THE TOUCHPAD IN VR
EVALUATING INPUT DEVICES IN VIRTUAL REALITY

by

Joeri van der Velden

Student ID 3928853

in partial fulfillment of the requirements for the degree of

Master of Science
in Game and Media Technology

at Utrecht University,

Supervisor: dr. W.O. Hürst
2nd supervisor: prof. dr. R.C. Veltkamp

Utrecht University

Abstract

Virtual reality (VR) has gained renewed interest in the consumer market after the technology required for high quality head-mounted displays (HMD) became affordable to mass-produce. VR is able to give the user unique experiences that no other medium can provide, but also requires a new set of design guidelines to create and interact with VR content. One crucial aspect in how we engage with VR applications is the input device. Today's common input devices are hard to adapt or unintuitive for this new medium. This thesis presents a framework to assist with designing experimental setups for evaluating input devices, both old and new, for different kinds of VR applications. The four main variables in this framework are the VR system, input device characteristics, VR application, and performance metrics. To verify the utility of our framework we use it to design an experimental setup that evaluates the performance and usability of the touchpad as an input device for a VR application focussed on image library management.This setup is then subjected to a user study. The results of that study show that the touchpad is able to provide users an intuitive interface for interacting with a VR image browser, but that multi-touch gestures are harder to perform than expected. Completing this user study demonstrates that the framework can assist in structuring research into input devices for VR.

Preface

It's an exciting time for fans of computer entertainment and virtual experiences, like myself. The new wave of consumer virtual reality devices picked up speed right before I started my masters at the Utrecht University, and has dominated my interests ever since. It's not often that you can contribute to the rise of a new medium. Being at university and doing this project has helped me look at developments through an academic lens.

I would like to thank my supervisor, dr. Wolfgang Hürst, for his guidance and my family for their support.

Joeri van der Velden
Utrecht, February 2017

Contents

1

INTRODUCTION

The virtual reality industry is rapidly growing thanks to recent investments by big technology companies and increasing popularity on the consumer market. VR technology still faces a variety of challenges, one of which is input methodologies. This thesis aims to contribute to this field of research by proposing framework that assists in evaluating input devices for VR. This introductory chapter will first discuss the history of VR and various existing input solutions, concluding with the thesis objectives.

Before diving in we would like to clarify some terminology. "Virtual Reality" has quite a broad meaning, especially in academic research. When referring in to a "VR system" or "immersive VR", we mean a system that makes use of a spatially tracked head-mounted display that fully occludes the user's view of his physical surroundings.

1.1. OVERVIEW OF THE VR INDUSTRY

We will describe how the modern VR industry was formed, the types of VR systems that are available today, and various common input solutions on the consumer market.

1.1.1. HISTORY

Virtual reality has seen a surge in consumer interest the past few years. The ideas for immersive VR technology itself are far older, and have been around since the early days of computers. Bringing these ideas to the market proved difficult. High profile devices like the Virtual Boy [1] were notable commercial failures, and are often pointed at for being the cause of VR fading from the consumer market for nearly two decades. Outside of the consumer space, VR research continued within academia. Specialized systems, such as the CAVE [2][3], were developed for scientific and engineering applications. It wasn't until the rapid growth of the smartphone industry that technologies like small, high resolution, low-latency displays, and reliable rotational and positional tracking became affordable to mass-produce. These were the key ingredients for making high quality immersive VR systems available to the consumer market in a manageable form factor.

The turning point for modern VR can likely be attributed to the rise of Oculus, a company specialized in VR hardware and software, starting with their successful crowd-funding campaign for the Rift HMD developer kit late 2012 [4]. The company was bought by Facebook in March 2014, shortly after introducing their second generation developer kit. In the same period, other major tech companies like HTC, Samsung, Google and Valve also entered the market. VR systems are now readily available to consumers for various mobile, game console and PC platforms.

Even though the technology was ready for market, there was still the question of content. The design rules for VR content were yet to be written. Developers soon learned that VR experiences were quite different from traditional media and offered great potential for immersive virtual experiences. Well designed VR experiences give the user a feeling of 'presence', the feeling of being in a different place. This new experience, that no other medium can provide, is one of several unique selling points of virtual reality.

1.1.2. MOBILE VR

The term "Mobile VR" is commonly ascribed to fully self-contained VR devices that are not tethered to any external computing platform. These often come in the form of smartphones that slide into holders to become makeshift HMDs. The mobility is its defining factor, the user can easily transport the device and use it in different locations. It also makes it one of the most accessible forms of VR, and thus has the most widespread adoption. Mobile VR does face some challenges in return. Namely, the limited computing power, tight thermal requirements and battery limitations, demand very optimized software and can cause devices to easily overheat under heavy load [5]. Additionally, technologies like positional tracking becomes a lot harder to implement, since there is no external tracking system that can be used as absolute point of reference. Positional tracking systems with "inside-out tracking", that perform positional tracking from the device itself, do exist but are not considered robust enough and up to par with the tracking quality of external systems.

1.1.3. ROOM-SCALE VR

"Room-scale VR" is a term assigned to current high-end VR systems with specific tracking characteristics. Namely the tracking space is 'room sized', the user can freely move within a space ranging up to about 4 by 5 meters. Currently, the most prominent product on the market with this feature is the HTC Vive, using the Lighthouse tracking system. It makes use of laser beacons for tracking, and theoretically supports a large amount of tracked objects within the same space with minimal computing overhead. While room-sized VR can provide some of the most compelling VR experiences, it requires the user to properly set up and configure the system every time the beacons move, making it hard to transport.

