Graduate School of Natural Sciences
Department of Information and Computing Sciences

Academic year 2010–2011

Gesture-Based Interaction Concepts
For Mobile Augmented Reality Applications

Casper van Wezel

Supervisor: dr. W.O. Hürst

In partial fulfillment of the requirements for the degree of Master of Science
This thesis describes my research into gesture-based interaction concepts for mobile augmented reality. This thesis can be seen as part of a larger study into different interaction and visualisation techniques for mobile augmented and virtual reality, in which multiple master students contribute. Although the work in this thesis has solely been done by me, I decided to write this thesis in the plural form instead of the singular form. The reason for this is, that I feel part of a group of students who worked together to make sure each one could make the most of their own master thesis. The collaboration was both helpful and educative, since we could share code but also discuss new ideas, concepts and approaches.

Parts of this thesis have been:

- Published in the 17th international conference on Advances in MultiMedia Modeling (MMM2011) [1].
- Presented at the symposium “Future of gaming” organized by A-Eskwadraat.
- Used actively in other MSc projects related to augmented reality and virtual reality.

The author and supervisor give permission to use this thesis for consultation and to copy parts of it for personal use. Every other use is subject to the copyright laws, more specifically the source must be extensively specified when using this thesis.
Abstract

Nowadays mobile devices are packed with high powered processors, cameras, touchscreens and a multitude of different sensors which can all be used to create natural interfaces for mobile applications. Still, current augmented reality applications, i.e. applications that use live images from the camera and augment them with virtual 3D objects to create the impression that the real and the virtual world are aligned, mainly rely on simple touchscreen gestures for interaction. In this experimental study, we utilize the different sensors to create more complex and engaging ways of interacting with an augmented reality on mobile devices. Finger-based interaction is the central interaction technique, yet we also explore a compass and accelerometer based approach.

In a first experiment we compare our created finger-based setup against the touchscreen and a compass-accelerometer based approach. It turned out that our finger-based interaction performed the worst, however the participants praised the engagement and fun-factor of the approach. Based on these findings we created a new table-top augmented reality with a game-like feeling and tested it in a second experiment. The results from the data analysis, questionnaire and interviews of the second experiment prove that the new form of augmented reality is a suitable platform for our finger-based interaction concepts.
First of all I would like to thank dr. Wolfgang Hürst for his support and discussions during the course of this thesis. I believe that without him I would not have been able to create such “great” interaction concepts and gesture-based system. A special word of appreciation goes out to all the guys over in room 684 of the Buys Ballot Laboratory. Without you guys creating this thesis would have been much less fun. I also owe a great deal of thanks to all my other friends for their moral support and advice in times of need. I would like to thank all the people that participated in the user-studies for their cooperation and I should also express my gratitude to my family and girlfriend for all their support, love and so on...

Final thanks goes out to everybody who has been involved in my study at Utrecht University during the last 6 years. You guys made these years totally awesome and unforgettable!
## Contents

| Abstract | iii |
| Acknowledgements | v |

### 1 Introduction

1.1 General introduction ........................................... 1
1.2 Augmented Reality ............................................... 3
   1.2.1 Short history of Augmented Reality ....................... 4
   1.2.2 Smartphones and Augmented Reality ....................... 5
1.3 Goal of this thesis ............................................ 6
1.4 Overview .................................................... 7

### 2 Related work

2.1 Tracking techniques ........................................... 9
2.2 Interaction techniques ....................................... 11

### 3 Finger tracking

3.1 Software ................................................. 13
3.2 Hardware ................................................. 14
3.3 Finger tracking on smartphones ................................ 14
   3.3.1 Retrieving camera images ................................ 14
   3.3.2 Decoding camera images ................................ 15
   3.3.3 Detecting and tracking the finger ....................... 16
      3.3.3.1 Color segmentation ................................ 16
      3.3.3.2 Marker detection ................................ 17
      3.3.3.3 Marker tracking with search windows .............. 18
      3.3.3.4 The final algorithm ................................ 19
3.4 Physical- and hardware-based limitations ..................... 19
   3.4.1 The human arm ......................................... 20
      3.4.1.1 Ranges of motion ................................ 20
      3.4.1.2 Arm lengths ...................................... 21
      3.4.1.3 Reachable workspace ............................... 22
   3.4.2 Limitations of the hardware ............................ 23
4 First experiment: Feasibility study

4.1 Touchscreen interaction concept

4.2 Device interaction concept

4.3 Finger-based interaction concept

4.4 User study

4.5 Results

4.5.1 Average times

4.5.2 Object selection

4.5.3 Menu selection

4.5.4 Translation

4.5.5 Questionnaire

4.6 Discussion

5 Revised augmented reality and finger tracking

5.1 Game board

5.2 Interaction styles

5.2.1 Translation - Pick up and release

5.2.2 Translation - Pushing

5.2.3 Scaling - Gesture-based

5.2.4 Scaling - Touchscreen

5.2.5 Rotation - One finger

5.2.6 Rotation - Two fingers

5.3 Visual feedback

6 Second experiment: Augmented reality board game interaction

6.1 Experiment setup

6.1.1 Level setup

6.2 Expectations

6.2.1 Translation

6.2.2 Scaling

6.2.3 Rotation

6.3 Results

6.3.1 Average times

6.3.2 Accuracy

6.3.3 Distance between the fingers and the camera
Contents

6.3.4 Performance of the natural feature tracker . . . . . . . . . . . . 59
6.3.5 Questionnaire and observations . . . . . . . . . . . . . . . . . 59
   6.3.5.1 Engagement . . . . . . . . . . . . . . . . . . . . . . . . 61
   6.3.5.2 Difference between mid-air and the board . . . . . . . 61
6.4 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61

7 Conclusion and future work 63
   7.1 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
   7.2 Future work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
      7.2.1 Markerless finger tracking . . . . . . . . . . . . . . . . . . 64
      7.2.2 3D finger tracking . . . . . . . . . . . . . . . . . . . . . . . . 64
      7.2.3 Implementation in a real game . . . . . . . . . . . . . . . . 65

A Appendix 67

Bibliography 71
Chapter 1

Introduction

This thesis describes the research into gesture-based interaction concepts for mobile augmented reality applications. This chapter starts with a general introduction into the field of gesture-based computing in section 1.1. Section 1.2 explains augmented reality in more detail and section 1.3 set forth the goals and objectives of this study. Section 1.4 specifies the outline of this thesis.

1.1 General introduction

In recent years, a new type of mobile device has become extremely popular. It is marketed as a “smartphone” and can best be described as a mixture of a portable computer and a phone. The production of the smartphone started around the year 2000 with the introduction of the Ericsson R380 \[2\]. This device was a combination of a personal digital assistant (PDA) with a normal cell phone and used a touchscreen for interaction. In years that followed more devices from different manufactures were launched. With each iteration the devices and underlying operating systems became more sophisticated and advanced.

The year 2007 can be seen as the foundation year for the smartphones of today with the release of the Nokia N95 and the Apple iPhone. The N95 had a wide range of feature such as GPS, a 5 megapixel camera with autofocus and LED flash, an accelerometer, 3G, Wi-Fi connectivity and TV-out \[3\]. The feature list of the first iPhone was less impressive than that of the N95, but nevertheless it is considered a “game changer” in the smartphone market. The iPhone was the first smartphone that featured a captive multi-touchscreen. The touchscreen gave Apple the opportunity to create a more natural user interface, changing the way users interact with their mobile devices. The implemented gestures, such as swiping or the “pinch to zoom” gesture can still be found on today’s smartphones. The features from the N95 and the multi-touchscreen from the iPhone became the standard for smartphones around the world.

Besides the introduction of gesture-based computing systems on mobile platforms, there was also the introduction of gesture-based gaming systems. In November 2006 Nintendo launched the Nintendo Wii, introducing a new way of gaming on consoles. The Wii is operated with the use of the Wii Remote, which is a handheld pointing device capable of tracking movements in all three dimensions. It makes use of accelerometers and infrared lights that are pickup up by a sensor bar to determine the spatial position. The information received from these devices made it possible
Chapter 1. Introduction

for game programmers to use physical gestures as input for their games. Gestures were made for all kinds of games, from sports games to first person shooters. The system turned out to be a huge success, with shipments running as high as 86.01 million units on 31-03-2011 [1]. Nintendo’s gesture-based systems was such a success that the other big console manufacturers, Sony and Microsoft, needed to develop their own systems. Thus the PlayStation Move and Microsoft Kinect entered the market, respectively in September and November of 2010. These gesture-based systems can be considered state of the art for consumer electronics, therefore we delved more into the inner workings of these systems.

The PlayStation Move is built up from different types of accelerometers, a magnetometer (for correcting errors in the inertial sensors) and a glowing orb on the top of the controller [5]. The glowing orb is used for tracking the distance between a camera, called the PlayStation Eye, and the controller. The distance can be calculated by analyzing the size of the orb’s emitted light on a captured image by the PlayStation Eye. The area is compared to known distance and size combinations of the spherical shaped orb [6]. To improve its robustness the orb has been outfitted with LED lights, which makes it possible for the orb to dynamically change the color to make sure that it is distinguishable from the user’s environment. The combination of the tracking algorithm and the accelerometer data creates a system that is capable of tracking the controller with an accuracy down to millimeters on the XY-plane and a few centimeters on the Z-plane according to the designers [7]. This makes the PlayStation Move far more accurate than the Nintendo Wii. This, however, is to be expected from a system that is released four years later and runs on much more powerful hardware.

The Microsoft Kinect is different from afore mentioned systems because “you are the controller”, indicating that the system is not controlled trough a “standard” game controllers [8]. The system is controlled through spoken voice commands and a natural user interface, which reacts to the user’s entire body without them having to wear sensors. Kinect consist of a horizontal bar containing a camera, depth sensor and multi-array microphone. The system is capable of tracking the user’s complete body in 3D, and features facial recognition and voice recognition. Kinect is implemented as a machine learning system that is trained to recognize up to thirty-one body parts [9][10]. The learning system is trained with a “vast and highly varied training set of synthetic images to ensure the system works for all ages, body shapes & sizes, clothing and hair styles”[10]. The incoming data from the camera and depth sensor is analyzed for body parts, after which the 3D body joint positions are calculated and used by a skeletal tracking algorithm to create the 3D body model [10]. The resulting 3D model is used for recognizing the numerous different gestures that the Kinect system can detect. The system received great reviews from many companies and gaming magazines, however it was also criticized for the latency between the movement and visualization on the screen. Nevertheless the impression remains that Kinect has raised the bar for motion controlled games and gesture-based gaming systems in general.
The year 2010 was also the year that tablet computers became extremely popular. A tablet is best described as a large smartphone. For the most part it has comparable hardware and runs the same OS, but it has a much larger screen, e.g. the iPad has a 9.7 inch (24.64 centimeter) screen, whereas new smartphones tend to have a screen size around 4 inch (10.16 centimeter). Although they are far less innovative than the PlayStation Move or the Kinect, the tablet computers are, just like smartphones, perfect examples of systems that can be controlled by gesture-based input and are extremely popular. The best known tablet is the Apple iPad, selling a million units in twenty-eight days and the iPad 2, launched in March 2011, selling almost a million units in its debut weekend [11][12].

The systems and devices mentioned above have created a wide-spread consumer interest, demand and more importantly acceptance of gesture-based interaction systems. We are now at a point in time were there are numerous devices available to consumers that largely or even solely depend on interaction with natural gestures.

In brief, this thesis is about natural gesture interaction. In this master thesis we follow the consumer interest in gesture-based systems. We will create and evaluate a gesture-based interaction system for augmented reality on smartphones. What makes this thesis special is that we are not going to create a gesture system using touchscreen technology. Instead we will make use of sensor and camera information to create a different way of interacting with the device. The camera is of special interest to us, since we are especially interested in finger-based interaction. Why we have chosen augmented reality and smartphones is further discussed in the following sections.

\subsection*{1.2 Augmented Reality}

Augmented Reality (further referred to as AR) applications augment the real world with virtual objects that display information that the user would normally not directly detect with his own senses. On a smartphone this means that virtual objects are combined with a video stream of the real world, which is perceived by the camera. Although AR has many different definitions, in theoretical literature the definition of Robert Azuma \[13\] is most commonly accepted. The definition states that an application is considered to be an AR application if it meets the following criteria:

1. It must combine the real and virtual world.
2. It must be interactive in real time.
3. The virtual world must be registered with the real world.

Numerous AR applications and systems have already been created. One of the most common and well known forms of AR is the augmentation of sport details on the playing field. An example that also conforms to the definition of Azuma is the line that is used to indicate the time of the previous contestant in speed skating.
Chapter 1. Introduction

The line is augmented on top of the ice, which gives the viewer an overview of the performance of the candidate compared to the best time. AR is also used in many other sports like rugby, soccer or ice hockey. Besides sports and other entertainment applications, AR is also used for more serious purposes such as the head-up displays in military aircrafts or a training aid for surgery by medical staff. Figure 1.1 shows some examples of AR applications.

