

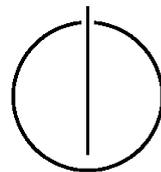
FAKULTÄT FÜR INFORMATIK

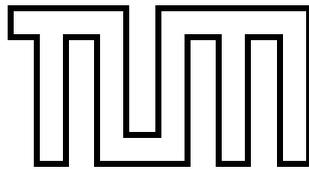
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatik

**User Distraction through interaction with a  
wall-sized display**

Alexander Plopski





FAKULTÄT FÜR INFORMATIK

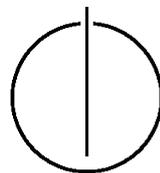
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatik

User Distraction through interaction with a wall-sized  
display

Ablenkung des Benutzers durch Interaktion mit einem  
Wand-großem Display

Author: Alexander Plopski  
Supervisor: Prof. Dr. Gudrun Klinker  
Advisor: M.Sc.-Inf Frieder Pankratz  
Dr. Kiyoshi Kiyokawa  
Date: May 14, 2012



I assure the single handed composition of this master's thesis only supported by declared resources.

München, den 14. Mai 2012

Alexander Plopski

---

## Acknowledgments

I would like to thank Prof. Haruo Takemura and Dr. Kiyoshi Kiyokawa for their support before my arrival in Japan and during the whole course of this thesis. I am grateful as well for the opportunity to conduct my research at their lab at the Osaka University. I am indebted to M.Sc.-Inf. Frieder Pankratz for his moral support during my preparation and project development. He always supported my progress and I would not have achieved these results without his suggestions and criticism. I am as well grateful to M.Sc.-Inf. Nicholas Katkakis for criticizing my ideas and goals, constantly pushing me towards higher aims and goals while pointing out weaknesses in my thought process. Last but not least I would like to thank Dr. Christian Nitschke for being my anchor in Japan, motivating me whenever I felt low and never letting me lose my spirit.

---

## Abstract

Wall-sized displays are conquering their place in the everyday applications in offices and command and control centers. They benefit from the capability to display more information and promote collaboration and exploration better than common single and double monitor setups. The development of suitable interaction techniques is lagging behind, because standards, e.g. mouse and keyboard for a desktop PC, are yet to be defined. With advancement in spatial recognition and reconstruction the research is shifting from simple pointing to spatial interaction. Researchers try to determine the best suitable interaction techniques for different tasks on such a screen. A technique must not only perform fast, but allow the user to perform tasks efficient by not requiring the user to focus completely and solely on the interaction. In this thesis we observe the applicability of different techniques for the drag and scale interaction on a huge display under the aspects of speed and attention requirements. We designed a test which allowed us to observe such application. Through a test series with ten participants we determined which techniques are best suited for this task. We discuss the emerging errors under the aspect of the repetition and missing and offer an explanation of possible causes of these.

---

## Zusammenfassung

Wand-große Bildschirme erobern ihren Platz an den Arbeitsplätzen und Kontrollzentren in aller Welt. Im Gegensatz zu normalen ein und zwei Monitor Systemen ermöglichen diese eine deutlich größere Menge an Daten gleichzeitig darzustellen und fördern die Zusammenarbeit und Erkundung des Datensatzes. Die Entwicklung erfolgreicher Interaktionstechniken hinkt dieser Entwicklung hinterher, insofern dass noch kein Standard, wie die Maus und Tastatur an gewöhnlichen Systemen, festgelegt worden ist. Durch Fortschritt der dreidimensionalen Erkennungs- und Rekonstruktionssysteme konzentriert sich die Forschung zusehendst vom Zeigen auf dreidimensionale Interaktion mit dem Monitor. Forscher versuchen die am besten geeigneten Techniken für die einzelnen Aufgaben zu entwickeln. Diese müssen nicht nur schnell ausführbar sein, sondern es dem Anwender auch erlauben die Aufgaben effizient auszuführen, indem sie nicht die gesamte Aufmerksamkeit auf sich ziehen. In dieser Arbeit untersuchen wir die Anwendbarkeit von fünf verschiedenen Techniken zum Skalieren und Verschieben von Objekten. Dabei achten wir nicht nur auf die Geschwindigkeit sondern auch die Ablenkung des Benutzers. Wir entwickelten einen Test der es uns erlaubte dies zu beobachten. Eine Testserie mit zehn Probanden erlaubte es uns die am besten geeigneten Techniken herauszufinden. Desweiteren werden die auftretenden Fehler unter dem Gesichtspunkt der Wiederholungs- und Versäumnungsfehler sowie die möglichen Ursachen diskutiert.

# Contents

Acknowledgements	iv
Abstract	v
<b>I. Introduction and Theory</b>	<b>1</b>
<b>1. Introduction</b>	<b>2</b>
1.1. Chapter Outline . . . . .	3
<b>2. Related Work</b>	<b>4</b>
2.1. Large Displays . . . . .	4
2.2. Spatial Input . . . . .	5
2.3. Attention Distribution . . . . .	6
<b>3. Experiment Design</b>	<b>7</b>
3.1. Tracking Volume and Interaction tracking . . . . .	7
3.2. User Interaction Techniques . . . . .	8
3.2.1. 2D Interaction . . . . .	8
3.2.2. Device Interaction . . . . .	9
3.2.3. Posture Interaction . . . . .	9
3.2.4. Two-Hand Interaction . . . . .	10
3.2.5. Head Interaction . . . . .	10
3.3. Mathematical Calculations . . . . .	11
3.3.1. Pointing Location . . . . .	11
3.3.2. Posture Detection . . . . .	12
3.3.3. Homography . . . . .	12
3.3.4. Singular Value Decomposition . . . . .	14
3.4. User Interface . . . . .	15
<b>II. Experiment Setup and Evaluation</b>	<b>18</b>
<b>4. Implementation</b>	<b>19</b>
4.1. Location and desired interaction processing . . . . .	19
4.2. User Interface . . . . .	20
<b>5. Experiments</b>	<b>21</b>
5.1. Preparation . . . . .	21

*Contents*

---

5.2. Evaluation . . . . .	22
5.2.1. Execution Time . . . . .	23
5.2.2. Game Performance . . . . .	26
5.3. User Questionary . . . . .	30
<b>6. Summary and Outlook</b>	<b>34</b>
<b>Appendix</b>	<b>36</b>
<b>Bibliography</b>	<b>37</b>

## **Part I.**

# **Introduction and Theory**

# 1. Introduction

When computers were first developed the only communication method were punch cards and users had to learn to interpret the output as well as the code to input the calculations. With the introduction of displays and the development of personal computers this interaction turned into typing the tasks and reading the system response from the display. Further development led to new input devices which reduced the stress of input and output. These new devices, such as mouse or the recent development of touchdisplays, led to new interaction techniques (such as click and drag) and user interfaces. This development went along with the development of new display devices. These include previously mentioned touchdisplays, high quality - low cost projector displays and high quality - high cost multidisplay wall-sized displays. All of these have their justification and applications at the beginning of the 22nd century. Current research in Computer Vision and more easily accessible high-quality reconstruction[19] and segmentation techniques[14] allow further development of interaction techniques and ubiquitous computing. The combination of such tracking and wall-sized displays allows for better cooperative surroundings and the use of natural motion to interact with the display. A multitude of interaction techniques based on Computer Vision recognition was developed (e.g. [38]). Other techniques focus on transferring interaction with well-known objects onto the screen (e.g. [36]). It has yet to be shown which interaction technique is favorable in general or whether or not multiple interaction techniques will be feasible, but some may perform better for specific tasks. This development aims to remove the consciousness of computers from our workplace.

Given the ease of access and the constantly reducing cost of pixels paired with the benefits of wall-sized monitors we believe that these will become more and more used in our everyday life. An experiment of an outdoor application was conducted by Yahoo in San Francisco. In movies and futuristic advertisements is the use of this technology everyday present. But to be viable the interaction with the display must be fast, easy to learn, allow interaction over a long period of time and we must be able to focus on other things and not only on the interaction itself. On a small display for example we can focus on reading a text but still comprehend and interact with other programs through fast glimpses due to short distance. On a bigger display on the other hand we most likely will have to turn our head or walk a wide distance with our eyes to observe the results of our interaction.

In this thesis we observe the attention requirements of multiple techniques for interaction with 2D objects on a wall-sized display. This interaction will involve dragging and scaling of these objects. The goal of this research is to provide developers with a comparison of different currently developed and used techniques based on their distraction from a time consuming task. This research was conducted at Prof. Takemura's laboratory at the Osaka University and was part of the exchange between the TU Munich and Osaka University.

## 1.1. Chapter Outline

Our research contributes to multiple areas of Human-Computer-Interaction. In chapter 2 we will outline current research in these areas, namely large displays, spatial input and attention distribution. We will focus on the theoretical design of our research in chapter 3. This chapter will explain our spatial tracking setup, the mathematical calculations necessary to compute the interaction location and the interaction task. Finally it will include the design of the user interface. The implementation will be described in chapter 4.1. We will explain the execution of our experiment in section 5.1 and evaluate the results in section 5.2. In the evaluation we will focus on the execution time, the error rates and the user questionnaire. Finally we will summarize our results and give an outlook on further research in chapter 6.

## 2. Related Work

In this thesis we focus on a subject which conjugates with multiple research areas of Human Computer Interaction (HCI), primarily large displays, spatial input and attention distribution.

### 2.1. Large Displays

Over the past years multiple approaches to create large-scale displays through combination and alignment of smaller displays have been introduced and observed. This research was strongly supported by the industry as well as the military. Multiple qualitative and quantitative experiments were conducted and have shown that the productivity and performance changes based on change in the visual effects. Ni et. al. [23] performed an in-depth studies of current development of large-scale display setups and outlined three basic devices which could be used to create such setups. Most people currently use multiple monitor setups, mostly double-monitor setups, at home and at work. This basic setup of two monitors can be extended to create a wall or a room of screens. Similarly LCD panels can be used to create such a wall. These are easy to setup designs but they suffer from bezel issues and are more expensive then the other approach. That is an alignment of projectors which benefit from having no bezel issues to cover and the disproportionality of device and screen size. A small projector can create a big screen. On the other hand projector alignment requires additional calibration algorithms to align the illumination and projection location of the devices, for example such as described in [25, 18]. Such setups allow furthermore not just the use of a standart wall-type displays but allow creation of immersive rooms, such as the CAVE or rounded displays. This requires a lot of computational power especially if the interaction is coupled with spatial input techniques. Therefore the usual approach is to use computer clusters to manage the computational workload.

With the continuous development of the aforementioned technologies the number of pixels achievable in such a wall is steadily increasing and the biggest wall up to today contains more than 200 million pixels. In the past studies on applicability of such monitors were conducted and Baudish et. al [8] found that they were better for information extraction from huge data sets compared to overview and zooming on a common display. In another study [10] Bi and Balakrishnan compared wall-sized screens to single and dual-monitor setups. They found out that the users preferred the huge display over their common workspace. They furthermore observed that users divided the screen into a central area, which acted as the area of focus and the surrounding peripheral region. Overall the users were more inclined to drag and rescale windows on a wall-sized screen while they would rather minimize and maximize windows on smaller configurations. Of course research does not focus only on comparisons with other display devices. While at first a

lot of focus was devoted to the task of pointing to the screen recently the focus shifted towards improvement of certain interactions and general usability of large-scale displays. In [7] Badusch et.al. observed the effects of a dense mouse pointer rather than a normal one. They found out that the dense mouse enabled the users to better track their current location thus reducing the time required to travel the screen. In [6] Badusch et.al. introduced another method for drag-and-drop interaction which they expected to perform better on large screens.

The ability of these displays to promote and enhance cooperative work and the ability to display and process huge amounts of information at a single screen has been recognized by military and industry. Now they are used in military, airspace or communication Command and Control Centers all over the world. Two examples of such application are the Interactive DataWall in the US Air Force Research Facility and Command and Control Center of AT&T.

### 2.2. Spatial Input

The bigger size of a wall-sized display inclines that standard interaction techniques, such as mouse and keyboard, are hard to transfer and sometimes not applicable at all. Increased use of touch displays for notebooks, personal computers, touch tables and for the latest generation of mobile phones, commonly known as smartphones, got the users accustomed to touching objects for selection and use of natural movements for interaction. This approach is rarely applicable when used with a wall-sized display. The disadvantage is evident. Due to the large size of the display the user would have to traverse a great distance to move an object on the screen and would be unable to observe the scene as a whole. Thus in this study we focused on techniques which enable interaction with a screen from a distance. Furthermore to enhance collaboration and reduce restrictions as much as possible we focused on techniques which could be used from different locations. Over the past years multiple techniques for the desired interaction have been developed and evaluated.