1.1.4. THE INPUT PROBLEM

The head-mounted display (HMD) is often the most notable part of a VR system, but it's not complete without an accompanying input solution. Common input devices such as mouse and keyboard, gamepads and touchscreens can be used with current VR systems to some extent, but are not considered intuitive or optimal solutions. A key difference is that the user's view is obstructed by the HMD, and thus cannot see his/her own hands or the physical input device he/she is interacting with (unless given some sort of virtual representation). Additionally, some VR systems allow the user to physically move around in a tracked space, so applications that want to take advantage of that cannot use stationary input devices like the keyboard.

For these reasons, a lot of research and development is being done to describe new input paradigms and create or adapt input devices that complement the variety of VR systems and their applications. However this community still lacks a common vocabulary for evaluating input devices in VR environments. This paper proposes a framework that unifies the process for assessing input device viability and performance evaluation.

1.2. INPUT TECHNOLOGIES IN THE CONTEXT OF VR

To start off, we look at the wide variety of input technologies being used in the consumer space, and their current use for VR systems in more detail.

1.2.1. COMMON INPUT DEVICES

As described earlier in 1.1.4, most common input devices today were not designed for VR. But they can be adapted to some degree.

MOUSE AND KEYBOARD

The mouse and keyboard are still the prevalent way of interacting with desktop platforms. Early VR demos also supported it to some degree. The keyboard is used for navigation, while the mouse controlled the camera yaw or a pointer projected onto the environment. The main downside of using mouse and keyboard is its rigid setup: the user is expected to always face the same direction, and requires a surface to rest the devices on. This limits the design scope for VR applications.

GAMEPAD

The gamepad, or game controller, has several advantages when it comes to VR input. It does not have the same setup restrictions as mouse and keyboard, allowing the user to freely rotate and look around a VR environment while operating it. Gamepads often offer a range of extra input options, such as buttons, analog sticks and shoulder triggers. Several consumer VR systems adopted it as the primary input method.

TOUCHSCREEN

The touchscreen has become one of the most common input devices today thanks to the prevalence of smartphones and tablets. However the default way of operating a touchscreen does not mesh well with VR systems simply because it requires the user to look at the screen to perform touch actions. Though, a touchscreen can still be adapted as a touchpad, which is further explored in 1.2.6.

REMOTES

A remote is a simple single-handed device with a number of controls, usually buttons, to interact with an application on a separate device like a TV or radio. Because they're relatively straightforward to combine with simple VR applications that mainly require navigating menus. Consumer VR devices like the Oculus Rift usually combine a remote with gaze controls described in subsection 1.2.5). Google's Daydream VR device also comes with a remote that tracks rotational movement for additional interactivity.

1.2.2. SPECIALIZED INPUT DEVICES

We define specialized input devices as input devices with specific applications in mind. For example steering wheels, feet paddles, and flight sticks. These are commonly stationary and simulate the driver's seat of a vehicle or operator cabin for machinery. These devices are often used by simulators for job training in professional environments, or for simulator games on the consumer market. Modern VR systems have shown to combine quite well with these kinds of controllers and contribute greatly to immersing the user further into the scenario presented by the simulation. However because of their specialized nature, they can be hard to adapt for other use cases.

1.2.3. TRACKED HANDHELD CONTROLLERS

One of the more successful input devices for high-end consumer VR systems so far is the handheld controller. It differs from the gamepad because it consists of two physically separate controllers for each of the user's hands, both tracked in the same space as the VR HMD. This allows the user to physically reach out and interact with objects in the virtual world, hence why this input device is often considered to be the most compelling solution. Examples are the HTC Vive wands and Oculus Touch controllers.

The handheld controller does have its limitations. Due to the need for external tracking, it's often limited to high-end VR systems that provide 'room-scale' tracking solutions as described in 1.1.3. This makes it hard to combine with mobile VR systems.

1.2.4. BODY TRACKING SYSTEMS

Body tracking systems that are intended for application input commonly use a camera or several depth cameras to track parts of the user's body, and apply these to virtual avatars or map movements to specific interactions. One example is the Microsoft Kinect [6] which estimates a user's pose to build up a virtual skeleton rig. Another example is the Leap Motion Controller [7], which does not track the full body but just the user's hands.

While a body tracking system seems to be the most comprehensive tracking method, it also faces several unique challenges. Tracking issues are quite common because of occlusion, making it hard for the system to estimate the pose of occluded body parts. Body tracking systems are often limited to a specific domain. For example, the Microsoft Kinect does full body tracking but does not have the resolution to estimate finger poses. Meanwhile the LeapMotion only tracks the user's hands. Another downside of using body tracking as an input solution is the lack of tactile feedback. Users often have problems triggering discrete actions, because the system has to infer the user's intent solely from a camera feed.

1.2.5. HEAD-TRACKED CONTROLS

Head-tracked controls, also called gaze controls, is the method of using the user's head direction as application input. For example, the user has to keep his head pointed towards a button in the virtual environment for several seconds to trigger its corresponding action. The input device in this case would be the VR HMD itself. This technique is commonly used in mobile VR, since it requires no additional hardware. The disadvantages are very clear however: the amount of possible input actions is very limited, and prolonged use of this method can be tiring for the user.

1.2.6. THE TOUCHPAD

The touchpad is an input device which most commonly serves as an alternative to mouse input for laptops, or as an alternative to buttons on other small consumer electronic devices like music players. Touchpads can come in several forms depending on its application, like a square flat surface that allows for swiping, or a circular touchpads for scrolling. A touchpad is similar to the touchscreen in terms of possible interactions, allowing for advanced multi-touch gestures when supported by the device. Some consumer VR systems already adopt touchpads, for example the HTC Vive wands use a circular touchpad instead of a thumbstick. It can also serve as a simple button replacement, like on the side of a Samsung GearVR HMD.