![Figure 1.1: Different AR applications. The top left image is from the Layar application on an Android smartphone. The top right image is from a commercial for a new MINI Cooper. The bottom left image illustrates the augmented line in an ice skating match. The bottom right image is a photo of the HUD in a military aircraft.](image)

3. Still image from http://www.youtube.com/watch?v=6L3DrxplMs

1.2.1 Short history of Augmented Reality

AR finds its origin in 1968 when a researcher named Ivan Sutherland created the first real AR system. The system used a see-through head-mounted display that showed wireframe drawings on top of the real world [14]. The system operated by tracking the head, which was either done through a mechanical system or through an ultra sound system. In 1975 Myron Krueger created the Videoplace system, in which users could interact with virtual objects in real-time [15]. The visual representation
occurred on a screen, where the user could see his silhouette and the virtual objects. Interaction was made possible through sensors in the floor, sensors on a table and a camera system. The first serious AR application was created by Tom Caudell and David Mizell in 1992 as employees for Boeing Computing Services. Their goal was to create an AR system to help improve manufacturing of aircrafts by displaying template information for cables and such on specific positions on a real-world object \[16\]. It was Caudell and Mizell who coined the term “augmented reality”. In 1993 a paper about an AR system called KARMA was published. This paper is widely cited and explains an AR setup in which a head-mounted display is used to show the internals of a laser printer \[17\]. This application is a good example of the maintenance and repair applications that have been created for AR. 1993 was also the year that the Global Positioning System (GPS) started to become operational. The GPS system is used by AR applications, such as Layar \[18\] and Wikitude \[19\], to display information about the surroundings of the user. In 1996 Jun Rekimoto introduced the 2D matrix markers, which are square shaped barcodes that can be used to align the coordinate systems of the virtual and the real world (figure 1.2) \[20\]. The idea of using a printable marker is still used widely throughout AR. In 1997 Azuma \[13\] released his survey on AR and Philippe Kahn invented the camera phone. In 1999 the ARToolKit was released by Hirokazu Kato and Mark Billinghurst under the open source GPL license. ARToolKit is a pose tracking library, which uses square fiduciary markers to track and align the real and virtual world. The system has six degrees of freedom and uses a template matching system for recognition. From 1999 onwards many different AR applications started to arise, AR games, AR browsers, collaborative AR experiences and AR guidance systems are some examples. In 2001 researches started to experiment with PDA’s as AR platform, which spiked the interest in more mobile AR solutions and soon researchers were also implementing applications on mobile phones. In the following years many other AR tracking systems and applications were developed and tested. In 2008 the application Wikitude for the Android platform was released. This application makes use of the built-in GPS and compass to display Wikipedia information on top of a camera stream. This AR browser proved to be an enormous success. 2008 was also the year that a natural feature pose tracking system was released for mobile phones \[21\]. The system was created by Daniel Wagner et al and was capable of tracking an image at a frame rate of 20 Hz. In 2009 another very popular AR browser was released called Layar, which uses the same principle as Wikitude. The system was capable of displaying all kinds of information about a person’s immediate surroundings. In March of 2010, work on this thesis started. A more detailed list of all the milestones in AR has been created for the ISMAR society \[22\].

### 1.2.2 Smartphones and Augmented Reality

With the introduction and availability of smartphones, AR has become interesting for a wide-spread public. AR applications such as Layar and Wikitude have been installed millions of times, indicating that AR is booming business. A main contributor to the success of these applications is the ease in which they can be
installed on smartphones. Smartphones have access to application stores, which make sure that end-users can install any application within a few “clicks”. Besides distribution benefits, AR on smartphones and other mobile devices has also more general benefits, i.e. mobile devices combine the input and output hardware into one device and interacting with mobile devices is socially accepted, therefore making AR applications intuitive to use. However, mobile devices also have disadvantages compared to head-mounted displays or other techniques that are linked to a desktop computer. Mobile devices have, according to [23]:

- Limited input options (no mouse/keyboard).
- Limited screen resolutions.
- Little graphics support.
- Reduced processing power.

We believed that these disadvantages could mostly be overcome by taking the time to think about interaction and interface design. We therefore decided that the benefits of using a mobile device outweighed the disadvantages, which made us decide to develop our interaction systems on a smartphone.

1.3 Goal of this thesis

At the moment of writing, gesture-based interaction is rarely used in an AR setting, while the naturalness of gesture-based interaction could potentially increase the overall experience of AR applications. As mentioned, the main focus of this thesis is on finger-based interaction. We believe that these kinds of gestures have an additional benefit, i.e. the user can use his own fingers that are present in the real world to manipulate objects that are in the virtual world, further entangling the two worlds. One of our envisioned applications is an AR browser in which the user can point, with his finger, to an object to get more information about it. These kind of ideas made us realize that with gesture-based interaction we can create more complex and possibly more intuitive interaction compared to the standard “tab”

interaction that is used in current AR applications. The main objective of this thesis is to determine if these initial ideas and assumptions are correct.

This thesis has one main experimental goal:

- Determine the feasibility of finger-based gestures for use in mobile augmented reality.

This global goal can be further specified into the following three ancillary goals:

- Determine the performance of finger-based interaction and compare it to touchscreen and other gesture-based interaction.
- Determine the user experience of finger-based interaction and compare it to touchscreen and other gesture-based interaction.
- Determine the hardware and physical limitations of using finger-based interaction in mobile augmented reality.

1.4 Overview

The remainder of this thesis is organized as follows. Chapter 2 set forth previous and current research in the field. Chapter 3 describes the finger tracking system that we have created. This system was mainly used as a proof of concept and to determine the physical- and hardware-based limitations that arise when using finger-based interaction on smartphones in an AR setting. Chapter 4 describes the first experiment, which is an initial study to test our finger-based interaction concept against an accelerometer- and compass-based approach and the touchscreen. Chapter 5 discusses the transition to the second experiment. We describe the newly created interaction concepts and adjustments to the AR. Chapter 6 describes the second experiment in which we tested the newly created concepts and AR. Chapter 7 gives a conclusion and discusses possibilities for future work.
Chapter 2

Related work

In the introduction we have already described some interesting gesture-based systems and AR applications. This chapter is written as an elaboration on that, describing research that is done in the field of mobile interaction and AR.

AR and mobile interaction are both very active research fields. There are multiple conferences in existence that are either dedicated to the subjects or have a part reserved for these topics. For AR the best known conference is the IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR). The conference has been held every year since 2002 and gets funding and support from large companies such as Qualcomm, Volkswagen and Samsung (all three were conference partners in 2010). The main conference involving mobile interaction is the International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI), which also gets major support from well-known companies. The papers published in this conference are not directly related to AR, nevertheless the conference remains interesting for people working in the field of interaction development or AR.

The ISMAR conference gives the best indication of what is currently taking place in AR research. According to a survey from the University of Canterbury [24], most papers are published in relation to tracking techniques. This makes up for 20.1% of the 313 papers that were included in this survey. Interaction techniques (14.7%) come in second place, which is the main research area of this thesis. Together with calibration/registration (14.1%) and display techniques (11.8%) these areas are considered to be core AR technologies.

2.1 Tracking techniques

The goal of tracking techniques is to follow the position and orientation of the user/device in space. This is an important part of many AR applications, since it is used for the alignment of the real and virtual world. Without a good tracking system registration problems may arise that could potentially destroy the illusion that the worlds are aligned. This makes tracking an interesting topic for AR research. At ISMAR over 80% [24] of the papers published about tracking were vision-based approaches, making the use of the camera for tracking the most popular variant.

The vision-based approach can be subdivided into two groups, the feature-based and model-based approach. In the feature-based research field, the square marker
approach of Jun Rekimoto [20] is one of the most dominant techniques. In the years after its introduction many different forms and techniques were developed for tracking a square marker. Each system had its own strength and weakness as showed in a comparative study performed by Xiang Zhang et al [25]. Other forms, such as a ring-based marker tracking system, were also researched [26]. Nevertheless the square-based marker is still used in AR applications to this day.

Besides tracking a square marker, Park et al showed that it is also possible to get pose information from natural features such as points, lines, edges or textures [27]. With the increase in processing power and advantages in the technique itself Daniel Wagner et al were capable of creating a system that could track multiple markers with a guaranteed frame rate [28]. The natural features they used were contrast points in everyday images, such as a picture of a cat or woman. Other systems use contour structure as natural features, making it possible to track hand sketches [29]. Overall natural feature systems are becoming more popular, mainly because they are considered to less obtrusive and provide a more natural experience [30].

According to the survey [24] model-based tracking techniques are a recent trend in vision-based tracking. The main difference with feature-based systems is that these systems use a predefined model for pose estimation and tracking. This makes model-based tracking systems work contrary to feature-based tracking systems. Feature-based tracking systems look for features in the input stream and try to interpret those, while model-based tracking systems use prior knowledge of the shape and appearance of a specific object to determine how that object fits best in the input stream. An example of a working real-time system was presented by H. Wuest et al [31]. The implemented system used a CAD model and a line tracking algorithm to create a model-based tracker that works under partial occlusion and illumination changes.

Besides vision-based tracking techniques, sensor-based or hybrid-based techniques were also studied. Sensor-based techniques make use of sensory information for tracking. Many different sensors can be used for this, e.g. magnetic, acoustic, inertial, optical and/or mechanical sensors [24]. This research area was mostly studied in the time of the predecessors (IWAR) of the ISMAR conference and therefore is no longer a very active field.

Hybrid-based tracking combines vision-based tracking with sensor-based tracking. Sensor information can, for example, be used to pinpoint the location of the user (GPS), making it possible for a vision-based tracking system to eliminate tracking targets that are not in the immediate surroundings. Vision-based tracking systems often have problems with rapid motions. A hybrid system that uses sensors to detect such a movement can more easily recover and can therefore be more stable [24]. Due to the huge interest in vision-based tracking systems, this research field is not very active.
2.2 Interaction techniques

Tracking and registration techniques make it possible to align the virtual and real world. Just like in any other application, interaction is important for the user experience, therefore many different interaction techniques have been developed. A good example of a study combining interaction with AR is the well-known tangible user interface (TUI) MagicCup [32]. A TUI makes interaction with virtual objects possible through interaction with physical objects. For example, in MagicCup the user is able to pick up a virtual object by placing a “magic cup” from the real world on the “position” of the virtual object in the real world. Other interaction concepts such as moving and deletion were also implemented. The goal of the MagicCup interface was to see if the interaction concepts could be used in a tabletop AR setting. Another well-known TUI system is the VOMAR application, which uses a real paddle to arrange furniture in an AR living room [33]. This system was also extended with speech recognition [34], which proved to be an intuitive extension.

Naturally researchers also turned to finger-based interaction systems as input for AR, since finger-based interaction is arguable a very natural way of interaction with an AR. Klaus Dorfmüller-Ulhaas et al created an finger-based interaction system in which a glove with retro reflective markers was used to track 4 joints in one finger [35]. The resulting finger tracker was used for interaction with an AR chess application on a desktop computer. Later applications were also able to track the hand without the use of a retro reflective glove or other artificial tools. An example of such a system is created by M. Lee et al, who used a skin color segmentation algorithm to detect the hand and finger [36]. They tested touching and pointing interaction concepts in three different AR applications. The system was (also) connected to a desktop computer and the AR was perceived through a see-through device that was held in the free hand.

Matthias Baldauf and Peter Fröhlich combined a Nokia N95 with a projector to create a device that could capture gestures and use them in existing applications [37]. The projector was used to project screen information onto any given surface. The system was wearable, which meant that the user had both hands free for interaction. To get the desired performance, the researches decided to let a server process the camera information and calculate the resulting gestures. A more recent wearable system is the SixthSense project from the MIT Media Lab [38] [39]. The system is widely known for its “futuristic” demo videos, in which they show how their natural interactions can change daily life. In figure 2.1 we see the setup of this system. The camera and projector are connected to a laptop that is placed inside a backpack, which needs to be worn at all times. Gesture interaction becomes possible through colored markers that need to be placed on the finger tips. The SixthSense project has published 5 papers, but never submitted a technical paper describing how the system works in-depth.
Thus far the described systems all use processing power from external devices to establish their finger-based interaction. In this thesis we only want to use the available processing power from the mobile device itself. Luckily, multiple papers have already proven that this is possible. Anders Henrysson et al created a system that is capable of tracking the fingers in 3D by using a Nokia 6680 and a port of the ARToolKit for Symbian [40]. The goal of the research was to test different input forms, under which a 2D and 3D finger tracking approach, for object translation and rotation. The 2D finger tracking was established through a simple frame-differencing tracking technique which required no additional markers. 3D tracking was done by placing a marker from the ARToolKit on the tip of the participant’s finger. The pose estimation from the ARToolKit provided the position and depth information. Byung-Kuk Seo et al created a mobile system, in which the user could interact with virtual objects that were augmented on the user’s palm [41]. The systems used a generalized statistical color model to detect hand colored pixels. The result was a silhouette of the hand and background noise, which was later removed using a distance transform algorithm. The result was an accurate hand silhouette from which the program could estimate not only the hand’s position and orientation, but also hand-based or finger-based gestures. The detected gestures were relatively simple, i.e. a closing gesture and an opening gesture, however the system proved that markerless tracking on a mobile device is feasible.

For our own research, we decided to use a similar approach to the SixthSense project, since the main goal of this thesis is to evaluate the feasibility of finger-based gestures. Hence, the focus is not on the computer vision aspect of recognizing and tracking the fingers and therefore a simpler and more computationally efficient method will suffice.

---

1. Image from the SixthSense project [39].
Chapter 3

Finger tracking

In this chapter we describe the creation of the finger tracking system. The finger tracking system contained no interaction concepts, since we used the application as a “proof of concept” rather than a fully functional application. We will use the finger tracking application to look more into the area in which our finger-based concepts will perform. We believe the insight in this area will help us better understand the physical- and hardware-based limitations of our approach.

3.1 Software

This thesis was one of the first to start working with AR on smartphones at Utrecht University, which meant that there was no platform available. Hence our first step was to decide for which OS we were going to create our applications. In March / April 2010 there were three main OSs available namely iOS, Android and Symbian. Symbian was used in other University projects and was criticized for its lack of good debugging tools and other difficulties. From searching the internet we knew that Apple’s iOS had great debugging tools and documentation, nevertheless we made the choice to go with the Android platform. There are several reasons why we felt that the Android platform would be more suited for our applications. Firstly, Google made the Android system open source, which meant that we could look up everything we wanted. Apple, on the other hand, has a very strict close source policy. Secondly, as an open source system Google attracted multiple manufactures who wanted to create smartphones for the system. This created competition among manufactures, reducing prices and reducing introduction time of new hardware. A huge disadvantage of Apple’s iOS is that Apple is the only producer of its hardware. The third advantage is that Android applications are programmed in Java, which was the main programming language taught at the university in the bachelor program. As these considerations obviously favor Android over iOS the decision was made to start programming for the Android platform.

The Android platform is based on the Linux kernel and is owned and maintained by Google Inc. Applications are written in Java and run in a customized Java virtual machine, called the Dalvik virtual machine. An application communicates through the Android SDK with underlying core libraries that are written in C. When developing applications the core libraries provide most of the useful functionality, such as a 3D Graphics (OpenGL ES) or a database system (SQLite). Applications can also (partly) be written in native-code languages such as C and C++. According to Google some applications might experience increased performance when using
native code [42]. At the start of this thesis Android version 2.0 was just released, however most phones still had OS version 1.5 or 1.6 installed on them. Android itself was also just starting to become popular in the Netherlands. Nowadays, Android is the number one platform on smartphones with worldwide sales of 36 million units in the first quarter of 2011 [43]. At this moment the newest version of the Android OS is 3.1, which is specialized for tablets. For smartphones the most current OS version is 2.3.4.