Jota et. al. [17] observed three different interaction metaphors with a screen from a distance. Whilst they expected that the different metaphors (grab, point, mouse) would perform best at a distance best suited for the respective metaphor they surprisingly found out that the pointing metaphor performed best in all tests.

As was stated before, classic interaction methods with big screens are often hardly, if at all, applicable to wall-sized screen interaction. Current research ranges from using all limbs for interaction, for example foot control by Scott et.al. [29] or Augsten et.al. [5], detecting muscle contractions to trigger interaction, Saponas et.al. [27], multidimensional input devices, e.g. soap [9], all the way to use of light sources to enable interaction through shadows and touch, Shoemaker et.al. [30]. Other techniques use a device to emulate the screen and let the user interact with it. Examples of that are the HIPerPaper by Weibel et.al. [36] or the table-screen interaction by Malik et.al. [20].

Currently most applications use raycasting to determine the location of interaction. The source of these rays may be the eyes, hands or pointing devices. Eyetriggered interaction has been studied for years, for example by Zhai in [39], Turner in [32] and Stellmach in [31]. But it suffers from being not very precise and requiring additional interaction devices. Attempts to apply purely gaze based interaction require the user to either use a delay as

an interaction trigger or blink. Both approaches are hardly usable in an actual application.

### **2.3. Attention Distribution**

It is general consensus that users have to devote their focus to interaction and lose focus of whatever task they were working on at that moment. This loss of attention can lead to two types of mistakes, a repetition mistake, where the user performs a recently finished task once again, and a miss mistake. The second type of possible mistakes can lead to grave results. In the past several accidents occurred because workers were disturbed during a safety check and skipped a test. In most cases researchers do not evaluate the focus required for the interaction. We suspect that it is assumed that the time required for an interaction is proportional to the focus required to perform it, but that must not necessarily be the case. A technique which performs slower may allow the user to multitask while a fast technique requires the user to fully focus to perform it.

Rogers et.al. [26] and Valk [33] have shown that a mismatched input device will reduce the performance. Other researchers [37, 28] have suggested that this mismatch does increase the attention requirements. McLaughlin et.al. [21] observed that the amount of attention did increase whenever a device was used for the input and it increased even further if the input device was mismatched to the task. Other studies [12, 35] have suggested that indirect input methods will require more attention than direct input. Indirect input means in this context that the user uses a device, for example a mouse. This method involves cognitive conversion of a small traversed distance with the device to a large traversal on the screen. Direct input does not require such conversion and could be touch screens, light pens or speech. These studies have shown that the amount of attention required for an interaction does scale with the age of the participants.

In another study Bragdon et. al. [11] evaluated the users ability to interact with a mobile device while performing different real-world tasks. Their goal was to determine the best way to interact with a mobile device. They found out that the performance speed of some interaction techniques did not vary significantly, even when a user had to perform an attention-saturating task. On the other hand the performance of the attention-saturating task dropped when the user performed concurrent tasks.

### 3. Experiment Design

The setup of the spatial interaction experiment consisted of three parts as shown in figure 3.1(a). The first task was to detect and determine the pointing direction of the interaction intentions. In a second step the captured data was analyzed to match a predetermined interaction method. In this step the intended interaction (point, drag, move) and the interaction location was determined and forwarded to the final part. In the final step the different tasks given to the user and the interaction with those tasks were displayed on a screen projected on the wall. The 2D-based system skipped the spatial capturing part and determined the interaction location and desired interaction directly from the users touch of the tablet. This framework is shown in figure 3.1(b). In our framework each part was handled by a designated program which communicated with each other through UDP communication. They are explained in detail in chapter 4.1.

#### 3.1. Tracking Volume and Interaction tracking

Over the past decades great advances were made in stereo model reconstruction, for example [19, 13] and current interaction systems begin to rely on stereo recognition to detect screen interaction, such as [32]. As it was not the goal of this project to create such a system we relied on the OptiTrack System to recognize and detect retroreflective targets to track the users. The OptiTrack System is an outside-in tracking system which consists of multiple synchronized cameras capable of emitting light in near infrared spectrum (figure 3.3(a)) and capturing its reflection within the tracking volume. The cameras recognize the reflection of the infrared light as white dots in a binary image. In our tests we first attempted to use 8 cameras placed at the four corners of the room but changed to 6 cameras placed as shown in figure 3.2. Before the tests the setup was calibrated by moving a single reflective marker through the tracked area. After enough data was recorded the system calibrated the location of the cameras in the world and their intrinsic parameters.

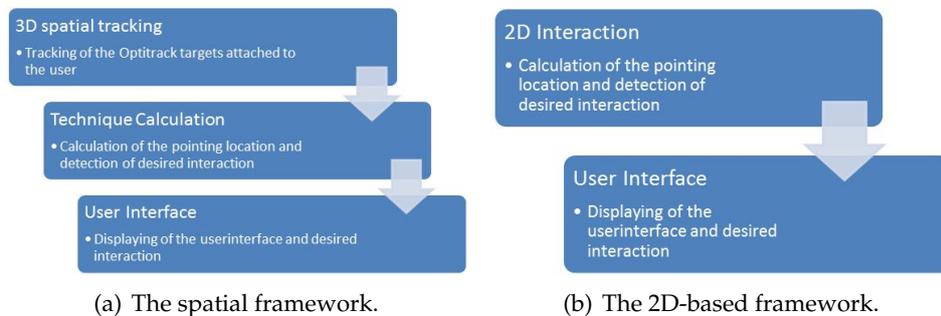


Figure 3.1.: Framework of the experiment.

For the purposes of tracking the pointing direction and postures the users were equipped with trackable targets (figure 3.3(b)), which were constructed from three rigidly fixed reflective markers in a distinguishable formation. The Optitrack system used the detected markers in each camera to reconstruct them in a 3D-world and detect the targets within it. That way the location and orientation of the targets was determined. To reduce the noise caused by minimal movement of the targets (body shivering) and possible tracking errors a Kalman filter was applied to the measurements. The system captured and processed at a speed of 60 fps.

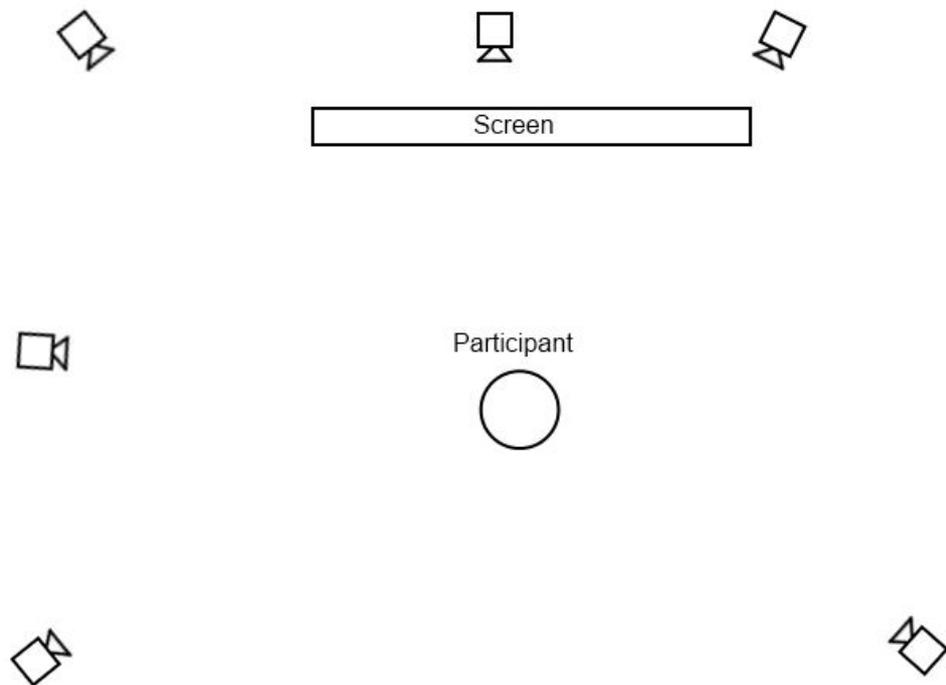


Figure 3.2.: The OptiTrack camera setup.

## 3.2. User Interaction Techniques

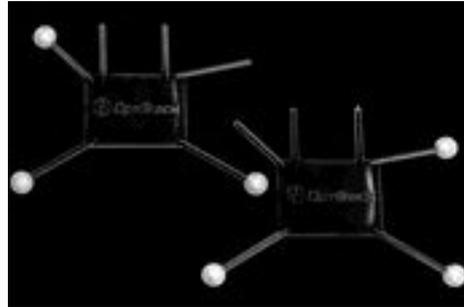
For our experiments we implemented five interaction techniques which were similar to those used in [22]. We had one representative of 2D interaction. The remaining four interaction techniques were based on recognition of the desired location and action in 3D, as shown in figure 3.3.

### 3.2.1. 2D Interaction

This interaction method relied solely on a 2D interface similar to [36]. A Samsung Galaxy Tab was used as the device of choice (figure 3.3(c)). The 2D interface used the TUIOdroid



(a) An OptiTrack camera.



(b) An OptiTrack trackable target.



(c) 2D



(d) Device



(e) Posture



(f) 2Hand



(g) Head

Figure 3.3.: Implemented interaction techniques.

([3]) interface to detect clicking on the tablet-display and sending data to the display interface through the network. The user was presented with a screen as shown in figure 3.4 . By clicking anywhere on the Tablet display the user could interact with the corresponding location of the wall-display. To initiate the drag or scale command the user had to additionally push on the blue area for scale or red area for drag.

### 3.2.2. Device Interaction

For this interaction technique the user was requested to point at the screen with the right hand and click on a device to trigger interaction (figure 3.3(d)). For our experiments the users were using their thumb to push the left and right buttons of a mouse. A smartphone or any other device could have been used as a replacement.

### 3.2.3. Posture Interaction

Posture interaction and gesture interaction are well used in current applications. For posture interaction the user has to assume a predefined pose to trigger the interaction. On the other hand the user user has to perform a predefined motion for gesture interaction. For example point to the screen with the palm of the hand and push the hand towards the screen. The posture interaction is shown in figure 3.3(e). The right hand was once again used for pointing at the screen. The left hand placed at around the height of one's shoulder and turned towardly the screen with the palm would trigger scaling. If the backhand faced the screen then dragging would be triggered.

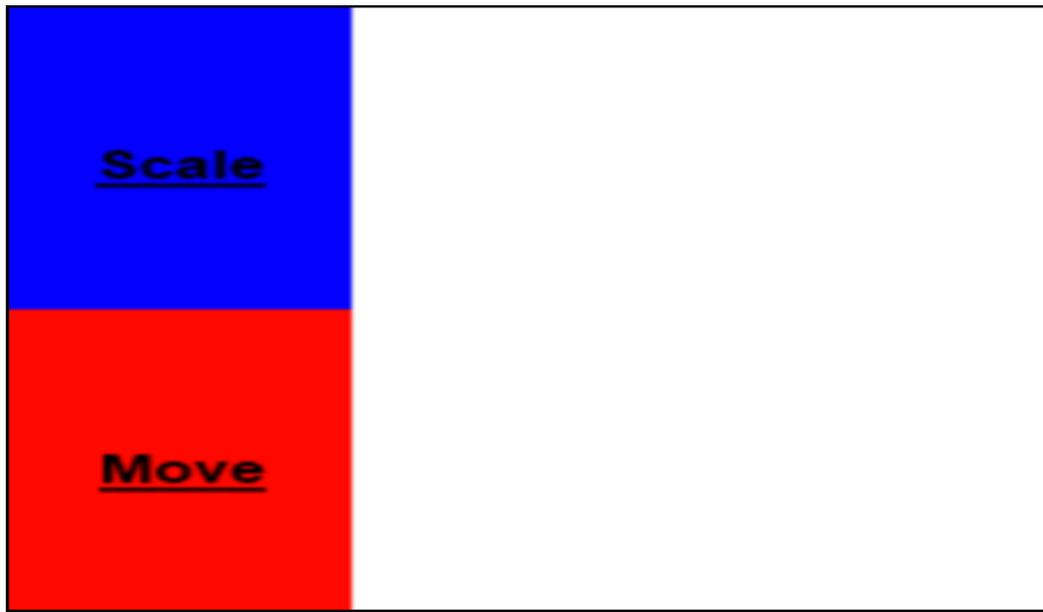


Figure 3.4.: The Interaction screen as displayed on the Samsung Galaxy Tab.

