get another attempt at grabbing the objects, but the system would not allow that. As such, the *Grab Pull* technique did not live up to the metaphor of grabbing and releasing objects in real life. This was an intentional design decision though, since none of the other techniques had the opportunity of retrying the attempt. If we had allowed *Grab Pull* to do so, we would most likely have ended up with a perfect hit rate for that technique.

6.3 Accuracy

We see that out of the 8 techniques, *Swipe* was the most accurate technique, having a mean of 12.4 pixels and 14.14 pixels from the center of each target for *Pull* and *Push* respectively. This is closely followed by *Throw*, with 15.07 pixels and 16.24 pixels for *Pull* and *Push* respectively, and *Grab*, with 16.8 pixels and 17.65 pixels for *Push* and *Pull* respectively. The *Tilt* technique trails far behind, with a mean distance of 28.03 pixels and 32.71 pixels from the center of each target for *Push* and *Pull* respectively. It is no surprise that *Tilt* is so far behind, since it requires the user to do a very subtle movement with the same hand they are pointing with. This usually leads to the cursor being moved away from its original placement since it is very hard to adjust the placement of the hand in such a way as to take into account the required activation movement. Several users attempted to place the cursor slightly below the target to compensate for the movement.

Even though we asked our participants to be as precise as possible in the *Accuracy-Study* and aim for the white cross in the middle of the target, we see that the target size actually does have an effect. This is likely related to the idea that the shape itself is still the target. Once the users aim as close as possible to the center of the shape, they prepare themselves to perform the gesture. The cursor also starts to deviate from the center since it is almost impossible to hold it completely still for the duration of the technique. Since it does not take much movement before the cursor leaves the small target's shape area, users must constantly realign the cursor with the center of the shape before performing the technique. This is not the case with the larger targets, where the cursor is allowed to deviate more from the center before it actually leaves the shape, and as such not prompting the need to realign the cursor towards the center, since it is still inside the shape. This suggests that users would initially aim towards the center of the cross and be as precise as possible. As they were to perform the gesture, as long as the cursor was still inside the shape, they would go ahead and perform the technique.

6.4 Summary

Initially, we had divided the techniques by assigning them different attributes. This is because we believed these attributes would heavily affect the results of these techniques compared to each other. We gave each technique two attributes, the amount of hands needed to perform the technique, and whether or not the phone was in movement during the activation of the technique. Table 6 shows the division that we made.

	Touch phone	Move phone
One handed	**Swipe**	Tilt
Two handed	Grab	**Throw**

Table 6. Attribute division for techniques, with the most efficient and accurate techniques highlighted

The data shows that *Swipe* is the most efficient and accurate technique out of all the four techniques. We then see that *Throw* follows closely behind, in both efficiency and accuracy. If we look at efficiency, we see that *Tilt* takes considerably less time to perform than *Grab*. In regards to accuracy, *Grab* is the third most accurate technique, with *Tilt* far behind as the least accurate technique.

While *Swipe* and *Throw* do not share any of the attributes we assigned them, they do have one thing in common: the cursor pointing hand is held still throughout the activation process. The *Tilt* technique requires the phone hand, which is also the cursor pointing hand, to be moved in order to activate, and *Grab* requires the user to open or close his cursor pointing hand, which causes a slight movement of the cursor when doing so. This seams to be the single largest contributing factor in whether or not the technique was efficient and accurate. Our findings suggest that the most accurate and efficient techniques are the ones were it is possible to hold the cursor still while performing the technique.

We then see that *Grab*, a two handed technique, is significantly more accurate than *Tilt*. This suggests that being capable of operating the mobile device in one hand and aiming with the other contributes to the stability of the cursor, making two hand gestures more accurate than one handed. This is also suggested by the findings in [21]. We also see that *Tilt* is significantly faster than *Grab*. This suggests that if cursor stability was not a problem, then one handed techniques might be faster to perform than two handed. In order to properly determine whether or not the amount of hands used to perform a technique has any effect on efficiency and accuracy, further research needs to be conducted.

7. CONCLUSION

In this paper, we present our results from a comparative empirical study between eight cross-device interaction techniques between smartphones and large displays. We performed two studies, comparing eight significantly different interaction techniques, in order to better understand what makes an interaction technique accurate and efficient.

Our results show that *Swipe* and *Throw* were the most successful techniques out of all eight techniques. The one common factor between the two techniques was the fact that the gestures needed in order to perform the technique did not include movement in the cursor pointing hand. This indicates that a large contributor to a technique's accuracy and efficiency is the ability to hold the cursor still while performing an interaction technique.

This comparative study provides a basic understanding of the different attributes that make a cross-device interaction technique between mobile devices and large displays successful in terms of accuracy and efficiency.

8. FUTURE RESEARCH

In the future, we would plan on extending this research by trying to better understand the effects the number of hands has on efficiency and accuracy of each technique. Another plan would be to create a locking mechanism and purely explore the efficiency of each technique. Once people no longer are afraid of missing, does that change how fast they perform each technique? We could also add try to measure how fast people learn and master each technique, and try to see how that is affected by the number of steps a technique has, or how complex it is. Is there any connection to the amount of hands needed to perform a technique?

These are all experiments we could conduct with some simple extensions and modifications of our current system and could provide some very interesting answers.

9. ACKNOWLEDGMENTS

We would like to thank Jeni Paay for her amazing guidance throughout both our final master's semesters. We would also like to thank Dimitrios Raptis for helping us analyze and understand all the data we gathered from our experiments. Jesper Kjeldskov also deserves a huge thanks, for making sure we had all the equipment we needed to conduct our experiments as well as giving us feedback on the experiments themselves. Finally, we would like to thank all the people who voluntarily took part in our experiments.

10. REFERENCES

1. Hrvoje Benko and Andrew D. Wilson. 2010. Multi-point Interactions with Immersive Omnidirectional Visualizations in a Dome. In *ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. ACM, New York, NY, USA, 19–28. DOI: http://dx.doi.org/10.1145/1936652.1936657

2. Richard A. Bolt. 1980. &Ldquo;Put-that-there&Rdquo;: Voice and Gesture at the Graphics Interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '80)*. ACM, New York, NY, USA, 262–270. DOI: http://dx.doi.org/10.1145/800250.807503

3. Sebastian Boring, Marko Jurmu, and Andreas Butz. 2009. Scroll, Tilt or Move It: Using Mobile Phones to Continuously Control Pointers on Large Public Displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7 (OZCHI '09)*. ACM, New York, NY, USA, 161–168. DOI: http://dx.doi.org/10.1145/1738826.1738853

4. Andrew Bragdon, Rob DeLine, Ken Hinckley, and Meredith Ringel Morris. 2011. Code Space: Touch + Air Gesture Hybrid Interactions for Supporting Developer Meetings. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. ACM, New York, NY, USA, 212–221. DOI: http://dx.doi.org/10.1145/2076354.2076393

5. Raimund Dachselt and Robert Buchholz. Throw and Tilt-Seamless Interaction across Devices Using Mobile Phone Gestures. Citeseer.

6. L. Gallo and A. Minutolo. 2012. Design and Comparative Evaluation of Smoothed Pointing: A Velocity-oriented Remote Pointing Enhancement Technique. *Int. J. Hum.-Comput. Stud.* 70, 4 (April 2012), 287–300. DOI: http://dx.doi.org/10.1016/j.ijhcs.2011.12.001

7. Luke Hespanhol, Martin Tomitsch, Kazjon Grace, Anthony Collins, and Judy Kay. 2012. Investigating Intuitiveness and Effectiveness of Gestures for Free

Spatial Interaction with Large Displays. In *Proceedings of the 2012 International Symposium on Pervasive Displays (PerDis '12)*. ACM, New York, NY, USA, Article 6, 6 pages. DOI:http://dx.doi.org/10.1145/2307798.2307804

8. David C. Hoaglin and Boris Iglewicz. 1987. Fine-Tuning Some Resistant Rules for Outlier Labeling. *J. Amer. Statist. Assoc.* 82, 400 (1987), 1147–1149. http://www.jstor.org/stable/2289392

9. Kaori Ikematsu and Itiro Siio. 2015. Memory Stones: An Intuitive Information Transfer Technique Between Multi-touch Computers. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications (HotMobile '15)*. ACM, New York, NY, USA, 3–8. DOI: http://dx.doi.org/10.1145/2699343.2699352

10. Mikkel R. Jakobsen, Yvonne Jansen, Sebastian Boring, and Kasper Hornbæk. 2015. *Human-Computer Interaction – INTERACT 2015: 15th IFIP TC 13 International Conference, Bamberg, Germany, September 14-18, 2015, Proceedings, Part III*. Springer International Publishing, Cham, Chapter Should I Stay or Should I Go? Selecting Between Touch and Mid-Air Gestures for Large-Display Interaction, 455–473. DOI: http://dx.doi.org/10.1007/978-3-319-22698-9_31

11. Bjarke M. Lauridsen, Ivan S. Penchev, Elias Ringhauge, and Eric V. Ruder. 2016. Comparison of Push Techniques for Cross-Device Interaction Between Mobile Devices and Large Displays. *Det Digitale Projektbibliotek (requires login at projekter.aau.dk)* (Jan. 2016), 69. http://goo.gl/vqP441

12. Andrés Lucero, Jussi Holopainen, and Tero Jokela. 2012. MobiComics: Collaborative Use of Mobile Phones and Large Displays for Public Expression. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 383–392. DOI:http://dx.doi.org/10.1145/2371574.2371634

13. Anders Markussen, Sebastian Boring, Mikkel R. Jakobsen, and Kasper Hornbæk. 2016. Off-Limits: Interacting Beyond the Boundaries of Large Displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5862–5873. DOI: http://dx.doi.org/10.1145/2858036.2858083

14. Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: A Mid-air Word-gesture Keyboard. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1073–1082. DOI: http://dx.doi.org/10.1145/2556288.2556964

15. Sven Mayer, Katrin Wolf, Stefan Schneegass, and Niels Henze. 2015. Modeling Distant Pointing for Compensating Systematic Displacements. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 4165–4168. DOI: http://dx.doi.org/10.1145/2702123.2702332

16. Brad A. Myers, Rishi Bhatnagar, Jeffrey Nichols, Choon Hong Peck, Dave Kong, Robert Miller, and A. Chris Long. 2002. Interacting at a Distance: Measuring the Performance of Laser Pointers and Other Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 33–40. DOI: http://dx.doi.org/10.1145/503376.503383

17. Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Pourang P. Irani, and Michel Beaudouin-Lafon. 2013. High-precision Pointing on Large Wall Displays Using Small Handheld Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 831–840. DOI: http://dx.doi.org/10.1145/2470654.2470773

18. Umar Rashid, Jarmo Kauko, Jonna Häkkilä, and Aaron Quigley. 2011. Proximal and Distal Selection of Widgets: Designing Distributed UI for Mobile Interaction with Large Display. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 495–498. DOI: http://dx.doi.org/10.1145/2037373.2037446

19. Jun Rekimoto. 1997. Pick-and-drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST '97)*. ACM, New York, NY, USA, 31–39. DOI: http://dx.doi.org/10.1145/263407.263505

20. Jürgen Scheible, Timo Ojala, and Paul Coulton. 2008. MobiToss: A Novel Gesture Based Interface for Creating and Sharing Mobile Multimedia Art on Large Public Displays. In *Proceedings of the 16th ACM International Conference on Multimedia (MM '08)*. ACM, New York, NY, USA, 957–960. DOI: http://dx.doi.org/10.1145/1459359.1459532

21. Julian Seifert, Andreas Bayer, and Enrico Rukzio. 2013. *Human-Computer Interaction – INTERACT 2013: 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part III*. Springer Berlin Heidelberg, Berlin, Heidelberg, Chapter PointerPhone: Using Mobile Phones for Direct Pointing Interactions with Remote Displays, 18–35. DOI: http://dx.doi.org/10.1007/978-3-642-40477-1_2

22. Daniel Vogel and Ravin Balakrishnan. 2005. Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, New York, NY, USA, 33–42. DOI:http://dx.doi.org/10.1145/1095034.1095041

23. Ulrich von Zadow. 2015. Using Personal Devices to Facilitate Multi-user Interaction with Large Display Walls. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15 Adjunct)*. ACM, New York, NY, USA, 25–28. DOI:http://dx.doi.org/10.1145/2815585.2815592

24. Robert Walter, Gilles Bailly, Nina Valkanova, and Jörg Müller. 2014. Cuenesics: Using Mid-air Gestures to Select Items on Interactive Public Displays. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 299–308. DOI: http://dx.doi.org/10.1145/2628363.2628368

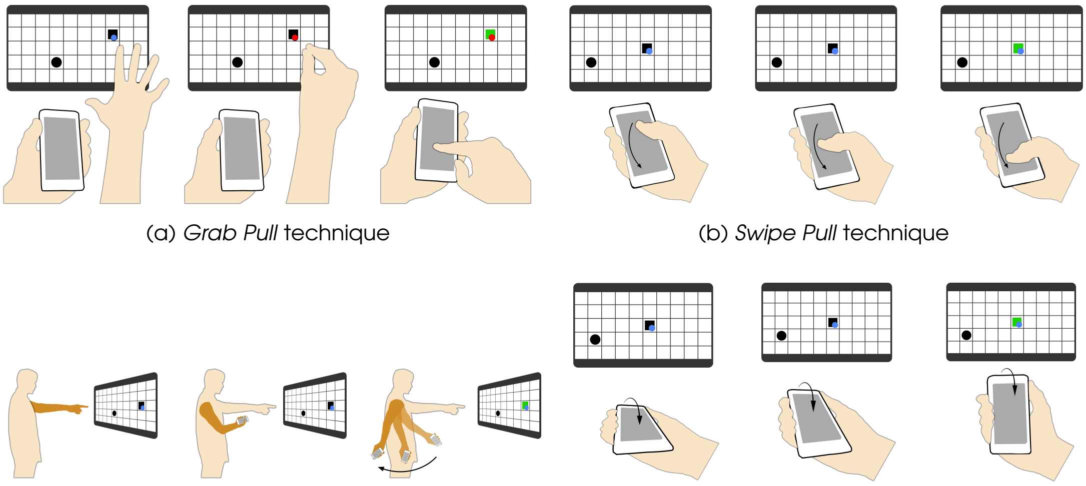
25. Koji Yatani, Koiti Tamura, Keiichi Hiroki, Masanori Sugimoto, and Hiromichi Hashizume. 2005. Toss-it: Intuitive Information Transfer Techniques for Mobile Devices. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1881–1884. DOI: http://dx.doi.org/10.1145/1056808.1057046

Differences Between Semesters

In this section we will discuss some of the differences between the first and second semester of our master thesis. Last semester's paper can be found in Appendix E.

Pull Techniques

The techniques used in the two studies this semester were twice the number of techniques we experimented with last semester which only compared 4 push techniques. The studies this semester also include the pull direction from the large display to the phone. Figure 1 shows the figures for all the pull techniques. We did not add them to the paper since we felt that we described the pull techniques sufficiently along with the push technique figures. If we had added them, they would have taken a lot of extra space, which we wanted to use for other things.



(a) *Grab Pull* technique

(b) *Swipe Pull* technique

(c) *Throw Pull* technique

(d) *Tilt Pull* technique

Figure 1: Illustrations of all the pull techniques that were used in the studies.

Improvements to the system

We made some improvements to the system compared to last semester, and one of the biggest improvements to the system was to replace the first generation Kinect with the newer Kinect for tracking bodies and detecting hands. This also meant that we were able to use the Kinect for Windows SDK 2.0 which gives the developer more features for example *"Thumb tracking, end of hand tracking, open and closed hand gestures"*[2].