The use of a standalone touchpad device for VR input has not been fully explored however. This solution would fit the constraints of mobile VR systems, while allowing for complex multi-touch interactions to interact with VR applications. Additionally this does not have to be a fully dedicated device, a touchscreen on a tablet or smartphone, combined with software to communicate with the VR system, could serve the exact same function as a dedicated touchpad. These are compelling reasons to explore the use of touchpad in VR, which we'll expand on in chapter 4.

1.3. THE INPUT DEVICE EVALUATION FRAMEWORK

The input devices presented in the previous sections all have varying degrees of utility for different VR applications. This makes them hard to evaluate and compare for new VR applications that might require novel input solutions. This is why we propose an input device evaluation framework for VR applications. This framework aims to support a variety input devices, for a range of common VR applications, in different user environments. The goal of the framework is to provide a clear path for input devices to identify and quantify their strengths and weaknesses in VR applications for different domains.

This research will describe the full framework, and implement a subset of its functionality to test the touchpad for use within a specific domain of VR applications.

1.4. THESIS OBJECTIVES

The thesis aims to achieve the following:

- Describe conceptual framework for evaluating input devices for VR applications
  - Outline requirements
  - Describe a process to assess input device viability
  - Describe steps for assessing the VR system
  - Describe steps for assessing the VR application
  - Discuss metrics
- Implement part of the framework into an experimental setup to evaluate the touchpad in VR
  - Apply framework
  - Describe design of application
  - Provide implementation details
- Evaluate whether the touchpad is a viable and intuitive input method for a VR application focussed on organizing an image library
  - Discuss goal and hypothesis
  - List metrics that are collected
  - Describe user study setup
  - Discuss results

1.5. THESIS OVERVIEW

Related work is discussed in chapter 2. We look at relevant papers in the area of input testing methodologies to construct our framework. We also explore work in the area of VR and touchpad input research.

In chapter 3 we describe the framework, containing the various elements for the VR system, input device, VR application and metrics to account for.

Next we describe the design of our experimental setup, along with a custom VR application for evaluating the touchpad. We briefly discuss results from an earlier user study we performed that compared the touchpad to various other input devices, to further motivate our choice of the touchpad as the test subject for the framework. The requirements, design concept and implementation details of the custom VR application are laid out in chapter 4. We will then use this application in the user study described in chapter 5, which collects metrics on the performance and usability of the touchpad for the VR application. Lastly the results will be discussed.

Chapter 6 will summarize our findings and recommend future work.

2

Related work

Before describing the framework we first explore the field of VR input, input evaluation research, touchpad input research.

2.1. VR input

We provided an overview of existing input solutions in section 1.2. To further examine how to evaluate input devices in VR, we need to identify what features and characteristics are important for a device to have in immersive VR applications.

VR has shown to be a paradigm shift for how we think about virtual applications. Within 3D environments, 3D interactions are quicker and more intuitive to work with [8] compared to traditional 2D interfaces. The definition of "VR" can be quite ambiguous in previous work. Not everything is applicable to the immersive head-tracked HMDs used in this research. However many important aspects of VR were researched early on in the 1990s. For example, the feeling of 'presence' [9]. Researchers also noted the importance of haptic feedback for VR input early on [10]. Haptic feedback enhances both accuracy and the feeling of experience of interacting with a virtual application [11] [12] [13]. This can be traced back to how important physical feedback is for when people interact with physical objects [14].

When designing or adapting applications for VR input, one of the key philosophies behind user interactions is to minimize energy spend, and maximize comfort. A demonstration of this can be found in the work of Mine et al. [15] where they convert an off-the-shelf modelling application, SketchUp, to be used within a CAVE system with tracked input devices, using their own UI adaptation. The goal is to eventually reach the point where users can operate "at the speed of thought".

Input devices for current high-end consumer VR mostly target gaming applications, and as such the design and functionality has elements comparable to those of traditional game controllers. Oculus outlined the design of their Touch controllers [16] at their annual conference. Buttons are still the primary method for giving the user reliable feedback to their actions.

2.2. Input evaluation methods

To motivate the design decisions of our framework we look at several input device evaluation studies. Our goal here is to identify methods for comparing input devices, how these were designed and with what goal in mind.

2.2.1. Performance and usability

Dang et al. [17] performed a usability study that compares voice, wand, pen and sketch input interfaces for use in a 3D air traffic control application. Their user study used task completion time and error ratio as the base metrics. Additionally participants filled out a survey that asked questions about ease of use, ease of learning, frustrations with respect to the user interface and the perceived level of difficulty of the task. These elements for performing usability studies seem to be the most common, often in context of a specific application or task. The following studies used similar approaches, by gather quantitative results through measuring task completion times, accuracy and error rate, along with a post-experiment questionnaire for all participants to gather the participant's thoughts on the device's ease of use:

- Kavakli and Thorne [18]) compare input devices (keyboard, mouse, joystick) for computer driving games.
- Zabramski [19] on input devices (mouse, pen, touch) for a shape tracing task.
- Woods et al. [20] on non-keyboard input devices (mice, trackballs, joysticks) for the health and safety industry.
- Turpin et al. [21] compare a variety of alternative input devices for people with disabilities.
- Thomas [22] evaluates three input techniques (mouse, head cursor, image-plane vision-tracked device) for use with a wearable augmented reality display.
- Accot and Zhai [23] compare input devices (mouse, tablet with stylus, touchpad, trackball and trackpoint) for a steering task.