3.2 Hardware

We created the finger-tracking system on a G1 Android Developer Phone (also called the HTC Dream, in this thesis referred to as the “G1”), which has a Qualcomm MSM7210A processor running at 528 MHz and 192 MB RAM [44]. The camera had a 3-megapixel camera that was capable of recording video if Android OS version 1.5 or higher was installed. This device was the first Android device that was released to the market [45]. After a while the university bought a Motorola Milestone (also called Motorola Droid, in this thesis referred to as the “Milestone”), which featured an ARM Cortex A8 600 MHz processor with 256 MB RAM and a 5 megapixel camera [46]. This device proved to be much faster than the G1. Besides performance advantages the device also featured a large 3.7 inch screen with a 480 x 854 pixels resolution. This was better than the 320 x 480 pixels from the G1. At the beginning of 2011 I bought a new phone, called the HTC Desire HD (in this thesis referred to as the “Desire”). This device features a Qualcomm MSM8255 Snapdragon processor running at 1 GHZ with 768 MB RAM. The device had a 4.3 inch screen with a resolution of 480 x 800 and an 8 megapixel camera. Once again this device proved to be much faster than the previous described Motorola Milestone.

3.3 Finger tracking on smartphones

In this thesis we restricted ourselves to using the hardware that is integrated in the smartphone. This means that our finger tracking approach relied on images captured by the camera and image processing to retrieve the location of the fingers.

3.3.1 Retrieving camera images

Image retrieval on an Android powered device proved to be difficult. The Android platform does not directly support “real time” image collection. The camera’s frame buffers are locked and can’t be accessed in java or native code. The SDK provides a camera API, which supports a preview “call-back” method capable of returning images from the camera in real time. However the images were stored in large arrays and unfortunately it was not possible to work with these prior to Android version 2.1. Every time a “preview image” was taken by the Android system it would allocate a new Java array. When the arrays were no longer needed, the system would remove them from the memory with a built-in garbage collection system. The preview call-back in combination with the garbage collection resulted
Chapter 3. Finger tracking

in the Android system spending around 500 milliseconds every second to clean up the useless arrays. This would make any real time image processing unworkable on an Android device, however in Android 2.1 there was a work-around. Android 2.1 contained an undocumented method that was capable of creating the arrays, which store the preview images, in front. This allowed us to create a system that reuses the arrays once we were done with our image processing. This system had the advantage that it did not trigger the garbage collector and could therefore return images at the maximum speed the hardware allowed.

The images returned by the preview call-back were still in a raw format and needed to be converted to a format that could easily be processed and viewed in the Android OS. Decoding these images in Java proved to be infeasible, because the decoding process would take up multiple seconds. To speed up the decoding process we delved into native code on Android, which turned out to be much faster when working with large arrays such as the raw image data.

### 3.3.2 Decoding camera images

The goal of the decoding step was to transform the raw image data, which was a byte array containing YCbCr color space values, to an integer array containing colors in the RGB color space. We wanted to do this for two main reasons. The first reason was ease of use, the RGB values are stored in the integer array as a single value, rather than 3 (not successive) values in the YCbCr array. This layout makes the RGB values easier to analyze and manipulate than the YCbCr values. The other reason is that the Android Bitmap class, which handles image representation, only works with RGB values. Using this format was required, since it was the only way we could easily see if our tracking algorithm worked.

The YCbCr color space is formatted according to the YCbCr 4:2:0 standard (in Android called YCbCr.420_SP or NV21). The 4:2:0 image format contains a plane of 8 bit luma (Y) samples followed by an interleaved Cb/Cr plane containing 8 bit 2x2 subsampled color difference samples. The Y samples are all found first in the byte array and are represented by unsigned characters. The Y samples are followed by unsigned characters containing interleaved Cb and Cr samples. The amount of Cb and Cr samples are as large as the amount of Y samples \[47\]. The YCbCr samples can be converted to RGB values according to the following code snippet \[48\]:

```
R = Y + ((359 * Cr) >> 8);
if (R < 0) R = 0; if (R > 255) R = 255;
G = Y - ((88 * Cb + 183 * Cr) >> 8);
if (G < 0) G = 0; if (G > 255) G = 255;
B = Y + ((454 * Cb) >> 8);
if (B < 0) B = 0; if (B > 255) B = 255;
```
Depending on the scene, running background applications and many other factors we were able to decode around 18 frames per second on the G1.

### 3.3.3 Detecting and tracking the finger

Given a camera image, the application must be able to detect and track the finger. We decided to use bright colors, which were attached to the fingertip, to make the finger more distinctive from its surroundings. This decision had the advantage that the finger tracking algorithm had degraded from finger-detection and tracking to color detection and tracking, which is computationally more efficient.

#### 3.3.3.1 Color segmentation

In computer vision, the goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze [49]. In our case we wanted to change the input image into a data structure that represented pixels that exceeded a certain color value. If a pixel exceeded this color threshold, we assumed that it is part of the colored marker.

In the implemented application we wanted to track bright red, green and blue colors. We used a color filter to determine if the color value was high enough. Every color had a unique color id, which we later used to get more information about the structure of the colored marker in the image. The result of the color filter was stored in an integer array with the same dimensions as the input image. The color filter was implemented in the following way:

```plaintext
if((red - green) >= red_threshold and
   (red - blue) >= red_threshold)
{
   // set red color id
}
else if((green - red) >= green_threshold and
         (green - blue) >= green_threshold)
{
   // set green color id
}
else if((blue - red) >= blue_threshold and
         (blue - green) >= blue_threshold)
{
   // set blue color id
}
else
{
   // mark pixel as "uninteresting"
}
```
The method checks if the red, green or blue values are dominant enough to be considered part of a red, green or blue marker and if so, the according color id is set in the array. The method simplified the image to an array that only contained the information about the position and color of brightly colored pixels. In the best case scenario, the array only contained the information about the markers and no background noise.

The described method had several benefits: the implementation was easy, the method only required one pass over the image and the method is not resource intensive. The method also has two downsides: it is not illumination invariant and it is not resistant to background noise. The quality of the tracking largely depended on the thresholds which were set experimentally to a default value. To prevent noise or illumination-related problems we would adapt these thresholds before we began an experiment.

### 3.3.3.2 Marker detection

The segmentation algorithm only worked on single pixels and therefore we did not yet know anything about the structure or organization of the colored markers, therefore we created an algorithm that returned information about the position and structure of the “blobs” of color in the array. The algorithm is sometimes referred to as a “region grower” and is capable of clustering related pixels together in groups or regions.

The algorithm takes an initial seed pixel and examines its neighboring pixels to determine whether the seed and the neighbor belong together. If the seed and neighbor belong together, then the algorithm adds the neighbor to the region of the seed and adds the neighbor to a stack of pixels that still need to be processed. This only happens when a pixel has not yet been added to a region. After all the neighbors of the seed have been processed, the algorithm picks a new seed from the stack. If the stack is empty, there are no more pixels that can be added to this region. The algorithm will start searching for a new and not yet processed “interesting” pixel. If it finds one, then a new region is created and that pixel is chosen as initial seed. It does this until it reaches the end of the image, when this happens all interesting pixels that are connected are grouped into regions. The last step is returning the regions that now contain information about the position, bounding box and size of a colored marker. This data will be used for recognizing gestures and visualization. The pseudo-code of the algorithm is given below:
for (every pixel)
{
  if (pixel.color is red, green or blue and not processed)
  {
    // label the pixel
    pixel.processed = true;
    // create a new region
    Region r = new Region();
    // create Queue
    Queue todo = new Queue();
    todo.add (pixel);
    // find all the pixels that belong to this region
    while (todo is not empty)
    {
      for (all eight neighbors of pixel)
      {
        if (neighbor is not processed and
            pixel.color is equal to neighbor.color)
        {
          neighbor.processed = true;
          r.add (neighbor);
          todo.add (neighbor);
        }
      }
    }
  }
}

3.3.3.3 Marker tracking with search windows

The program was now capable of detecting colored markers in the image and returning information about the structure. In most cases the markers occupied a small section of the image, which meant that we did a lot of unnecessary processing, therefore we introduced search windows which allowed us to specify an area in which the algorithm must be performed.

Before we could use our search windows we needed to know which areas in the image were relevant to the application. In our application these were the areas that contained the colored markers, therefore we first needed to detect them before we could use this method. Detection is done by performing the combination of the color filter and region grower algorithm on the whole image. From this we got the position and structural information of all the markers in the image. In our final application this is done every two seconds to ensure that we also detect newly introduced markers.

Once we knew where the markers were we could start tracking them. For each marker we combined the previous location with the bounding box information,
which gave us an area in which the marker would be likely to appear in the next frame. The calculated area was supplied to the region grower, which would then only be performed on a specific part of the image. The region grower was implemented in such a way that it was possible to “grow” outside the borders of the search window. This had the advantage that we could detect the marker even if only one pixel of the marker was in the view of the search window. The new position and structural information from the region grower was used to update the information of the tracked marker. If all the markers were lost or no markers were found we would (re)start the detection phase.

3.3.3.4 The final algorithm

The final color tracking algorithm is a combination of 3 techniques:

- Color filter, used to determine if a pixel belongs to the marker.
- Region grower, used to retrieve structural information.
- Search windows, to reduce the amount of unnecessary work therefore boosting the performance.

The region growing technique was chosen, because it could easily be combined with the color filter and search window approach limiting the amount of overhead generated by the function calls. Another advantage was that the combination of color filter and region grower only required one traversal over the image to collect the required structural information. In this traversal the region grower would access pixels multiple times, however real calculations and assignments were only done on non-processed pixels.

The final speed of the algorithm was considered to be more than enough for the purpose of our finger-tracking system on the G1. The final speed of the algorithm was somewhere between 8 - 15 frames per second and was highly affected by the amount of brightly colored pixels that needed to be processed and combined into regions. For this reason we have no formal speed results. On the Milestone the speed climbed to more than 20 frames per second and on the Desire the application ran in real-time (around 26 frames per second).

3.4 Physical- and hardware-based limitations

Once we had our finger-tracking application we could start looking into the physical and hardware-based limitations of our envisioned AR systems. In our envisioned applications we assumed that the user held the mobile device in one hand and used the other free hand for interaction. The range the user could cover with his or her hand was important for the interaction design, since we did not want to create an interaction concept in unreachable space. Besides the reach of the human arm we also needed to consider the range of the camera, especially because this was our main input for the finger tracking. Both studies combined gave us more insight in the limitations of our AR applications.
Chapter 3. Finger tracking

3.4.1 The human arm

The human arm is one of the upper limbs on a human body. It consists of bones, joints, muscles, nerves and blood vessels. In anatomy the arm refers to the segment between the shoulder and the elbow, however in this thesis we will refer to the arm as the entire upper limb from shoulder to wrist. This is the common, but also historical definition of the arm. The upper arm contains one bone, the humerus. This bone joins the scapula at the shoulder (glenohumeral joint) and the ulna and radius bones in forearm. The shoulder is a ball-and-socket joint, which allows for movement of the arm in a wide circular plane. The elbow is a hinge joint between the end of the humerus and the beginning of the radius and ulna bones. The radius and ulna bones can rotate around each other, which allows for additional range of motion. The radius and ulna are connected to the elbow and the wrist (carpus). In turn the wrist connects to the metacarpus which connects the finger bones to the wrist.

All the mentioned body parts contribute to the movement range of the human arm. In this thesis we are interested in the reachable workspace of the human arm. This is defined as the volume within which all points can be reached by a chosen reference point on the wrist [50]. The end of the human arm is the beginning of the wrist, therefore if we know the reachable workspace we can get a clear understanding of the maximum reach of the human arm. We are not interested in the wrist or hand, since these will be used for finger interaction and can therefore not contribute to the bounding volume. Every position within this volume is a place where finger interaction can take place. It is impossible to reach outside the volume and therefore no interaction can take place there.

3.4.1.1 Ranges of motion

The aforementioned volume is influenced by two factors: the ranges of motion (ROM) of each joint and the length of the bones in the arm. In figure 3.1a we can see the layout of the human arm and its joints. The shoulder consists of an inner and an outer joint. The inner joint has two perpendicular rotations and one translation, while the outer joint has three perpendicular rotations. The elbow consists of a joint with two perpendicular rotations, however in the arm model we only have one, because the radioulnar joint (supination / pronation) does not influence the maximum reach and can therefore be removed from the model. We can further simplify the model by removing the inner shoulder joint. The inner shoulder joint can be used for precise calculations by biomechanics experts and physiotherapists [51] [52], however in our case a shoulder that is represented as a joint with three rotations is enough. This simplified model now coincides with the standard goniometric measurement technique to determine the extent of a shoulder injury in physiotherapy [53].

The physiotherapist can, for example, measure the ranges of movement of the different parts of the arm with the use of a (double-armed) goniometer. The given
Chapter 3. Finger tracking

Figure 3.1: The left image (a) gives an overview of the different joint in the human arm, while the right image (b) gives an overview of the different movements and corresponding names of the human arm.\footnote{Images from [51] and [52].}

angles are always related to the plane in which the movement works. The extension ($\phi_{Fmin}$) and flexion ($\phi_{Fmax}$) are measured in the sagittal plane, the elevation trough adduction ($\phi_{Amin}$) and abduction ($\phi_{Amax}$) in the coronal plane and the internal ($\phi_{Rmin}$) and external ($\phi_{Rmax}$) rotations in the transverse plane. The extension ($\phi_{EFmin}$) and flexion ($\phi_{EFmax}$) of the elbow are measured in the sagittal plane \cite{52}. Table 3.1 gives an overview of a ROM study performed on 109 males.

The angles in this table give us an indication of what can be considered natural ranges of movement. However we must note that these values can vary considerably among individuals as they are affected by age (visible in table 3.1), sex and possible injuries \cite{54}. An informal test with two male and two female participants gave enough conformation that the ROM values below are a good indication, since they did not differ extremely from table 3.1.

3.4.1.2 Arm lengths

Besides the ROM, the length of the arm segments also contributes to the maximum reach of the arm. The arm segments (the arm and the forearm) can be measured, but also computed from the anthropometric table \cite{56}. According to this table the arm segments are relative to a person’s height ($h$). The length of the humerus is $d_{\text{humerus}} = 0.185h$ and the forearm is $d_{\text{forearm}} = 0.146h$, therefore the total length of the arm

\footnote{Images from [51] and [52].}
Chapter 3. Finger tracking

### Table 3.1: The ranges of motion for different parts of the arm. The angles represent the mean ± one standard deviation [55].