For this semester's studies the system will detect the height from the floor to the head joint and use this in the mapping between a participant's pointing hand and the cursor on the screen. By doing this we avoid situations from last semester where a participant had to stand on a chair in order to interact with the system.

The implementation and robustness of the techniques have been improved and especially for the *Grab*, *Throw*, and *Tilt* techniques. The pointing, for all techniques, is much more responsive and without as much jitter as last semester.

The registration of when a hand is opened or closed for *Grab Push* and *Grab Pull* is much more robust and does not produce nearly the same amount of false positives as before. This is both a result of the new Kinect, as well as a much better angle to capture the state of the participants' hands. By lowering the position of the Kinect, instead of having it as close as possible to the large display, the Kinect had a much better overview of the hand. This meant that it was much more capable of determining the correct state of the hand from what it could see, as the profile of the hand was that much more pronounced.

The *Throw* technique can be performed in an overhand, underhand, or sideways style for both the *Pull* and *Push* direction. This was because we changed the definition of a *Throw*. In our previous semester, *Throw* was defined as a sequence of positions of the hands that needed to be fulfilled. This semester, these sequences were replaced and instead we defined *Throw* as a movement from one direction to another (backwards or forwards) of the hands, which had to exceed a certain speed. This lead to a much more open, as well as more precise, version of the *Throw* technique.

We also improved the *Tilt* technique by improving its interpretation of what a tilting movement was from the readings on the accelerometer. While this did lead to considerably less false positives, it did mean that the tilting movement needed to be much more pronounced,. This was hard for our participants, as from our observations, people felt very uncomfortable with the *Tilt* technique and performed it very slowly, so that

[2] *Kinect for Windows SDK 2.0 Features* https://msdn.microsoft.com/library/dn782025.aspx

they could be as precise as possible with it. This would unfortunately not trigger the technique, so participants often had to perform it multiple times, each one with slightly more power, until it finally triggered.

Improvements of the demo videos

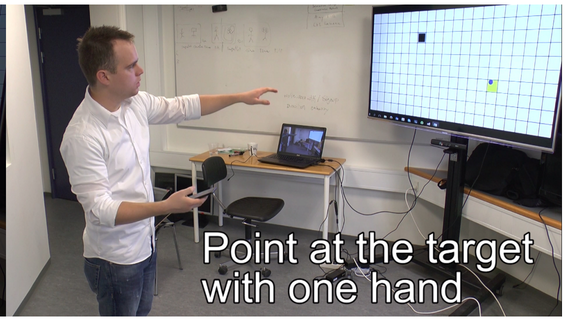
When we did the experiment last semester we also had demonstration videos for the participants to look at if they were in doubt of how to perform a certain technique. The way we did it then was to have the videos run in a loop on the small screen while the test started on the big screen. This meant that participants were able to start the test even after just seeing the title of the technique they were suppose to do and last semester we saw people jumping straight into using the technique without having looked at the video of the technique. When this would happen they would simply spend more time in the beginning trying to figure the technique out for themselves. If they were not successful performing the technique they would finally turn their attention to the demonstration of the technique. During the experiment some participants noted that they did not notice the video of the technique running in a loop on the small screen.

The quality of the videos we used last semester were not good and people often complained that they were not clear and that they could not clearly see how the technique should be performed. The videos were shot from two different perspectives and the phone appeared very small on the screen and was not easy to see. The tempo of the videos and the cuts to white text on black background made the video seem unconnected and possibly a little hard to follow.

For this semester we changed the procedure for how we introduced the participants to the techniques and we also changed all the demonstration videos. The procedure was changed such that participants had to watch the technique being performed two times before the test would start and they could start using the technique themselves. Another change we made to the procedure were to have the video play on the large screen initially and after two iterations on the large screen the video played in a loop on the small screen. The videos were changed so they are easier to understand and visibly more clear than were the case previously, for example, with the interaction on the phone being hard to see. In the new videos a technique is explained in a slower pace and without cuts to a black screen with white text and we zoomed in on the phone as can be seen in Figure 2c. Four screen shots of the Push Throw technique demo video is shown in Figure 2.

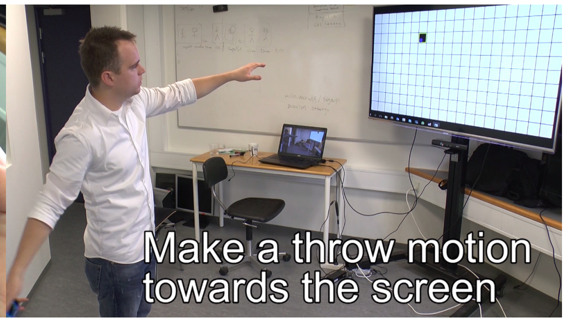
This lead to a much smoother experience, both for us and the participants. Last semester, we had to intervene during almost every experiment in order to further explain the techniques to the participants. This time, we very rarely had to intervene and further explain to the participants what they had to do or what they were doing wrong.

(a)



(b)

(c)



(d)

Figure 2: The images here shows the screen at different times during the demonstration video. The video was shown to the participant before each technique test starts. In (a) the technique is being presented with the direction and the name of the technique. In (b), (c), and (d) the video pauses and the participant will be able to read the instructions on the screen.

Improvement to the analysis

In last semester's paper, the results were not cleaned for erroneous attempts which meant that we counted all attempts. Some attempts would be clear errors such as (1) a participant unintentionally activating a technique with the cursor in a corner of the screen opposite the target, or (2) a participant would spend too much time on an attempt due to the system not registering the technique. These two kind of errors would affect the logged data of distance and time and the statistical method used last semester was also not correct for a within subjects design. Last semester we also did not have a proper study were we looked at the accuracy of each technique, so our results regarding accuracy are quite misleading.

Another thing of great importance in regards to the results was that we did not actually perform a correct analysis on the data we generated in our last semester. As such, our numbers are not correct and we cannot say with confidence if the effects we measured were significant or not. We went to much further lengths this semester in order to make sure we performed the correct analysis. We consulted regularly with people who had a much larger understanding of statistics, which greatly helped us get the correct model and a much more precise and clear understanding of what the numbers meant. This also means that the analysis themselves are correct this time around, and we can say with much certainty that the results we show are, in fact, significant and correct.

Concluding Remarks

This thesis deals with the theme of cross-device interaction between mobile devices and large displays. We approached this theme by finding and implementing 8 different interaction techniques and then performing a comparative study between them.

One of the ideas behind this thesis was to see if we could find any attributes to these techniques that made them successful in regards to accuracy and efficiency. We initially believed that the some of the important attributes to the success of a technique would be whether or not the phone was in motion during the technique and the amount of hands used to perform the given technique.

Our results did show that there might be some association between the amount of hands and the accuracy and efficiency of each technique. There seems to be some indication that one handed techniques are faster to perform that two handed, but that two handed techniques are more accurate. These are only indications though and in order to more conclusively say that this is indeed the case, more research must be conducted with these specific attributes in mind.

A much more important attribute though came up and that was the ability to hold the cursor still while performing the technique. The four most successful techniques, the *Push* and *Pull* versions of *Swipe* and *Throw*, all had that in common. Users where capable of keeping the cursor still while activating and performing the technique. The other four techniques, the *Push* and *Pull* versions of *Tilt* and *Grab*, had movements on the cursor pointing hand, causing the cursor to move during activation instead of keeping it stable.

In the future, we would like to extend our research by examining more closely the relationship between amount of hands and the efficiency and accuracy of each technique. This could be done by implementing different techniques were the focus is much more on the amount of hands and the role of each hand while performing the given technique. Having techniques were the sole role of one of the hands is for aiming, like *Throw*, and others were each hand has some gesture it has to perform in order to activate the technique, such as *Grab*. A research like this could lead to a much more clear understanding of how the amount of hands affects the performance of a interaction technique.

Summary

In this master thesis, two experimental studies were created with the purpose of comparing eight different cross-device interaction techniques allowing users to push and pull data to and from a large display using a smartphone. The studies complements existing research in the area of cross-device data transfer between handheld devices and large displays by providing a quantitative comparison of different techniques which combines finger pointing, pointing with phones, and one- and two-handed interactions.

To perform the studies and collect data, a system was developed using a Microsoft Kinect, a large display, and a smartphone. The studies were conducted in the usability lab and ran for four weeks with a total of 84 people participating.

This thesis builds on knowledge gained in the authors' 9th semester pre-thesis, and further development of the system in the 10th semester thesis has resulted in a scientific paper presenting the studies and the results. The paper concludes that four techniques (*Pull* and *Push* versions of the *Swipe* and *Throw* techniques) were more successful than the other four techniques (*Pull* and *Push* versions of the *Grab* and *Tilt* techniques). Because the best techniques, *Swipe* and *Throw*, have different attributes it might be useful for interaction designers to know which techniques to consider when creating systems that use this kind of cross-device interaction. Since the techniques have different defining attributes they can be used in different situations and contexts.