In order to support a wide range of input devices, our framework should have a general method of quantifying input device performance. One such method is provided by Fitt's Law [24], which is represented as a formula that provides a performance rating based on the time a user needs to move from a certain start position to a goal object using a specific input device. Vertegaal [25] used Fitt's Law to compare eye tracking input techniques (manual click and 'dwell time' click) with mouse and stylus input. They also calculated error rates to give an indication of how accurate these input devices were relatively. Finally, participants also filled out a questionnaire that rated each input device on how fast, accurate and easy-to-learn they were in their experience. Beaton et al. [26] applied a variant of Fitt's Law for comparing mouse, trackball and thumbwheel input devices for 3D display workstations.

2.2.2. USER-CENTERED RESEARCH

Research on input devices do not always aim to explore the performance or usability of the device itself, but rather what it can tell about the user. Our framework aims to describe a workflow for doing research with input devices within VR environments in general. Input devices can be studied to measure a user's reaction to the application itself and what they're experiencing, which is useful for games in particular. For example, Sykes and Brown [27] measure emotion through the pressure with which users depress the buttons on a gamepad. Input devices can also be used to evaluate a user's physical or mental abilities. Mahmud and Kurniawan [28] use various input devices (mouse, touch screen and tablet with stylus) with older people to measure decline in cognitive ability as they age.

2.2.3. INTERFACES

When doing research on the usability of input devices, the interface with which you interact with the target application is of great significance. The most common interface is the simple 2D graphical user interface, where users can select, drag, and scroll 2D interface elements. For VR, spatial interfaces become interesting to explore, where the user is able to move through a 3D environment. Bowman et al. [29] provide a survey of usability evaluation in virtual environments, where the user is expected to move through the physical environment while performing tasks.

2.2.4. EVALUATION METHODS

Bowman et al. [29] compiled a list of usability evaluation methods that have been applied in the context of virtual environments:

- Cognitive Walkthrough: stepping through the common tasks that a user would perform and evaluating the interface's ability to support each step.
- Formative Evaluation: an observational and empirical evaluation method that presents users with tasks and collects informal (user reactions, comments) or formal (task time, error rate) data.
- Heuristic Expert Evaluation: combine data from several usability expert evaluations of the interface, not involving representative users.
- Post-hoc Questionnaire: a written set of questions used to obtain demographic information and interests of users. Can be more consistent than an interview.

- Interview / Demo: a more in-depth interview with the user that can obtain more information than a questionnaire, at a deeper level of detail.
- Summative or Comparative Evaluation: similar to formative evaluations, but instead compares two or more configurations of the user interface.

They identify two major approaches to evaluating usability methods: the testbed evaluation and sequential evaluation. In the testbed evaluation the methods are evaluated in a generic context instead of a specific application. The sequential evaluation on the other hand employs applications-specific guidelines and domain-specific user tasks. The testbed evaluation method is considered a good way to develop a more general understanding of usability methods, while sequential evaluation can be useful to produce an interface tailored to a specific application.

2.3. TOUCHPAD RESEARCH

In chapter 4 we will describe the design of an experimental setup for evaluating the touchpad in VR, using our framework as basis. Here we will first look into previous research concerning touchpads for similar applications. Previous research that used the touchpad rarely did so in combination with virtual reality devices. However, there are several examples where it was combined with an AR (augmented reality) system, which has similarities with VR in terms of constraints and mobility. Feiner et al. [30] explored the concept of combining a pen-screen with their mobile AR system in the early days of mobile computing, highlighting the various challenges with displays, software and tracking. On the software side it's interesting to look at how touchpads could work with 3D interfaces. For example, Medeiros et al. [31] use a touchscreen tablet, which can target a virtual object within a 3D environment and then manipulate it using common gestures. Similar solutions were used by Wang and Lindeman [32], who present a 3D user interface system with a non-occlusive HMD and an arm-mounted tablet to use as an input solution along with a wand. Mine et al. [15] build a system to control VR user interfaces using hand-tracked touch surfaces for an immersive 3D modelling application, and share how they adapted their interface design to this new input system.

When combining VR with multi-touch touchpads, we want to let the user to interact with the virtual world by performing finger gestures. When compared with touchscreens, one notable difference is the 'direct' and 'indirect' manipulation of elements, as in onscreen elements are either directly under your finger, or you move objects on a different screen using finger motions on the touchpad. Nagashima et al. [33] explore this aspect by using a touchpad to manipulate virtual objects in an AR environment. They compare direct (absolute) and indirect (relative) movement of virtual objects. Their findings suggest that the direct method worked best for selecting objects, but still recommended the indirect method because it allowed the user to look at the actual AR objects. Also the difference in performance of the two methods becomes smaller as the touchpad surface becomes smaller. Knoedel and Hatchet [34] explored similar RST (rotations, scaling and translations) multi-touch techniques for the touchpad, and how much difference in user performance there is between direct and indirect touch interactions. They found that even though direct touch has faster completion times, indirection interaction has better efficiency and precision.

3

A FRAMEWORK FOR EVALUATING INPUT DEVICES FOR VR APPLICATIONS

This chapter discusses the conceptual design of a framework that assists with evaluating input devices in virtual reality applications. In the first section we describe the various elements that the framework consists of. These elements are then detailed in the following sections.

3.1. ELEMENTS OF THE FRAMEWORK

The goal of this framework is to construct an experimental setup that can be used to evaluate various input devices in VR. The target of the evaluation can vary, it can be about the device's usability or performance, or to evaluate the user's abilities with the device. Figure 3.1 shows a global overview on how the framework is structured. The VR system, input device characteristics, VR application and relevant metrics will be discussed in more detail in the coming sections.