#### 3.2.4. Two-Hand Interaction

This interaction method imitated natural interaction similar to those of tables, tablets and mobile phones. The user was pointing towards the screen with both hands. The pointing locations were shown as a green and a black circle. To trigger the interaction the user had to press a button with the right foot. By pointing at the interaction object with only one hand while pushing the interaction button the user initiated the drag command. By pointing at the object with both hands the scale command would be selected. The scaling changed according to the absolute distance changes of the markers. For the other interaction techniques the scaling changed according the distance to the center of the object.

#### 3.2.5. Head Interaction

Rising interest in gaze interaction suggested to observe this interaction technique as well, but gaze based interaction focuses the user's attention only on one task\location at the same time, thus no multitasking is possible. As an alternative we suggest head-based interaction. The user can point to the screen with the head, and observe other areas with his eyes, before returning the attention to the pointed location. The interaction can be triggered for example through secondary devices, gestures, postures or speech. In our case to trigger the drag command the user could point to the screen either with the front or the back of the palm of the left hand. Analogous the right hand triggered the scaling.

### 3.3. Mathematical Calculations

To simplify the calculations we set up the world in such a way that the screen was orthogonal to the Z-Axis. Before the users interactes with the screen the location of the corers was calculated by pointing at them from a three meter distance. The location and orientation data received from the optitrack system was averaged. the location was given as a point

$$P_r = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}$$

where  $p_i$  represents the translation along the axis i.

The orientation data was given as a quaternenion.

Given a quaternenion  $q_i$  with

$$q_i = q_x, q_y, q_z, q_w,$$

$(q_x, q_y, q_z)$  represents the axis around which the object will be rotated and  $q_w$  represents the angle  $\alpha$ , with  $\alpha = 2 * \cos^{-1}(q_w)$ .

This angle reresentation can be transformed into a matrix form M where

$$M = \begin{pmatrix} q_x^2 - q_y^2 - q_z^2 + q_w^2 & 2 * (q_x * q_y - q_z * q_w) & 2 * (q_x * q_z + q_y * q_w) \\ 2 * (q_x * q_y + q_z * q_w) & q_y^2 - q_x^2 - q_z^2 + q_w^2 & 2 * (q_y * q_z - q_x * q_w) \\ 2 * (q_y * q_z + q_x * q_w) & 2 * (q_x * q_z - q_y * q_w) & q_z^2 - q_x^2 - q_y^2 + q_w^2 \end{pmatrix}.$$

#### 3.3.1. Pointing Location

The pointing rays were casted along the Z-axis. Given the normalization factor

$$N = \sqrt{M_{13}^2 + M_{23}^2 + M_{33}^2}$$

where  $M_{ij}$  represents the entry of the matrix M in row i and column j this ray can be formulated as

$$R_z = \begin{pmatrix} \frac{M_{13}}{N} \\ \frac{M_{23}}{N} \\ \frac{M_{33}}{N} \end{pmatrix}.$$

The screen plane was represented by the normal vector

$$V = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.1}$$

and a point

$$P_w = \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix}$$

which was located in the same plane as the wall.

The interaction point  $P_I$  of the ray  $R_z$  with the wall can be calculated through equation 3.2.

$$\begin{aligned} t &= \frac{V(P_w - P_r)}{VR_z} \\ P_I &= P_r + tR_z \end{aligned} \quad (3.2)$$

After the corners of the screen were calculated the homography  $H$ , which transformed their location to match the pixel coordinates of the display, was determined as explained in chapter 3.3.3. During the tests the point  $P_I$ , the intersection of the pointing ray and the screen, was multiplied by  $H$ , as shown in equation 3.3.  $P_{2D}$  represents the pixel coordinates of the pointing location.

$$P_{2D} = HP_I \quad (3.3)$$

### 3.3.2. Posture Detection

The desired postures were detected whenever the user was holding the hand above a pre-defined threshold. In this case the user had to point to the screen with the palm, thus the observed ray was along the Y-axis of the tracking target.

Given the normalization factor

$$N = \sqrt{M_{12}^2 + M_{22}^2 + M_{33}^2}$$

the observed ray  $R_y$  was described as

$$R_y = \begin{pmatrix} \frac{M_{12}}{N} \\ \frac{M_{22}}{N} \\ \frac{M_{32}}{N} \end{pmatrix}.$$

To determine whether the user triggered an interaction or not the angle  $\alpha$  between  $V$  (equation 3.1), the normal of the screen, and  $R_y$  was calculated by equation 3.4.

$$\alpha = \arccos(R_y * V). \quad (3.4)$$

After transforming  $\alpha$  into degree space we observed whether it fits within the defined thresholds. The desired interaction (drag, scale, point) was sent to the interface if the user was pointing within the screen.

### 3.3.3. Homography

The following calculations are explained in detail in [16].

#### Direct Linear Transformation

The Direct Linear Transformation(DLT) algorithm allows to calculate the homography of two given point clouds  $P_1$  and  $P_2$ , which contain at least four corresponding point  $p_{1,i} \leftrightarrow$

---

### 3. Experiment Design

---

$p_{2,i}$ . The points are given in homogeneous coordinates, that means that a point (x,y) is displayed as

$$p = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

The relationship of  $p_{1,i} \leftrightarrow p_{2,i}$  can be expressed by equation 3.3. But  $H * p_{1,i}$  must not necessarily be equal to  $p_{2,i}$ , the result may be scaled. This means that both vectors,  $p_{2,i}$  and  $H p_{1,i}$  point in the same direction. To better denote this relation the equation can be written as

$$p_{2,i} \times H p_{1,i} = 0. \quad (3.5)$$

The homography matrix is a  $3 \times 3$  matrix. Given that  $h^{jT}$  denotes the j-th row of H equation 3.3 can be written as

$$H P_I = \begin{pmatrix} h^{1T} P_I \\ h^{2T} P_I \\ h^{3T} P_I \end{pmatrix}. \quad (3.6)$$

Through equation 3.6, equation 3.5 can be transformed into

$$p_{2,i} \times H p_{1,i} = \begin{pmatrix} y_{2,i} h^{3T} p_{1,i} - h^{2T} p_{1,i} \\ h^{1T} p_{1,i} - x_{2,i} h^{3T} p_{1,i} \\ y_{1,i} h^{2T} p_{1,i} - y_{2,i} h^{1T} p_{1,i} \end{pmatrix} = 0. \quad (3.7)$$

Since  $h^{jT} p_{1,i} = p_{1,i}^T h^j$  one can extract  $h^j$  to create a single matrix multiplication 3.9, where

$$0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.8)$$

and  $\|h\| = 1$ .

$$A_i h = \begin{pmatrix} 0^T & -p_{1,i}^T & y_{2,i} p_{1,i}^T \\ p_{1,i}^T & 0^T & -x_{2,i} p_{1,i}^T \\ -y_{2,i} p_{1,i}^T & x_{2,i} p_{1,i}^T & 0^T \end{pmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \quad (3.9)$$

The matrix A is a  $3 \times 9$  matrix and h is a 9-vector which can be transformed into the matrix H. The third row of the matrix  $A_i$  can be omitted because it can be obtained up to a scale from a sum of  $x_{2,i}$  times the first row and  $y_{2,i}$  times the second row of  $A_i$ . As the matrix H has 8 degrees of freedom (it is up to a scale of  $H_{3,3}$ ) at least 8 equations are needed in the matrix A. This is achieved by combining the matrices  $A_i, i = 1, \dots, n$ , into a single  $2n \times 9$  ( $3n \times 9$ , if the third row is kept) matrix A. Thus at least four points are necessary. The resulting equation can be solved through a SVD-Decomposition (chapter 3.3.4) of the matrix A, with

$$UDV^T = A. \quad (3.10)$$

U and D are orthogonal and V a diagonal matrix. The last column of the matrix V equals the vector h. After transforming h into H and rescaling it, so that  $H_{3,3} = 1$ , H could be used in equation 3.3. But this calculation of H suffers from the quadratic pixel coordinates in

the matrix  $A$ , thus the values have different weights onto the calculation. To solve this, an additional step is necessary.

### Normalized Direct Linear Transformation

The normalized DLT algorithm extends the previously explained DLT algorithm by defining

$$\tilde{H} = T'^{-1}HT. \quad (3.11)$$

$T'$  is a transformation of the points  $P_2$  and  $T$  a transformation of the points  $P_1$  where

$$P'_2 = T'P_2 \quad (3.12)$$

$$P'_1 = TP_1, \quad (3.13)$$

thus

$$P_2 = T'^{-1}HTP_1. \quad (3.14)$$

The matrices  $T'$  and  $T$  will remove random scaling and improve the final results. In the following the calculation of matrix  $T$  will be explained, the matrix  $T'$  is calculated analogous.

The matrix  $T$  is a  $3 \times 3$  matrix which can be decomposed into a scaling component  $S$  and translation component  $t$ , as shown in equation 3.15.

$$T = \begin{pmatrix} S & t \\ 0 & 0 & 1 \end{pmatrix} \quad (3.15)$$

$t$  is the negative centroid of all points within  $P_1$ . After translating all points by  $t$  the centroid is  $(0,0,1)$  thus the center of the X-Y coordinate system. To account for scaling the x and y coordinates of the translated points are scaled uniformly by the factor  $s$  so that their average distance from the center is  $\sqrt{2}$ , therefor the average point would be  $(1,1,1)^T$ . This results in the matrix

$$T = \begin{pmatrix} s & 0 & t_x \\ 0 & s & t_y \\ 0 & 0 & 1 \end{pmatrix} \quad (3.16)$$

After the matrices  $T$  and  $T'$  have been determined the basic DLT algorithm is used to calculate the matrix  $\tilde{H}$ . Finally the matrix  $H$  can be obtained through the equation 3.11.

### 3.3.4. Singular Value Decomposition

The Singular Value Decomposition (SVD) is used primarily to derive the solution of over-determined systems of equations. The SVD decomposes a given matrix  $A$  into three matrices  $U$ ,  $D$  and  $V$  so that  $UDV^T = A$  holds. The matrices  $U$  and  $V$  are orthogonal and the matrix  $D$  is a diagonal matrix with non-negative values. This decomposition can be achieved so that the entries of  $D$  are in descending order. This version of the decomposition is mainly used.

Given a  $m \times n$  matrix  $A$ , where  $m \geq n$ , then  $U$  will be an  $m \times n$  matrix with orthogonal columns,  $D$  is an  $n \times n$  diagonal matrix and  $V$  is an  $n \times n$  orthogonal matrix. The matrix  $U$  has the following properties

- $U^T U = I_{n \times n}$ ,
- $\|Ux\| = \|x\|$ ,
- $U U^T \neq I$ , unless  $m = n$ .

The matrix  $D$  contains the singular values of the matrix  $A$ . These are the square roots of the eigenvalues of  $A^T A$  while the columns of the matrix  $V$  are the eigenvectors of  $A^T A$ . From the definition  $A = U D V^T$  and the fact that  $V$  is orthogonal and therefore  $V^T = V^{-1}$  follows that  $A^T A = V D U^T U D V^T = V D^2 V^T = V D^2 V^{-1}$ . This is the definition of the eigenvalues. Therefore  $D^2$  contains the eigenvalues and the columns of  $V$  are the eigenvectors.

The exact calculations required to compute the SVD are explained in detail in [15] and an implementation can be found in [24]. The code for the implementation is available at [1].

#### 3.4. User Interface

The user-interface was designed in Java and is shown in figure 3.6. The screen was divided into a game and a task area. The game area is displayed on the left side of the screen. The users were asked to participate in a game resembling GuitarHero and devote all their attention to it. The tiles could spawn at a uniform or a random distance. Given random spawning of tiles the probability to spawn a new tile was constantly at 3% for each screen update cycle. The distance ranged between 1 and 31 pixels (the system would definitely spawn a new tile at 31 pixels distance). The probabilities to spawn a tile for each distance are shown in figure 3.5. Thus the expected distance between two tiles is 20,367 pixels. We had three levels of delays between each update cycle (20ms, 15ms and 10ms), thus three different speeds for the tiles (slow, intermediate and fast). By pressing the "Ctrl"-button with the left foot the user could remove a tile when it reached the red line.