2016

AALBORG UNIVERSITY

APPENDICES

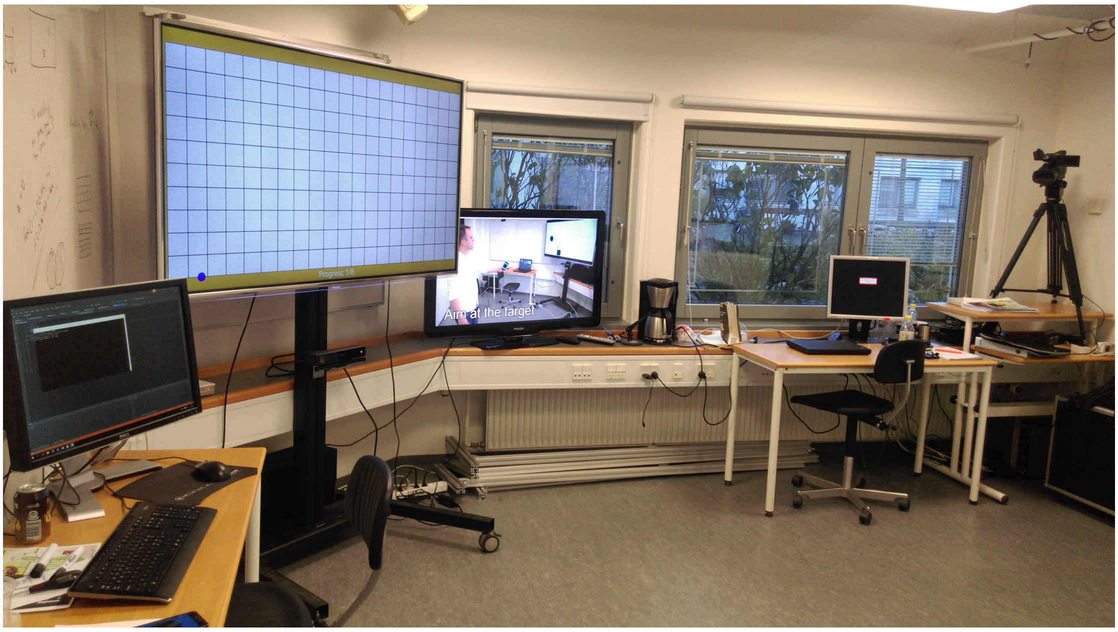
Contents

2016

AALBORG UNIVERSITY

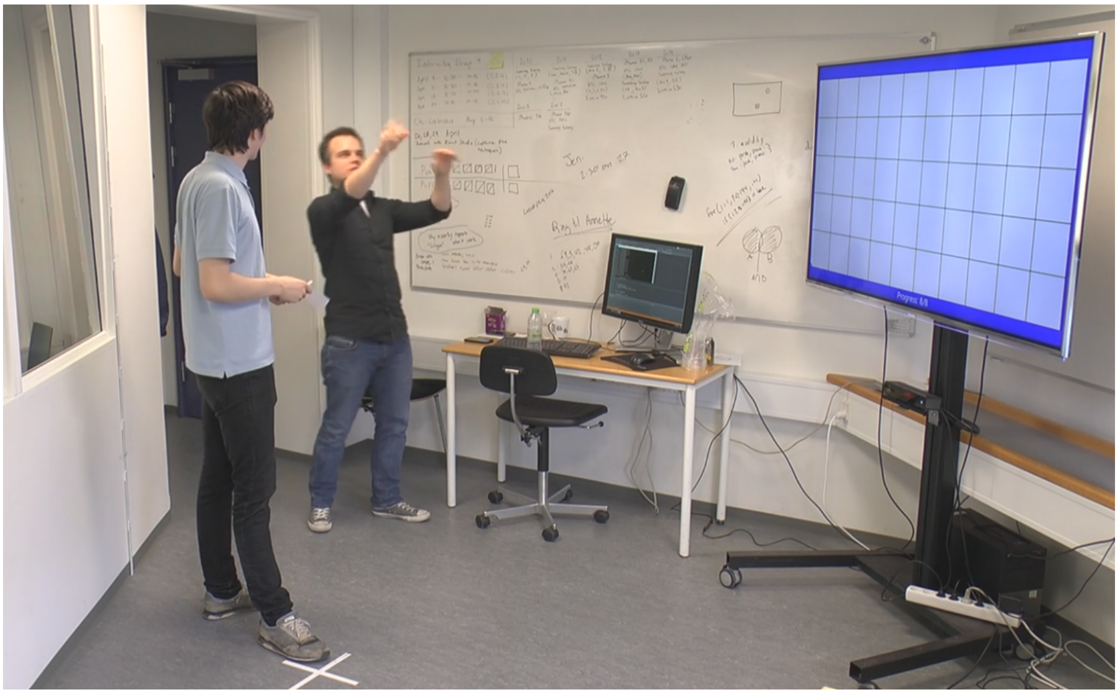
Appendix A: Images

Setup

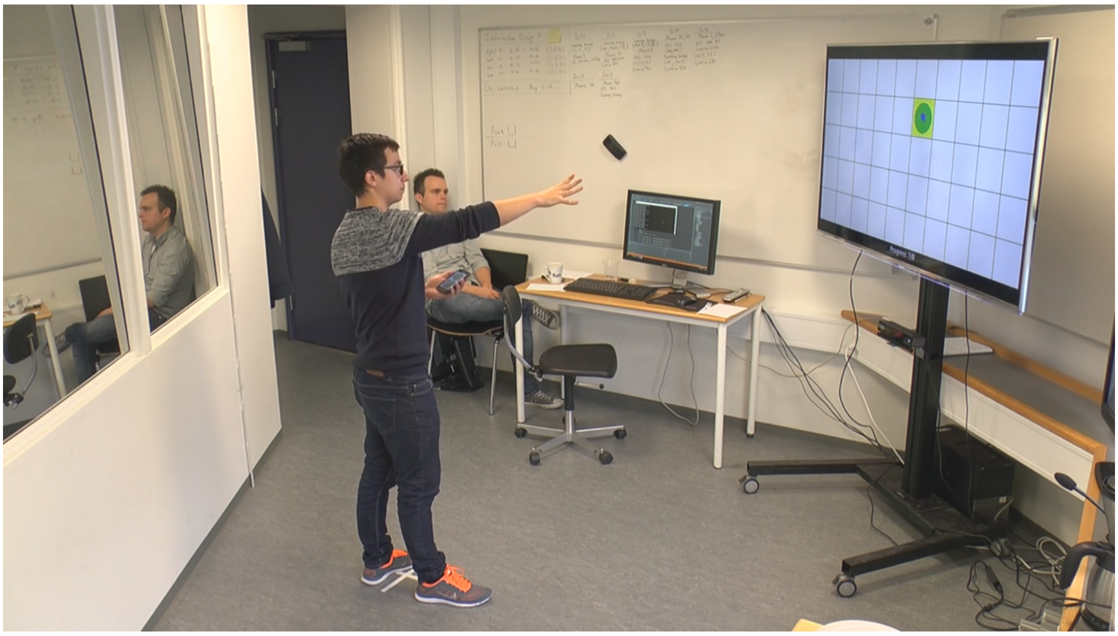


Presenting the experiment

Discussing the experiment

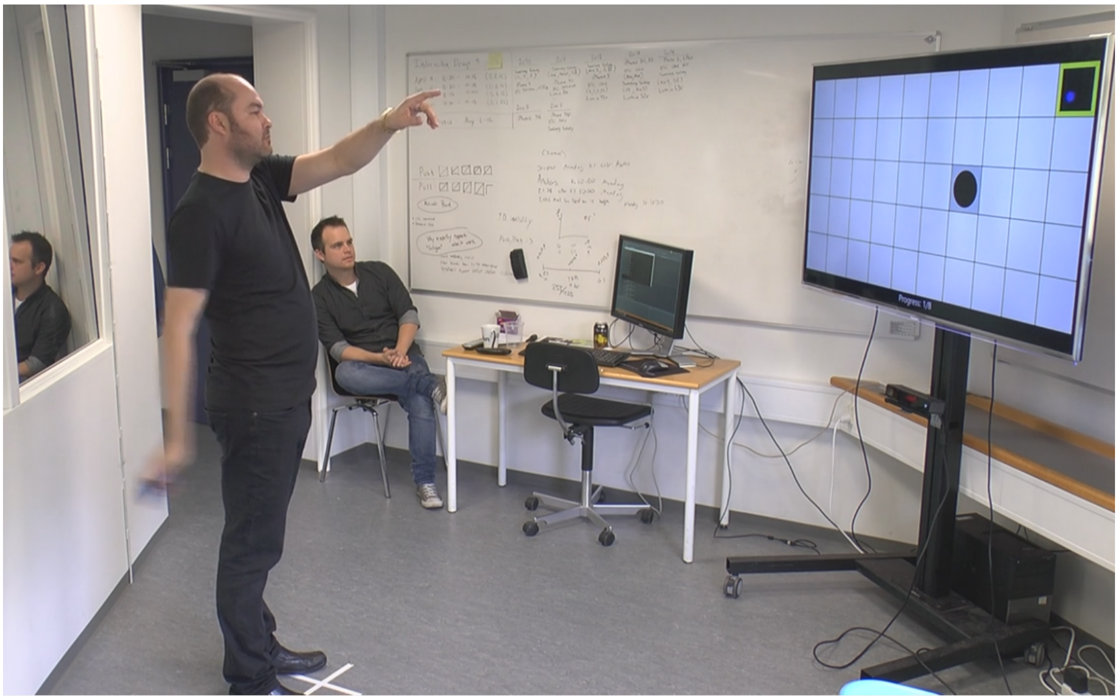


Grab technique

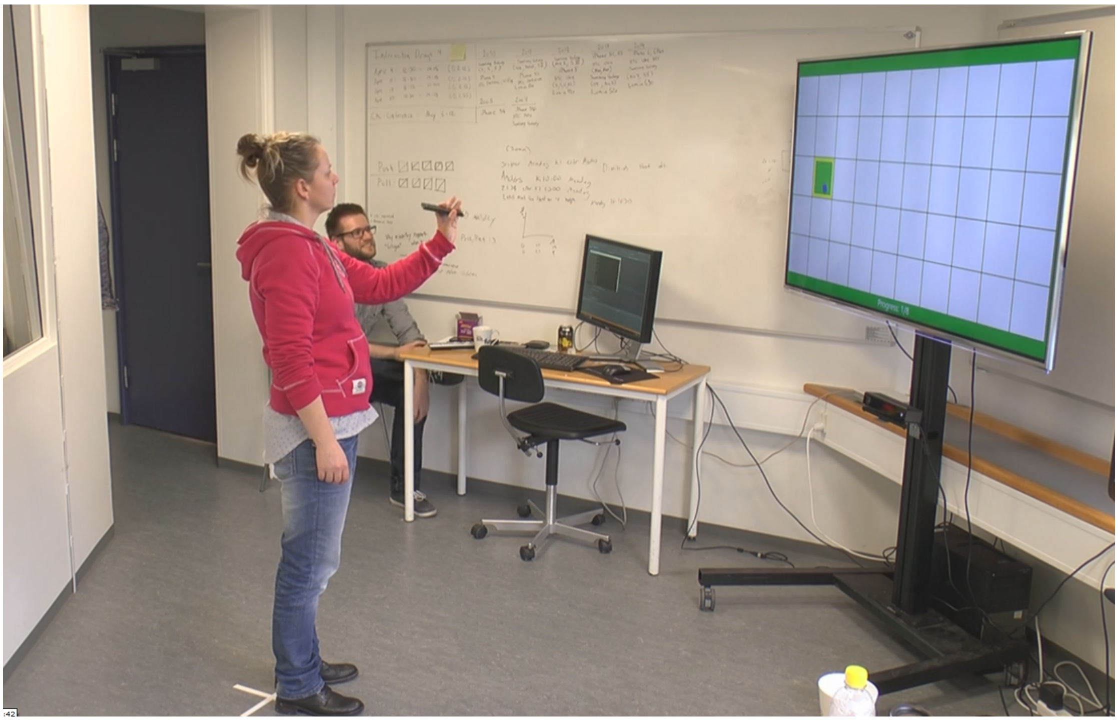


Swipe technique

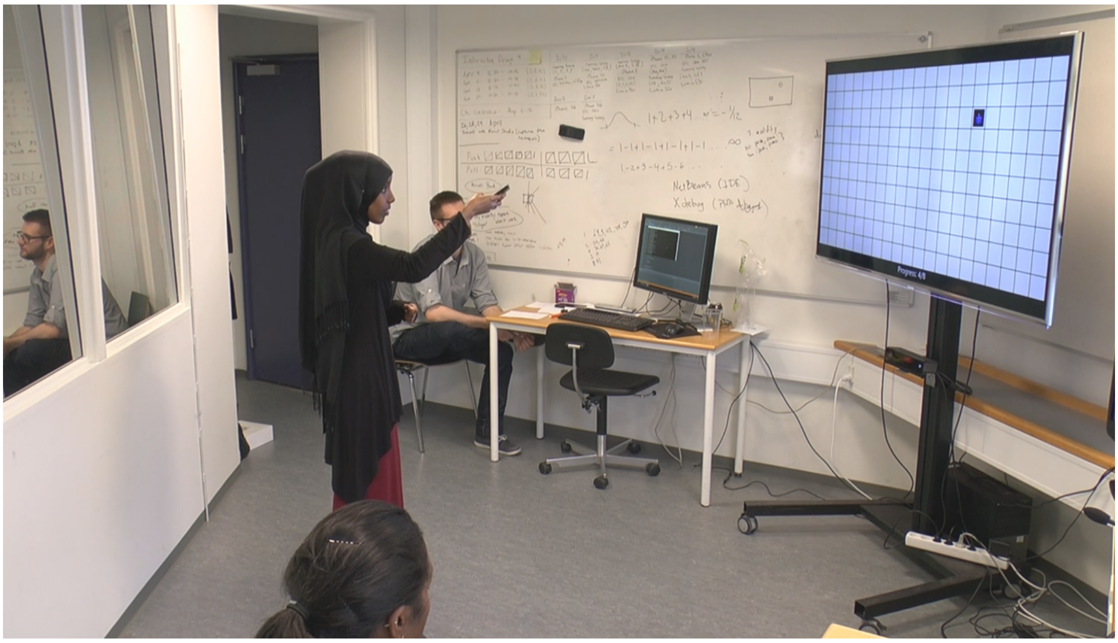
Throw technique



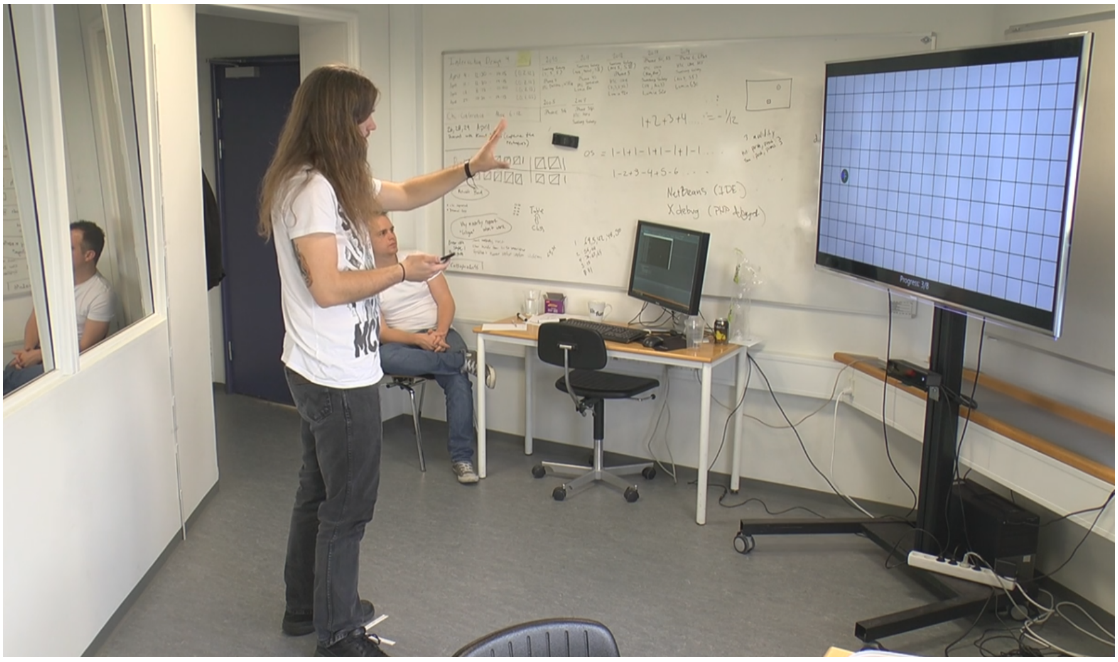
Tilt technique



Class observer



Accuracy study



Appendix B: Demographic Questionnaire

Gesture Test

Age: ____

Height: ____

Gender: ☐ Male ☐ Female

Left or right handed: ☐ Left ☐ Right

Smartphone user: ☐ Yes ☐ No

How long have you had one? ___________________________

What device do you currently own? _____________________

Have you ever played any motion sensor based games such Wii games, Kinect games, or PlayStation Move games?