3.2. VR SYSTEM

The VR system is a combination of the head-mounted display, tracking system, and computational platform that simulates and renders the 3D environment. Since the focus of this framework is the evaluation of the input device, we assume the VR system consists of off-the-shelf products. This allows us to discuss some of the constraints and how it affects our experimental setup.

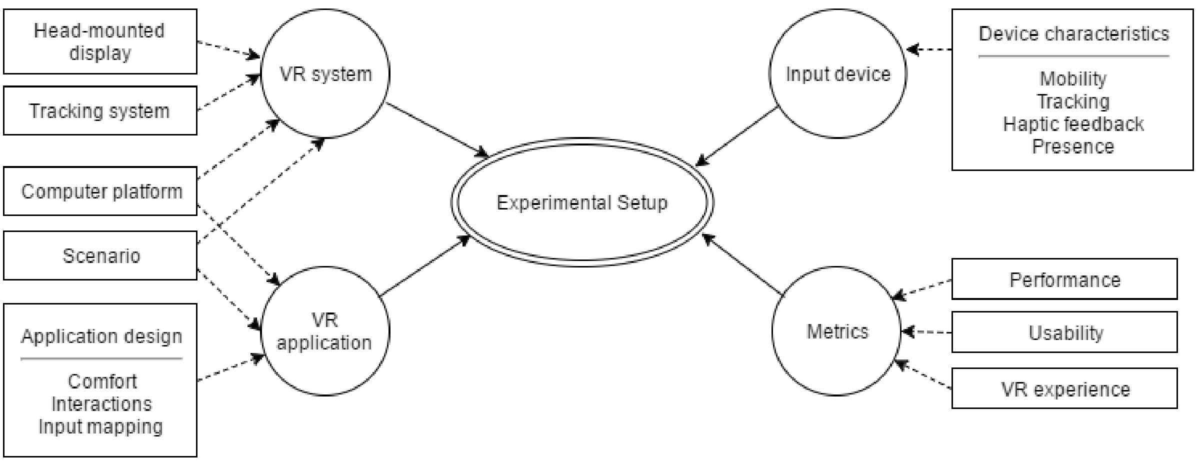

Figure 3.1: Global overview of the elements of our framework.

3.2.1. HEAD-MOUNTED DISPLAY

Modern HMDs on the consumer market have several aspects that affect how the VR system can be used, and the types of VR applications it supports. We look at the following:

- Field-of-view and resolution: this determine how well the user can spot objects in his virtual surroundings and how well the user can discern small visual details (like for example, small text).
- Tethered or untethered: tethered devices are usually connected to more powerful computers, while untethered devices allow for greater freedom of movement.
- Camera: a pass-through camera can allow the user to view his/her environment without adjusting the HMD, or be used for computer vision tasks.
- Eye-tracking: this technology can be used for all sorts of additional interactions within VR, like providing context for input actions.

3.2.2. TRACKING SYSTEM

Tracking systems for modern VR are, perhaps surprisingly, not that different from each other in terms of tracking quality. This is because the tracking needs to be practically indistinguishable from natural movement to our eyes before we subconsciously accept it, without getting feelings of nausea. Any sub-par implementation would be an unviable product. We can still identify a few notable differences between tracking systems in terms of features:

- Positional tracking: while on-board 6-DOF (degrees of freedom) IMU for small electronic devices is pretty commonplace (for example in smartphones), these sensors are usually subject to drift. This is especially noticeable in VR since the user's position and rotation in the physical world will gradually go out of sync with those in the virtual world. High-end VR systems commonly utilize external tracking sensors to correct for this drift through sensor fusion. Mobile VR devices limit themselves to rotational tracking to keep their position in the virtual world predictable.
- Tracking space: the physical volume in which the user is tracked. This mostly concerns tracking systems that use external tracking sensors. Even when the system supports a large space, an experimental setup that attempts to mimic conditions of the average consumer should not assume that the full extent of this space is available.
- Modular device support: as briefly touched upon in section 1.1.3, some tracking systems can inherently support a large number of tracked devices with minimal computational overhead. This allows the user to add additional markers or sensors to other objects in his physical environment, and have these tracked in the virtual simulation. This is especially interesting in the context of our framework, since having an input device spatially tracked in the VR application can open up a whole new range of interactions.

3.2.3. COMPUTER PLATFORM

Rendering detailed virtual environments, especially at the resolutions and refresh rates required by the HMD, is still quite challenging to do even with the exceedingly powerful hardware we have in today's smartphones. Heat produced by the device is still on of the main issues of small electronics (see section 1.1.2). Desktop platforms have less of a problem since these can disperse heat more quickly and efficiently, but come with the drawback of limited mobility.

3.2.4. SCENARIO

The scenario is the use case that serves as basis for the evaluation and determines the type of VR system that are viable for the experimental setup. For example, in an office environment you can expect users to have access to more capable computer platforms compared to when they're travelling by plane.

3.2.5. ASSESS VR SYSTEM

We described several elements in the previous sections that are relevant to assessing the VR system for an experimental setup using our framework. To illustrate how this assessment can be performed, we take the Oculus Rift VR system launched in 2016 as an example.

EXAMPLE: OCULUS RIFT

The HMD has about a 90 degrees field of view. The display resolution is around 1080x1200 pixels per eye. For reference, to achieve 'retinal resolution' (the resolution at which a human with perfect vision can discern details) at a similar field of view, the display would have to feature around 16000x16000 pixels [35]. This means that small text or distant objects can be hard for the user to see. The HMD is tethered, and does not provide a camera or eye-tracking.