On the right hand side the user was asked to complete several tasks. In the task-area the user was given two different types of tasks. In figure 3.7 the possible tasks are shown as they were seen by the user. A drag-task given to the user is displayed in figure 3.7(a). During the drag task it was not required to scale the target. If the users accidentally scaled the target they had to rescale it to a similar size. The second task, as displayed in figure 3.7(b), requested the user to scale the target and fit it in between of the displayed red rectangles. During this task it was not required to drag the target. For the dragging task the starting location and the goal location were randomly assigned within the right  $\frac{2}{3}$ rd of the screen. On the other hand the starting location was fixed to be at a random location around the center of the screen for the scaling task. The scaling task required the user to scale the target to a random size of at most  $\frac{1}{3}$ rd of the screen. To ensure that the user faced a similar amount of scaling and dragging tasks the first task was selected with a probability of 50%. After that the probability shifted by 10% towards the other task and so on.

Distance	Propability	Distance	Propability
0	0	16	0.018997536
1	0.03	17	0.01842761
2	0.0291	18	0.017874781
3	0.028227	19	0.017338538
4	0.02738019	20	0.016818382
5	0.02655878	21	0.01631383
6	0.02576202	22	0.015824415
7	0.02498916	23	0.015349683
8	0.02423949	24	0.014889192
9	0.0235123	25	0.014442517
10	0.02280693	26	0.014009241
11	0.02212272	27	0.013588964
12	0.02145904	28	0.013181295
13	0.02081527	29	0.012785856
14	0.02019081	30	0.01240228
15	0.01958509	31	0.401007069

Figure 3.5.: The propabilities of spawning a tile.

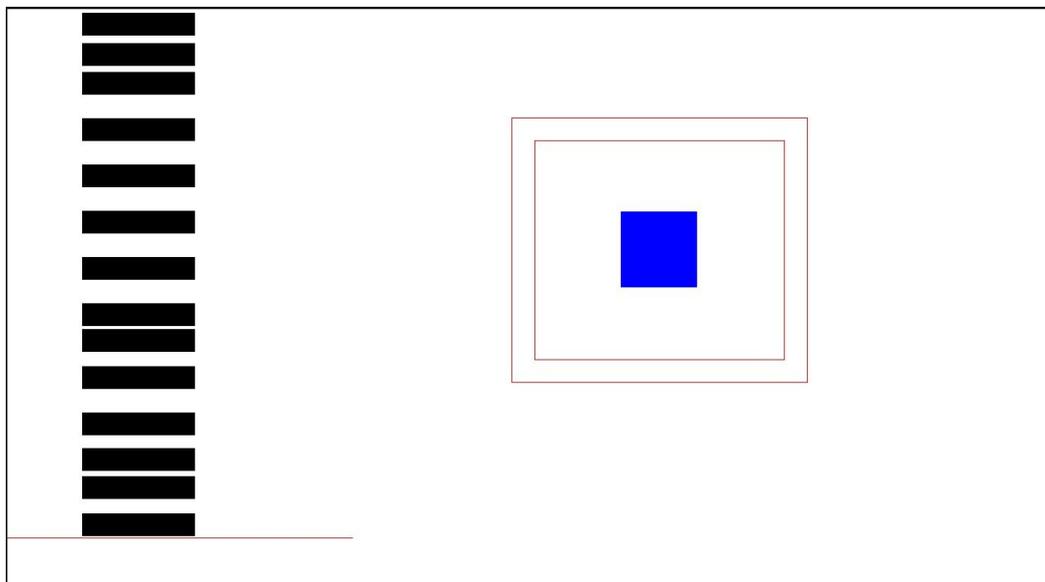


Figure 3.6.: The user interface displayed at the wall-sized screen.

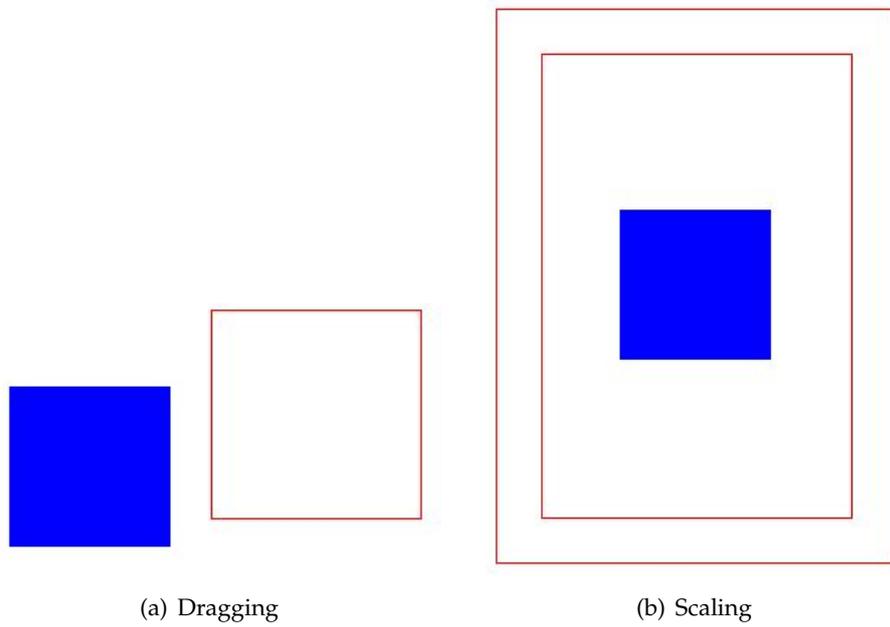


Figure 3.7.: Different tasks given to the user.

## **Part II.**

# **Experiment Setup and Evaluation**

## 4. Implementation

### 4.1. Location and desired interaction processing

The communication of the tracked poses to the calculation part in figure 3.1(a) was done by the Virtual Reality Peripheral Network (VRPN) [4]. After embedding it into our C++ code we registered two targets into our listener. This listener received poses from the OptiTrack software and forwarded them to computation. We used the mathematical library of OpenCV [2] to calculate the mathematical part of our implementation. For each test a different class was designed. During the initialization these classes performed the calculation required for online communication. During the runtime the program listened to new target poses from the network and performed the calculations described in section 3.3.

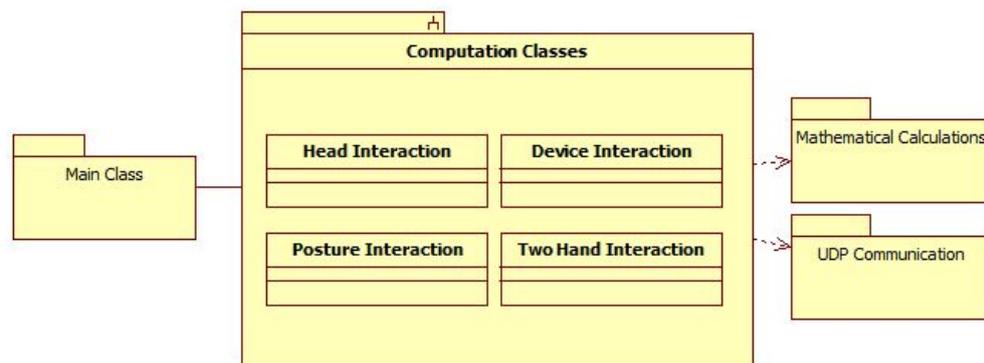


Figure 4.1.: Class diagram of the interaction calculations.

These results were sent to the user interface via UDP. The results were encoded into the following messages

- move  $X_R Y_R$
- scale  $X_R Y_R$
- drag  $X_R Y_R$
- TwoHandInteract  $X_L Y_L X_R Y_R$
- TwoHandMove  $X_L Y_L X_R Y_R$ .

$X_R$  and  $Y_R$  denote the pixel coordinates the right hand is pointing at. Analogous  $X_L$  and  $Y_L$ .

## 4.2. User Interface

The interface consisted of three main threads - the communication, the game and the task area as shown in figure 4.2. At the start of the program the threads were initialized and their dependency on the execution of each other was established. Whenever a thread shut down the program would end.

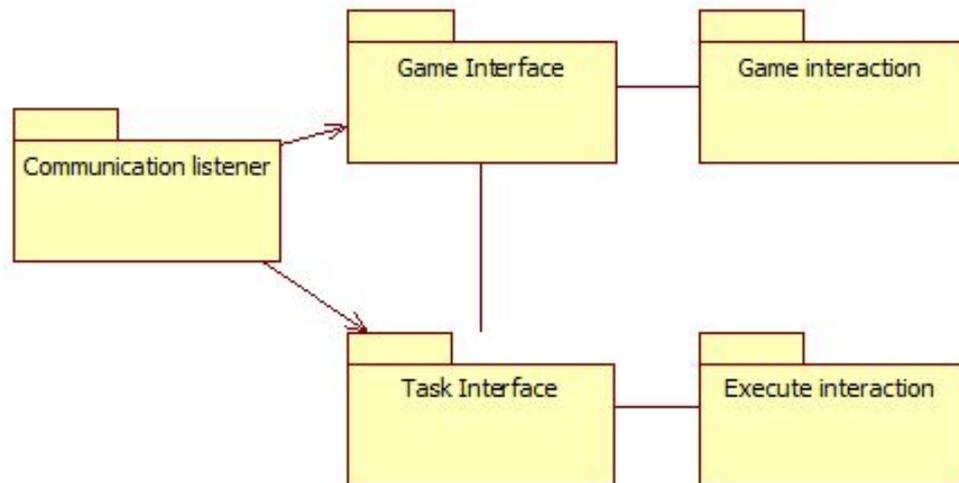


Figure 4.2.: The Java Classes for the user interface

**Communication Thread** The communication listened for messages from the tablet or the spatial processing part. The message was disassembled into the respective parts (interaction and pointing location(s)). Depending on the interaction it simulated a mouse event or called the respective function within the task thread.

**Task Thread** The task thread constantly checked whether a task was completed and created a new randomized task if necessary. It shut down after a predetermined period of time or when the appropriate amount of tasks was completed. The execution interaction thread listened for any mouse events or messages function calls from the communication thread and updated the objects' parameters. Both threads saved the data whenever the user interacted with the screen or a new task was assigned.

**Game Thread** The game thread was constantly checking whether the task thread was still alive and shut down otherwise. During the execution time the thread would check whether a tile must be removed or a tile must be spawned. Finally it drew the game interface. The game interaction thread would catch the pressing of the "Ctrl"-button and execute the removal of a tile. Both threads saved their actions to a file.

## 5. Experiments

### 5.1. Preparation

Ten males between 23 and 32 took part in our tests. The mean age was 25.8 with a standard deviation of 2.638 years. The median age was 25. All users were well used to computer work and computer games. One user was wearing glasses. One user was left-handed, but was using his right hand for pointing tasks in our tests. During everyday work he was using the same setups as other users, e.g. right-handed mouse control. Four participants were westerners, five were Japanese and one was Vietnamese.

During the tests the users were seated approximately three meters away from the screen as shown in figure 5.1. In front of the participant was a 2 x 3 meter screen, which was projected by a Sanyo PDG-DWL2500 projector, and a keyboard was placed at his feet to function as buttons. The keyboard had all buttons but the "Ctrl"-button, to interact with the game, and the Num-block's "Enter"-button, for the two-handed interaction, removed so to not obstruct the interaction.

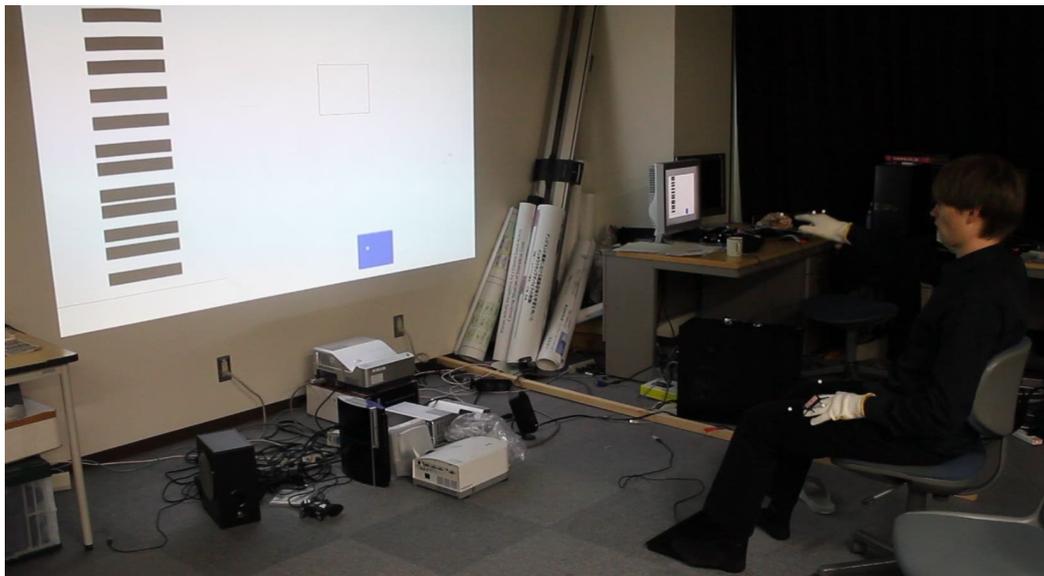


Figure 5.1.: A participant taking the test.