<table>
<thead>
<tr>
<th>Joint</th>
<th>Angle (N = 109)</th>
<th>≤ 19 year (N = 53)</th>
<th>&gt; 19 year (N = 56)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shoulder joint</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extension</td>
<td>62.3° ± 9.5°</td>
<td>67.5° ± 8.0°</td>
<td>57.3° ± 8.1°</td>
</tr>
<tr>
<td>Flexion</td>
<td>166.7° ± 4.7°</td>
<td>168.4° ± 3.7°</td>
<td>165.0° ± 5.0°</td>
</tr>
<tr>
<td>Adduction</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Abduction</td>
<td>184.0° ± 7.0°</td>
<td>185.4° ± 3.6°</td>
<td>182.7° ± 9.0°</td>
</tr>
<tr>
<td>Internal rotation</td>
<td>68.8° ± 4.6°</td>
<td>70.5° ± 4.5°</td>
<td>67.1° ± 4.1°</td>
</tr>
<tr>
<td>External rotation</td>
<td>103.7° ± 8.5°</td>
<td>108.0° ± 7.2°</td>
<td>99.6° ± 7.6°</td>
</tr>
<tr>
<td><strong>Elbow joint</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extension</td>
<td>0.6° ± 3.1°</td>
<td>0.8° ± 3.5°</td>
<td>0.3° ± 2.7°</td>
</tr>
<tr>
<td>Flexion</td>
<td>142.9° ± 5.6°</td>
<td>145.4° ± 5.3°</td>
<td>140.5° ± 4.9°</td>
</tr>
</tbody>
</table>

is $d_{\text{arm}} = 0.331h$ ($d_{\text{arm}} = d_{\text{humerus}} + d_{\text{forearm}}$). The resulting arm length is an approximation since many factor influence the growth of the arm. According to the “Centraal Bureau voor de Statistiek”, the average male length in the Netherlands is 1.803 meters [7]. This means that the average length of the male arm in the Netherlands is 0.5968 m ($0.331 \times 1.803$). The average female length is 1.674 m, which means that the average arm length is 0.5541 m ($0.331 \times 1.674$).

### 3.4.1.3 Reachable workspace

The ROM values combined with the arm lengths can be used in computer simulations to get a clear image of the reachable workspace of the arm. The computer simulation can, for example, use a kinematic model of the arm and joints to iterate over the angles to get the needed workspace information [51] [52]. Figure 3.2 gives a three dimensional insight into the reachable workspace / volume that we can use for our finger interaction concepts.

As mentioned, this thesis will contain two different experiments. In the first experiment we will be testing a setup in which the user has to stand upright and interact with the system. In this experiment the workspace will be equal to the workspace sketched above and in figure 3.2. In the second experiment the participants will sit behind a table and interact with the system. This will change the workspace slightly, since the table blocks parts of the reachable space. We assume that the ranges of motion do not change, which implies that the only difference between the two experiments is a blockage of the reachable workspace on the z-axis. This cutoff will be different for each table and chair combination, therefore we do not further specify this blockage of the z-axis.
3.4.2 Limitations of the hardware

For our experiments we used 2 mobile devices, the Motorola Milestone and the HTC Desire HD. The G1 was only used for creating the finger-tracking system. The Milestone and Desire were, on the date of purchase, part of the high-end segment of the mobile smartphone market. This implies that the cameras on the devices are considered to be of high quality. The Milestone had a 5 megapixel camera and the Desire had an 8 megapixel camera, however the resolutions for video capturing are always much lower to get an acceptable frame rate. For our interaction concepts it is important to identify the hardware capabilities and limitations. We have identified two points of interest:

- The viewing angle, which limits our interaction concepts in the horizontal and vertical plane.
- The resolution, which determines at which distance finger interaction becomes feasible.

3.4.2.1 Viewing angle

To get an idea of the ranges we can cover with the cameras on both devices, we did an informal experiment in which we calculated the viewing angle in the horizontal and vertical plane. These angles are important since they restrict the area in which we can use our finger interaction concepts.

In the experiment (figure 3.3) the devices were placed in a holder perpendicular to the ground. The holder was placed at a certain distance ($d_{\text{camera}}$) from a wall which

---

1. Images from [51].
was also perpendicular to the ground. The mobile phone was placed in landscape view, since all our envisioned AR applications make use of this view. The aim of the experiment was to find the outermost points that the camera can still see in the horizontal and vertical plane. The height and width, retrieved from these points, combined with the distance to the wall gives us a good indication of the angles in both planes. The angles can be calculated with the following formulas:

\[
\alpha = 2 \times \arctan \left( \frac{\text{width}}{2d} \right)
\]

\[
\beta = 2 \times \arctan \left( \frac{\text{height}}{2d} \right)
\]

Table 3.2 contains the results of the experiment. When we average the values from the small experiment, we can conclude that the Desire has a horizontal viewing angle of 45 degrees and a vertical viewing angle of 35 degrees. The Milestone has a horizontal viewing angle of 55 degrees and a vertical viewing angle of 42 degrees.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>d\text{camera}</th>
<th>Width</th>
<th>Height</th>
<th>(\alpha)</th>
<th>(\beta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desire</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.3 m</td>
<td>0.25 m</td>
<td>0.19 m</td>
<td>45.24</td>
<td>35.14</td>
</tr>
<tr>
<td>2</td>
<td>0.5 m</td>
<td>0.40 m</td>
<td>0.31 m</td>
<td>44.59</td>
<td>34.45</td>
</tr>
<tr>
<td>Milestone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.3 m</td>
<td>0.32 m</td>
<td>0.24 m</td>
<td>56.14</td>
<td>43.60</td>
</tr>
<tr>
<td>2</td>
<td>0.5 m</td>
<td>0.51 m</td>
<td>0.38 m</td>
<td>54.04</td>
<td>41.61</td>
</tr>
</tbody>
</table>

Table 3.2: The results from the viewing angle experiment on the Desire and Milestone. The d\text{camera}, width and height columns are in metres.

1. Images created by Rebecca Grootelaar.
3.4.2.2 Resolution

The resolution of the camera determines in large parts the possibilities for interaction. The interaction methods will all depend on position shifts (in pixels) of the tracked finger(s). From this shift we can determine that a finger is moved, how fast it has moved and in which direction. A high resolution means that the image contains more pixels and therefore our finger tracking method can become more accurate. The downside is that a higher resolution means more processing time for the tracking and visualization methods. From experience we had learned that a higher amount of frames per second tends to make the interaction concepts more stable / robust and therefore easier to use. We determined that a resolution of \(320 \times 240\) pixels was the best balance between resolution and performance. This resolution might seem low but interaction take place just behind the mobile device, therefore a high resolution is not directly necessary.

3.4.2.3 Combining the viewing angles with the resolution

Now that we know the viewing angles and the resolution we can discuss the area in which finger tracking is feasible based on the hardware. As mentioned, the interaction concepts will depend on the position of the marker in pixels. When interacting with the camera, the fingers will have a certain distance to the camera. This distance is critical for the performance of the finger tracking algorithm. The combination of viewing angles and distance to the camera will create a plane in which the interaction will take place. The camera resolution is fixed, which implies that the same amount of pixels is mapped onto the different areas of the planes. This means that if we have a large plane, the finger has to move further to get the same shift in pixels as in a smaller plane. This can either improve or degrade tracking performance. We assume a motion in which we move away from the lens and therefore increase the area of the plane. At the beginning of this motion the area of the plane will be very small, therefore small movements of the finger will become large shifts in the position of the finger for the finger tracking system. The distance from the camera is too small to make it useable for tracking. If we continue the motion and increase the distance between it and the camera, the finger tracking will become more robust and reliable, however if we move too far away the tracking quality starts to degrade again, because the finger has become too small to accurately track it. This means that there is a certain zone in which tracking performs the best. An informal experiment, in which we measured the distance between the camera and the fingers, was performed to identify the transition areas and corresponding zones. Table 3.3 contains the zones and their corresponding usefulness.

Based on table 3.3 we can conclude that we can expect the best performance in the area with a distance ranging from 10 cm to 42 cm from the camera of the mobile device. For our interaction design it is important to know how far the user has to move their finger to change one pixel in the image. The formula below can be used to calculate this “movement versus pixel” value. In the formula, the angle value
Table 3.3: The results from the experiment in which the goal was to determine which zones are (un)usable for finger interaction. The beginning of zone and end of zone columns contain the distances from the finger to the camera in metres.

<table>
<thead>
<tr>
<th>Zone number</th>
<th>Beginning of zone</th>
<th>End of zone</th>
<th>Usable / Unusable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 m</td>
<td>0.075 m</td>
<td>Unusable</td>
</tr>
<tr>
<td>2</td>
<td>0.075 m</td>
<td>0.100 m</td>
<td>Transition to usable</td>
</tr>
<tr>
<td>3</td>
<td>0.100 m</td>
<td>0.420 m</td>
<td>Usable</td>
</tr>
<tr>
<td>4</td>
<td>0.42 m</td>
<td>0.555 m</td>
<td>Transition to unusable</td>
</tr>
<tr>
<td>5</td>
<td>0.555 m</td>
<td>∞</td>
<td>Unusable</td>
</tr>
</tbody>
</table>

refers to the viewing angle of the camera in the horizontal or vertical plane. The resolution value refers to the resolution in that plane.

\[
\text{pixel movement} = \frac{(\sin \left( \frac{\text{angle}}{2} \right) \times \text{distance to camera}) \times 2}{\text{resolution}}
\]

When we use the Desire smartphone and the distance to the camera is 10 cm, then a shift of one pixels maps to a horizontal finger movement of 0.24 mm \(((\sin(45 / 2) \times 0.10) \times 2) / 320\) and a vertical finger movement of 0.25 mm \(((\sin(35 / 2) \times 0.10) \times 2) / 240\). In the 42 cm case a shift of one pixel maps to a horizontal finger movement of 1.34 mm \(((\sin(45 / 2) \times 0.42) \times 2) / 320\) and a vertical finger movement of 1.05 mm \(((\sin(35 / 2) \times 0.42) \times 2) / 240\). These values illustrate the difference between finger movement just in front and far away from the camera. The values are used for implementing and designing finger-based interactions. Note that the values are linked to the mobile device.

3.4.3 Combining the physical- and hardware-based limitations

In previous sections we have studied the inner workings of the human arm and the hardware on which our AR application will run. Both sections described the possibilities and the restrictions that they impose on the AR application. When creating an AR application, and thus combining both chapters, some additional limitations arise. As mentioned the device is held in one hand and the user interacts with the other free hand. The visual feedback from the AR application is displayed on the screen, which means that the user always has to look at the screen to see if he or she is interacting with the virtual objects. This means that the eyes of the user have to be focused on the screen at all times. The “least distance of distinct vision” (LDDV) is a term used by optometrists to describe the closest distance in which a person with normal vision, called 20 / 20 vision, can comfortably look at something. The LDDV of adults with normal vision is around 25 centimeters [57]. This value changes when people become older or have eye related health problems or injuries. We adopt the LDDV distance as the average distance that a user will hold the device, since people will naturally go for a comfortable position. This
position has consequences for the previously explained workspace (figure 3.2), since we can remove the part that is too close to the body. We can also remove the part in which the hand is behind the body, since we cannot see the visual feedback from the device there. After these removals, the workspace can be described as a sphere with the radius equal to the arm length positioned at shoulder joint minus a sphere with a radius of 25 centimeter positioned at the center of the eyes, whereby every point of the spheres behind the coronal plane is removed. All the points that lie in the volume described by the definition are part of set A.

With the information from table 3.2 we can also describe the volume in which the finger interaction will perform best. The best performing space can be described as a sphere with the radius of 67 centimeter (LDDV plus the end of the good performance zone) positioned at the center of the eyes minus a sphere with a radius of 35 centimeter (LDDV plus the beginning of the good performance zone) positioned at the center of the eyes, whereby every point of the spheres behind the coronal plane is removed. The “beginning of the good performance zone” can be included here, because the eyes and the device are always aligned, since the user can only use the finger-based concepts if he looks at the device. All the points that lie in the volume described by the definition are part of set B.

When we calculate the intersection of set A and set B ($A \cap B$) we end up with a description of the volume in which our interaction concepts will perform best. This volume has been kept in mind when designing and testing the interactions further described in this thesis and should be seen as a guideline rather than a strict volume in which the concepts must perform.
In this experiment we wanted to compare two interaction concepts against known touchscreen interaction in an AR setting. The first concept was the described finger tracking approach. The other interaction concept depended on the movement of the device. It utilized the accelerometer and compass sensors to determine how the device is rotated or tilted. Known AR applications such as Layar and Wikitude also utilize these sensors to display and update the world according to where the user is currently pointing. In our case we also used this information to interact with the AR. We have named this concept “Device” interaction, since it reacts to the movement of the device.

To create a better comparison we decided to use only 2D finger tracking, since the touchscreen per default only returns 2D information. The device concept returned values that gave the orientation and tilt of the device, which together can also be seen as a form of 2D information. The colored marker used in this test was a bright (almost fluorescent) green dot. In this study we restricted ourselves to the three tasks of selecting virtual objects, selecting entries from context menus and translation of 3D objects into 2D. These tasks were chosen because they represent a “natural flow” for translation in our envisioned application. If we wanted to translate an object from point A to B, we would first have to select the object, then select the translation operation from a menu and then translate the object to point B. A pie menu style was chosen because it is proven to be more effective for pen and finger-based interaction compared to standard list style menus [58]. A pie menu also takes advantage of shorter access times (Fitts’ Law) and better orientation, since all the menu elements are equally far away from the center.

4.1 Touchscreen interaction concept

For the touchscreen-based interaction the three tasks were implemented in the following way:

- Selection and menu selection was done by pressing on the screen. This conforms to normal touchscreen use and is also implemented in Layar and Wikitude.

- Translation was done by dragging an object from place A to place B. This is also a common way of relocating an object on touchscreen devices.