☐ Yes ☐ No

How often? ☐ Daily ☐ Once a week

☐ Once a month ☐ Other: ____________

Appendix C: Example of raw test log

```
[18:20:29]: Started new gesture practice. Type: Tilt Direction: Pull
[18:20:29]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:21:37]: Started new gesture test. Type: Tilt Direction: Pull
[18:21:38]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:21:43]: Target: Hit  Shape: Correct TC: (17,04) CC: (17, 04) JL: Short Pointer position: (1067.7,270.2).
[18:21:49]: Target: Miss Shape: Correct TC: (08,08) CC: (09, 07) JL: Medium Pointer position: (575.5,441.8).
[18:21:50]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:21:56]: Target: Hit  Shape: Correct TC: (09,00) CC: (09, 00) JL: Long Pointer position: (1156.8,62.3).
[18:21:57]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:22:03]: Target: Miss Shape: Correct TC: (18,06) CC: (18, 05) JL: Short Pointer position: (1118.2,325.2).
[18:22:04]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:22:12]: Target: Miss Shape: Correct TC: (02,04) CC: (02, 03) JL: Long Pointer position: (313.8,396.9).
[18:22:13]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:22:22]: Target: Miss Shape: Correct TC: (09,00) CC: (10, 00) JL: Medium Pointer position: (608.1,26.3).
[18:22:29]: Target: Miss Shape: Correct TC: (04,03) CC: (05, 03) JL: Short Pointer position: (338.1,186.9).
[18:22:30]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:22:35]: Target: Hit  Shape: Correct TC: (06,00) CC: (06, 00) JL: Medium Pointer position: (805.0,30.1).
[18:22:40]: Target: Miss Shape: Correct TC: (00,02) CC: (01, 02) JL: Long Pointer position: (137.1,266.0).
[18:22:41]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:22:50]: Target: Miss Shape: Correct TC: (10,05) CC: (10, 04) JL: Medium Pointer position: (664.1,304.5).
[18:22:51]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:22:56]: Target: Miss Shape: Correct TC: (09,04) CC: (09, 03) JL: Medium Pointer position: (1130.2,462.8).
[18:22:57]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:22:58]: Target: Miss Shape: Correct TC: (19,03) CC: (11, 00) JL: Short Pointer position: (690.5,0.2).
[18:22:58]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:22:59]: Target: Miss Shape: Correct TC: (02,00) CC: (05, 00) JL: Long Pointer position: (723.6,54.8).
[18:23:10]: Target: Hit  Shape: Correct TC: (07,04) CC: (07, 04) JL: Long Pointer position: (903.1,574.6).
[18:23:11]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:23:19]: Target: Hit  Shape: Correct TC: (06,05) CC: (06, 05) JL: Medium Pointer position: (393.6,342.5).
[18:23:20]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:23:25]: Target: Hit  Shape: Correct TC: (06,03) CC: (06, 03) JL: Short Pointer position: (802.5,433.0).
[18:23:26]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:23:30]: Target: Hit  Shape: Correct TC: (00,00) CC: (00, 00) JL: Long Pointer position: (33.0,29.1).
[18:23:31]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:23:37]: Target: Hit  Shape: Correct TC: (01,02) CC: (01, 02) JL: Short Pointer position: (172.4,298.7).
[18:23:39]: Started new gesture practice. Type: Throw Direction: Pull
```

```
[18:24:32]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:24:41]: Started new gesture test. Type: Throw Direction: Pull
[18:24:54]: Target: Hit  Shape: Correct TC: (06,07) CC: (06, 07) JL: Medium Pointer position: (391.5,455.6).
[18:25:00]: Target: Hit  Shape: Correct TC: (03,02) CC: (03, 02) JL: Short Pointer position: (221.8,162.1).
[18:25:01]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:25:08]: Target: Hit  Shape: Correct TC: (08,03) CC: (08, 03) JL: Long Pointer position: (1009.8,445.6).
[18:25:31]: Target: Miss Shape: Correct TC: (06,00) CC: (04, 00) JL: Short Pointer position: (561.6,120.4).
[18:25:32]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:25:40]: Target: Hit  Shape: Correct TC: (06,05) CC: (06, 05) JL: Medium Pointer position: (404.0,327.4).
[18:25:41]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:25:46]: Target: Hit  Shape: Correct TC: (08,03) CC: (08, 03) JL: Medium Pointer position: (1073.4,395.1).
[18:25:47]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:26:04]: Target: Hit  Shape: Correct TC: (03,03) CC: (03, 03) JL: Long Pointer position: (206.6,197.4).
[18:26:10]: Target: Hit  Shape: Correct TC: (06,08) CC: (06, 08) JL: Short Pointer position: (378.0,512.7).
[18:26:11]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:26:14]: Target: Hit  Shape: Correct TC: (00,01) CC: (00, 01) JL: Long Pointer position: (49.7,180.9).
[18:26:15]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:26:18]: Target: Hit  Shape: Correct TC: (05,00) CC: (05, 00) JL: Long Pointer position: (350.1,19.3).
[18:26:22]: Target: Hit  Shape: Correct TC: (14,04) CC: (14, 04) JL: Medium Pointer position: (881.5,273.9).
[18:26:23]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:26:26]: Target: Hit  Shape: Correct TC: (00,04) CC: (00, 04) JL: Long Pointer position: (47.2,548.0).
[18:26:27]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:26:32]: Target: Hit  Shape: Correct TC: (14,06) CC: (14, 06) JL: Long Pointer position: (871.4,387.6).
[18:26:33]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:26:35]: Target: Hit  Shape: Correct TC: (02,01) CC: (02, 01) JL: Medium Pointer position: (323.7,192.6).
[18:26:38]: Target: Hit  Shape: Correct TC: (04,03) CC: (04, 03) JL: Short Pointer position: (576.9,425.9).
[18:26:43]: Target: Hit  Shape: Correct TC: (07,03) CC: (07, 03) JL: Short Pointer position: (898.0,402.1).
[18:26:44]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:26:47]: Target: Hit  Shape: Correct TC: (13,01) CC: (13, 01) JL: Short Pointer position: (827.7,79.1).
[18:26:48]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:26:52]: Target: Hit  Shape: Correct TC: (02,03) CC: (02, 03) JL: Medium Pointer position: (296.2,427.7).
[18:26:54]: Started new gesture practice. Type: Swipe Direction: Pull
[18:27:50]: Started new gesture test. Type: Swipe Direction: Pull
[18:27:51]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:27:56]: Target: Hit  Shape: Correct TC: (02,04) CC: (02, 04) JL: Short Pointer position: (157.7,273.1).
[18:28:00]: Target: Hit  Shape: Correct TC: (11,08) CC: (11, 08) JL: Medium Pointer position: (698.4,520.1).
```

```
[18:28:01]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:28:04]: Target: Hit  Shape: Correct TC: (00,00) CC: (00, 00) JL: Long Pointer position: (67.8,68.0).
[18:28:05]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:28:10]: Target: Hit  Shape: Correct TC: (01,06) CC: (01, 06) JL: Short Pointer position: (99.3,400.2).
[18:28:11]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:28:14]: Target: Hit  Shape: Correct TC: (07,04) CC: (07, 04) JL: Long Pointer position: (920.2,558.8).
[18:28:15]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:28:19]: Target: Hit  Shape: Correct TC: (10,00) CC: (10, 00) JL: Medium Pointer position: (654.7,34.8).
[18:28:25]: Target: Hit  Shape: Correct TC: (15,03) CC: (15, 03) JL: Short Pointer position: (953.1,214.5).
[18:28:26]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:28:29]: Target: Hit  Shape: Correct TC: (03,00) CC: (03, 00) JL: Medium Pointer position: (406.4,68.3).
[18:28:33]: Target: Hit  Shape: Correct TC: (09,02) CC: (09, 02) JL: Long Pointer position: (1158.0,312.5).
[18:28:34]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:28:38]: Target: Hit  Shape: Correct TC: (09,05) CC: (09, 05) JL: Medium Pointer position: (584.0,333.6).
[18:28:39]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:28:42]: Target: Hit  Shape: Correct TC: (00,04) CC: (00, 04) JL: Medium Pointer position: (49.0,554.4).
[18:28:43]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:28:48]: Target: Hit  Shape: Correct TC: (00,03) CC: (00, 03) JL: Short Pointer position: (43.1,219.7).
[18:28:49]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:28:52]: Target: Hit  Shape: Correct TC: (07,00) CC: (07, 00) JL: Long Pointer position: (907.7,63.9).
[18:28:56]: Target: Hit  Shape: Correct TC: (02,04) CC: (02, 04) JL: Long Pointer position: (310.3,531.9).
[18:28:57]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:29:02]: Target: Hit  Shape: Correct TC: (13,05) CC: (13, 05) JL: Medium Pointer position: (810.8,341.9).
[18:29:03]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:29:06]: Target: Hit  Shape: Correct TC: (03,03) CC: (03, 03) JL: Short Pointer position: (436.4,431.7).
[18:29:07]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:29:13]: Target: Hit  Shape: Correct TC: (19,00) CC: (19, 00) JL: Long Pointer position: (1181.6,25.5).
[18:29:13]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:29:16]: Target: Hit  Shape: Correct TC: (08,02) CC: (08, 02) JL: Short Pointer position: (1036.8,317.8).
[18:29:18]: Started new gesture practice. Type: Pinch Direction: Pull
[18:30:44]: Started new gesture test. Type: Pinch Direction: Pull
[18:30:45]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:30:54]: Target: Hit  Shape: Correct TC: (02,05) CC: (02, 05) JL: Short Pointer position: (142.4,341.7).
[18:31:04]: Target: Hit  Shape: Correct TC: (11,01) CC: (11, 01) JL: Medium Pointer position: (691.6,86.3).
[18:31:05]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:31:11]: Target: Hit  Shape: Correct TC: (00,04) CC: (00, 04) JL: Long Pointer position: (40.3,566.9).
```

```
[18:31:12]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:31:22]: Target: Hit  Shape: Correct TC: (01,03) CC: (01, 03) JL: Short Pointer position: (105.4,225.3).
[18:31:23]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:31:31]: Target: Hit  Shape: Correct TC: (07,00) CC: (07, 00) JL: Long Pointer position: (890.3,58.7).
[18:31:32]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:31:42]: Target: Hit  Shape: Correct TC: (10,09) CC: (10, 09) JL: Medium Pointer position: (642.6,570.4).
[18:31:48]: Target: Miss Shape: Correct TC: (15,06) CC: (08, 00) JL: Short Pointer position: (494.2,0.0).
[18:31:49]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:31:55]: Target: Miss Shape: Correct TC: (03,04) CC: (04, 00) JL: Medium Pointer position: (488.9,0.0).
[18:32:05]: Target: Hit  Shape: Correct TC: (09,02) CC: (09, 02) JL: Long Pointer position: (1153.0,305.1).
[18:32:06]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:32:13]: Target: Hit  Shape: Correct TC: (09,04) CC: (09, 04) JL: Medium Pointer position: (577.6,278.5).
[18:32:14]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:32:21]: Target: Hit  Shape: Correct TC: (00,00) CC: (00, 00) JL: Medium Pointer position: (52.6,80.3).
[18:32:22]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:32:32]: Target: Hit  Shape: Correct TC: (00,06) CC: (00, 06) JL: Short Pointer position: (17.1,402.7).
[18:32:33]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:32:39]: Target: Hit  Shape: Correct TC: (07,04) CC: (07, 04) JL: Long Pointer position: (920.9,571.3).
[18:32:46]: Target: Miss Shape: Correct TC: (02,00) CC: (03, 00) JL: Long Pointer position: (397.9,0.0).
[18:32:47]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:32:55]: Target: Hit  Shape: Correct TC: (13,04) CC: (13, 04) JL: Medium Pointer position: (813.8,275.1).
[18:32:56]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:33:03]: Target: Hit  Shape: Correct TC: (03,01) CC: (03, 01) JL: Short Pointer position: (415.5,210.2).
[18:33:04]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:33:14]: Target: Hit  Shape: Correct TC: (19,09) CC: (19, 09) JL: Long Pointer position: (1201.8,582.5).
[18:33:15]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:33:22]: Target: Hit  Shape: Correct TC: (08,02) CC: (08, 02) JL: Short Pointer position: (1034.6,339.9).
[18:33:24]: Started new gesture practice. Type: Tilt Direction: Push
[18:34:32]: Started new gesture test. Type: Tilt Direction: Push
[18:34:37]: Target: Hit  Shape: Correct TC: (03,00) CC: (03, 00) JL: Medium Pointer position: (442.3,79.3).
[18:34:38]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:34:50]: Target: Hit  Shape: Correct TC: (18,08) CC: (18, 08) JL: Long Pointer position: (1144.4,528.5).
[18:34:59]: Target: Hit  Shape: Correct TC: (04,06) CC: (04, 06) JL: Long Pointer position: (277.9,428.9).
[18:35:00]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:35:03]: Target: Hit  Shape: Correct TC: (03,00) CC: (03, 00) JL: Short Pointer position: (429.1,73.2).
[18:35:04]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
```

```
[18:35:11]: Target: Hit  Shape: Correct TC: (14,07) CC: (14, 07) JL: Medium Pointer position: (873.5,455.6).
[18:35:12]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:35:19]: Target: Hit  Shape: Correct TC: (00,01) CC: (00, 01) JL: Long Pointer position: (18.2,188.5).
[18:35:25]: Target: Hit  Shape: Correct TC: (07,01) CC: (07, 01) JL: Long Pointer position: (896.3,211.9).
[18:35:26]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:35:35]: Target: Miss Shape: Correct TC: (10,06) CC: (11, 06) JL: Short Pointer position: (685.2,380.4).
[18:35:44]: Target: Hit  Shape: Correct TC: (18,00) CC: (18, 00) JL: Medium Pointer position: (1127.7,46.9).
[18:35:45]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:35:48]: Target: Hit  Shape: Correct TC: (02,02) CC: (02, 02) JL: Long Pointer position: (347.9,312.2).
[18:35:49]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:35:57]: Target: Hit  Shape: Correct TC: (08,00) CC: (08, 00) JL: Short Pointer position: (538.0,31.5).
[18:36:05]: Target: Hit  Shape: Correct TC: (15,07) CC: (15, 07) JL: Medium Pointer position: (940.4,461.5).
[18:36:12]: Target: Miss Shape: Correct TC: (02,02) CC: (02, 01) JL: Long Pointer position: (126.2,114.3).
[18:36:13]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:36:16]: Target: Hit  Shape: Correct TC: (05,03) CC: (05, 03) JL: Medium Pointer position: (689.6,431.1).
[18:36:17]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:36:23]: Target: Hit  Shape: Correct TC: (15,03) CC: (15, 03) JL: Short Pointer position: (925.3,232.5).
[18:36:24]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:36:27]: Target: Hit  Shape: Correct TC: (05,03) CC: (05, 03) JL: Short Pointer position: (683.6,435.3).
[18:36:32]: Target: Hit  Shape: Correct TC: (01,00) CC: (01, 00) JL: Medium Pointer position: (197.9,58.3).
[18:36:36]: Target: Hit  Shape: Correct TC: (04,01) CC: (04, 01) JL: Short Pointer position: (535.8,193.3).
[18:36:39]: Started new gesture practice. Type: Swipe Direction: Push
[18:37:38]: Started new gesture test. Type: Swipe Direction: Push
[18:37:43]: Target: Hit  Shape: Correct TC: (07,01) CC: (07, 01) JL: Medium Pointer position: (885.6,154.0).
[18:37:44]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:37:51]: Target: Hit  Shape: Correct TC: (13,08) CC: (13, 08) JL: Short Pointer position: (833.8,519.2).
[18:37:56]: Target: Hit  Shape: Correct TC: (18,05) CC: (18, 05) JL: Short Pointer position: (1126.6,328.9).
[18:37:57]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:38:02]: Target: Hit  Shape: Correct TC: (02,00) CC: (02, 00) JL: Long Pointer position: (289.7,31.9).
[18:38:03]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:38:08]: Target: Hit  Shape: Correct TC: (12,07) CC: (12, 07) JL: Medium Pointer position: (766.7,455.2).
[18:38:13]: Target: Hit  Shape: Correct TC: (03,03) CC: (03, 03) JL: Medium Pointer position: (205.1,199.8).
[18:38:14]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:38:17]: Target: Hit  Shape: Correct TC: (08,03) CC: (08, 03) JL: Long Pointer position: (1046.0,418.1).
[18:38:22]: Target: Hit  Shape: Correct TC: (05,02) CC: (05, 02) JL: Short Pointer position: (704.3,292.1).
[18:38:25]: Target: Hit  Shape: Correct TC: (02,03) CC: (02, 03) JL: Short Pointer position: (287.8,410.5).
```

```
[18:38:26]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:38:30]: Target: Hit  Shape: Correct TC: (17,01) CC: (17, 01) JL: Long Pointer position: (1068.4,89.2).
[18:38:37]: Target: Hit  Shape: Correct TC: (06,08) CC: (06, 08) JL: Long Pointer position: (405.5,522.0).
[18:38:38]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:38:41]: Target: Hit  Shape: Correct TC: (07,02) CC: (07, 02) JL: Medium Pointer position: (915.3,282.9).
[18:38:42]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:38:46]: Target: Hit  Shape: Correct TC: (01,02) CC: (01, 02) JL: Long Pointer position: (94.9,142.8).
[18:38:47]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:38:51]: Target: Hit  Shape: Correct TC: (05,02) CC: (05, 02) JL: Medium Pointer position: (664.4,293.1).
[18:38:52]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:38:58]: Target: Hit  Shape: Correct TC: (16,07) CC: (16, 07) JL: Short Pointer position: (1004.2,450.0).
[18:38:58]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:39:02]: Target: Hit  Shape: Correct TC: (01,01) CC: (01, 01) JL: Long Pointer position: (161.8,163.3).
[18:39:03]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:39:06]: Target: Hit  Shape: Correct TC: (12,06) CC: (12, 06) JL: Medium Pointer position: (749.0,385.6).
[18:39:07]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:39:10]: Target: Hit  Shape: Correct TC: (03,02) CC: (03, 02) JL: Short Pointer position: (409.2,290.6).
[18:39:12]: Started new gesture practice. Type: Throw Direction: Push
[18:40:21]: Started new gesture test. Type: Throw Direction: Push
[18:40:22]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:40:39]: Target: Hit  Shape: Correct TC: (17,05) CC: (17, 05) JL: Short Pointer position: (1073.9,337.3).
[18:40:48]: Target: Hit  Shape: Correct TC: (08,01) CC: (08, 01) JL: Medium Pointer position: (510.9,87.0).
[18:40:49]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:40:56]: Target: Hit  Shape: Correct TC: (09,04) CC: (09, 04) JL: Long Pointer position: (1154.7,543.2).
[18:40:57]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:41:08]: Target: Hit  Shape: Correct TC: (18,03) CC: (18, 03) JL: Short Pointer position: (1118.6,221.6).
[18:41:08]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:41:16]: Target: Hit  Shape: Correct TC: (02,00) CC: (02, 00) JL: Long Pointer position: (325.5,59.9).
[18:41:17]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:41:26]: Target: Hit  Shape: Correct TC: (09,09) CC: (09, 09) JL: Medium Pointer position: (579.2,580.4).
[18:41:32]: Target: Hit  Shape: Correct TC: (04,06) CC: (04, 06) JL: Short Pointer position: (272.1,396.2).
[18:41:33]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:41:38]: Target: Hit  Shape: Correct TC: (06,04) CC: (06, 04) JL: Medium Pointer position: (775.3,541.9).
[18:41:43]: Target: Hit  Shape: Correct TC: (00,02) CC: (00, 02) JL: Long Pointer position: (65.4,306.5).
[18:41:44]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:41:49]: Target: Hit  Shape: Correct TC: (10,04) CC: (10, 04) JL: Medium Pointer position: (634.1,280.5).
```

```
[18:41:50]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:41:55]: Target: Hit  Shape: Correct TC: (09,00) CC: (09, 00) JL: Medium Pointer position: (1148.0,58.2).
[18:41:56]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:42:08]: Target: Hit  Shape: Correct TC: (19,06) CC: (19, 06) JL: Short Pointer position: (1168.7,413.9).
[18:42:09]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:42:13]: Target: Hit  Shape: Correct TC: (02,04) CC: (02, 04) JL: Long Pointer position: (295.8,553.5).
[18:42:29]: Target: Hit  Shape: Correct TC: (09,04) CC: (09, 04) JL: Long Pointer position: (1148.8,552.9).
[18:42:30]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:42:35]: Target: Hit  Shape: Correct TC: (06,04) CC: (06, 04) JL: Medium Pointer position: (387.1,279.0).
[18:42:36]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:42:40]: Target: Hit  Shape: Correct TC: (06,01) CC: (06, 01) JL: Short Pointer position: (789.1,186.5).
[18:42:41]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:42:48]: Target: Hit  Shape: Correct TC: (00,09) CC: (00, 09) JL: Long Pointer position: (32.3,582.5).
[18:42:49]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:43:00]: Target: Hit  Shape: Correct TC: (01,02) CC: (01, 02) JL: Short Pointer position: (172.1,284.3).
[18:43:05]: Started new gesture practice. Type: Pinch Direction: Push
[18:44:07]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:44:14]: Started new gesture test. Type: Pinch Direction: Push
[18:44:21]: Target: Hit  Shape: Correct TC: (07,02) CC: (07, 02) JL: Short Pointer position: (467.4,156.3).
[18:44:22]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:44:27]: Target: Hit  Shape: Correct TC: (07,04) CC: (07, 04) JL: Medium Pointer position: (906.3,557.9).
[18:44:28]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:44:33]: Target: Hit  Shape: Correct TC: (02,03) CC: (02, 03) JL: Long Pointer position: (139.6,213.0).
[18:44:36]: Target: Miss Shape: Correct TC: (16,03) CC: (18, 03) JL: Long Pointer position: (1116.2,200.0).
[18:44:37]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:44:41]: Target: Miss Shape: Correct TC: (03,03) CC: (05, 00) JL: Medium Pointer position: (694.6,35.5).
[18:44:49]: Target: Hit  Shape: Correct TC: (09,00) CC: (09, 00) JL: Long Pointer position: (1214.0,0.0).
[18:44:55]: Target: Hit  Shape: Correct TC: (07,02) CC: (07, 02) JL: Short Pointer position: (889.9,311.0).
[18:44:56]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:45:01]: Target: Hit  Shape: Correct TC: (05,02) CC: (05, 02) JL: Medium Pointer position: (346.9,154.9).
[18:45:06]: Target: Hit  Shape: Correct TC: (09,06) CC: (09, 06) JL: Short Pointer position: (588.4,381.7).
[18:45:09]: Target: Miss Shape: Correct TC: (19,07) CC: (18, 04) JL: Medium Pointer position: (1103.0,275.2).
[18:45:10]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:45:15]: Target: Hit  Shape: Correct TC: (07,01) CC: (07, 01) JL: Short Pointer position: (919.5,175.5).
[18:45:21]: Target: Hit  Shape: Correct TC: (01,04) CC: (01, 04) JL: Long Pointer position: (183.9,555.3).
[18:45:25]: Target: Hit  Shape: Correct TC: (03,02) CC: (03, 02) JL: Short Pointer position: (425.0,302.1).
```

```
[18:45:26]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:45:32]: Target: Hit  Shape: Correct TC: (16,02) CC: (16, 02) JL: Medium Pointer position: (1004.2,156.3).
[18:45:35]: Target: Miss Shape: Correct TC: (16,08) CC: (16, 02) JL: Short Pointer position: (992.0,171.8).
[18:45:36]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:45:40]: Target: Hit  Shape: Correct TC: (01,02) CC: (01, 02) JL: Long Pointer position: (163.9,322.4).
[18:45:41]: Changed grid size. Grid height: 10 Grid width: 20 Cell height: 61.4 Cell width: 60.7
[18:45:46]: Target: Hit  Shape: Correct TC: (16,01) CC: (16, 01) JL: Long Pointer position: (993.7,101.5).
[18:45:47]: Changed grid size. Grid height: 5 Grid width: 10 Cell height: 122.8 Cell width: 121.4
[18:45:50]: Target: Hit  Shape: Correct TC: (03,02) CC: (03, 02) JL: Medium Pointer position: (421.7,306.8).
[18:45:50]: Test session ended.
```