The tracking system uses external cameras for full 6-DOF spatial tracking, and is mainly recommended to be used in a 'forward facing' setup where the user does not move from the same spot, and faces in the same general direction. Placing objects behind the user in the virtual world could be problematic. The system does not provide a way to track third-party devices.

The Rift only supports desktop and laptop platforms, requiring high-end processor and graphics processor components according to the recommended specifications. This means this VR system is mainly suited for (home) office setups.

FUTUREPROOFING

This part of the framework is the most likely to need updating in the future, as technology is rapidly progressing. Computers are able to perform better in increasingly smaller packages. Advances in computer vision could enable solid positional tracking without the need of for external tracking sensors. The visual quality of HMDs is improving with each generation of products. Tethered devices might soon become history too. For now these elements remain relevant to assessing the viability of a VR system.

3.3. INPUT DEVICE CHARACTERISTICS

To evaluate an input device within VR, we need to make a surface assessment of its strengths and weaknesses. These characteristics affect how compatible it is with the scenario, VR system and VR application used in the experimental setup. We identify four characteristics for the input device that are important in the context of immersive VR. Namely:

- Mobility: relevant for mobile VR, see section 1.1.2.
- Tracking: spatial tracking is integral to VR. The input device can leverage or affect the tracking system.
- Haptic feedback: input devices can be used to provide feedback to the user, by stimulating the user's sense of touch. This only becomes more important in the context of VR.
- Presence: the ability of VR to fully immerse the user in the virtual environment is one of its biggest selling points. The input device could enhance this even further.

,

We assign ratings to each, allowing us to create a simple overview of the input device's expected utility within a scenario:

- Single plus (+): device characteristic appears suited for (and is usually designed for) this scenario.
- Single O (o): device characteristic does not appear to give any particular advantage or disadvantage for this scenario.
- Single minus (-): device characteristic is at a disadvantage or poorly suited for this scenario.
- Double minus (- -): input device is incompatible with this scenario because of this device characteristic.

We will detail what each rating means for each device characteristic in the following sections.

3.3.1. MOBILITY

The mobility of the device is determined by whether the device is:

- Handheld or placed on surface
- Wireless, wired, or otherwise dependent on another device in the vicinity
- Heavy or easy to carry for prolonged sessions

The ratings assigned to assess the mobility of a device have the following meaning:

- Single plus (+): device is easy to move, and can be handled for prolonged sessions in this scenario.
- Single O (o): device is not particularly mobile but would not likely underperform in this scenario.
- Single minus (-): device impedes the user in this scenario, but can still be used.
- Double minus (- -): device cannot be practically used in this scenario.

3.3.2. Tracking

Input devices can support various forms of tracking in the context of VR.

- Onboard inertial measurement unit (IMU): Often only supports rotational tracking. Positional tracking is usually relative (prone to drift over time) or too inaccurate (GPS) for VR.
- External tracking: usually combined with onboard IMUs. The external tracking system can be, but does not have to be, the same as that of the targeted VR system.
- No tracking: common for input devices that weren't designed with VR in mind.

The ratings assigned to assess the tracking of a device have the following meaning:

- Single plus (+): device is natively tracked by the VR system in this scenario.
- Single O (o): device can be partially tracked (onboard IMU for example), or could be mounted to another tracked object (like the HMD).
- Single minus (-): device is not tracked.
- Double minus (- -): device negatively affects the tracking system and should not be used in this scenario.

3.3.3. Haptic Feedback

The importance of haptic feedback for input devices was described in section 2.1. In terms of haptic feedback devices have the following options:

- Touch surfaces, buttons and other physical controls: these give feedback through simply feeling the action of touching or pressing on the given control.
- Vibration motors: usually used by events within the application to give feedback to the user, or to enhance the feedback of the user's input actions.
- No haptic feedback: common for input devices that use optical body tracking and as such do not have direct contact with the user's body.

The ratings assigned to assess the haptic feedback of the device have the following meaning:

- Single plus (+): device supports haptic feedback like vibration, rumble or other physical stimuli to the user based on input.
- Single O (o): device provides basic haptic feedback in the form of buttons or touch surfaces as points of contact.
- Single minus (-): device provides no haptic feedback.
- Double minus (- -): device provides feedback, but in a form that makes it impractical for this scenario.

Table 3.1: Example of rating device characteristics in an office environment with a mobile VR system

Input Device	Mobility	Tracking	Haptic Feedback	Presence
Mouse and Keyboard	-	-	o	-
Gamepad	+	o	o	o
Remote	+	o	o	o
LeapMotion	o	o	-	+
Kinect	- -	o	-	+

3.3.4. PRESENCE

Enhancing the feeling of 'presence' as described in section 2.1 is often key to the success of an input device within VR applications. The characteristic is harder to describe and consists of elements from the other characteristics. Our approach is to ask the following questions:

- Does the input device impede the user's movements in any way the user does not expect?
- Is the device within the same virtual space as the user and tracked with great accuracy?
- Does the device provide realistic feedback to events within the virtual world?

The ratings assigned to assess the added 'presence' of the device have the following meaning:

- Single plus (+): device likely improves the VR experience.
- Single O (o): device does not particularly add to the VR experience, but does not hurt it either.
- Single minus (-): device is detrimental to the VR experience.
- Double minus (- -): device completely breaks the immersion of the VR experience.