All tests were performed in a single session with a short break of approximately five minutes whenever the user finished the tests for one interaction technique. For each interaction technique the tests were preceded by a five minute practice session. The test execution was as follows:

1. Comparison test - slow

2. 1st interaction technique
3. 2nd interaction technique
4. 3rd interaction technique
5. Comparison test - intermediate
6. 4th interaction technique
7. 5th interaction technique
8. Comparison test - fast.

During a comparison test the user had to participate in the game for two minutes but no drag or scale tasks had to be done. The order of the interaction techniques and tests for each technique was randomized for each participant. After the participants completed all tests they were asked to answer a questionnaire.

To evaluate the amount of attention devoted to the interaction and execution speed we performed four tests for each interaction technique. Three tests required the users to perform six different scale\drag tasks. During these tests the tiles moved at a random distance and the speed was set to fast, intermediate or slow. The final test required the user to perform the tasks for a period of five minutes, which was much longer than the expected execution time of the other tests. During this test the speed of the tiles was set to "slow" and their distance was uniform.

Based on previous research we formulated four hypothesises for our experiment

- H1** Within the free-pointing interaction techniques the amount of degrees of freedom used for the interaction will determine the speed of performance.
- H2** The head-based interaction will perform slower than the other techniques but the error-rate will not be higher than for other techniques.
- H3** The 2-Hand interaction will perform worse than the posture interaction.
- H4** The purely 2D-based interaction technique will put additional cognitive mental load on the user thus scoring lower than the device based interaction.

## 5.2. Evaluation

During the evaluation of the experiment we observed two types of performances, the speed of the execution and the performance of the game. We observed two types of possible errors during the game, the repetition error, when a user attempts to remove a tile whenever no tile reached the goal, and the miss error, when the user missed a tile.

Two groups of observations will often look different but the results will be close enough that no significant statistical difference exists. To calculate the difference of our observations an Analysis Of variance (ANOVA) test was performed. A common type of the ANOVA test is the F-test. The F-value is given as

$$F = \frac{\text{variation among sample means}}{\text{variability within samples}}$$

where our observations are divided into samples by interaction techniques and the performed test. Therefore we had a two-way ANOVA. If the F-value is low the observations are similar, otherwise they are statistically different. The resulting F-value can be transformed into a percentage value which gives the probability that the observations are actually the same. This is the case if the probability is above 0.05 (5%).

### 5.2.1. Execution Time

We observed that the users needed more time to complete a test whenever the speed increased. The users required on average 57.6 seconds to complete the tasks at the slowest speed. The intermediate speed resulted in just a four second longer execution time of 61.9 seconds. On the other hand the difference to the highest speed was as large as ten seconds. The highest speed had an average execution time of 67.8 seconds. The statistical probability that these observations are not relevant due to a random error was 5.41% but the difference between the slowest and the fastest speed had a probability of only 0.19%. We can conclude that the speed did have a significant influence onto the performance time. These results show that the users attempted to focus primarily at the game and perform well in it. Of course we were interested not only in the performance time depending on the game speed but the interaction technique as well. The average time required to complete six tasks with each technique are shown in figure 5.2. Figure 5.3 displays the required times for each technique and interaction speed. Similar results can be observed within the five minute test in figure 5.4.

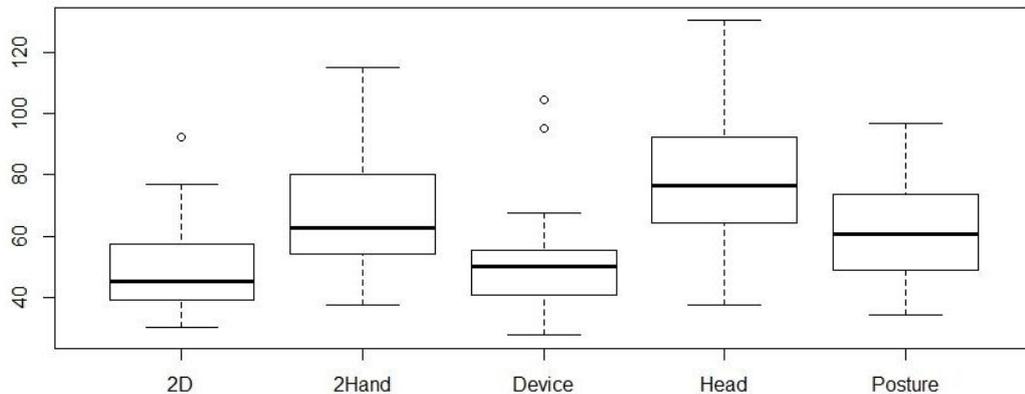
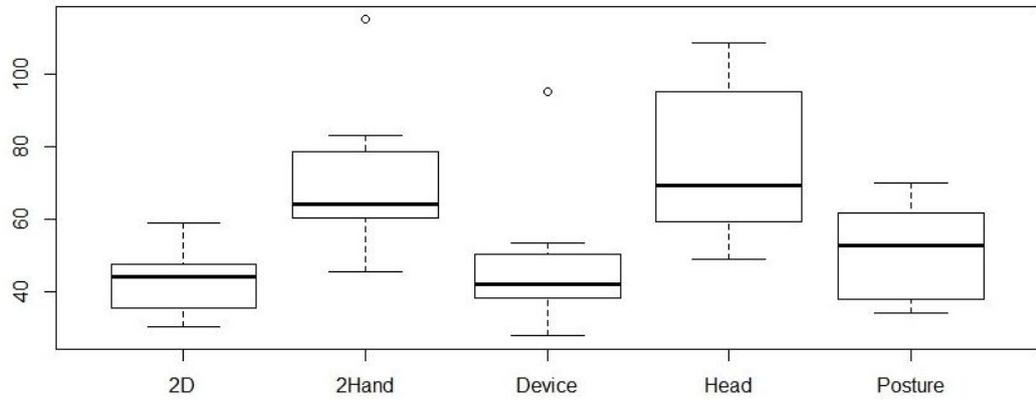
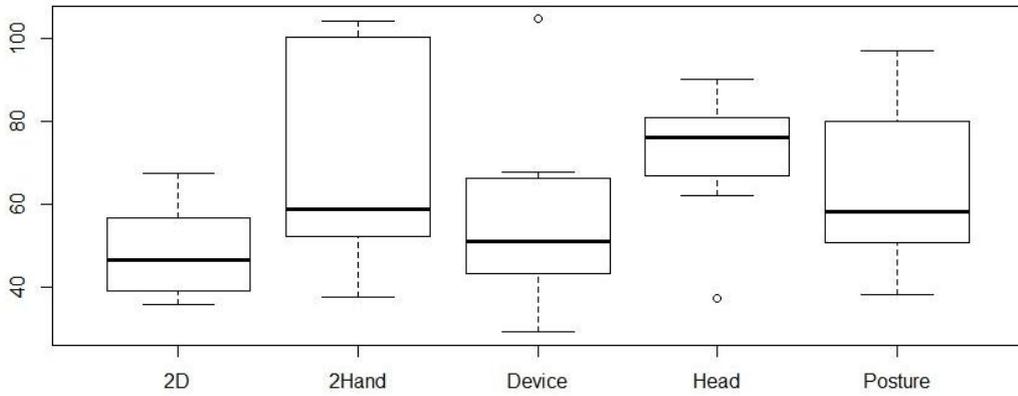


Figure 5.2.: Times required to perform six tasks for each interaction technique.

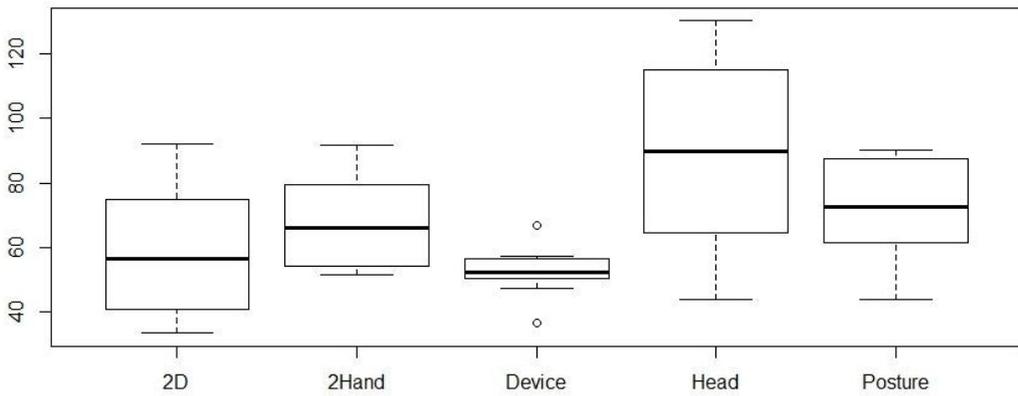
From figures 5.2 and 5.4 we deduce that the technique of choice does have a great impact on the performance speed. The 2D and Device based techniques performed best. Followed by the posture and two-hand interaction. The head-based interaction achieved the worst performance. These results at least partially support the hypotheses H1 and H2 and partially reject the hypotheses H3 and H4. To determine the correctness of hypotheses



(a) slow



(b) intermediate



(c) fast

Figure 5.3.: Times required to perform six tasks for each interaction technique and speed

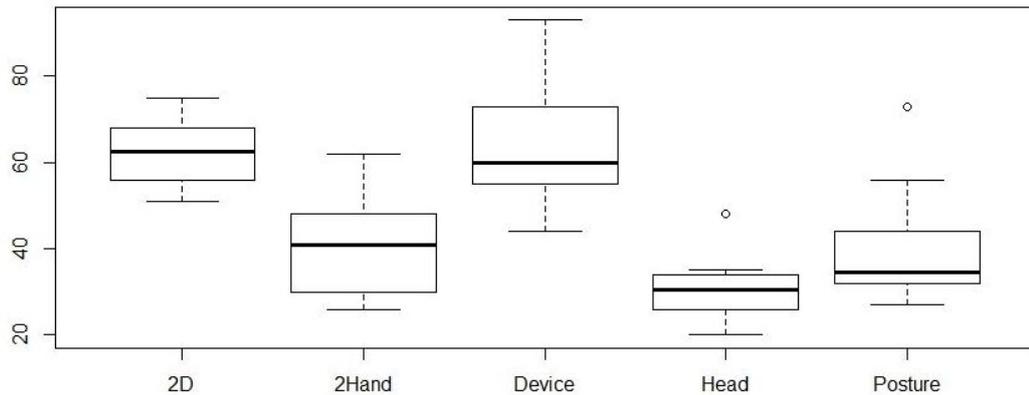


Figure 5.4.: Amount of tasks completed within a five minute period.

H2-H4 an observation of the game performance is necessary. This will be discussed in chapter 5.2.2 .

Multiple sources can be seen as possible reasons for this result. The first is the amount of possible degrees of freedom (DOFs). We observe primarily the translational DOFs and take rotational DOFs into account only when they are the primal sources of input. The 2D and device interaction share the property of only four DOFs. The 2D interaction combines two DOFs for the right and two DOFs for the left hand. The device interaction combines three DOFs for the right (pointing) hand and two DOFs for the interaction hand. But practically the user had to translate the hand only in one dimension to switch the desired interaction thus we can assume that the device interaction had only four DOFs in total.

The posture interaction had theoretically only four DOFs, three DOFs for the pointing hand and one rotational DOF to reach the interaction postures after previously placing the hand in a fixed location. During our experiments however we often observed that users lowered and raised the left hand time and time This results in a total of six to seven DOFs. The two-hand interaction had seven DOF's in total. These were a combination of three DOFs for each hand and one DOF for the right foot which was required to trigger the interaction.

The final position was taken by the eight to ten DOF head interaction. During the design of the experiments we assumed that only four DOFs would be required for this interaction method, two rotational DOFs for each hand and two DOFs for the head pointing. However we witnessed the same occurrences as with the posture interaction, namely the users lowered and raised the hands instead of simply rotating them. Therefore the interaction enclosed three to four DOFs for each hand and two DOFs for the head. This supports the hypothesis H1.