Appendix D: Example of raw comment log

[09:58:48]: Push throw starting
[10:00:56]: Changing pointing arm
[10:01:08]: Missed the second to last one due to aiming
[10:04:06]: No problems with the grab push
[10:05:16]: The cursor jumps down
[10:05:28]: The pointer has been positioned on the target and the tilt is performed and the pointer will then jump down, up or to the sides.
[10:06:09]: He is doing better than most participants by the looks of it
[10:06:29]: Only missed a few
[10:08:41]: Did great with the swipe
[10:12:44]: No issues with the throw pull either
[10:15:41]: No system errors. Didn’t lock the pointer during the grab pull
[10:17:07]: The pointer jumes a little
[10:17:32]: Right jump
[10:17:40]: Small jump down
[10:17:51]: Some small jumps. But like I wrote earlier, it’s small jumps after the pointer was centered on the target and the technique was performed

Comparison of Push Techniques for Cross-Device Interaction Between Mobile Devices and Large Displays

Bjarke M. Lauridsen
Aalborg University
blauri10@student.aau.dk

Ivan S. Penchev
Aalborg University
ipench14@student.aau.dk

Elias Ringhauge
Aalborg University
eringh10@student.aau.dk

Eric V. Ruder
Aalborg University
eruder10@student.aau.dk

ABSTRACT
In recent years, research into cross-device interaction techniques has increased. Much of this research focuses on interaction between mobile devices and large displays. We contribute to this body of knowledge with an empirical comparison of four different push techniques - *Pinch*, *Swipe*, *Throw*, and *Tilt* for interaction between mobile devices and large displays. We report on success rate, efficiency and accuracy. We also present the ease of use of techniques as perceived by users. We show that *Swipe* was the most effective in terms of success rate, efficiency and accuracy. Furthermore, *Swipe* gathered the highest score, in regards to ease of use, by users. Participants also reported that *Pinch* was the most fun to use.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous; See http://acm.org/about/class/1998/ for the full list of ACM classifiers. This section is required.

Author Keywords
Interaction Techniques; Cross-Device Interaction; Natural User Interaction; Kinect; Mid-air Gestures; Smartphones; Large Displays; Data Transfer

INTRODUCTION
The evolution of ways people interact with the digital world is noteworthy considering the short life-span of computing. How we use our devices, which devices we use, and the context in which we use them has been continually under transformation. From portable personal computers originally considered mostly for specialized field applications such as accountancy, military use, or for sales representatives, which addressed mobility of a person's workspace, to modern hand-held devices which presents their users with such degrees of freedom that ultimately workspaces are becoming more ubiquitous. In non-work context people are now connected mostly everywhere, which aids us in a search for information or in communicating, changing the way we interact. This expansion has not only increased mobile computing due to greater convenience, but also made it widespread.[6]

As numerous divergent devices are being adopted in different domains and contexts, understanding cross-device interaction is currently becoming more important and relevant; after all, people take their hand-held devices into situations where other technologies are active. This ubiquitous presence of devices means that they can be used to enhance everyday situations in all kind of places. Imagine if a public display could morph from a one-way broadcast device that merely shows visual content to a two-way interaction device that provides a more engaging and immersive experience. This emergence of cross-device communication opportunities prompts a need to understand how different interaction techniques perform in use, i.e. in terms of how easily, quickly and accurately, or in terms of how enjoyable or satisfying it is to interact in this way.

Research in the area of cross-device interaction is increasing with the changing trends. Earliest examples are in the late 90's, within ubiquitous computing, with Rekimoto's work. He argued for what he called multi computer user interface and that interaction techniques must overcome the boundaries among devices in multi-device settings[16].

Recent HCI research has focused on how to include natural modality more in cross-device interaction, contributing to what should be know as cross-device natural user interaction. Some researchers used spatial information [11, 12], others used touch [19], or combined touch with air gestures [4]. But we still have limited understanding of how to design cross-device natural user interaction techniques and we lack empirical studies of this.

Inspired by the opportunities presented by such challenges, this paper reports on a empirical study between four different cross-device natural user interaction techniques for data transfer between mobile devices and large displays. We discovered that out of the four techniques we developed and implemented, *Swipe* was the most effective technique. We also show that even though *Pinch* is not as effective as the others, users described it as a fun technique.

RELATED WORK

Public displays are an inherently visual medium with graphics and animations that are increasingly used to visualize data. However the general case is that presented data can only be viewed and not interacted with. The current trend is a change from non-interactive information broadcasters to active medium information exchange. Combining this with the recent trend of increased number of hand-held device per capita an opportunity for new cross-device applications in public space arises, providing new opportunities for exploring cross-device interaction (CDI), in this context.

In a historic perspective, one of the earliest working cross-device applications is by Myers et al. [13]. One of the applications realized within their *Pebbles project* is SlideShow Commander that utilized Personal Devices Assistants (PDA) to control a PowerPoint presentation running on other computer or laptop. It was possible not just moving between slides, but also scribbling and writing on the PDA slides, while annotations are shown on the presentation for the audience. However the idea of cross-device has deeper roots in ubiquitous computing. Rekimoto's work on *pick-and-drop technique* is one of the earliest examples for exploring a technique that spawns between multiple devices. The technique " *allows a user to pick up an object on a display and drop it on another display as if he/she were manipulating a physical object.*" [15]. These two early work examples, even though in different fields, are related and would provide a foundation for additional research.

An example of such research is that of Boring et al. [2] who not only build a cross-device application but explored the implications of different techniques on it. Boring et al. explored the transfer of data from a large public display onto a mobile device. They created a method of transferring data from a large screen by using the camera on the mobile device. The user would take a picture of whatever content they were interested in, after which the application would query the content server with the picture taken from the user. Through visual analyses, the content server would determine what content the user was interested in and would return that content to them. They show that there is a need for enabling data exchange between mobile devices and public displays. In another study, Boring et al. [3] investigated cross-device interaction between large displays and mobile phones. Investigating three different interaction techniques in order to continuously control a pointer on a large screen from a mobile device. *Move* and *Tilt*, two of the three interaction techniques, enabled faster selection time compared to the last one, *Scroll*, but at the cost of higher error rates. They showed that different interaction techniques have certain strengths and weaknesses, and depending on the context and use, certain techniques are more effective.

Boring et al.'s idea [3] of how to control a public display is only one side of cross-device interaction, a different idea from Nielsen et al. [14] uses collaboration surface made from multiple devices to investigate the use of multiple device together, by allowing a number of devices being put next to each other and " pinched " together to form a larger collaborative workspace. In order to expand on the idea of common workspace, a movement from use of multiple devices to build one large display was needed. Schmidt et al. [18] proposed a cross-device interaction style for mobiles and surfaces where one can use multiple phones to interact with a digital surface.The researchers point out that *"natural forms of interaction have evolved for personal devices that we carry with us (mobiles) as well as for shared interactive displays around us (surfaces) but interaction across the two remains cumbersome in practice"*. In order to overcome this they propose the use of mobiles as tangible input on the surface in a stylus like fashion.

A combination of the ideas is presented by Skov et al. [20] illustrating six different cross-device interaction techniques for the case of card playing. A player can see their own cards on their phone and use three different techniques for playing a card from the hand-held to the tablet, which is placed on a table. In the other direction, i.e. when drawing a card, the player also has three techniques to choose from. The study aims to quantitatively evaluate each of the techniques and shows that there is a difference in time and number of errors between the techniques. They recorded two types of errors, namely, interaction errors and play errors. The number of interaction errors shows how difficult it is for a user to perform a given technique while play errors represent the errors related to the game and is recorded when the user plays a wrong card. The difference in interaction errors is apparent, especially between two of the techniques for playing a card.

The research above illustrates two different direction movement in cross-device interaction, what they have in common, is data transfer between devices. Hamilton and Wigdor's work [7] aggregates much of the works above and clearly articulates the data transfer. They create a prototype framework for cross-device applications by combining a number of interaction techniques for data transfer, chaining tasks, and managing interactions sessions. Data transfer is a challenge and as such there are different approaches for solution. Marquardt et al. [12] study cross-device interaction on tablets with a natural modality, by involving spatial information through proxemics.Based on the constructs of f-formation, micro-mobility, and co-present collaboration, they build their prototype with the idea of support for fluid and minimally disruptive interaction in document transfer. Bragdon et al.[4] propose using a combination of air gestures and touch.They aim to design, implement and test a system that allows, a group of users, to interact using air gestures and touch gestures. The purpose is to increase control, support democratic access, and share items across multiple personal devices such as smartphones and laptops where the primary design goal is fluid, democratic sharing of content on a common display.

INTERACTION TECHNIQUES

In this section we illustrate and describe techniques that can be used to push information from mobile devices to large displays with the purpose of empirically comparing them to each other. We want to find intuitive techniques that allow a user to walk up and use a large display. The techniques are characterized by allowing the users to interact with a large screen in a natural way using one or both hands and their mobile phone.

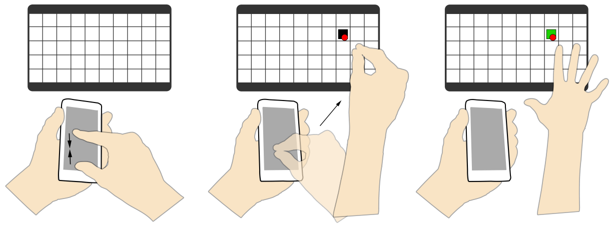
All the techniques were found in the literature and chosen based on a set of criteria outlined in Table 1. We are also interested in examining the effect of each technique on targets of two different sizes, large and small, relative to the size of the screen because the need for precision of an application is not always the same. Sometimes an applications needs to allow the user to place some data in really specific locations, where sometimes a general approximation is enough. We chose four techniques named *Pinch*, *Swipe*, *Throw* and *Tilt* because they fulfill different aspects of the criteria requirements in Table 1 and allow us to compare them to each other in an experimental setup with simple tasks.

Criteria	Description
Number of hands	There must be both one-handed and two-handed techniques.
Previously used	To avoid designing and testing a set of novel techniques, we had the criterion that all techniques must have been used by others before we would use them.
Complexity	The techniques must differ in their complexity and therefore we included techniques with different amount of steps.
Natural feel	There must be a natural feel to the techniques in some way.
Time	The time it takes to perform the different techniques must be different.

Table 1. This table describes the set of criteria.

In the following we explain why these four techniques were chosen and how they should be performed.

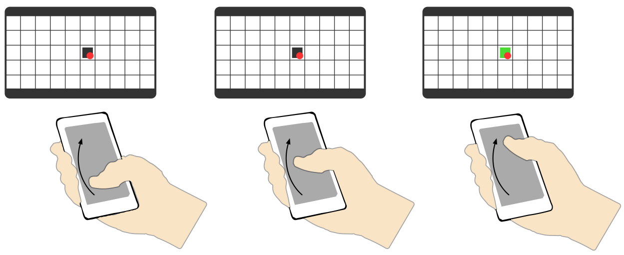
The *Pinch* technique (Figure 1) is used in [8] by Ikematsu et al., as part of a drag-and-drop method for moving data objects between devices. Chen et al. uses a pinching gesture in [5] for cross-device interaction between a smartphone and a smartwatch to control volume. Benko and Wilson [1] used the *Pinch* technique for interacting with omnidirectional visualizations in a dome. This technique is again a combination of the pointing technique used by Scheible et al. [17] and the aforementioned pinching techniques. The reason for including this technique was to imitate the natural action of picking up a real object e.g. piece of paper, and then moving it to another location. With *Pinch* we get a two handed technique which requires the user to perform a series of steps and are thus considered more complex and time consuming than the one handed *Swipe* and *Tilt*. The *Pinch* technique is performed by *1*) holding the phone in one hand and making a pinch gesture on the phone with the other hand (fig. 1a), subsequently closing the hand; *2*) pointing at a target on the large display with the closed hand (fig. 1b); and *3*) opening the hand to complete the technique (fig. 1c).