3.3.5. ASSESS INPUT DEVICE CHARACTERISTICS

Using the above device characteristic ratings, we can create an overview of how compatible an input device would be with a specific scenario. As an example, we rate several input devices mentioned in section 1.2. Our scenario in this case is located at in an office environment, at a mobile workplace, with a mobile VR system (as described in section 1.1.2). See table 3.1.

From this table you could identify that, for example, the gamepad and remote are well suited for this scenario given their advantages in mobility, while a Kinect system would be impractical.

3.4. VR APPLICATION

The VR application is central to the experience. Designing for interaction in VR has several differences from those of classic computer applications. The details and purpose of the application are purposefully left ambiguous in order for our framework to support a wide variety of applications. For this reason, the following subsections generally target how VR affects application design.

3.4.1. APPLICATION DESIGN

Here we describe how VR dictates several best practices for comfort, the types of interactions that are common in VR, and the points to consider with input mapping. While there are many other aspects in which the use of VR affects the overall application design, this goes beyond the scope of this thesis.

COMFORT

Motion sickness has been one of the biggest problems for VR system- and software designers to tackle. The inherent issue is the mismatch of visual and vestibular input that humans normally use to balance themselves [36]. The severity of the effect can vary per person, some users feel fine and can use VR systems for extended periods, while others can be very sensitive to any form of vestibular mismatch. The community of VR software developers have been sharing a variety of techniques [37] to maximize the comfort for the vast majority of users.

Our framework supports VR applications that take at least the most important guidelines into account:

- Maintain consistent application performance: modern consumer VR hardware uses displays with above-average frame refresh rate and very low latency processing of tracking date. The VR application should keep up with the hardware and avoid drops in performance wherever possible.
- Maintain sensory consistency: ideally, the user's head should always be in control of the absolute position of the virtual camera. If the application requires the user to move through a large virtual environment, linear acceleration or teleportation is the most common solution. Avoid tilting, rolling or bouncing the virtual camera.
- Maintain a consistent reality: there are a variety of ways that an application can 'break' a user's VR experience by failing to meet the user's expectations. For example, by drawing objects that are far away in front of objects that are close.

INTERACTIONS

Interacting with a VR application can differ from traditional computer applications in several big ways. Common 2D graphical user interfaces have to be adapted or replaced depending on the VR system and input device. The following interactions are generally considered 'intuitive' in VR, given the right input device:

- Pointing: using a tracked device, point at the virtual object of interest to provide context to an action.
- Looking: similar to pointing, but using the user's gaze direction.
- Grabbing: common for tracked hand controllers, the ability to reach out in the virtual space and touch an object.
- Pressing: buttons or touch surfaces within reach of the user's hand that can trigger discrete actions, usually in concert with the other interaction methods.

INPUT MAPPING

Mapping input actions to specific application interactions is an important step. Not all mappings are necessarily intuitive in VR. For example, when performing a grab interaction, a button operated with the index finger might be more intuitive for the user compared to using a button operated by the thumb. The correct mapping is highly dependent on the input device and VR application. Because of this it is difficult to provide a generic case.

3.4.2. COMPUTER PLATFORM

The computer platform that the VR application aims to run on can affect its design. While desktop platforms are capable of rendering rich virtual worlds or perform computationally heavy simulations, mobile VR platforms are more limited and require the application to be designed within performance constraints.

3.4.3. SCENARIO

We described how the scenario affected the VR system earlier. It also determines the type of VR application used in the setup. This can be, for example, using a VR architecture tool for engineering buildings in an office environment, or a communication app to talk with contacts in VR while travelling via public transport.

3.4.4. ASSESS VR APPLICATION

The elements described above can be one of the more complex elements of building the experimental setup. Since few examples of 'mainstream' VR applications exist today, designing a VR application requires iteration and regular user testing to reach a state in which it is useful and representative for evaluating an input device, and could likely be considered a research topic on its own. Instead of a definite assessment of a VR application for the experimental setup, its design should be considered a variable in the experiment and discussed along with the results.

3.5. METRICS

Most of the common methods for evaluating input devices, as explored in section 2.2, are relevant when doing the evaluation with a VR system. We condense this to three metrics for the framework, namely performance, usability, and experience:

3.5.1. PERFORMANCE

Speed and accuracy are the two major components of input device performance. How fast is the user able to complete the task, and how many errors did the user make while doing so. Using this data, one can construct a performance rating to compare different input devices, or different users.

Several options for quantifying this metric are:

- Time span recording: record the time between the start and end of the task.
- Distance recording: how far did the user have to travel with the input device.
- Error recording: how many errors did the user make relative to the amount of task completions.

In section 2.2.1 we described several methods for calculating a performance rating. The chosen method depends on the type of evaluation, but the common factors that we outlined above are often the same.

3.5.2. USABILITY

The usability of a device can be described as the ease of use and learnability. How long does it take before the user is able to operate the device with confidence and to its full potential? Like most previous work, we quantify this metric through questionnaires, letting the user assign a rating that describes their own perceived performance. Questions can be tailored to particular characteristics of the experience being in VR, but that depends on the focus of the evaluation.

3.5.3. VR EXPERIENCE

Quantifying the VR experience is similar to usability, since it's highly subjective to the user. We identify several factors that contribute to the VR experience that can vary between different input devices:

- Presentation: did the user enjoy the representation of the input device inside the VR environment (if any)?
- User immersion: did the user feel engaged by the task and did the input device contribute to that?
- Presence: did the user feel 'present' inside the environment?
- Spatial reasoning: the fully immersive 3D environment allows for more in-depth research of the user's spatial awareness.