The interaction flow will be as follows: the user selects an object by pressing somewhere in the bounding box of the object on the touchscreen. This selection evokes
Chapter 4. First experiment: Feasibility study

a context menu with three options from which the user must select the translation mode. Translation is done by pressing on the object and then moving the finger to a certain position on the screen. The object follows the finger until the user releases the touchscreen. The objects in the AR are all related to the real world, which means that if the device is moved, the box stays in the position that is related to the real world. The real world is observed by the camera and viewed live by the user. The user can stop translating by pressing on a stop button in the lower left-hand corner of the screen. This gives the object its final position in the real world.

As said, this concept conforms to standard touchscreen interaction and therefore we believe that this concept will be intuitive to most people. In terms of accuracy we can expect the concept to be reliable and accurate for menu selection, since we are in full control of the interface design. We expect that the accuracy of normal selection is good, but that problems may arise when objects are small, placed close to each other or are (partially) overlapping because they are placed behind each other in the 3D world. There are two potential problems that can reduce the accuracy when interacting with small or closely positioned objects: the user's finger could be blocking the screen and therefore the user can't exactly see where he is clicking, or the finger can be too thick to select these small / close objects. We expect the accuracy for translation to be good but not perfect, since the user cannot precisely see where he is placing the object, because the finger is in the way. However, the user can easily make small adjustments in the placement, which would increase the accuracy.

4.2 Device interaction concept

As mentioned, the device concept depends on the sensors built into the mobile device. We utilized the accelerometer and the compass to change the visual projection of the scene, e.g. if the user rotates around his own axis the visual projection will also rotate and might therefore show completely new objects. The device concept makes use of the change in visual representation to create a new form of interaction. For the three tasks this will result in the following interaction concepts:

- Selection and menu selection happens when the user places a virtual object in the center of the screen. If the object is in the center a progress bar starts filling up. Once filled the object or menu item is selected. This means that a user is able to select a certain object by tilting the device and turning around his own axis.

- In translation mode an object will stick to the center of the screen, which means that the user can relocate the object by tilting and turning the device. An object is placed at its new location by pressing anywhere on the touchscreen.

To make the concept more intuitive we placed a reticule in the center of the screen to display where the interaction took place. The progress bar would take 1.25 seconds
to fill up. This value was selected based on our own experiences with the concept.

Our expectation is that this concept is slower than the touchscreen-based interaction. This is not only because it takes a while to fill the progress bar, but also because the user has to maneuver the device to make sure the reticule rests on the object he wants to select. In the touchscreen case this is not necessary, since the user directly presses on the object he wants to select. In terms of accuracy we expect that this concept works slightly better for selection, since the user has more control over the position of the reticule than he does with the placement of the finger. We expect that small or overlapping objects can be selected more accurate. Selection requires that the user holds the device stable for 1.25 seconds which might be critical, since the user is required to hold the device in mid-air without any support. We expect translation to be intuitive and easy to control. Compared to the touchscreen it might perform better, since the user only has to move the device rather than move the device and also drag the object to the new location. The accuracy might be lower than in the touchscreen case, since releasing the object (by pressing the touchscreen) might introduce small changes in the position of the device and therefore reduce accuracy. These errors are not introduced in the touchscreen case, since the object is already placed on its final location. The user only confirms this location by pressing the stop button.

4.3 Finger-based interaction concept

For the finger-based approach we made use of the colored marker tracking algorithm explained in section 3.3.3. Interaction for the three tasks was implemented in the following way:

- Selection and menu selection happens when the finger “touches” the virtual object, i.e. when the bounding box of the colored marker collides with the bounding box of the virtual object. When the bounding boxes collide, a progress bar starts filling up. This will take 1.25 seconds and once filled the object or menu item is selected.

- In translation mode an object can be moved by pushing the finger against the object. This means that if you push from the left, the box moves to the right and vice versa. An object is placed at its new location by pressing anywhere on the touchscreen.

We expect that this concept takes more time than the touchscreen-based concept, since it takes a while to fill the progress bar, but also because people need time to move their finger to the object they want to select. The expected time for translation will be equal or slower than the touchscreen-based concept. Just like in the touchscreen case a user has to perform two operations: he must tilt / turn the device and use the finger to push the box in the right direction. We expect the accuracy to be comparable to the touchscreen approach, since a user can make small changes to the object’s final position, which is also possible with the touchscreen approach.
Gesture-based interaction can be a very powerful way for human-computer interaction, however in our mobile finger tracking application there are many shortcomings and potential problems. The most important one for our interaction tasks is the range the camera covers. As described in section 3.4.2.3 there is an “optimal” range distance from the camera. If the finger is held closer or farther away from the camera, the tracking quality starts to decrease. Another potential problem arises when we want to move a box by pushing it from the side. This might turn out to be difficult depending on which hand is used, e.g. pushing it from the right using the left hand. This action might result in an awkward hand position or even force people to switch the hand in which they hold the device.

4.4 User study

The main goal of the user study was to evaluate the interaction concepts described above. We used 18 participants of which 12 were male and 6 female, 5 participants at ages 15-20, 8 at ages 21-30, 1 at ages 31-40, 1 at ages 41-50 and 3 at ages 51-52. For the finger-based concepts we asked participants to place the marker on their nail or on the bottom of the tip of their index finger. Only one participant decided to place the marker on his nail. Prior to the experiment we asked participants to choose the hand in which they would hold the device during the experiment. Switching hands was not allowed. Eleven participants chose to hold the device in their right hand and seven participants chose their left hand.

We decided to use a within-group user study in which each participant tested each interface concept and task. The interfaces were presented in a different order to eliminate possible learning effects. The tasks were presented in a fixed order: first the object selection, then menu selection and lastly translation. This order was chosen because this would also be the natural order in a real usage case. We created an introductory test for each task, in which the supervisor of the experiment could demonstrate the interaction style and goal of the experiment. After the introductory test the participant could try the interaction style in one practice test. After this test the real tests would start. For the object and menu selection there were three tests, while the translation experiments contained four tests. For the translation task we decided to create two tests that required the participant to translate the object to the right and two tests that required the participant to translate the object to the left. In this way we could guarantee that participants would not be influenced by the hand that they used to hold the device. Participants were told that the first two tests were not part of the experiment and that the tests needed to be completed as fast and accurate as possible. The object selection tests could be rated as easy (in which three objects were separated from each other), medium (in which one object was separated from two partially overlapping objects) and hard (in which all three objects were overlapping). In each menu selection test the participants were asked to select either the up, right or bottom menu item. In the translation tests the participants were asked to translate an object to an indicated target position. In two of the four tests the target was placed to the left at an angle of 35 and 130
degrees. In the other two tests the target was positioned 35 and 130 degrees to the
right. In the AR application the view of the camera was set to 72.5 degrees, which
meant that the goal in the 35 degree tests was directly in view. In the 130 degree
tests the participant had to rotate around their own axis to reach the target since it
was not directly in view. At the start of a translation test the participant were told
in which direction to translate the object. The order of the tests was randomized
for each participant to avoid order-related influences on the results.

During the experiment we logged the following information:

- Time: the time it took the participant to complete a test.
- Accuracy: whether or not they were successful in fulfilling the task. For
  translation we also logged the distance to the target.
- Sensor data: the data generated by the compass and accelerometer.
- Tracking data: the data generated by our finger tracking algorithm.

Mobile computing is often used for entertainment and leisure applications, therefore
we asked the participants to fill in a questionnaire in which they could rank the
concepts based on accuracy and performance, but also on aspects such as fun and
engagement. Remarks, comments and other feedback of the participant during and
after the experiment were written down and taken into account during the analysis
of the questionnaire.

4.5 Results

4.5.1 Average times

For our AR application it is important that users can perform tasks fast and ac-
curate. Figure 4.1 shows how long the participants took to complete the tests,
averaged over all participants for each task. We can clearly see that touchscreen in-
teraction is the fastest for object selection and menu selection. This conforms to the
expectation we described in the previous section. The device and finger concepts
take longer for selection, but still seem to be in a reasonable range. For translation
it is clear that the device interaction is much faster than the others.

From our tests we can conclude that there is no difference between participants who
held the device in their left or right hand.
4.5.2 Object selection

Figure 4.2 gives an overview of the average times for object selection per level of difficulty. We can see that again touchscreen interaction is the fastest concept. The finger-based concept seems to be slightly faster than the device concept, although in the hard level the concept is clearly much slower. From the data we can see that this is because participants need time to adjust the marker to the correct size for selection. A large marker makes it impossible to select the correct object if the objects are close together, therefore people try to decrease the marker size by moving their arm backwards. To do this and reorientate on the scene, will take some time and explains the increase in time.

The participants performed the tasks in the easy and medium level almost flawlessly with only one (probably) accidental mistake in the easy level with the touchscreen.
However, in the hard level we started to see mistakes in all the interaction concepts. The touchscreen concept performed worst with 5 errors in 17 tests (one test was not recorded) compared to 2 errors in 18 tests for the finger-based concept. The extra time for the finger tracking concept pays out with an increase in accuracy. The lower accuracy of the touchscreen concept conforms to our expectation that overlapping objects are critical for the touchscreen.

Due to an unforeseen design error, the device concept had an unfair disadvantage and therefore we can’t compare the success/error results to the touchscreen or finger-based concept. If a participant was unintentionally holding the device on an object at the start of a test, the progress bar would immediately start filling which in some cases resulted in an accidental error if the participant was not fast enough to remove the reticule from the object. Six participants in total had this error. When we remove these accidents from the dataset we can see that the average time increases from 10,618 second to 14,741 seconds. This is still in line with the time for the tests done at easy and medium difficulty. The number of errors would decrease to 0, which means that all the errors were made in tests with the initialization problem. From this we can conclude that holding the device in a particular position over a longer period of time does not result in accuracy problems.

4.5.3 Menu selection

In figure 4.3 we can see the average times for the different menu selection levels. We see that for all three interaction concepts it did not matter what the position of the menu item was. This was to be expected since we used a pie menu, in which all the items have equal distance from the center.

![Average times of the menu selection tests](image)

**Figure 4.3:** Average time in milliseconds it took participants to complete the menu selection tests.

The device concept performs worst with 3 errors over all 54 tests (18 participants times 3 tests per participant). The finger-based approach contained one error and the touchscreen concept had no errors. This result is pleasing, since almost all tests were done correctly.
4.5.4 Translation

In figure 4.4 we can see an expected increase in time when the participant must move outside the initial view (130 degree tests). We can clearly see that the device-based concept is the fastest. This can be explained by the aforementioned number of operations, since, compared to the other concepts, the device concept only requires one operation, i.e. relocate the device so that the object is in the correct position. In the other two concepts the participant must not only move the device, but also the object through touchscreen or finger gestures.

![Average times of the translation tests](image)

In figure 4.5 we can see the (3D) Euclidean distance from the final position of the object to the target. An Euclidean distance smaller or equal to 2.0 is considered to be on the target, however a smaller distance is more accurate. In figure 4.5 we can see that the finger-based concept is very inaccurate compared to the other two. A look into the data reveals that the high distance value is mostly caused by people prematurely pressing the touchscreen and therefore placing the object far away from the target position. The touchscreen and device accuracy are comparable. The device concept performs less accurate in the 35 degree tests, while it outperforms the touchscreen concept in the 130 degree test. The data revealed no explanation for this, however it could be that during the touchscreen interaction people lost motivation to accurately place the object, because they had already spent around 40 seconds to get to the target. As for translation, we can say that the device concept is most promising, since it is the fastest approach and also has a good accuracy.
4.5.5 Questionnaire

In the questionnaire the participants were asked to rank the interaction styles used in the experiment. In table 4.1 we see how the participants ranked the interaction styles for each task, based on performance.

<table>
<thead>
<tr>
<th></th>
<th>Object selection</th>
<th>Menu selection</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>D</td>
<td>F</td>
</tr>
<tr>
<td>Ranked 1st</td>
<td>18</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ranked 2nd</td>
<td>-</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>Ranked 3rd</td>
<td>-</td>
<td>8</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 4.1: Ranking of the interaction styles based on performance.

We can see that the data from the experiment conforms to what participants experienced. All participants ranked the touchscreen-based interaction best for both selection tasks. In the translation case the device interaction concept scored best with 15 participants ranking it first. Surprisingly, the finger-based concept with its low performance was picked as second best 8, 7 and 9 times.

In the second part of the questionnaire the participants were asked about the overall performance (which concept is the best for all tasks), fun and both. The results are shown in table 4.2.

<table>
<thead>
<tr>
<th></th>
<th>Performance</th>
<th>Fun</th>
<th>Performance and fun</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>D</td>
<td>F</td>
</tr>
<tr>
<td>Ranked 1st</td>
<td>11</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>Ranked 2nd</td>
<td>3</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Ranked 3rd</td>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 4.2: Ranking of the interaction styles based on performance, fun and both.
In table 4.2 we can see that the touchscreen-based concept was considered the best concept based on performance, but the worst concept based on enjoyment. The finger-based concept was considered the most fun concept to use by 13 participants. This already became clear during the experiments, since we frequently heard comments like “fun” or “cool”. Nevertheless the handling of the finger-based concept was also criticized as being “challenging”. When we asked to rate both fun and performance we can see that the touchscreen and finger-based concept were chosen 8 and 7 times respectively. This indicates that there is no clear winner, however most participants commented that it depends on the application. They found the finger-based interaction style more suitable for games or other type of leisure applications, while the touchscreen interaction style was preferred for AR and more serious applications.

4.6 Discussion

In this experiment we have seen that a combination of touchscreen interaction with device input would suit best. For the selection tasks the touchscreen proved to be the best concept, generating the best results in the shortest amount of time. Touchscreen interaction was also rated by the participants as the best concept for the selection tasks. The device concept showed promising results for the translation tasks, in which it achieved the best results in the shortest amount of time. This concept was also rated best for translation by the participants.

Finger-based interaction turned out to have the worst performance. Besides performance we also identified two other problems during the experiments:

The first problem was identified based on the participant’s comments. We denoted that some people felt uncomfortable holding the device horizontal for a long period of time. The described feeling was unexpected, but not surprising since the human arm is normally not held in an unsupported horizontal position for a long period of time (the test took 15 to 20 minutes without the questionnaire). When the arm is held in such a position it might become fatigued and painful. The discomfort might be caused by poor posture and/or the limited amount of physical exercise we do with the muscles that are needed to hold the device in a horizontal position [59]. This problem can be compared with the touchscreens “gorilla arm” and should in any case be seen as a general problem for these kind of AR applications on mobile devices.