Another reason could be that the users were using large muscle parts or muscles they rarely use for the interaction. That was particularly visible with the head based interaction. Although it required the users only to rotate the head, many complained that they felt more fatigue than with other techniques due to constantly using their neck muscles, something they were not used to in such a short period of time.

The third reason can be how well the users were accustomed to the technique. The best performing techniques resembled interaction with desktop computer system, because

they use familiar elements. On the other hand head interaction was absolutely new for all participants.

A final possible explanation could be the difficulties users faced for fine adjustments with the head interaction. Only small head movements resulted in large movements on the screen. Furthermore the users could not control the head direction as precise as their finger locations on the tablets, which resulted in an increased execution time and probably attention demands as well.

### 5.2.2. Game Performance

We observed that the users made more errors with increased speed, as shown in figure 5.5.

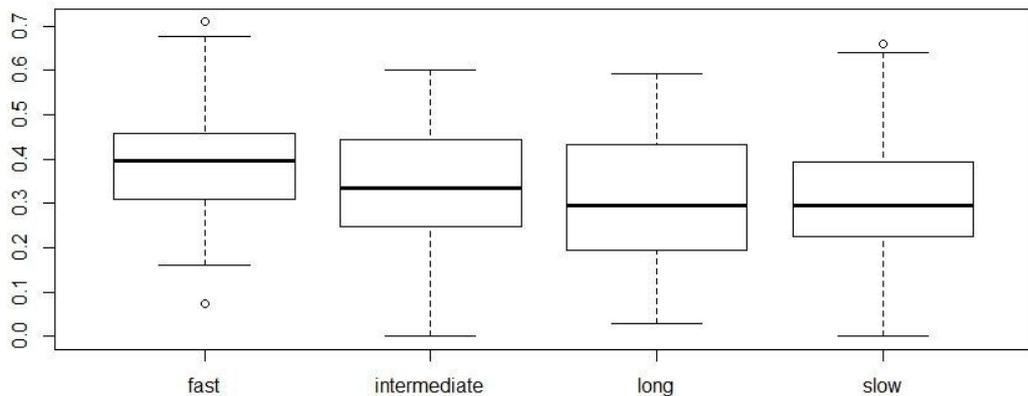


Figure 5.5.: Overall error rates based on speed.

The spikes towards the top in the error rates were caused by a user who in general performed very bad. On the other hand the spikes towards the bottom were caused by a user who performed close to perfection during all tests. Results without these participants can be seen in figure 5.6. For completeness we will include these participants in all further evaluations.

Interestingly neither the speed ( $F 0.392$ ,  $pr 0.76$ ) nor the choice of the interaction technique ( $F 1.549$ ,  $pr 0.209$ ) had any significant impact on the repetition error as shown in figure 5.7. This suggests that users attempted to observe and predict the next time they have to interact before focusing on the interaction. Another explanation could be that users reflexively continuously pressed the button. But the increased execution time and the increased overall error rate suggest otherwise.

Due to the constant repetition error the overall error must be a result of the different miss error rates ( $F 11.19$ ,  $p < 0.0001$ ). These are shown in figure 5.8. It displays that surprisingly the longterm test had a higher miss rate than the slow test. These observations suggest that the users devoted a certain period of time to the interaction task whenever they focused on it.

The interaction techniques had a significant impact on the misserror rates ( $F 5.373$ ,  $pr 0.00169$ ). This relation is shown in figure 5.9. We can observe that the error divides interaction techniques into two groups.

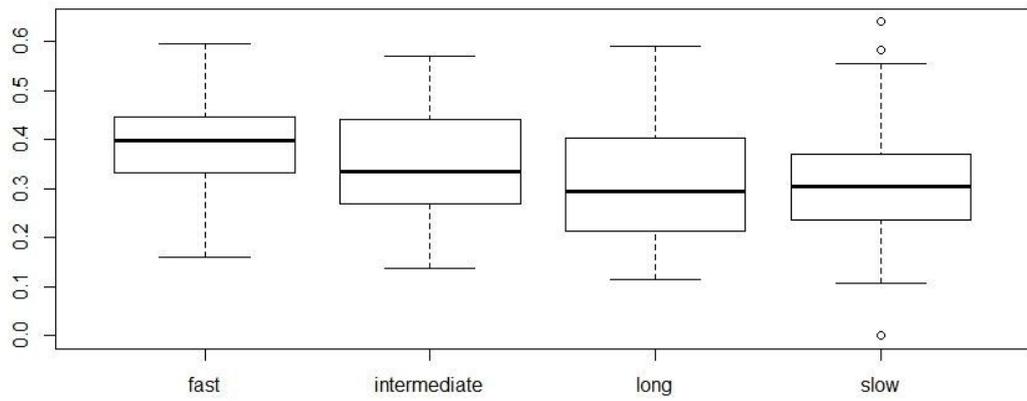


Figure 5.6.: Overall error rates based on speed without the best and worst performing participants.

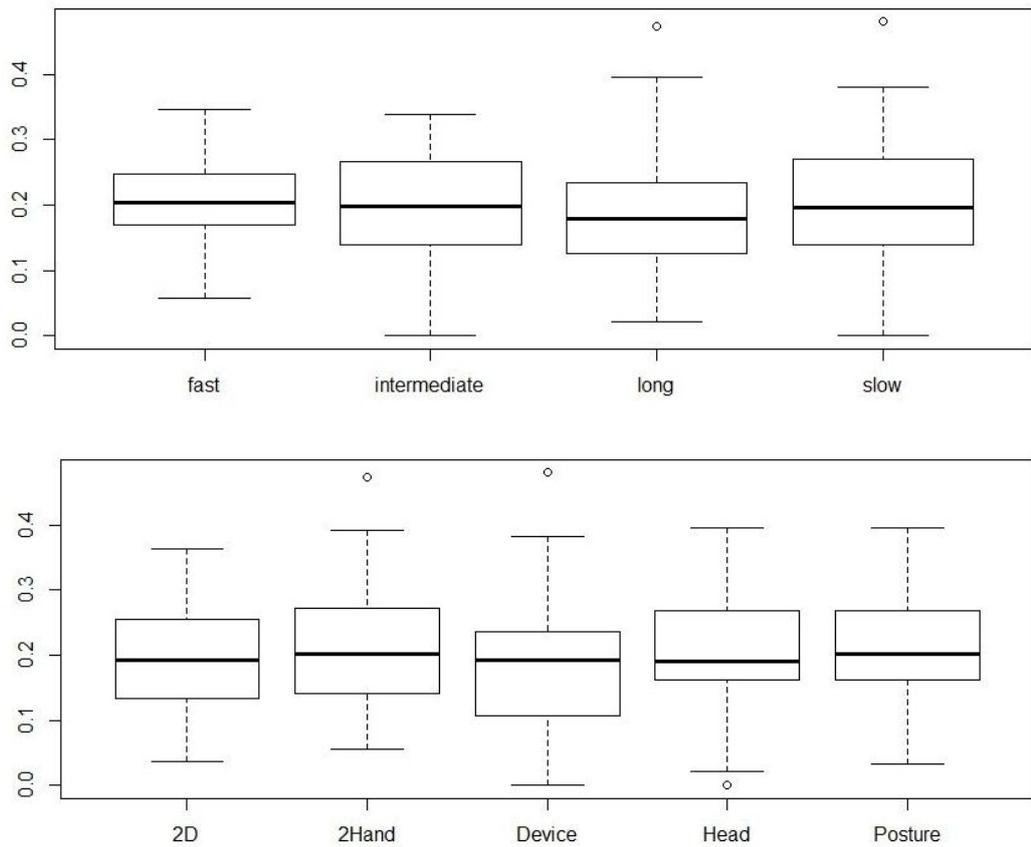


Figure 5.7.: The repetition error dependent on game speed and interaction technique.

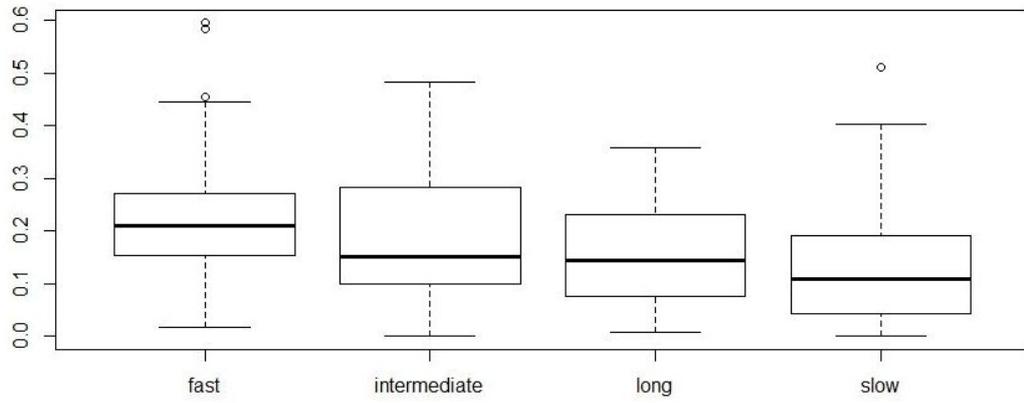


Figure 5.8.: The miss error dependent on game speed.

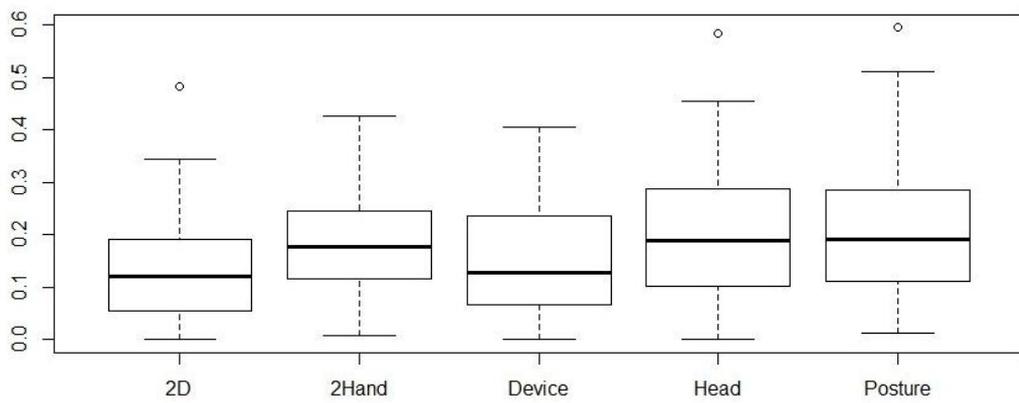


Figure 5.9.: The miss error dependent on interaction technique.

The first group consisted of the 2D and device interaction. This result is equal to the performance time. But the second group included all other methods of spatial interaction. This result supports our hypothesis H4. One would expect that the users who required more time to solve a task made less errors, but our observations show that mostly this is not the case. Figure 5.10 shows an example of the observed distributions. Weighing the error rate by the time made the results more prone to random errors (F 3.34, pr 0.02) but did not change the overall results significantly, as shown in figure 5.11.

As the pointing of the device and posture interaction was the same, the different error rates must be a result of the differences in the way to choose the desired interaction. We suspect that the additional local information of the left hand, which had to be processed during the posture interaction, was the reason for the difference.

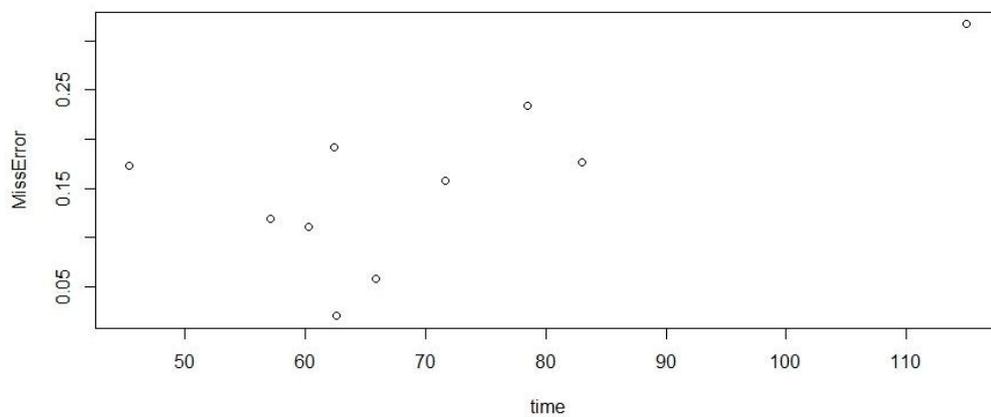


Figure 5.10.: The miss error rates for the 2-Hand interaction during the slow speed test.