(a) (b) (c)

Figure 1. (a) First step of the *Pinch* technique is to make a pinch gesture on the phone using the index finger and thumb. (b) Second step is to move and use the pinched hand as a pointer. (c) Third step is to release by opening the hand.

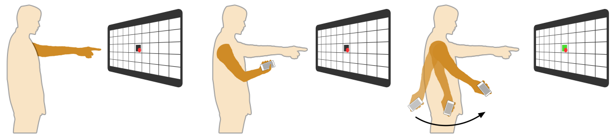
The *Swipe* technique (Figure 2) is used by Bragdon et al. [4] in Code Space, a system using the Kinect and smartphones to support developer meetings. Bragdon et al. describe the technique as: *"cross-device interaction with touch and air pointing"* and the swipe motion is described as *"flicking up on the touch screen"*. This technique was chosen because of its simplistic design and the low level of complexity. Only one hand is needed and the amount of effort and time required to execute this technique is minimal compared to other techniques. *Swipe* is copied exactly as described in [4]. The *Swipe* technique is performed by *1*) pointing at a target on the large display with the phone in a stretched arm, and *2*) making a forwards swipe motion with the thumb on the phone's screen (fig. 2).



(a) (b) (c)

Figure 2. The *Swipe* technique is performed by using the thumb to swipe from (a) to (c).

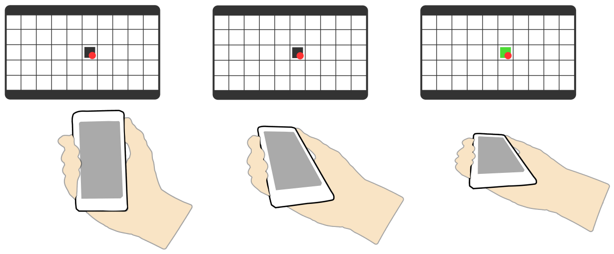
The *Throw* technique (Figure 3) is a combination of a technique for pointing [17] i.e. using a hand as a cursor in mid-air, and a throw technique described by Walter et al. [21] used in a system for sharing information on large public displays. We chose to include this technique based on its natural and playful design. The technique mimics the real world scenario of throwing something like a ball somewhere or to someone. *Throw* is two handed, more complex than aforementioned techniques, and takes a little longer to execute because of the increased number of steps. The *Throw* technique is performed by *1*) pointing at a target on the large display with one hand (fig. 3a); *2*) holding the phone in the other hand and selecting data (fig. 3b); and *3*) making a swinging motion towards the large display (fig. 3c).



(a) (b) (c)

Figure 3. (a) First step of the *Throw* technique is to point at the screen. (b) Second step is to select data. (c) Third and final step is to do a swinging motion with the phone towards the screen.

The *Tilt* technique (Figure 4) is used in a collaborative application by Lucero et al. [9] to transfer an object from a large display to the user's smartphone. Boring et al. [3] use the tilt technique when moving a pointer on a display using a phone and though not the same application, the execution of the technique is the same. We chose this technique because it is one handed, relatively low complexity, and much like the *Swipe* it is generally easy to use. When the direction is reversed, *Tilt* is an exact copy of the way Lucero et al. describe the technique. The *Tilt* technique is performed by *1*) pointing at a target on the large display with the phone in a stretched arm, and *2*) making a forwards tilt with the phone (fig. 4).



(a) (b) (c)

Figure 4. The *Tilt* technique is performed by doing a forward tilt motion with the phone from (a) to (c).

As mentioned, the techniques have been used in previously published papers where they were part of applications or systems, facilitating interaction between devices such as phone to display interaction and vice versa. The described techniques are different in the way they are performed but also the number of hands that are required to make them work. We chose two techniques that are one-handed (*Swipe* and *Tilt*) and two techniques that are two-handed (*Throw* and *Pinch*). Also, to vary the complexity we chose two techniques that require more steps (*Throw* and *Pinch*), and two techniques that require less steps for the user to complete the technique (*Swipe* and *Tilt*).

All the techniques make use of some combination of pointing in mid-air and touch gestures on the smartphone screen. The mid-air pointing interaction is achieved by using the Microsoft Kinect for Windows which uses a depth camera making it possible to track a user's hand in mid-air. As for the touch gestures, smartphones have an accelerometer and a touchscreen, making it possible to detect motion input and detect contact between e.g. a finger and the screen. These technologies are, in combination with each other, used to recognize the four techniques described in this section.

EXPERIMENT

The four cross-device interaction techniques mentioned above were implemented and then evaluated in a lab study in order to judge their performance compared to each other. We are interested in knowing whether or not the different techniques with different target sizes have an effect on the efficiency, accuracy, and ease of use of pushing information to a large display. Therefore, we developed an application that would allow us to run experiments and test the effect of the different techniques and target sizes. We utilized a Microsoft Kinect, a 65' inch Panasonic television with a 1920×1080 resolution and a Samsung Galaxy SII to create the experimental application. An overview of the experiment setup can be seen in Figure 5.

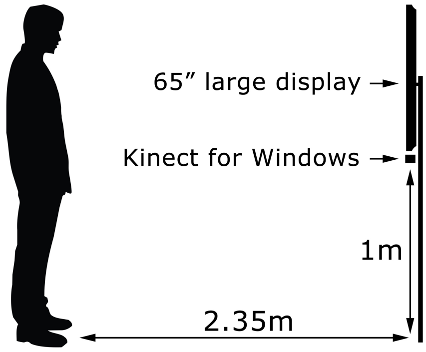


Figure 5. An overview of the entire setup of our experiment.

Implementation

The 4 techniques (*Pinch*, *Swipe*, *Throw*, and *Tilt*) were implemented in order to push data onto the large display. They were implemented in a simple and short target practice application, where the goal was to "hit" the target on the display with the shown technique.

A grid system was implemented in the test application, where each cell of the grid is a possible target. As mentioned, we were interested in measuring the effect of different target sizes on the different techniques, therefore the grid is implemented in two different sizes. The grid system can have large cells, where the grid is 5×10 cells and each cell is 61 pixels or 7.3 cm wide, or it could also have small cells, where the grid is 10×20 cells and each cell is 122 pixels, or 14.6 cm wide. The target is located in one of those cells, and scales accordingly to the size of the cell(See Figure 6a).

A red dot works as the pointer in the screen; it is the location that would be hit when the user performed the given technique. The yellow highlighted cell is the cell in which the pointer is currently located inside. This is extra feedback for the user so he knows exactly were he will hit once he performs the technique.

The developed mobile application was simple. It showed two shapes, a circle and a square, which the user could choose to push to the display. The display would tell which shape was the correct one by having that shape as the target in one of the grid cells. We chose two shapes so that it did not become a search problem with users spending too much time searching for the correct shape. We wanted the user to spend some time orienting him or her self with the phone and not just simply performing the gesture without paying any attention to the phone at all. The phone screen can be seen in Figure 6b.

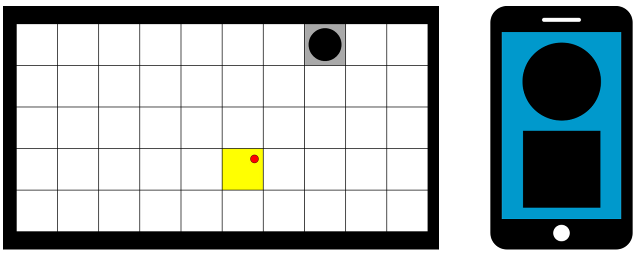
Users would control the pointer on the large display with their hands. Which ever hand was closest to the screen would determine the position of the pointer on the large screen. This meant that users could switch hands whenever they pleased at any point during the test.

The *Pinch* technique was implemented with the help of the Kinect and the touch screen on the mobile phone. This technique started by having the user pinch the shape on the screen of the mobile phone and close his or her hand around it, as if to grab it. The Kinect would then look for a opening of the hand motion, on the pointer hand, and take that as the target point.

The *Swipe* technique was implemented with the touch screen of the phone. Here, we detected when a significant swipe happened on the screen, and then use the pointer location to place the shape that was swiped up onto the screen.

The *Throw* technique was also implemented with the help of the Kinect and the accelerometer on the mobile phone. The Kinect looked at the user to recognize when a user moved the mobile phone from 10 centimeters behind the hip to 10 centimeters in front of the hip. At the same time, the phone detects when a significant change in the accelerometer happened, so to not simply detect an unintentional wave of the arm. The Kinect would then use the position of the other hand to see where on the screen the user intended to perform the *Throw* technique towards.

The *Tilt* technique was implemented mostly with the accelerometer of the phone, by checking for a significant change in the z and y axis of the accelerometer, as if tilting the phone forward.



(a) (b)

Figure 6. (a) The large display screen of the application, with the red dot in the highlighted square as the starting point and the black circle as the target. (b) The phone screen showing the two shapes.

Experimental Design

The experiment was conducted as a within-subject research, with the four different interaction techniques and two target sizes as independent variables.

The within-subject research was chosen because we wanted to minimize the amount of subjects needed in order to get a significant result. We also believed that the learning effect would not be as pronounced since the four techniques are very different from each other. We chose to investigate techniques because we were interested in learning about the way people interact with large displays, and the two target sizes to investigate precision. Sometimes users need to be as precise as possible, and sometimes they just need to be able to interact with a large display.

For the dependent variables, different measures of completion time and hit success were used, as well as a short questionnaire to get the user's satisfaction with regard to the given interaction technique. Which technique started the test was randomized in order to mitigate the learning effect on the entire set of tests. In the end, the *Pinch* gesture started 26.4% of all tests, *Swipe* started 22.7% of all tests, *Throw* started 24.5% of all tests, and *Tilt* started 26.4% of all tests. All of this was automatically logged, and every test session was also video recorded in order to be able to go through them in case we wanted to go into detail in one of the test sessions.

A simple logging mechanism was developed, which created a unique file for each user and outputted all attempts into that file. In the end, the result was a list of 53 files, one for each test participant, where each file would have a list of attempts and target size switches. Each attempt would have a time stamp, whether the user hit the target or not, whether he selected the correct shape, were the target was, and where the participant hit. These where the following measures that we were able to deduce from the log files that were generated:

Total Time: This was the time each user spent completing the test for a given interaction technique. This was measured from the time each user had hit his first target after a practice period of three tries until he had hit his last target. There were a total of 18 targets, plus the first target used for calibration.

Time per target: This was the time each user spent hitting each of the targets. This was measured as the time since each user last hit a target until he hit the next one.

Hit success: Whether or not each user hit the given target. Current pointer and target position (in both cell and pixels coordinates) were also recorded in order to give a precise measure of accuracy for each attempt in terms of *distance to target*.

Ease of use: Each user was given a questionnaire after having gone through each interaction technique. There were 6 questions, all taken from the USE questionnaire [10]. These were asked to get an understanding of how useful and easy to use each technique was. The 6 questions were the following:

- It is easy to use
- Using it is effortless
- It is easy to learn to use
- I can use it successfully every time
- I quickly became skillful with it
- I learned how to use it quickly

Users were able to rate their answers to each question on a 7 point Likert scale. We also wrote down any comments made during the experiment and combined them with the questionnaire responses to get a better understanding of the user's response to each of the techniques.

Participants

In total, 53 people took part in our experiment, which was conducted in a usability lab. The participants where between 20-45 years old (M: 24.4, SD: 4.3) and were between 1.63 and 1.95 meters tall (M: 1.82, SD:7.8). 88.7% of users were right handed, 90.6% were male, and 96.2% of them were smartphone users. Of those who owned smartphones, they had owned one for 2-15 years (M:5, SD:2.1). They were recruited through a mixture of our social network and recruitment posters around the campus.

The Experiment

Each test subject was taken into the usability lab and given a short introduction to what we were doing and why. We then explained how the system worked and what they had to do. We would hand them a phone, ask them to stand on a marked cross, so that the distance to the screen would always be the same, and start the test.

The application chose at random one of the four techniques and displayed a short explanatory movie of how to perform the technique on a screen right beside the main application display(See Figure 7a).

(a) (b)

Figure 7. (a) The main screen on the left with the tutorial video screen on the right. (b) The experiment in progress, as seen from the right.

The user would then have three practice attempts, in order to get familiar with the technique. Nothing was logged during the practice phase. A shape would appear, either a square or a circle, at one of the cells in the grid. The user would have to choose the correct shape on the phone and perform the technique with that shape selected. The shapes on the phone (fig. 6b) would randomly change positions, so that the user would have to check the phone after every technique. The target would also randomly change size from small to large or vice-versa. In reality, the target sequences where hard coded by us in such a way that there was an equal distribution of small and large targets. We also made sure that there was an equal distribution of distances between each target. We classified them as short jumps, medium jumps, and long jumps. A short jump was 2 large cells(4 small cells), a medium jump was 4 large cells(8 small cells), and a large jump was 6 large cells(12 small cells). After a practice phase of 3 practice targets, a calibration start target would be shown. This is so that we could calculate the distance between all other targets correctly. The user would then go through the rest of the test (18 targets), going through a total of 22 targets.

That means that our experiment had the following list of conditions:

- Technique (4)
- Target size (2)
- Target jump distance (3)
- Repetition (3)

This means that each user had a total of $4 \times 2 \times 3 \times 3 = 72$ targets.

After going through every target for one technique, the user would then be asked to fill in the short questionnaire regarding the technique just tried in terms of how natural it felt based on ease of use measures.

This entire process would be repeated four times in total, once for each technique. After that, we presented them with a short demographics questionnaire, in order to better understand the user. We asked them about their age, height, if they were left or right handed, if they had a smartphone, for how long, if they had any experience with a Kinect, Wii, Playstation Move, or any other similar air gesture based technologies, and how often they used them. Finally, we thanked them for their time. The entire test took on average 15 minutes.

RESULTS

We will now present the results that we achieved through out our experiment and also how they were achieved. We will first present our findings in respect to success rate(based on hit success), then in respect to efficiency(based on time per target), and finally in respect to accuracy(based on distance to target). Finally, we will look at the questionnaires and show the significant findings there in terms of ease of use. We will discuss these results later, in the Discussion section. Each of the four interaction techniques, *Pinch*, *Swipe*, *Throw* and *Tilt* were completed 18 times per participant. For each target size, each technique was performed 477 times.

Success Rate

In this section we will present the results related to the success of hitting a target. We will perform an analysis of the hit success rate and the effect of each technique with respect to target sizes.

The success rate's mean and standard deviation, M(St.D.) for each technique for small(S) and large(L) target sizes can be seen in Table 2 and in Figure 8 the results are shown as a graph.

Hit Success Means

	Pinch (n = 477)	Swipe (n = 477)	Throw (n = 477)	Tilt (n = 477)
S	0.65 (0.48)	0.91 (0.29)	0.83 (0.37)	0.58 (0.49)
L	0.78 (0.41)	0.97 (0.18)	0.94 (0.25)	0.78 (0.41)

Table 2. Mean hit and standard deviation for each technique per target for small(S) and large(L) targets.