The second problem was observed during the experiments. As explained in section 3.4.2 there is a range in which our finger-based interaction concepts are likely to perform best. The minimum value of this range is 10 centimetres, however during the experiment we observed participants who held the finger much closer to the camera. As a result, the finger was interpreted as very large, which made selection and translation tasks much harder if not undoable. Some participants figured out that holding the finger at a larger distance from the camera gave more control, yet the majority kept trying to complete their tasks with a close distance to the camera.
Based on the results we can conclude that finger-based interaction is not suited for serious AR applications, because it lacks the needed performance. Nevertheless the participants were enthusiastic about the finger-based interaction. This enthusiasm can be seen in the ratings, where finger-based interaction was highly rated in terms of fun, engagement and entertainment. In leisure and other entertainment applications performance might not be as critical as for serious applications, since entertainment value is more important. For example the PlayStation Move and Microsoft Kinect do not per default make games easier to operate, but they are a huge success because of their entertainment value. This observation combined with the questionnaire made us realise that finger-based interaction might be more suited for entertainment applications.

Based on these reasons we created a second experiment that combined a tabletop AR with a game-like feeling. A tabletop AR will augment virtual objects on top of a marker that is placed on a table. This creates a strong physical connection, since it looks like the virtual objects are positioned on the table. A tabletop AR is chosen because it can potentially reduce the two discovered finger-based interaction problems. The ergonomic problem might be reduced, because the participants are able to rest their arms on the table, which should reduce the fatigued feeling. The distance problem might be reduced as well, because the physical connection might persuade people to interact closer to the table. This increases the distance between the finger and camera, which should increase the ease of operation and possible the accuracy. The next chapter describes the implementation of the tabletop AR and the new interaction concepts we introduced for the second experiment.
Chapter 5

Revised augmented reality and finger tracking

In this chapter we describe the adjustments we made to the finger-tracking application to make sure it looked and felt more like an entertainment application. We added a pose estimation and tracking algorithm to the AR, so it was capable of creating the illusion that the virtual objects were placed on the table. We also created new finger-based interaction concepts for use in the second experiment.

5.1 Game board

The tabletop augmented reality requires an additional marker tracking system. This marker tracking system will be used to align the virtual object with the table, which creates the illusion that the objects are placed on the table. We decided that the visualisation should look like a board game from real life. We chose a game board setting because we believed our interaction concepts could nicely line up with real life board game interaction concepts. For example in a real board game, translation occurs by picking up the object and placing it on a new position. These kinds of concepts can be mapped to the finger-based gestures, which make our finger-based concepts work and look natural in a board game setting. A board game is also directly recognized as an entertainment or leisure application, which is what we are aiming for with overall feeling of the second experiment.

The marker required by the new marker tracking system was always placed on the table. This implies that our interaction concepts would always take place between the table/marker and mobile phone. This opens the possibility that the hand could be blocking a part of the marker, making it hard for the tracking system to align both worlds. If the marker would be lost during the experiment, the visual representation would not line up with the finger tracking data, which would render our interaction concepts useless. As a result, the quality of the marker tracking system is critical to the performance of the complete AR system.

During the development of the tabletop AR, we looked at multiple open source marker tracking algorithms. We decided that extending an already implemented and optimized marker tracking system would suit better in the overall objective of this thesis. Implementing such a system from scratch would have no additional research benefits, since it has already been done multiple times. There were multiple systems for Android that could be considered as a good option. The best known systems are the NyARToolkit 60 and AndAR 61, which are both based on the
well-known ARToolKit. Both systems provide the developer with a framework around the frequently used square matrix marker tracking algorithm. We decided to do a comparative study to see which tracking system performed best. In this study we looked at the frames per second, the tracking quality and the extensibility of the systems. The AndiAR framework proved to be better than the NyARToolkit, because it was faster and thus performed more smoothly. However, in October 2010 Qualcomm, a wireless telecommunications research and development company, launched their own AR SDK for Android [62]. The SDK contained the same kind of functionality as the other toolkits, but also added a natural feature tracker. The performance of this SDK is unmatched by the other frameworks and consequently we decided to implement our finger tracking framework in the Qualcomm AR SDK.

We decided to use the natural feature tracking system above the standard algorithm for the following reasons:

1. The natural feature tracker was capable of tracking all kinds of images, hence we could take any image (e.g. a photo) and turn it into the marker that could be used for tracking.

2. The natural feature tracker worked by searching for points with high contrast in the camera image. If the points, found in the camera image, fitted a predefined target / marker, the system could calculate the needed transformation matrices. This meant that if we created a marker with lots of these high contrast points, the tracking became more stable. If we also distributed the points throughout the whole image, the marker could still be found even if there was partial occlusion. This is a great advantage over the normal tracking algorithms, which lose their ability to track the marker if the thick black border is occluded.

The natural feature tracker had an additional benefit that we could use any image as marker. This means we could make the marker itself look like a game board, which further improves our game board setup. We decided to make the marker look like “Ludo” (in Dutch “Mens erger je niet”). This game is well known in Europe and consequently all participants should be able to relate to it. The Ludo game is normally played on a board, which makes the board an ideal target for the natural feature tracker. In figure 5.1a, we see the Ludo board used in the experiment. In Figure 5.1b, we see the distribution of the high contrast points in the created board. The Ludo board was created in such a way that it had lots of high contrast points throughout the image. The Qualcomm “My trackables” application gave the image a 5 out of 5 stars, ensuring us that the created board had great tracking capabilities.

5.2 Interaction styles

Besides the new tracking algorithm, we also needed to come up with new (and improved) interaction concepts for the second experiment. To align the first and second experiment, we decided to test the atomic operations: translation, scaling
and rotation on the board and scaling and rotation in mid-air. If we would have selected a more specialized operation, such as pointing, we would limit our research to select application groups that can utilize the pointing gesture. With the high-level operations: translation, scaling and rotation we address a wider field, because these operations are the basics of many applications. We decided we would only compare finger-based concepts in the second experiment. Using this approach, we hope to gain more insight in the feasibility of the finger-based concepts, rather than more insight about the performance compared to other methods. We also decided to create two interaction concepts for each atomic operation, so we could better conclude on the feasibility of an atomic operation. For example if we had chosen one interaction concept and that concept contained an unseen design error or other problem, we could redo the entire experiment, since we could not conclude on the feasibility of that operation, which would make our conclusions weaker. In the process of creating the two interaction concepts, many other concepts were discarded due to complexity or occurrence of ambiguities. The interaction styles that seemed most promising based on intuitiveness and ease of use were implemented for use in the second experiment. For all the finger-based concepts we used a red and/or green marker. The green marker was placed around the thumb and the red marker was placed around the index finger.

### 5.2.1 Translation - Pick up and release

This translation interaction concept is based on the natural movement of the hand while it moves a pawn, on a game board, to a certain position. It consists of three acts, i.e. picking up the pawn, moving it to a new location and releasing it. This behavior is implemented in the following way: If the bounding box of the red and green markers is close enough to the object, then the object is picked up. When
moving around, the object will stick to center between the red and green bounding box. If the user moves the bounding boxes away from their common center, i.e. move the fingers away from each other, then the object is released and fixed to its new position. This interaction concept is named “Pick up and release” or in short puar.

5.2.2 Translation - Pushing

This translation interaction concept is based on the “pushing” concept from the first experiment. The user is able to move a virtual object by pushing with the finger against the object. Internally this works by pushing the red bounding box against the bounding box of the virtual object. When pushed the object will be relocated in the direction it is pushed. The green marker will be ignored by the system. This means that this interaction concept only works with the index finger.

5.2.3 Scaling - Gesture-based

This scaling interaction concept is based on the “pinch-to-zoom” concept that is extensively used in touchscreen applications. It consists of two gestures, one for enlarging the object and one for shrinking the object. Shrinking works by pushing the outer edges of the object inwards, i.e. pushing with the bounding box of the red and green marker against the bounding box of the object. The enlarging process works in two steps. First the user must place the markers against each other outside the object, i.e. place the bounding boxes of the red and green marker against each other and then move inwards. The object can now be enlarged by increasing the distance between the fingers. Switching between the two gestures can be done by releasing the object, i.e. making sure that the bounding boxes of the markers are not touching the object or by removing the markers completely from the screen.

5.2.4 Scaling - Touchscreen

This scaling concept uses only one gesture for shrinking or enlarging an object. An object can be shrunk or enlarged by decreasing or increasing the distance between the red and green marker. This means that this interaction concept combines the two gestures from the gesture-based scaling concept. The drawback is that we now have no way to stop scaling, e.g. the dimensions of the object are directly linked to the distance and therefore no other gesture can be made. To overcome this problem we use the touchscreen to stop scaling. This behaviour is implemented in the following way: if the bounding box of the red and green fingers are close enough to the object, the object gets selected for scaling. Once selected the user can change the dimensions of the object by changing the distance between his fingers. If the user taps anywhere on the touchscreen, the object is deselected again.

5.2.5 Rotation - One finger

With this interaction concept we can rotate an object with the use of the red marker. The first step is to select the object with the red and green markers. When this is
done, the user can rotate the red marker around the object to rotate the object in a
fixed plane. A protractor will be drawn around the object to give more information
and context to the rotation. The green marker is ignored by the system. When
the user taps anywhere on the touchscreen, interaction stops and the object will no
longer respond to the red marker.

5.2.6 Rotation - Two fingers

This interaction concept is based on the natural way of rotating an object. In real
life we can pick up an object and use our fingers and wrist to rotate an object.
We modelled this in our application as follows: if the bounding box of the red or
green markers is touching the bounding box of the object, the object can be rotated.
The rotation of the hand, as seen in figure 5.2a or figure 5.2b, will be mapped to
rotations on a fixed plane of the object. The angles between the red and green
markers are used for the rotation. If the user “releases” the object, i.e. making sure
the markers do not collide with the bounding box of the object, then interaction
will stop and the object is fixed.

Figure 5.2: On the left the game two finger rotation in the board style AR (a). On the
right the game two finger rotation in the mid-air style AR (b).

5.3 Visual feedback

All interaction concepts, except the pushing concept, require the participants to
first select the object. This selection is done by grabbing the object with both
fingers. Grabbing a virtual object might be strange, because the fingers have no
haptic feedback. Hence, we developed a system that visualizes when an object is
grabbed and released. If interaction is about to occur or if the object has just been
released, the object has a blue glow. This glow is aligned with the bottom of the
object and can be seen in figure 5.3a. When the participant grabs the object, the
object gets selected and the glow turns green (figure 5.3b). When the object is
deselected (by moving the fingers away from the object), the glow turns blue again.
This is a good indication that the interaction has stopped and that the fingers can
move freely again.
To make the visualization more attractive we used 3D models of objects that one would normally find in the Ludo game. We created a pawn model and a dice model. The pawn model was used in the translation and scaling tasks on the board, while the dice model was used for scaling in mid-air and the rotation concepts. These models were chosen because they relate to the objects used in the real world Ludo game and therefore complement the AR experience.
Chapter 6

Second experiment: Augmented reality board game interaction

In the second experiment we tested the newly created AR and interaction concepts. Section 6.1 describes our experiment setup and section 6.2 describes our expectations. Section 6.3 discusses the results from the collected data, the observations, the questionnaire and comments that the participants made. Section 6.4 discusses the results from the experiment.

6.1 Experiment setup

The second experiment in this thesis consists of two parts. The first part is a follow up study on the first experiment, in which we evaluate the remaining atomic operations scaling and rotation. The second part consists of an evaluation of the atomic operations translation, scaling and rotation in the new tabletop AR setting. We gave this AR a game-like feeling and as a result we refer to it as the (game) board style AR. In the first experiment we evaluated the AR browser or “mid-air AR”, which meant that the participants needed to hold the device upright while interacting with virtual objects that were augmented in mid-air without a clear physical connection. The new AR setting uses a natural feature tracker to track an image placed on a table. In this experiment this image will be a Ludo (in Dutch “Mens erger je niet”) game board. The virtual objects will be aligned with the Ludo image, which gives the impression that the objects are positioned on top of the game board. This is completely different from the first experiment, since we now have a clear connection to the physical world.

The experiment used a within-group study, i.e. each participant tests all interfaces. The experiment starts with a demo scene in which the participant can get acquainted with the AR and the game board. The participant is told to push the natural feature tracker to its extremes by moving fast and/or covering up parts of the marker. In this way the participant is more aware of the capabilities and limitations of the feature tracking algorithm.

After the introductory scene we start with the experiments for each atomic operation. The scaling and rotation operations will be tested in the mid-air and board style, while translation will only be tested on the board. We did not investigate translation for the mid-air style, since we have already tested it in the first experiment and it would require information from the accelerometer and compass. We decided to turn off the accelerometer and compass, because they introduce jitter in
the system and are not needed for the other operations. If we disable the compass and accelerometer we can only translate within the reaches of the screen, which makes for very small translations and a less interesting research. The scaling and rotation operations are done on the spot and thus have no need for additional sensor information. Turning off the sensor data gives the scaling and rotation operations a slight advantage compared to the translation from the first experiment, since jitter from the compass and accelerometer is removed.

Each interaction concept will be explained in a practice test in which the participant can try out the interaction. The goal is to get the participant familiar with the interaction concept, because we already know from the previous experiment that some training is necessary. After the practice level the “real” tests start. Translation will consist of three tests, while scaling and rotation will contain two tests (four in total, because we have the mid-air tests and the board game tests). The participants will be informed of this setup and will know which tests are excluded from the results. To eliminate learning effects the mid-air and game board styles will be switched for each participant. This gives a total of eight combinations, which means that a multiplication of eight participants would suit best in this study. To further eliminate the learning effects, the real levels will be randomized for every atomic operation. The atomic operations will not be randomized or switched since the interaction styles are completely different from each other. Consequently the order will not influence the results.

During the experiment we logged all variables that were potentially of interest to evaluation. The following variables were logged in every frame of the experiment:

- Time: the time it took a participant to complete the assignment.
- Accuracy: the accuracy with which the assignment was completed.
- Finger tracking data, this data provided information about how the participants moved their fingers and at what distance from the camera during the experiment: This distance data was of special interest, since it could be used to verify if the distance problem was solved by changing the AR.
- Game board tracking data: this data provided information about the quality of the natural feature tracker.