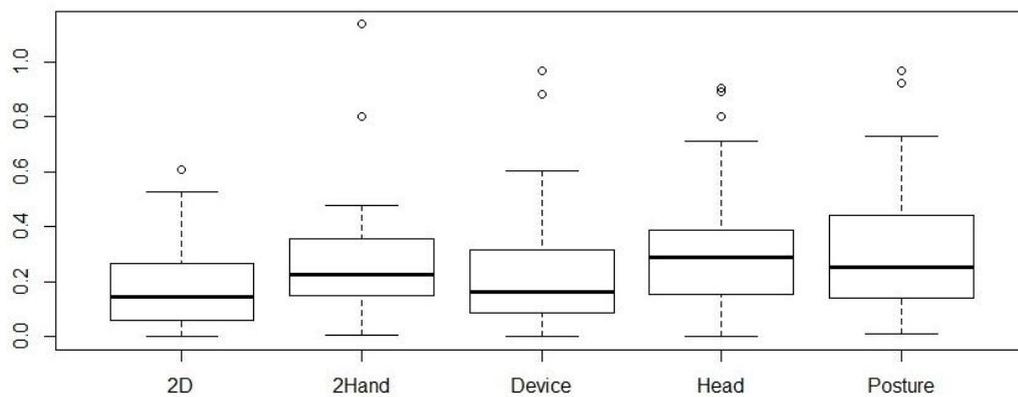


Figure 5.11.: The miss error weighted by performance time.

Interestingly we observed that for some users the error rates were very similar for all interaction techniques. But for others we could clearly distinguish efficient and inefficient techniques.

Another unexpected result was the error rate for particular tests. We observed that the

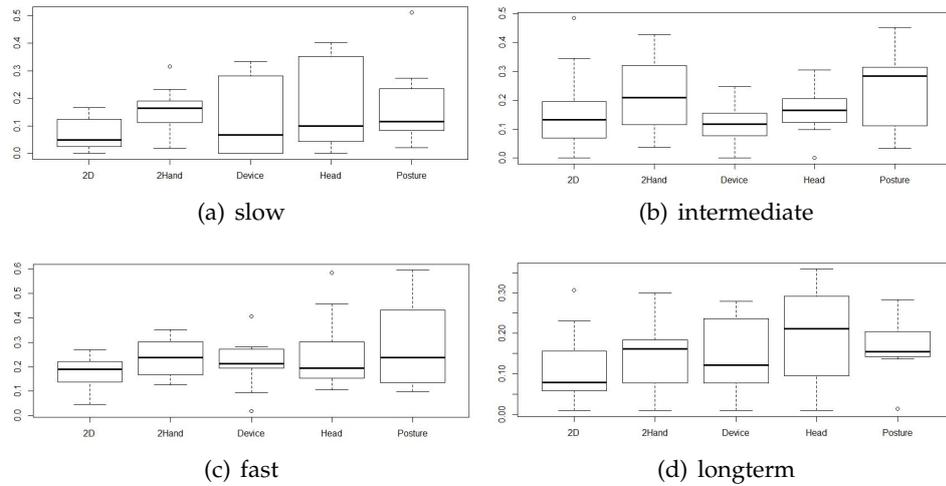


Figure 5.12.: Errorrates for each test.

error rates for each test in combination with the interaction techniques, as shown in figure 5.12, show no statistical differences. But nonetheless we can observe multiple error levels for the slow, intermediate and longterm tests. All techniques performed equally well for the fast test. The participants performed worst during the longterm test. Nonetheless we could not find any performance changes within.

These results show that our hypothesis H1 and H2 were confirmed, while the hypothesis H3 and H4 were rejected.

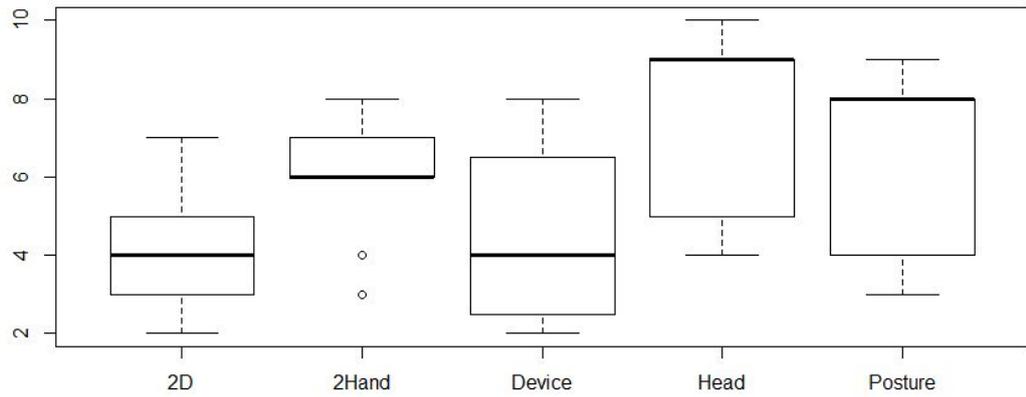
### 5.3. User Questionary

After the tests the users were asked to answer the following questions for each interaction technique:

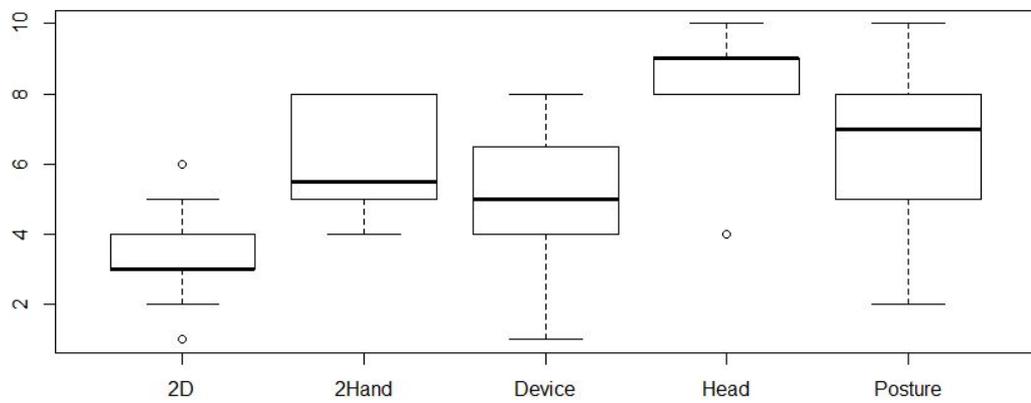
1. How mentally demanding was the interaction method?
2. How physically demanding was the interaction method?
3. How much did you feel distracted by the interaction tasks?
4. What do you think, how high was your error rate?
5. How did you like the interaction method?
6. How frustrating was the interaction?

Participants could rate each question on a scale of 1 to 10 where 1 was very little and 10 was very much. Our results show that users felt a connection between mental and physical load, thus rating the results very similar for all techniques (figure 5.13).

We can see that the device interaction is the only exception in our results. The reason for that seems to be that users had no problems to trigger the interaction because they were



(a) Q1: How mentally demanding was the interaction method?



(b) Q2: How physically demanding was the interaction method?

Figure 5.13.: Similar results for the mental and physical demands.

familiar with the use of devices for computer interaction, thus only very little mental load was necessary. On the other hand the pointing task required to use the whole arm, thus tiring the users. However some users noted that placing the arm in a fixed location, thus reducing the amount of movement required for pointing, would greatly reduce the fatigue and improve the performance.

The amount of focus required for the interaction is shown in figure 5.14 . As expected this result was very similar to the previous questions.

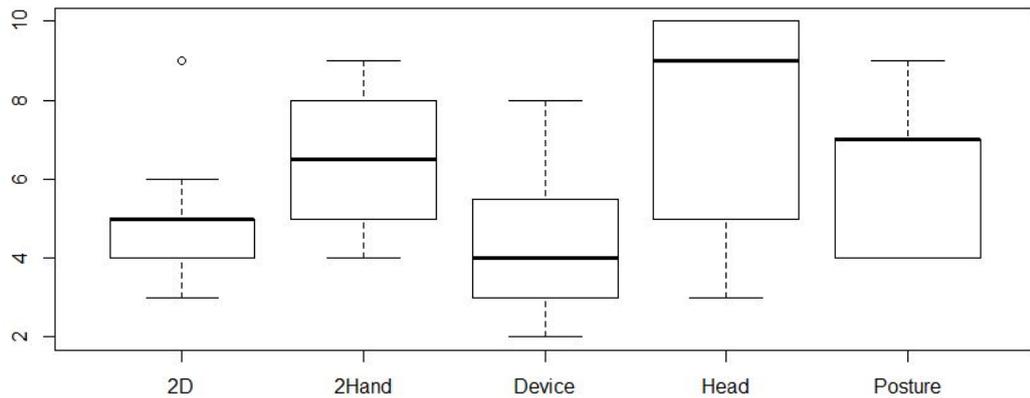


Figure 5.14.: Q3: How much did you feel distracted by the interaction tasks?

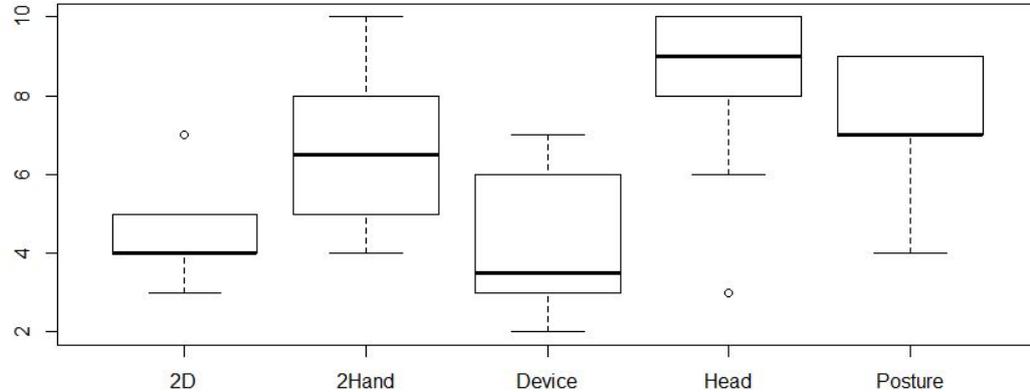


Figure 5.15.: Q4: What do you think, how high was your error rate?

The participants expected that their error rates corresponded to figure 5.15. We believe that this is a combination of higher attention demands coupled with more time spent for each interaction. But as was shown before, that was actually not the case.

The biggest surprise were the answers to question five (figure 5.16). We can see that despite the expected high error rate for the two-hand interaction technique the users liked it very much. This suggests that users are in general ready to accept minor errors and time requirements as long as they feel comfortable and used to an interaction method. On the other hand the posture interaction was widely disliked. After looking into the personal

results and voting we observed that users who had no problems attaining the poses rated this method very high, among the best scored techniques of our experiment. On the other hand the other half rated it very low. These users had troubles assuming the correct poses and sometimes initiated the wrong interaction method.

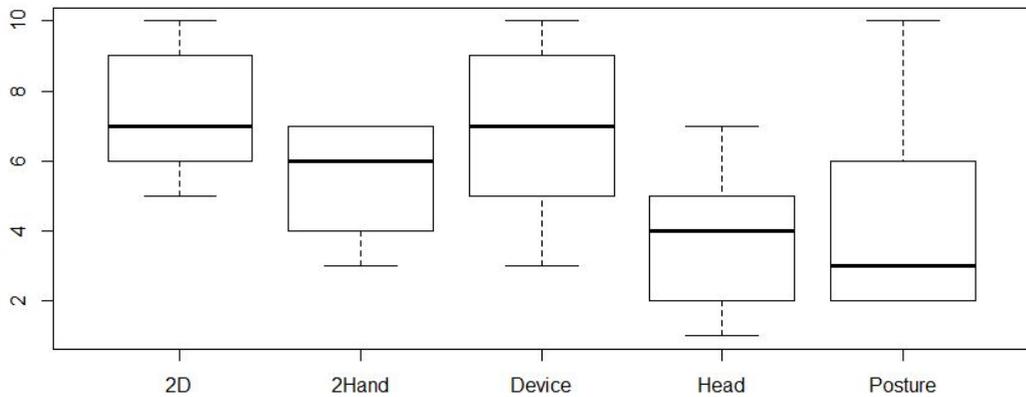


Figure 5.16.: Q5: How did you like the interaction method?

The answers to the final question (figure 5.17) were expected for the reasons stated previously. First of all, the head, two-hand and posture interaction techniques were new to the users. Secondly users who had troubles with the correct poses were expected to feel more frustrated. Finally users had problems for fine adjustments with the head interaction and felt that they had to use previously underused muscles.