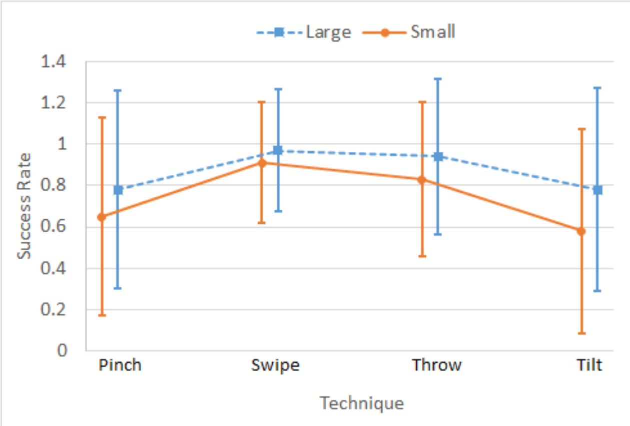


Figure 8. Mean and standard deviation for each technique in regards to hit success rate per target.

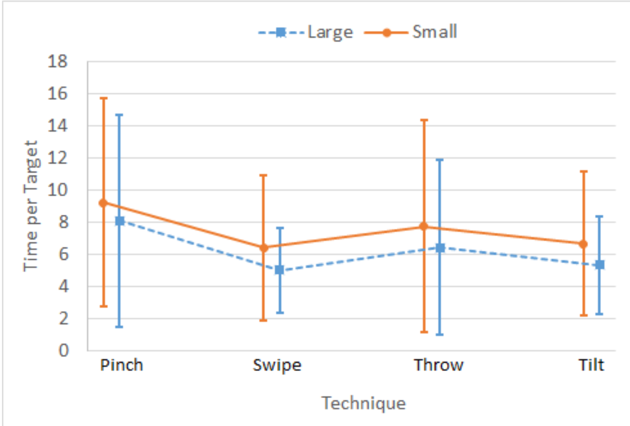


Figure 9. Mean and standard deviation for each technique in regards to time per target.

In order to see the effect of each technique on the hit ratio per target for the different target sizes, we performed two different one-way ANOVA's, where we split the data between the two different target sizes. We then performed a post-hoc pairwise LSD test to see where the significant difference were.

For the the small target, we got $(p < 0.001)$, $(F(2.722, 1295.674) = 62.754)$, (Greenhouse-Geisser correction: 0.907). The pairwise test showed that all techniques were significantly different. *Pinch* and *Swipe* had $(p < 0.001)$, *Pinch* and *Throw* $(p < 0.001)$, *Pinch* and *Tilt* $(p = 0.031)$, *Swipe* and *Throw* $(p = 0.001)$, *Swipe* and *Tilt* $(p < 0.001)$ and finally, *Throw* and *Tilt* had $(p < 0.001)$.

For the large target, we got $(p < 0.001)$, $(F(2.472, 1176.749) = 42.773)$, (Greenhouse-Geisser correction: 0.824). The pairwise test showed that all of the techniques, with the exception of *Pinch* and *Tilt*, were statistically different from each other. The results were as following: *Pinch* and *Swipe* $(p < 0.001)$, *Pinch* and *Throw* $(p < 0.001)$, *Pinch* and *Tilt* $(p = 1.000)$, *Swipe* and *Throw* $(p = 0.025)$, *Swipe* and *Tilt* $(p < 0.001)$, and finally *Throw* and *Tilt* $(p < 0.001)$.

Efficiency

In this section we present the efficiency results which defines the amount of time spent performing a technique. We perform an analysis of the efficiency and the effect of each technique with respect to target sizes.

Table 3 shows the mean time per target in seconds for each of the techniques as well as their standard deviation for both target sizes.

Time per Target Means

	Pinch (n = 477)	Swipe (n = 477)	Throw (n = 477)	Tilt (n = 477)
S	9.23 (6.48)	6.41 (4.49)	7.73 (6.60)	6.67 (4.49)
L	8.09 (6.60)	5.01 (2.66)	6.42 (5.43)	5.33 (3.04)

Table 3. Mean time and standard deviation for each technique per target for small(S) and large(L) targets.

To get the effect of each technique on the time for the different target sizes, we performed two one-way ANOVA's, one for the small target and another for the large target. A post-hoc pairwise LSD test to see where the significant difference were.

For the the small target, we got $(p < 0.001)$, $(F(2.740, 1304.290) = 26.523)$, (Greenhouse-Geisser correction: 0.913). The pairwise test showed that all techniques were significantly different except *Swipe* and *Tilt*. *Pinch* and *Swipe* had $(p < 0.001)$, *Pinch* and *Throw* $(p < 0.001)$, *Pinch* and *Tilt* $(p < 0.001)$, *Swipe* and *Throw* $(p < 0.001)$, *Swipe* and *Tilt* $(p = 0.354)$ and finally, *Throw* and *Tilt* had $(p = 0.004)$.

For the large target, we got $(p < 0.001)$, $(F(2.221, 1057.144) = 44.539)$, (Greenhouse-Geisser correction: 0.740). The pairwise test showed that all of the techniques, with the exception of *Swipe* and *Tilt*, were statistically different from each other. The results were as following: *Pinch* and *Swipe* $(p < 0.001)$, *Pinch* and *Throw* $(p < 0.001)$, *Pinch* and *Tilt* $(p < 0.001)$, *Swipe* and *Throw* $(p < 0.001)$, *Swipe* and *Tilt* $(p = 0.077)$, and finally *Throw* and *Tilt* $(p < 0.001)$.

Accuracy

Finally, we will perform an analysis of the accuracy and the effect of each technique with respect to target sizes. Here, we took three different measures of accuracy; the distance between where the user hit and the target cell as well as taking the x and y axis independently. These were all measured in pixels. This was because there were signs that certain techniques might miss in a specific direction, and we wanted to see if the data supported that. An overview of the distance mean and standard deviation in pixels can be seen in Table 4 and Figure 10.

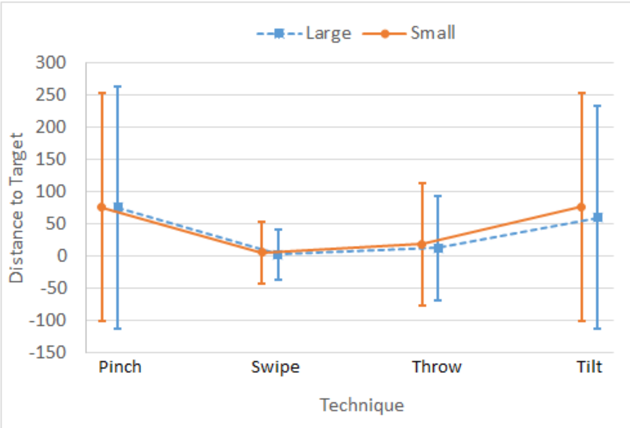


Figure 10. Mean and standard deviation for each technique in regards to distance.

Distance Means

	Pinch (n = 477)	Swipe (n = 477)	Throw (n = 477)	Tilt (n = 477)
S	75.97 (176.15)	5.40 (47.81)	18.60 (95.29)	76.51 (177.45)
L	75.41 (187.58)	2.29 (38.82)	12.37 (81.36)	59.88 (172.22)

Table 4. Mean and standard deviation for the distance for each technique per target for small(S) and large(L) targets.

We performed two one way ANOVA's, one for each target size, on the distance data.

For the the small target, we got ($p < 0.001$), ($F(2.341, 1114.249) = 37.504$), (Greenhouse-Geisser correction: 0.780). The pairwise test showed that all techniques were significantly different except *Pinch* and *Tilt*. *Pinch* and *Swipe* had ($p < 0.001$), *Pinch* and *Throw* ($p < 0.001$), *Pinch* and *Tilt* ($p = 0.961$), *Swipe* and *Throw* ($p = 0.008$), *Swipe* and *Tilt* ($p < 0.001$) and finally, *Throw* and *Tilt* had ($p < 0.001$).

For the large target, we got ($p < 0.001$), ($F(2.176, 1036.004) = 33.315$), (Greenhouse-Geisser correction: 0.725). The pairwise test showed that all of the techniques, with the exception of *Pinch* and *Tilt*, were statistically different from each other. The results were as following: *Pinch* and *Swipe* had ($p < 0.001$), *Pinch* and *Throw* ($p < 0.001$), *Pinch* and *Tilt* ($p = 0.171$), *Swipe* and *Throw* ($p = 0.015$), *Swipe* and *Tilt* ($p < 0.001$) and finally, *Throw* and *Tilt* had ($p < 0.001$).

X and Y Distance Means

	Pinch (n = 477)	Swipe (n = 477)	Throw (n = 477)	Tilt (n = 477)
S XD.	54.33 (140.87)	3.78 (40.15)	10.32 (81.84)	49.23 (151.36)
L XD.	55.32 (159.88)	1.88 (31.74)	10.32 (74.01)	49.23 (157.35)
S YD.	42.90 (110.30)	2.00 (26.16)	8.00 (49.37)	39.3 (99.82)3
L YD.	37.41 (104.18)	1.14 (22.37)	4.33 (34.21)	22.86 (73.75)

Table 5. Mean and standard deviation for distance on the X-Axis(XD) and distance on the Y-Axis(YD) for each technique per target for small(S) and large(L) targets.

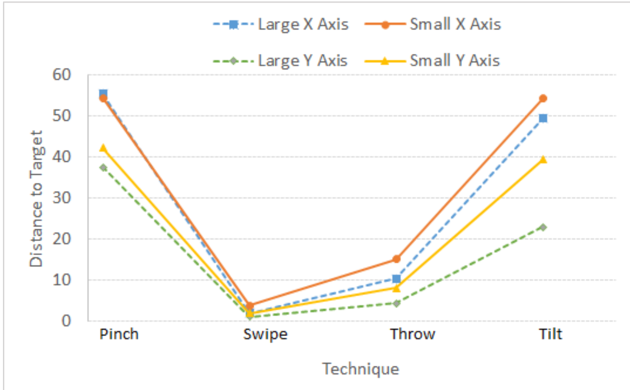


Figure 11. Mean and standard deviation for each technique in regards distance on the x-axis and y-axis.

To get a better understanding of the distance we decided to examine the distance in regards to the x-axis and in regards to the y-axis. The results can be seen in Table 5 and Figure 11

Results of Ease of Use Questionnaire

Our questionnaire was based on USE, which used Likert scale 4.2, when encoding the data we did it as continuous variable, as such *"strongly disagree"* got a value of 1, and *"strongly agree"* a value ot 7. After that the cumulative value per technique, based on the different questions, was calculated, the data was ploted and presented in Figure 12.

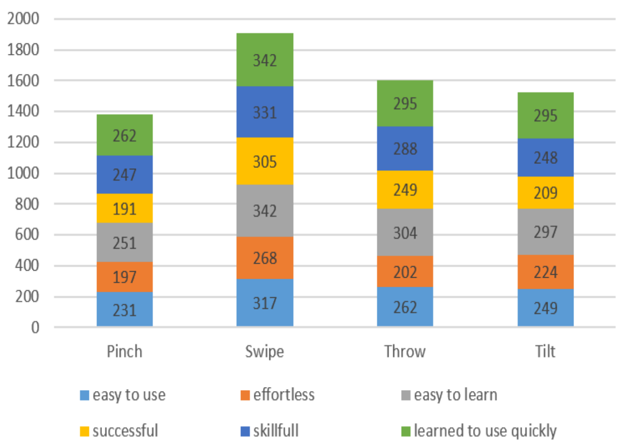


Figure 12. Cumulative values of survey questions per technique

A One-Way MANOVA was applied ($F(18, 574.66) = 5.118$, $p < 0.000$) which showed that there is statistical differences between each techniques. In order to specify where this differences lie we perform an post hoc test, we summarize the results in regards to each question from the survey, as follow: *1)* "It is easy to use" *Swipe* is statistically different, however there is no statistical difference between *Throw*, *Tilt*, and *Pinch*; *2)* "Using it is effortless" *Swipe* is statistically different, however there is no statistical difference between *Throw*, *Tilt*, and *Pinch*; *3)* "It is easy to learn to use" *Swipe* and *Pinch* are statistically different, however there is no statistical difference between *Tilt*, and *Throw*; *4)* "I can use it successfully every time" *Swipe* and *Throw* are statistically different, there is no statistical difference between *Tilt*, and *Pinch*; *5*) "I quickly became skillful with it" *Swipe* and *Throw* are statistically different, however there is no statistical difference between *Pinch*, and *Tilt*; *6*) "I learned how to use it quickly" *Swipe* is statistically different, however there is no statistical difference between *Throw*, *Tilt*, and *Pinch*. The results show that the ratings received by the techniques differ in the different aspects of the areas covered by the survey.

DISCUSSION

When discussing our results, it is important to redefine the terms that we use. When talking about success rate, we mean whether or not the technique hit the given target. When we mention efficiency, we are talking about the time it takes to successfully perform that technique. When talking about accuracy, we mean the distance the attempt was from the target(in pixels). It is also important to note that the standard deviation in some of our measures are quite high. This is primarily because the experimental system was not robust enough to get a perfect reading on the users intention to perform a technique. Sometimes, the system would misunderstand the gesture a user made, so that it either activated too early, or did not activate all. Activation means that the system interpreted the gestures the user performed as an attempt to hit the target. This is further discussed in Section 6.7

When looking at the results on the effect of the four techniques, *Pinch*, *Swipe*, *Throw* and *Tilt*, as well as the two target sizes, small and large, on the time per target, the results tell a rather interesting story.

Success Rate

If we look at the results in regards to the effect of the technique on the success rate of each attempt, it is interesting to note that the two techniques that were not significantly different from each other were *Tilt* and *Pinch*. These two techniques both used the hand that controlled the pointer to activate the technique. When tilting the phone forward, usually the hand would move together with the phone causing the pointer to displace itself from the users intended position. When releasing the *Pinch*, the Kinect would sometimes reevaluate the location of the hand joint, now that it could see the entire hand, which would also cause the pointer to displace itself from the intended position. *Pinch* and *Tilt* were also the techniques that had the largest amount of activation errors due to the implementation of the system. Sometimes, users would show large amount of their palms to the Kinect during a *Pinch*, even though their hand was closed, causing the Kinect to interpret that as an opening of the hand and activate the technique. *Tilt* would sometimes activate if the user moved the mobile around too quickly, especially when orienting the pointer up and down on the screen.

Swipe and *Throw* both had reasonably high success rates. *Throw* did not require the user to actually move the pointer hand while activating the technique. While *Swipe* did require the user to perform some movement on the hand that was used as a pointer, it was very little movement. This is also a technique all smartphone users are familiar with, since a lot of applications use some form swiping to activate some functionality.

Efficiency

There was a significant difference between all techniques, with the exception of *Swipe* and *Tilt*. These were the two one handed techniques that we chose. The range of movement needed in order to activate these two techniques was rather limited, the full motion could be achieved quite quickly and is quite similar for both of them. This is why they are not statistically different from each other. *Swipe* and *Tilt*, are on average, at least a second faster then the other two techniques. Their standard deviations are also smaller, which means that users were more consistent, with regards to how long it took to hit each target, with these two techniques.