6.1.1 Level setup

In tables 6.1, 6.2, 6.3 and 6.4 we see the setup of the different levels. The scaling levels for the mid-air and the board style AR are identical. This was also the aim for the rotation levels, yet due to the different visual representation an error was not noted during the “dry experiments”. This means that the rotation angles are not equal, but this has no drastic impact on the experiment, since we can still fully analyze the interaction concepts. The level numbers from the tables are used in the discussion of the results below and also in the graphs.
6.2 Expectations

In the first experiment we have seen that finger-based interaction lacked the needed performance for serious applications. Besides performance we also identified 2 problems, the ergonomic and finger distance problem. We expect that both problems are solved of reduced by switching to the board game AR style. The ergonomic problem should be reduced because the participants can now rest their arms on the table. This should reduce stress on the arms and could result in better performance, since the arms are held in a more stable position. We expect that the physical connection with the table solves or at least reduces the distance problem from the first experiment, because we expect participants to interact closer to the table, because the virtual objects are positioned there. Overall we expect participants to favor
Chapter 6. Second experiment: Augmented reality board game interaction

the game board AR. This is not only because we expect that it performs better or because we solved or reduced the problems, but also because we expect that the combination of the game board marker and the visualization gives a more engaging experience compared to the mid-air style AR.

6.2.1 Translation

We created two translation interaction concepts for this experiment, i.e. the puar concept and the pushing concept. The pushing concept is adopted from the last experiment, where it proved to be inaccurate for long translations and decent for short translations. In this experiment translation will only be done on the board, which means that we have shorter translations compared to the previous experiment and no jitter from the accelerometer or compass. Therefore we expect that the pushing concept is more stable and has a good accuracy score, but we expect that the pushing concept performs worse than the new puar concept. The main reason is that in the pushing concept, the participant has to constantly correct the direction in which the pawn is moving. This implies that the user reorients the finger multiple times in one translation task, which is not necessary in the puar concept since it automatically follows the fingers once picked up, thus we expect that the puar concept is faster than the pushing concept. In terms of accuracy we expect both methods to perform equally well, since both concepts enable accurate positioning. Nevertheless, pushing has a slight advantage, because it can more easily be used for small corrections.

6.2.2 Scaling

Considering scaling, we have two interaction concepts, i.e. the gesture-based scaling concept and the touchscreen-based scaling concept. Both interaction concepts will be tested on the board and in mid-air. Performance-wise we expect the same performance between the two styles, since the interaction concepts react in precisely the same way. The only difference is the visualization, which we expect to be better for the board game style as explained before.

We consider the gesture-based scaling concept to be more natural, however we expect it to be slower than the touchscreen concept, because the participant loses time when he has to switch between the enlarge and shrink operation. In terms of accuracy we expect the concepts to perform equally well, since both concepts allow for the same amount of control over the final size. In the touchscreen case the interaction stops when the touchscreen is pressed, which gives the participant great control over the size of the object. In the gesture concept the participant can stop scaling at any moment by moving the fingers in the opposite direction of the operation he is performing, e.g. if the participant is shrinking an object, then increasing the distance between the fingers will terminate the shrink operation. We expect that this gives the same amount of control and as a result the same amount of accuracy.

For touchscreen scaling we expect that the combination of holding the device, in-
teracting with the fingers and operating the touchscreen will be considered to be annoying by some participants. This interaction concept is more prone to errors when the mobile device is not held in a stable position, however we expect that most participants will have no problems finding such a position.

6.2.3 Rotation

As for rotation, we also have two interaction concepts, i.e. the one finger rotation and the two finger rotation. Both are tested in the mid-air and board style AR and for the same reason as the scaling tasks, we do not expect big differences between the two styles. The one finger concept can be seen as a remote operation, while the two finger approach is more natural and modelled after rotation in real life. We expect the two finger rotation concept to be much slower for the board and mid-air style AR, because the two finger rotation concept is limited by the rotational freedom of the wrist, while the one finger rotation concept has no limitations and can therefore rotate freely. In terms of accuracy we expect both concepts to perform equally well, because they both allow for small rotations which are needed for exact placement. In terms of accuracy, we expect the two finger rotation concept to be slightly more accurate than the one finger rotation concept, because the one finger rotation concept is more reactive, e.g. when the finger is held close to the object, a small change in the position of the finger has a big influence on the rotation of the object. This makes it more prone to errors and therefore we expect it to perform slightly worse than the two finger rotation concept.

6.3 Results

The results below are from the within-group study described in section 6.1. A total of twenty-four participants, of which twenty males and four females, participated in this experiment. The experiment itself took around 45 minutes. This includes an introduction, the questionnaire and an informal conversation afterwards. The ages of the participants ranged from 18 till 25 years with an average of 21,34 years. The participants were all from Utrecht University. The next part of this chapter will discuss the average times of the participants. Section 6.3.2 will do the same for the accuracy. Section 6.3.3 discusses the distance the finger was held from the camera and section 6.3.4 discusses the performance of the natural feature tracker. These two were both considered critical for the performance of our interaction concepts. Section 6.3.5 discusses the questionnaire and observations we made during the experiment. Finally, section 6.4 sums up the conclusions from this experiment.

In the evaluation we make use of the built-in t-test of Microsoft Excel for significance analysis. Significance was proven, if the chance that the data was random, was less than 5% (p<0.05). We tested significance for the different concepts: per level, per AR style and overall. This results in multiple tests for each interaction concept and therefore the exact chance values are only mentioned when significance is proven.
6.3.1 Average times

Figure 6.1 shows the average time the participants spent in each level. Blue indicates the translation levels, green indicates the scaling levels and red indicates the rotation levels. For scaling and rotation the darker colors correspond to levels that have been performed in the mid-air style AR, while the brighter colors correspond to levels done in the game board style AR. To get the best insight into the times of the different interaction concepts we have removed error values from the dataset. Examples of removed tests are a test that took 202 seconds or a test that took 3 seconds and was still on its initial position. We did not remove slow participants or failed tests, we only removed tests that undoubtedly went wrong and would influence the results in a positive of negative way. A total of 22 from the 528 tests were removed.

![Average times per level](image)

**Figure 6.1:** The average times in milliseconds of each level in the experiment. A larger version of this graph can be found in appendix A.

6.3.1.1 Translation

In the translation levels we can see that no concept can be considered to be better based on time. We see that for small translations the pushing concept seems more suitable, since it is faster in level 1 and slightly in level 3. The puar concept seems more suitable for translation over larger distances, but the advantage is only 2.6 seconds.

In level 3 of the pushing concept, the people who held the device in the right hand experienced a slight disadvantage. This could explain the time difference between level 1 and 3, because the majority of participant held the device in the right hand. The right to left movement in level 3 signifies that the participants would first have to move the hand to the right and then push the pawn to the left. If they approached the object from the top, the left hand needed to be rotated about 180 degrees to push the object. The last experiment taught us that this movement could feel awkward and could therefore add time to level 3 for the right-handed participants.
However an analysis of the data for the right-handed users showed that they were on average not slower than the left-handed users. A significant difference between the two concepts could not be proven.

6.3.1.2 Scaling

In figure 6.1, we can see a time difference between the gesture concept and the touchscreen concept. The gesture concept is on average slower for both interaction styles. Based on the figure, we can also conclude that it does not matter if the interaction takes place in mid-air or on the board, since the average times are almost equal. Upon inspection of the gesture concept we can see that shrinking an object takes longer, while the touchscreen concept takes more time for enlarging objects. We have not found an explanation for this in the data and significance could also not be proven. Nevertheless, the touchscreen concept proved to be significantly faster than gesture concept in the mid-air (p = 0.027) and board style AR (p = 0.020).

6.3.1.3 Rotation

In figure 6.1, we can see that the rotation with one finger is faster than rotation with two fingers. In the one finger case, the longer rotation is a bit faster. Interaction on the board is slower for the one finger rotation. The reason behind this could be that the visualization, which was a top down view, because the marker laid on the table, made it harder to see if the rotation was correct.

Overall the one finger rotation proved to be significantly faster than two finger rotation in the mid-air (p = 0.003) and board style AR (p = 0.009). One finger rotation proves to be especially suited for long rotations, since it is significantly faster in the first level of the mid-air (p = 0.005) and board style AR (p = 0.018).

6.3.2 Accuracy

To test the performance of the interaction concepts, we considered not only the time it took the participants to complete the tests, but also the “distance” to the goal. This distance gives an indication of the accuracy of the implemented concepts. In each test the participant had a clear goal, however accuracy itself remains subjective, e.g. what is considered by participant A to be accurate could be considered “not precise” by participant B. Therefore, rather than a hard goal / distance value, we have created a range in which we consider the test to be successfully performed. This range is subdivided into two subsets, i.e. the range in which a tests is considered accurate and the range in which a test is classified as really accurate or perfectly positioned. These ranges have experimentally been determined and may differ for each test.
Chapter 6. Second experiment: Augmented reality board game interaction

Figure 6.2: The accuracy of each level in the experiment. The columns are subdivided into two parts, the bottom part is the “perfect score” and the top part is the “accurate score”. Together they form the tests that were considered successful. A larger version of this graph can be found in appendix A.

The experiment contained 528 tests from which almost 70% (69.70%) were completed successfully (error values included). If we look at all the operations we can see that the translation operation is most accurate, while the scaling and rotation operations seem comparable.

6.3.2.1 Translation

In figure 6.2 we see that most puar translation tests were performed successfully. The puar concept has a success rate of 83.34%, while the pushing concept has a success rate of 81.95%. If we look at the “perfect” range we see a difference, puar scores a rate of 31.95% while the pushing concept scores 23.61%. As for the pushing concept, most perfect scores are found in level 3, in which the translation was really small. The puar concept has a more equal distribution of perfect scores among the levels. Based on this we conclude that the puar concept is generally more accurate than the pushing concept. However, if puar is not suited for the envisioned application, then pushing could also be used, since it has almost the same success rate. A significant difference between the two concepts could not be proven.

6.3.2.2 Scaling

In figure 6.2 we can see the scaling operations seem to have almost the same accuracy. If we look at the gestures-based scaling concept, we see that it has a success rate of 61.46%. The gesture concept in mid-air is a bit more accurate with a success rate of 64.58% versus the board with 58.34%. In the touchscreen concept it is the other way around with mid-air scoring a success rate of 64.58% and the board scoring 68.75%, with an overall success rate of 66.67%. The touchscreen concept seems to be best suited for shrinking objects on the board, because only four participants failed the test (success rate of 87.50%).

When we look at the perfect ranges we see that the gesture concept scores low on
the board (25%) compared to the mid-air (45,83%) tests. As for the touchscreen concept, the performance is in both cases 35,42%. When we combine the mid-air and board style AR, we see that both gesture scaling and touchscreen scaling score a 35,42% perfect score rate.

The results indicate that the participants understood and could operate both interaction concepts. If accuracy is the main objective than the mid-air style could best be chosen, since it proved to work significantly better than the board game approach for all scaling levels (gesture-based scaling: \(2,157 \times 10^{-6}\)), touchscreen-based scaling: \(9,479 \times 10^{-9}\)). A significant difference between the two scaling concepts could not be proven.

6.3.2.3 Rotation

The accuracy of the rotation concepts seems to follow the same trend as the scaling levels. There is no clear winner and the average accuracies are really close. The one finger rotation has an accuracy rate of 63,54%, while the two finger rotation concept performs slightly better with a success rate of 67,71%.

In the mid-air style AR the one finger rotation has a 64,58% accuracy rate and on the board an accuracy rate of 62,5%. This means that the one finger approach performs equally well in both AR styles. This is to be expected, since the one finger rotation is a “remote” operation, i.e. there is no real connection with the object. This means that the focus lies more on the rotation operation itself, rather than the actual interaction with the object. Therefore it matters less if the object is manipulated in mid-air or on the board.

Considering the two finger rotation concept, the accuracy rate in mid-air is 72.92% and on the board 62.5%. The mid-air concept clearly performs better. A reason for this could be that the mid-air rotation style gives more freedom in the way people can rotate, i.e. the visualisation could allow participants to rotate more easily in mid-air compared to the board.

When we look at the perfect ranges we see that both operations score a success rate of 37,50%. When we look closer into the one finger rotation concept we see that the mid-air style scores a success rate of 39,58% and the board style 35,42%. Considering the two finger concept, the success rate in mid-air is 43,75%, while the board style scores a 31,25% success rate. In both cases the board seems to perform worse. We assume that this has to do with the top-view visualization, which makes it slightly harder to see of the rotation is perfect or not.

A significant difference between the two concepts could not be proven.
6.3.3 Distance between the fingers and the camera

During the experiment we logged the amount of pixels of each finger marker for every frame. This information gives us insight in the distance the fingers were held from the camera. This distance is important, because we can argue that holding the fingers close to the camera makes most interaction concepts inoperable. During the previous experiment, in which we only looked at mid-air style interaction, we noticed that the participant had the tendency to hold the finger close to the camera. This observation was also made during this experiment especially in the mid-air tests.

Figure 6.3 is a plot of the amount of pixels of the green and red markers in the different levels. Note that the one finger rotation levels are plotted, but in these we mainly used the red marker. The green marker might even be hidden from the camera, since it was not used for interaction. The amount of pixels is linked to the visibility of the colored marker and could therefore be noisy. To cope with this problem, all the experiments were done in the same room with the same lighting conditions and threshold settings, therefore the plot in figure 6.3 gives an accurate view on the distance between the finger and the camera during the experiment.

![Figure 6.3: The distances between the fingers and the camera per level. A larger version of this graph can be found in appendix A.](image)

From figure 6.3 it is immediately clear that the participants held their finger much closer to the camera in the mid-air style tests. For the red \( (p = 3.197 \times 10^{-17}) \) and green \( (p = 5.119 \times 10^{-17}) \) marker the distance between the finger and the camera is significantly larger in the board style AR. The significance and the high values for the mid-air concepts versus the low values of the board concepts indicate that, without a physical connection, people tend to move their hand closer to the camera. As mentioned in section 3.4.2.3, a larger distance from the camera results in smaller markers which could potentially increase the accuracy of the system, because small movements and adjustments become possible.

From the accuracy and time data we must conclude that “the closer finger distance” to the camera had, on average, no negative effects in this experiment. Nevertheless, some participants commented that they felt that they had more control over the interaction concepts if they moved farther away from the camera. This is also what we observed during the mid-air experiments. As a result, we conclude that the
distance to the camera still remains critical for interaction, although a direct effect was not visible in the experiment data.