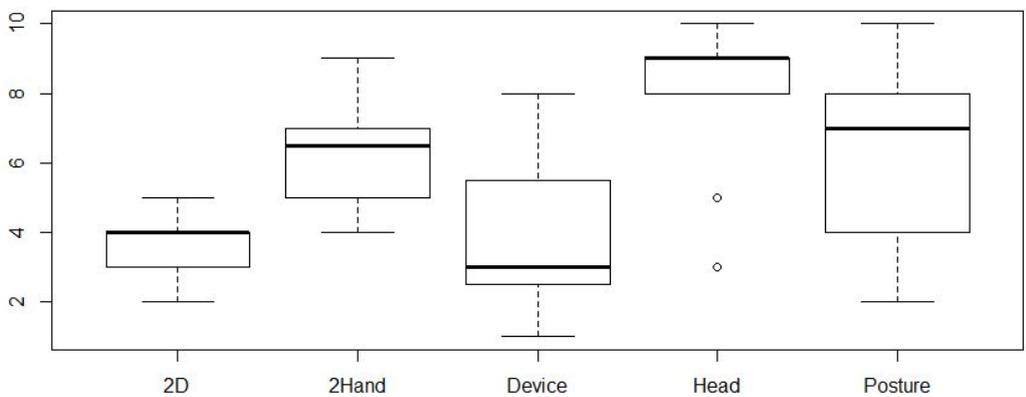


Figure 5.17.: Q6: How frustrating was the interaction?

## 6. Summary and Outlook

During our research we have implemented multiple techniques for wall-sized screen interaction. Our goal was to observe the user's ability to multitask at a wall-sized screen and the impact the interaction techniques had on the performance in regard of execution time and error rates. All techniques resulted in higher user error rates than when they had to focus only on one task. We were surprised that the secondary error was not influenced by any factors of our experiment, thus we conclude that this error is the result of multitasking.

In regard to the more relevant miss error we found out that interaction techniques which involve a device and the motion of only a small portion of our muscles resulted in less errors than techniques based on spatial interaction. These techniques performed faster as well. Among the purely spatial interaction based techniques we observed that they achieved similar error rates, as was expected.

In terms of speed we see three groups, where the familiarity and amount of degrees of freedom are the main dividing factors. The device and 2D interaction techniques scored best, followed by the posture and two-hand interaction and finally the head interaction.

Despite these results we believe that further testing of the posture technique is required due to the different impressions the users had of this technique. For further testing we must define new postures, which are easier to acquire and thus reduce the mental load on the users. The high ratings it achieved among the users who had no problems with the predefined postures suggest that it may perform far superior than in our experiment with a different set of postures.

The close results of the device and 2D interaction suggest that both are equally useful and should be preferred over other techniques. One must not forget that users complained that all techniques which involved spatial pointing were tiring and lead to faster exhaustion. On the other hand the tablet device required the users to use both hands for the holding task and put additional stress through its increased weight. Therefore we suggest that device interaction should be chosen over the 2D interaction. It requires the user to carry a by far lighter device and requires only one hand for the holding task. The other hand can be used for other tasks, thus promoting collaboration and interhuman communication.

We believe that further research in computer vision reconstruction algorithms will allow us to detect even small pose changes thus enable the reproduction of interaction techniques resembling our current device techniques and make these obsolete, similar to the clicking technique of Vogel and Balakrishnan [34]. A question in this aspect could be: How much influence did the haptic feedback of the devices have on the performance results? Some improvements which enable the detection of fine pose changes will allow to reduce the fatigue of the users, as the pointing task will involve less muscles and as a side-product increase the performance speed as well.

# Appendix

## List of Figures

3.1. Framework of the experiment. . . . .	7
3.2. The OptiTrack camera setup. . . . .	8
3.3. Implemented interaction techniques. . . . .	9
3.4. The Interaction screen as displayed on the Samsung Galaxy Tab. . . . .	10
3.5. The propabilities of spawning a tile. . . . .	16
3.6. The user interface displayed at the wall-sized screen. . . . .	16
3.7. Different tasks given to the user. . . . .	17
4.1. Class diagram of the interaction calculations. . . . .	19
4.2. The Java Classes for the user interface . . . . .	20
5.1. A participant taking the test. . . . .	21
5.2. Times required to perform six tasks for each interaction technique. . . . .	23
5.3. Times required to perform six tasks for each interaction technique and speed	24
5.4. Amount of tasks completed within a five minute period. . . . .	25
5.5. Overall error rates based on speed. . . . .	26
5.6. Overall error rates based on speed without the best and worst performing participants. . . . .	27
5.7. The repetition error dependent on game speed and interaction technique. . .	27
5.8. The miss error dependent on game speed. . . . .	28
5.9. The miss error dependent on interaction technique. . . . .	28
5.10. The miss error rates for the 2-Hand interaction during the slow speed test. .	29
5.11. The miss error weighted by performance time. . . . .	29
5.12. Errorates for each test. . . . .	30
5.13. Similar results for the mental and physical demands. . . . .	31
5.14. Q3: How much did you feel distracted by the interaction tasks? . . . . .	32
5.15. Q4: What do you think, how high was your error rate? . . . . .	32
5.16. Q5: How did you like the interaction method? . . . . .	33
5.17. Q6: How frustrating was the interaction? . . . . .	33

# Bibliography

- [1] Implementation of the svd algorithm. Available online at <http://www.nr.com/webnotes/nr3web2.pdf>.
- [2] Open source computer vision (opencv). Available online at <http://opencv.willowgarage.com/wiki/>.
- [3] tuiodroid - open source tuio app for android. Available online at <http://code.google.com/p/tuiodroid/>.
- [4] Virtual reality peripheral network. Available online at <http://www.cs.unc.edu/Research/vrpn/>.
- [5] Thomas Augsten, Konstantin Kaefer, René Meusel, Caroline Fetzer, Dorian Kanitz, Thomas Stoff, Torsten Becker, Christian Holz, and Patrick Baudisch. Multitoe: high-precision interaction with back-projected floors based on high-resolution multi-touch input. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology, UIST '10*, pages 209–218, New York, NY, USA, 2010. ACM.
- [6] Patrick Baudisch, Ben Bederson, and Alexander Zierlinger. Drag-and-pop: Accelerating drag-and-drop on large screens, 2002.
- [7] Patrick Baudisch, Edward Cutrell, George Robertson, and Pravin Santiago. High-density cursor: An improved mouse cursor that accelerates target acquisition, 2003.
- [8] Patrick Baudisch, Nathaniel Good, Victoria Bellotti, and Pamela Schraedley. Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, CHI '02*, pages 259–266, New York, NY, USA, 2002. ACM.
- [9] Patrick Baudisch, Mike Sinclair, and Andrew Wilson. Soap: a pointing device that works in mid-air. In *Proceedings of the 19th annual ACM symposium on User interface software and technology, UIST '06*, pages 43–46, New York, NY, USA, 2006. ACM.
- [10] Xiaojun Bi and Ravin Balakrishnan. Comparing usage of a large high-resolution display to single or dual desktop displays for daily work. In *Proceedings of the 27th international conference on Human factors in computing systems, CHI '09*, pages 1005–1014, New York, NY, USA, 2009. ACM.
- [11] Andrew Bragdon, Eugene Nelson, Yang Li, and Ken Hinckley. Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11*, pages 403–412, New York, NY, USA, 2011. ACM.

- [12] N. Charness, P. Holley, J. Feddon, and T. Jastrzembski. *Light pen use and practice minimize age and hand performance differences in pointing tasks*, pages 373–384. *Human Factors* 46.
- [13] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 27(3):98:1–98:10, August 2008.
- [14] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, sept. 2010.
- [15] Gene H. Golub and Charles F. van Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)*. The Johns Hopkins University Press, 3rd edition, 1996.
- [16] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [17] Ricardo Jota, João M. Pereira, and Joaquim A. Jorge. A comparative study of interaction metaphors for large-scale displays. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, CHI EA '09*, pages 4135–4140, New York, NY, USA, 2009. ACM.
- [18] Alessandro Lai, Alessandro Soro, and Riccardo Scateni. Interactive calibration of a multi-projector system in a video-wall multi-touch environment. In *Adjunct proceedings of the 23rd annual ACM symposium on User interface software and technology, UIST '10*, pages 437–438, New York, NY, USA, 2010. ACM.
- [19] Yebin Liu, C. Stoll, J. Gall, H.-P. Seidel, and C. Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1249–1256, june 2011.
- [20] Shahzad Malik, Abhishek Ranjan, and Ravin Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *ACM SIGGRAPH 2006 Sketches, SIGGRAPH '06*, New York, NY, USA, 2006. ACM.
- [21] Anne Collins McLaughlin, Wendy A. Rogers, and Arthur D. Fisk. Using direct and indirect input devices: Attention demands and age-related differences. *ACM Trans. Comput.-Hum. Interact.*, 16(1):2:1–2:15, April 2009.
- [22] Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11*, pages 177–186, New York, NY, USA, 2011. ACM.
- [23] Tao Ni, Greg S. Schmidt, Oliver G. Staadt, Mark A. Livingston, Robert Ball, and Richard May. A survey of large high-resolution display technologies, techniques, and applications. In *Proceedings of the IEEE conference on Virtual Reality, VR '06*, pages 223–236, Washington, DC, USA, 2006. IEEE Computer Society.

- [24] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, 2007.
- [25] R. Raskar, M.S. Brown, Ruigang Yang, Wei-Chao Chen, G. Welch, H. Towles, B. Scales, and H. Fuchs. Multi-projector displays using camera-based registration. In *Visualization '99. Proceedings*, pages 161–522, oct. 1999.
- [26] W. A. Rogers, A. D. Fisk, A. C. McLaughlin, and R. Pak. *Touch a screen or turn a knob: Choosing the best device for the job*, pages 271–288. *Human Factors* 47. 2005.
- [27] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, UIST '09, pages 167–176, New York, NY, USA, 2009. ACM.
- [28] W. Schneider and A. D. Fisk. *Degree of consistent training: improvements in search performance and automatic process development*, pages 160–168. *Perception Psychophysics* 31. 1982.
- [29] Jeremy Scott, David Dearman, Koji Yatani, and Khai N. Truong. Sensing foot gestures from the pocket. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, UIST '10, pages 199–208, New York, NY, USA, 2010. ACM.
- [30] Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. Shadow reaching: a new perspective on interaction for large displays. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 53–56, New York, NY, USA, 2007. ACM.
- [31] Sophie Stellmach, Sebastian Stober, Andreas Nürnberger, and Raimund Dachsel. Designing gaze-supported multimodal interactions for the exploration of large image collections. In *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*, NGCA '11, pages 1:1–1:8, New York, NY, USA, 2011. ACM.
- [32] Jayson Turner, Andreas Bulling, and Hans Gellersen. Combining gaze with manual interaction to extend physical reach. In *Proceedings of the 1st international workshop on pervasive eye tracking & mobile eye-based interaction*, PETMEI '11, pages 33–36, New York, NY, USA, 2011. ACM.
- [33] A. M. Valk. *An experiment to study touch screen "button" design*, pages 127–131. *Proc. Hum. Factors Ergonomics Soc.* 29.
- [34] Daniel Vogel and Ravin Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, UIST '05, pages 33–42, New York, NY, USA, 2005. ACM.
- [35] N. Walker, D. A. Philbin, and A. D. Fisk. *Age-related differences in movement control: Adjusting sub-movement structure to optimize performance*, pages 40–52. *J. Gerontology: Psych. Sciences* 52. 1997.

- [36] Nadir Weibel, Anne Marie Piper, and James D. Hollan. Exploring pen and paper interaction with high-resolution wall displays. In *Adjunct proceedings of the 23rd annual ACM symposium on User interface software and technology*, UIST '10, pages 455–456, New York, NY, USA, 2010. ACM.
- [37] C. D. Wickens. *Processing resources in attention*, pages 63–98. *Varieties of Attention*, R. Parasuraman and D. R. Davies Eds. Academic Press, New York, 1984.
- [38] ByungIn Yoo, Jae-Joon Han, Changkyu Choi, Kwonju Yi, Sungjoo Suh, Dusik Park, and Changyeong Kim. 3d user interface combining gaze and hand gestures for large-scale display. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, CHI EA '10, pages 3709–3714, New York, NY, USA, 2010. ACM.
- [39] Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (magic) pointing. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, CHI '99, pages 246–253, New York, NY, USA, 1999. ACM.