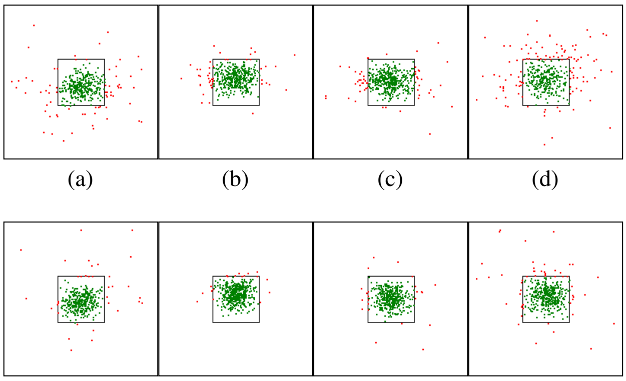
Looking at the two other techniques, *Pinch* and *Throw*, their times also reflect the range of motion needed in order to activate each technique. *Pinch* requires the user to pinch the shape on their phone, lift their hand up, direct it on the screen, and then finally let go. This can be seen in its mean, where it takes almost 1.59 seconds longer to perform than the second longest technique, *Throw*. *Throw* also requires a considerable range of motion in order to activate: point with one arm, select the shape on the phone with the other arm, bring your arm back and then finally swing it forward. Both two handed techniques take significantly longer to perform than their one handed counter-parts.

We noticed that users would spent relatively little time getting into the general vicinity of the target, and would spend most of their time per attempt getting the pointer on top of the actual target. This was more pronounced for the small target, were users would perform smaller, more careful adjustments in order to not overshot the target, which can be seen when comparing the mean times of each technique for the two different target sizes.

Accuracy

If we look at the results regarding the distance from the target, it paints quite a clear image of which of the techniques are more accurate. *Swipe* is by far the most accurate of the four techniques. It is so precise, that it is actually more precise for the small target than all other techniques are for the large target. Figure 13 is an image that shows the location of each hit compared to the given target cell.

Figure 13 shows that *Swipe* and *Throw* have a large concentration of hits inside the target cell, where *Pinch* and *Tilt* have quite a spread of hits outside the target cell. This shows us that users are capable of hitting the target with *Swipe* and *Throw* more consistently and accurately than with the other two techniques by considerable amounts.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 13. Hit location illustration for each technique. Green is a hit, red is a miss (a) Pinch (Small target) (b) Swipe (Small target) (c) Throw (Small target) (d) Tilt (Small target) (e) Pinch (Large target) (f) Swipe (Large target) (g) Throw (Large target) (h) Tilt (Large target)

Also, an interesting thing to notice is that there is a trend where the largest distance from the target is located on the x axis. This can be seen in Table 5, where the distance is broken into the two different axis. For all four techniques, the x axis has a larger mean distance from the target then the y axis. Further research should be done though if one were to make some conclusive statement in regards to the effect of the different axis on the accuracy of the techniques.

Target Size

If we compare the effect of the target size on the different techniques, it does not have that big of an impact on the success rate of each technique compared to each other. *Swipe* and *Throw* are still the techniques with the highest success rate. Our results also show that target size does not change the efficiency of each technique compared to each other. The same is true for the distance from the target cell. Our results point towards a tendency were the size of the target does not influence the measured effectiveness of each technique relative to each other. This of course would require further research, since a sample size of two target sizes is not enough.

Easy of Use

Looking at the questionnaire, we can look at each question and see there is a trend. If the user gave it a high score, then he strongly agreed with the given question. We can then say the accumulated scores of each technique for all questions show a tendency towards the user agreeing that the given technique was easy to use and hence more natural to use. The higher the score, the easier the user felt the given technique was to perform. This means that in general, users considered *Swipe* much easier to use then the other techniques. *Throw* and *Tilt* were considerably close to each other, while *Pinch* trailed quite significantly behind. It is interesting to note that *Throw* and *Tilt* were so close to each other in ease of use, even though *Throw* outperformed *Tilt* considerably, both in regards to time and hit success, as well as the consistency of the technique, shown by the standard deviation.

User Comments

There is also a qualitative aspect to take into account here, which is not reflected well in the surveys or the test results, but were recorded on video and notes during each experiment. For example, a large finding was that correct mapping between the direction in which the user is pointing and the pointer icon on the screen is critical to the success of the application. It is extremely important for the experience of the user to have as close to absolute mapping as possible. Eleven users mentioned having trouble reaching all areas of the screen, but almost all users showed sign of trouble, by for example standing on their toes or stretching their arms as far as possible. One user got so frustrated that she asked for a chair to stand on. With more or different sensors, placed on hands, fingers and phones, we could have had a more precise pointer by being able to determine the direction of the phone and the arm and not solely rely on the position of the users hands. This would most likely lead to more precise results, because the the mapping between the pointer and the users pointing direction would be much closer to a absolute mapping.

In regards to the mobile phone, four users complained that the screen was too small when performing a *Pinch*, making it hard to precisely select the correct shape. Four other users complained that the screen was too large when performing a swipe, since it was hard to reach the correct shape with their thumbs while still maintaining precision with the pointer. Four users mentioned that it was hard to orient themselves with the phone while performing the *Throw* technique, having to break their flow to look down on the phone to select the correct shape. Three users mentioned the same problem with *Swipe*, whenever the targets were too high. This sometimes lead to the mobile phone covering up the target and making it hard for the user to orient themselves to the large display. This was an effect of the relative mapping though, since it was hard to see the screen when their arm was stretched far above their head in order to reach the high targets. The same error could have occurred with *Tilt*, since the user might also end up lifting the phone in front of their field of view, but none of the users mentioned it there.

There is also a the learning aspect of each technique. Six people actually mentioned that the *Pinch* technique was hard to learn, and a large portion of the participants had to be told more specifically how to perform it. The same held true for the *Throw* technique, a large number of the participants had to be told that they had to perform a slightly larger motion in order for the application to understand that a throw motion was being attempted. Very few of the participants had to have further instructions on how the *Swipe* technique worked, and few people needed further help with the *Tilt*. This is most likely a combination of the complexity of some of the techniques as well as the tutorial movies not being descriptive enough. This also lead to frustration, were users thought they were performing the technique correctly and nothing was happening. Four users mentioned being frustrated by the *Pinch* technique, while three users got frustrated with the *Tilt* technique

The fatigue effect is also something to take into consideration. 13 people mentioned being fatigued through out the test and 11 of them first mentioned it during the *Throw* technique. Some users commented that it was because one arm had nothing to do but be uplifted and point to the screen, while the other arm performed all the motion. One user mentioned it would not have been that noticeable if the pointing arm had some motion to perform. We have hopefully accounted for this by rotating the order in which the techniques are performed.

Finally, there is also the fun aspect to take into consideration. Nine users actually mentioned having fun while performing the *Pinch* technique. They compared it to casting a spell or causing explosions on the screen. Three other users mentioned that this technique was especially interesting. It is still worth remembering though that *Pinch* was, by large, the hardest technique for users to learn.

Limitations

There are of course some limitations to the system we developed. Firstly, the intention with the *Tilt* technique was that the users would point and tilt with the phone, but because of our implementation, it was possible for users to point with one hand and tilt the phone with the other. The same holds true for the *Swipe* technique, where users were able to point with one hand and swipe with the other.

The way the system detects an open hand is not very robust: sometimes, depending on the profile of the hand, it misreads the users intentions and believes the user opened his hand.

The system also has a very narrow definition of what throwing means. This is something that can be seen when users were told to "throw" the data from the phone to the screen. Some would perform a much larger tilt motion, others would perform a baseball-like pitching motion.

The Kinect also had some problems determining where the different arm joints were. If the elbow joint was directly behind the hand joint from the Kinects perspective, it would cause the pointer to move erratically since the Kinect was not absolutely sure were the hand joint was. Another problem occurred when the user put their two hands close to each other. The Kinect would have problems determining where the individual hand joints were located.

CONCLUSION

We conducted a study on cross-device interaction techniques, where our focus was on pushing data from a mobile device onto a large display. We compared four different techniques (*Pinch*, *Swipe*, *Throw* and *Tilt*) with two different target sizes. Our main concern was investigating the success rate, efficiency, accuracy and ease of use of the user while using the different interaction techniques.

Our findings show that *Swipe* was the technique with the highest success rate, and at the same time, the most efficient and accurate technique. This was also the technique the users felt easiest to use. We also found that *Pinch* was considered to be a fun and entertaining technique by many of our users. Finally, we also found that the mapping between the screen and the users pointing direction is critical to the applications success. Great care should be taken to achieve as close to an absolute mapping as possible.

FUTURE RESEARCH

Based on our experiment, in which we used two target sizes, we can not determine the difference in effect between targets of different sizes on a large display. As such we would suggest more research including a greater varity of target sizes.

Whereas our experiment only consider push techniques, we would strongly suggest looking at pull techniques for cross-device natural user interaction with large displays. With pull techniques we imagine research investigating the opposite direction i.e. pulling information from a large display to a mobile device. The techniques which may be preferred for pushing information to a large display might not be the best choice for pulling information.

Our experiment has exclusively been concerned with specific measures like success rate, efficiency and accuracy for each technique, and we could suggest for future research that other measures be included in experiment. This includes, but is not limited to, measures on user experience and which techniques users prefer to use for interaction with displays located in public places.

We focused only on the interaction between mobile phones and large displays. In the future though, the range of different personal devices will probably be much more widespread than today, and we suggest further research in this area. An example of this research is the interaction between large displays and devices such as smart watches, tables, and smart glasses.

As a final suggestion, a framework for cross-device natural user interactions might help, for example, developers and researchers with techniques, guidelines and designs for interacting with large displays in the future.

ACKNOWLEDGMENTS

We would like to thank Jeni Paay for the excellent guidance we received throughout the entire project. We would like to thank Dimitrios Raptis for helping us set up the experiment as well as understanding all the data we got out of it. We would also like to thanks Jesper Kjeldskov for making sure we had all the resources we needed. Finally, we would like to thank all the test participants in our experiment.

REFERENCES

1. Hrvoje Benko and Andrew D. Wilson. 2010. Multi-point Interactions with Immersive Omnidirectional Visualizations in a Dome. In *ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. ACM, New York, NY, USA, 19–28. DOI: http://dx.doi.org/10.1145/1936652.1936657

2. Sebastian Boring, Manuela Altendorfer, Gregor Broll, Otmar Hilliges, and Andreas Butz. 2007. Shoot & Copy: Phonecam-based Information Transfer from Public Displays Onto Mobile Phones. In *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology (Mobility '07)*. ACM, New York, NY, USA, 24–31. DOI: http://dx.doi.org/10.1145/1378063.1378068

3. Sebastian Boring, Marko Jurmu, and Andreas Butz. 2009. Scroll, Tilt or Move It: Using Mobile Phones to Continuously Control Pointers on Large Public Displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7 (OZCHI '09)*. ACM, New York, NY, USA, 161–168. DOI: http://dx.doi.org/10.1145/1738826.1738853
4. Andrew Bragdon, Rob DeLine, Ken Hinckley, and Meredith Ringel Morris. 2011. Code Space: Touch + Air Gesture Hybrid Interactions for Supporting Developer Meetings. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. ACM, New York, NY, USA, 212–221. DOI: http://dx.doi.org/10.1145/2076354.2076393
5. Xiang 'Anthony' Chen, Tovi Grossman, Daniel J. Wigdor, and George Fitzmaurice. 2014. Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 159–168. DOI: http://dx.doi.org/10.1145/2556288.2556955
6. Larry Francis. 1997. Mobile Computing: A Fact in Your Future. In *Proceedings of the 15th Annual International Conference on Computer Documentation (SIGDOC '97)*. ACM, New York, NY, USA, 63–67. DOI: http://dx.doi.org/10.1145/263367.263378
7. Peter Hamilton and Daniel J. Wigdor. 2014. Conductor: Enabling and Understanding Cross-device Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2773–2782. DOI: http://dx.doi.org/10.1145/2556288.2557170
8. Kaori Ikematsu and Itiro Siio. 2015. Memory Stones: An Intuitive Information Transfer Technique Between Multi-touch Computers. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications (HotMobile '15)*. ACM, New York, NY, USA, 3–8. DOI: http://dx.doi.org/10.1145/2699343.2699352
9. Andrés Lucero, Jussi Holopainen, and Tero Jokela. 2012. MobiComics: Collaborative Use of Mobile Phones and Large Displays for Public Expression. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 383–392. DOI:http://dx.doi.org/10.1145/2371574.2371634
10. AM Lund. 2001. Measuring Usability with the USE Questionnaire. STC Usability SIG Newsletter. (2001).
11. Nicolai Marquardt. 2011. Proxemic Interactions in Ubiquitous Computing Ecologies. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*. ACM, New York, NY, USA, 1033–1036. DOI: http://dx.doi.org/10.1145/1979742.1979691
12. Nicolai Marquardt, Ken Hinckley, and Saul Greenberg. 2012. Cross-device Interaction via Micro-mobility and F-formations. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 13–22. DOI: http://dx.doi.org/10.1145/2380116.2380121
13. Brad A. Myers. 2001. Using Handhelds and PCs Together. *Commun. ACM* 44, 11 (Nov. 2001), 34–41. DOI:http://dx.doi.org/10.1145/384150.384159
14. Heidi Selmer Nielsen, Marius Pallisgaard Olsen, Mikael B. Skov, and Jesper Kjeldskov. 2014. JuxtaPinch: Exploring Multi-device Interaction in Collocated Photo Sharing. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 183–192. DOI: http://dx.doi.org/10.1145/2628363.2628369
15. Jun Rekimoto. 1997. Pick-and-drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST '97)*. ACM, New York, NY, USA, 31–39. DOI: http://dx.doi.org/10.1145/263407.263505
16. Jun Rekimoto. 1998. Multiple-Computer User Interfaces: A Cooperative Environment Consisting of Multiple Digital Devices. In *Proceedings of the First International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture (CoBuild '98)*. Springer-Verlag, London, UK, UK, 33–40. http://dl.acm.org/citation.cfm?id=645968.674728
17. Jürgen Scheible, Timo Ojala, and Paul Coulton. 2008. MobiToss: A Novel Gesture Based Interface for Creating and Sharing Mobile Multimedia Art on Large Public Displays. In *Proceedings of the 16th ACM International Conference on Multimedia (MM '08)*. ACM, New York, NY, USA, 957–960. DOI: http://dx.doi.org/10.1145/1459359.1459532
18. Dominik Schmidt, Julian Seifert, Enrico Rukzio, and Hans Gellersen. 2012. A Cross-device Interaction Style for Mobiles and Surfaces. In *Proceedings of the Designing Interactive Systems Conference (DIS '12)*. ACM, New York, NY, USA, 318–327. DOI: http://dx.doi.org/10.1145/2317956.2318005
19. Julian Seifert, Adalberto Simeone, Dominik Schmidt, Paul Holleis, Christian Reinartz, Matthias Wagner, Hans Gellersen, and Enrico Rukzio. 2012. MobiSurf: Improving Co-located Collaboration Through Integrating Mobile Devices and Interactive Surfaces. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces (ITS '12)*. ACM, New York, NY, USA, 51–60. DOI: http://dx.doi.org/10.1145/2396636.2396644

20. Mikael B. Skov, Jesper Kjeldskov, Jeni Paay, Heidi P. Jensen, and Marius P. Olsen. 2015. Investigating Cross-Device Interaction Techniques: A Case of Card Playing on Handhelds and Tablets. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction (OzCHI '15)*. ACM, New York, NY, USA, 446–454. DOI: http://dx.doi.org/10.1145/2838739.2838763

21. Robert Walter, Gilles Bailly, Nina Valkanova, and Jörg Müller. 2014. Cuenesics: Using Mid-air Gestures to Select Items on Interactive Public Displays. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 299–308. DOI: http://dx.doi.org/10.1145/2628363.2628368