6.3.4 Performance of the natural feature tracker

As mentioned the ability of the natural feature tracker to detect our game board was critical, therefore we decided to register whether or not the board was detected during each frame the experiment. In the experiment the board was successfully detected in 92.30% of the 297484 frames. This includes the startup of each level (in which the participant needs to reorient), but excludes the practice levels. The practice levels are not included, because they are not representable for the rest of the experiment. From the given percentage we can conclude that the performance of the natural feature tracker is good even under partial occlusion from the hand.

6.3.5 Questionnaire and observations

At the end of the experiment we had an informal conversation with the participants to get more information about how they experienced the different interaction styles. During this conversation we asked the participants to rate the different interaction styles from 1 to 10, with 1 being bad / very unpractical and 10 being perfect / very useful.

<table>
<thead>
<tr>
<th>Interaction style</th>
<th>Average grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation - pushing</td>
<td>7.02</td>
</tr>
<tr>
<td>Translation - puar</td>
<td>7.27</td>
</tr>
<tr>
<td>Interaction on the board</td>
<td></td>
</tr>
<tr>
<td>Scaling - gesture-based</td>
<td>6.65</td>
</tr>
<tr>
<td>Scaling - touchscreen</td>
<td>7.15</td>
</tr>
<tr>
<td>Rotation - one finger</td>
<td>7.52</td>
</tr>
<tr>
<td>Rotation - two fingers</td>
<td>5.58</td>
</tr>
<tr>
<td>Interaction in mid-air</td>
<td></td>
</tr>
<tr>
<td>Scaling - gesture-based</td>
<td>6.67</td>
</tr>
<tr>
<td>Scaling - touchscreen</td>
<td>7.17</td>
</tr>
<tr>
<td>Rotation - one finger</td>
<td>7.40</td>
</tr>
<tr>
<td>Rotation - two fingers</td>
<td>5.33</td>
</tr>
</tbody>
</table>

Table 6.5: Overview of the grades of the different interaction styles averaged over all participants.

Table 6.5 gives an overview of the grades of the different interaction styles averaged over all participants. All interaction concepts, except one, score at least a satisfactory in the Dutch grading system. Scaling was rated slightly better in the mid-air style, while rotation was rated slightly better in the board style. The grades for the board and mid-air are almost equal, because only three participants found that the differences were large enough to give the interaction styles different grades.
Besides the grading we used three open questions to get some more information about the way people thought about the different interaction styles. The main goal of these questions was to get a conversation started about the experiment. The used questions are:

1. What is your overall impression of the finger tracking concept for interacting with an AR?

2. Do you see yourself playing a board game with finger-based gestures?

3. Do you think that the finger interaction increases the fun or engagement factor compared to the touchscreen or other input concepts?

In general, the participants answered positively to the questions. They liked the interaction styles and 3 even asked if the application was downloadable from the Android market. Naturally, there were also negative remarks. Below we summarize the important remarks, observations and comments:

In general participants were positive about the created setup. Frequently we heard exclamations like “very cool” or “wow”, but of course also frustration sounds when things didn’t work as expected. Some concepts seemed to work better than others. Three participants mentioned that the frustration factor of some concepts was too high to make them practical at this point. The two finger rotation was mentioned as an example. Two participants mentioned that they think that the touchscreen is better for these kinds of operations. One of them also questioned the practicality of the concept, while the other still found them “usable and original”. A total of five participants complained that holding the device and operating the touchscreen was annoying. A holder or mount for the device was opted as a solution for this problem. The big disadvantage of a holder is that we lose the mobility of the device, therefore this was never considered as an option.

Overall, the participants were positive and found the system “fun to use”. One participant mentioned that he thought that this form of interaction was not yet possible and that it “seemed like something from the future”. One called it a “good addition to the touchscreen, as some interaction concept work faster with finger interaction than they would do with the touchscreen or even the mouse”. In this case one finger rotation was given as an example.

One participant said that “the mix between AR and the natural operations feels convincing and emotionally good”, which was one of the main goals of the interaction concepts. Other participants also mentioned that the combination of the real and virtual world was done nicely. Three participants mentioned that the visual feedback from the interaction concepts helped them to perform the tasks. One even mentioned that the visual feedback was “great”, which he also found “important” in these types of applications. Two participants explicitly mentioned that they preferred the board game setup, because they felt more in control of the visual representation, i.e. the natural feature tracker reacted to the position of the device.
and changed the visualisation accordingly, thus giving the participants control over the visualisation.

Nine participants mentioned that the concepts take some getting used to, which was to be expected since interaction with the fingers is not (yet) an everyday task. One of them said “sometimes it took some practice, but after that it worked great”. This comment sums up the impression that the concepts left behind after the experiment, i.e. participants needed some time to see how the concepts work and what the best approach is, but after that they could all complete the tasks we gave them. A result of this is that the participants felt that finger-based interaction has potential for future applications. Games were frequently mentioned as a good platform for our system.

6.3.5.1 Engagement

All participants agreed that the engagement and fun factor can be comparative or higher, than that of touchscreen applications. In the puar levels, two participants were observed who made a movement like they were holding a real pawn in their hands, i.e. they first picked it up, moved the hand with the pawn upwards and placed it down again by moving the hand downwards. Another four participants commented that they looked at the game board marker to see if the objects were actually there or if they were placed at the correct position. Both are indications that the participant felt engaged with the AR. Two participants commented that it even felt strange that they were holding objects that were not really there, while it looked like they did. One participant commented: “with sound it almost becomes real” indicating that the participant felt engaged with the created setup.

6.3.5.2 Difference between mid-air and the board

The participants mentioned that interaction-wise there was no real difference between the board and mid-air styles, nevertheless the board was considered to be slightly easier by some. One participant mentioned that interaction on the board was easier, because “the visualization automatically adjusts to the position of the mobile phone”. Therefore he had the impression that he could better focus on the finger interaction, since he didn’t have to focus on the position of the device.

It seems that overall the board AR made a bigger impression on the participants, since all comments related to engagement or fun were made with respect to the board style AR.

6.4 Discussion

In this experiment we evaluated different finger-based interaction concepts in a game board and a mid-air style AR. We can conclude from the data analysis and especially the informal conversations that participants liked the created setup. This can also be confirmed by the grades that the participants gave our interaction concepts.
When we look at the data and the feedback from the participants we can conclude that:

Considering translation, the puar concept was thought to be slightly better, because the participants gave it a somewhat better grade and seemed more positive about the naturalness of the interaction concept. Data-wise there was no observable difference in performance between the two concepts.

As for scaling, the touchscreen-based interaction concept was considered to be slightly better, because the participants gave it a better grade and the data told us that the performance of the touchscreen concept is slightly better. On the other hand, we must note that the intuitiveness and ease of use of the touchscreen was questioned by multiple participants. Scaling itself can best be done in the mid-air style, since it performs significantly better in this setting.

With regard to translation, the one finger rotation interaction concept was considered to be better, because the participants gave it a better grade and because the concept was clearly faster, while having almost the same accuracy.

The fun factor and engagement of the created ARs was considered to be higher or comparable to that of touchscreen applications. Multiple participants mentioned that they felt more engaged with our AR setups than with “normal games” on a mobile device. We can conclude based on the comments that especially the board game AR is considered to be more engaging and fun. An indication of this was also visible in the data, because the participants held the fingers much farther away from the camera in the board game AR, compared to the mid-air AR. This means that the participants were holding the fingers closer to the board, indicating that the connection to the real world made them think that interaction actually takes place on the board.

Finger-based interaction seems to be more suitable for the game board AR style, since participants were far more positive about this setup and the experiment data proves that the interaction concepts work good on the board. We believe that the participants were more pleased with the board AR, because the connection to the real world gave the interaction concepts more context, since it was clearer how and why we wanted to use the concepts. We also know from the data analysis that the participants held the fingers further away from the camera, which has the potential of making the interaction concepts more robust.
Chapter 7

Conclusion and future work

7.1 Conclusion

In this thesis we have designed, implemented, evaluated and discussed different interaction concepts for mobile AR applications. The main focus was on finger-based interaction, which proved to be more suited for entertainment applications than for serious applications.

In the first experiment we used an AR that was based on the widely known AR browsers. Founded on a comparative study, between finger-based interaction, device-based interaction and touchscreen-based interaction, we came to the conclusion that touchscreen-based interaction was more suited for selection tasks, while the device-based approach worked best for translation. The finger-based interaction concept lacked the needed performance for serious application, such as AR browsers. Based on the questionnaire and interviews with participants we came to the conclusion that finger-based interaction improved fun and engagement and could best be used in an entertainment or leisure application, such as an AR game.

In the second experiment we used a tabletop AR application with a game-like setting. We used a comparative study to get further insight in the feasibility of finger-based interaction. In the questionnaire and interviews it became clear that the tabletop setup was considered to be better suited for our finger-based concepts. Translation on the board could best be done with the use of the puar concept. Our more natural gesture-based scaling concept proved to perform less than the touchscreen scaling concept. As for rotation, the one finger concept proved to perform much better than the two finger concept.

The main experimental goal of this thesis was to determine the feasibility of finger-based gestures for use in mobile augmented reality. The results from the second experiment show that the accuracy of finger-based gestures is good, but the best proof for feasibility comes from the participant reactions and comments. In the second experiment, all participants agreed that the use of finger-based interaction improved fun and engagement, which is especially important for entertainment and leisure applications. This enthusiasm was also seen in the first experiment, although from that we can conclude that finger-based interaction is not suited for all mobile applications. Especially applications that need fast interaction with high accuracy, could better use the touchscreen or other gesture methods, such as the device gestures.
In summary, we can state that there is no “one size fits all” solution for finger-based interaction in mobile AR applications. Each interaction concept has its positive and negative sides, but we can conclude from the reactions of the participants and the results from the experiments that finger-based interaction systems do have future potential for mobile AR applications.

7.2 Future work

In this thesis we have developed and tested multiple interaction concepts and ARs. During the creation and implementation of these systems, we have uncovered multiple other possibilities for other interesting research topics. In this section we spend some attention to the most promising ideas or concepts.

7.2.1 Markerless finger tracking

In the experiments we have seen that participants were able to interact with the AR in a natural way, through the use of finger-based gestures. We used colored markers to make detection and tracking of the fingers easier. The main disadvantage is that participants are interacting in a natural way with artificial markers attached to their fingers. Although participants did not complain, this could distort the illusion that interaction happens naturally. We believe that the system can be made more natural if the fingers could be detected without the use of the colored markers. Fortunately, another university student named Willem-Jan Spoel decided to focus his master thesis on markerless finger tracking on mobile devices.

7.2.2 3D finger tracking

In this thesis we focused on 2D finger-based interaction concepts. A logical next step is testing multiple 3D interaction concepts to see if that works better than the 2D concepts.

In the one finger mid-air style rotation experiments, the participants had to rotate the object around the y-axis that is aligned with the touchscreen (figure 5.2b). During these tests we were confronted with participants that wanted to rotate the object by moving the finger in a circular pattern that is parallel to the perpendicular axis of the touchscreen. This gesture didn’t work in our system, because we had no accurate depth information and therefore no 3D tracking. This example illustrated that 3D tracking could be more natural than our 2D finger tracking approach. Besides improving current concepts, 3D information could also be used to create new interaction concepts or operations, such as 3D translation or 3D rotation. For these reasons we believe that 3D tracking would be worth researching.

In this thesis we investigated the possibilities for 3D interaction with our system. Just like the PlayStation Move we could use the amount of pixels to determine the distance to the camera. We created a simple setup that used this technique, however we discarded it because it did not provide the necessary sensitivity for
making smooth 3D gesture concepts. The number of pixels found is affected by the lightning conditions, shadows and so on. This makes the 3D information noisy and inaccurate, however for simple gestures, the technique can be used as proven by Joris Dekker. He used the amount of pixels and resulting 3D information to simulate mouse clicks. To reduce false positives he created an additional verification system. In this thesis we made the decision not to depend on noisy 3D data, because we believed that it would degrade the performance of the interaction concepts, rather than improve them.

Notable is that in the upcoming months smartphones with dual camera’s will be released. The HTC Evo 3D [63] and the LG Optimus 3D [64] are especially designed to display and create images and video’s with 3D depth information. The dual camera setup makes for a great way to establish 3D interaction if one is able to get access to both camera’s at the same time. We can expect that the resulting depth information is much more robust than the “amount of pixels”-approach described above and should therefore allow for smooth 3D interaction.

7.2.3 Implementation in a real game

In the questionnaire of the first experiment we have seen that finger-based interaction scores high on entertainment. This was the reason we wanted to add a more game-like feeling to the second experiment. A good follow-up experiment would be to implement a full game with the interaction concepts from the second experiment. The main goal could be to test the performance of the interaction concepts when used in real applications, rather than in the test beds we created.

It would also be interesting to consider different ways of switching between the operations (translation, scaling and rotation) in a real game. In our experiment each level was initialized to work with a certain gesture and operation, however in a real game all operations and gestures need to be accessible. Switching between operations is not a straight forward task. We created two systems to test if it was possible to implement a switching system for the second experiment. The first system used virtual buttons, which could be selected with the finger. This system was very basic and worked decently, however it had no additional value for the experiment. The second system we created would switch operations based on the number of fingers visible in the camera image. If one finger was found, the system selected the translation operation, if two fingers were found the scaling operation got selected and three fingers selected the rotation operation. We tried to improve this system for several weeks, also adding a way of creating objects (five fingers), but eventually we decided that it was too unstable to use in a real experiment or application. In the experiment we decided to abandon switching, but it remains vital for a real game or application. A small study in which different switching operations are compared should give a game or application programmer more insight in which operations are suitable and can be used in a real AR application.

The combination of gesture-based interaction systems and AR proved to be an
interesting research area. There are still many possible research projects in this area, therefore I hope this thesis will serve as an inspiration to all thinking of or just started working in this research field. I wish students and other researchers the best of luck with their own gesture-based or AR related projects.
Appendix A

This appendix contains three graphs from the second experiment. They have been enlarged to fit on one A4 page per graph, therefore the graphs will start on the next page.
Figure A.1: A larger version of the times graph from section 6.3.1
Figure A.2: A larger version of the accuracy graph from section 6.3.2
Figure A.3: A larger version of the distance graph from section 6.3.3.


[37] M. Baldauf and P. Fröhlich, “Supporting hand gesture manipulation of projected content with mobile phones,”