3.5.4. ASSESS METRICS

The metrics to choose as the dependent variables in your experiment depend on the research you're performing. When, for example, you take the Leap Motion Controller [7], you could compare how fast users from different age ranges are able to solve tasks using performance metrics, or you could compare how users found their VR experience improved compared to other input methods. In essence, this does not differ much from the common approach to non-VR input device research. VR mainly adds more factors to account for.

3.6. DISCUSSION

Starting with a scenario and desired input device, this framework provides assessments for the choice of VR system, input device, VR application and metrics. Together this should prove sufficient for designing an experimental setup for evaluating an input device in VR. While this framework alone is not an fully exhaustive in terms of considerations for setting up an experiment, it's intended to serve as a set of re-usable guidelines to standardize research efforts into input methods and virtual reality.

4

Design of a experimental setup for evaluating the Touchpad in VR

In section 1.2.6 we described the reason for using the touchpad as our test case for the framework, supported by previous work explored in section 2.3. We further support our choice of the touchpad by briefly discussing a preliminary study. After that we describe our scenario, followed by the design of the experimental setup supported by our framework from chapter 3. Finally we will describe the design and implementation details of the VR application.

4.1. Preliminary study

Early on in this thesis project we performed a preliminary study that compared several different input methodologies, namely the mouse, gamepad, touchpad and head-tracking (gaze-based input). The goal of this comparison was to test the implementation of a VR image browsing application and identify strengths and weaknesses of the aforementioned input methods. The report of the study can be found in appendix C. While the touchpad was not a clear winner, it still did well in terms of performance. This prompted us to further explore the touchpad in VR, which lead to this research.

4.2. The Scenario

The touchpad is interesting to evaluate as a VR input device because of its mobility, general availability and ease of use. We hypothesise it will mesh well with mobile VR use cases, which so far have relied on gaze-tracking or a simple remote for input. Using a touchpad in this scenario would greatly expand the possible actions for the user, and thus allow for more complex application controls. As discussed in section 1.2.6, there are several types of touchpads, of which a standalone touchpad in the form of a re-purposed smartphone or tablet touchscreen would be the most interesting to explore given its wide availability. The results would be directly applicable to the current consumer market, without the need for introducing new hardware.

With the choice of the touchpad clear, we need to identify a type of application to simulate in the experimental setup. The preliminary study used an image browsing application, since managing virtual datasets is a common task on any computing platforms, and requires many interactions that are common in other productivity tools. There are no off-the-shelf image browsing applications for VR, so we will implement our own. Details can be found in section 4.5.

In summary, our scenario now consists of using a touchpad, in the form of a tablet or smartphone, used for a VR image browsing application. For the VR system, we will use the Oculus Development Kit 2, which supports nearly all features found in modern VR systems. As for our metrics, we are mainly interested in the performance and usability of the touchpad within our scenario. Now that our scenario is complete, we can construct the experimental setup using the framework described in chapter 3.

4.3. Applying the framework

In this section we will go over the four elements of the framework, with the scenario we described above. We assess the VR system, input device characteristics, VR application and metrics to motivate the design of our

Table 4.1: Touchpad characteristics ratings

Input Device	Mobility	Tracking	Haptic Feedback	Presence
Touchpad	+	-	o	o

experimental setup.

4.3.1. VR SYSTEM

The VR system we choose for our scenario is an Oculus Rift Development Kit 2. Using the points laid out in section 3.2 we create an assessment.

HEAD-MOUNTED DISPLAY

The HMD features a 100 degree field of view, with a single 1920x1080 display. The relatively low resolution means small text will be hard to read so this should be avoided in our application design. The HMD is tethered so the user is expected to stay within the same area.

TRACKING SYSTEM

The tracking system consists of a single camera positioned in front of the user, providing 6 DOF spatial tracking. However this only works when the HMD is within the camera's view, meaning the user can turn his/her head about 90 degrees in either direction assuming they stay stationary facing their body in the direction of the camera.

COMPUTER PLATFORM

The Rift only supports Windows desktop platforms. This means we will not likely run into any performance constraints, when using a computer that satisfies the recommended specifications.

SCENARIO

Since the computer platform is required to be a PC desktop, and the HMD is tethered, our setup isn't as 'mobile' as the scenario we're simulating. However the experiment doesn't depend on the user being mobile to obtain results.

4.3.2. INPUT DEVICE CHARACTERISTICS

Following the rating system described in section 3.3, we score the different characteristics of the touchpad in table 4.1. We detail the ratings in the following subsections.

MOBILITY

The touchpad in our scenario is given a positive rating because it is easy to carry, and can be used for prolonged sessions. Communicating input actions from the touchpad hardware to the computer system running the VR application happens wirelessly.

TRACKING

The touchpad is not tracked within the application in our use case, thus resulting in a negative rating in this case. Though it is possible to use the inbuild IMU of the touchpad hardware we use for 3-DOF rotational movement tracking, we choose not to use it to focus on the touch interactions.

HAPTIC FEEDBACK

Haptic feedback comes in the form of the touch surface. The user is able to feel what kind of actions they are performing. Beyond that no other haptics are used.

PRESENCE

We do not consider the touchpad to add much to the VR experience itself, nor does it impede on it. The touchpad is itself not visible within the virtual environment in this particular use case, the user is expected to operate it 'blindly'.

4.3.3. VR Application

The VR image browsing application is specifically developed for our experimental setup. Instead of using the points provided by the framework in section 3.4 to assess an existing solution, they will serve to motivate the design of our own VR application.

Comfort

An image browsing application should support a wide demographic. This makes it important for the VR experience to be as comfortable as possible to prevent any form of motion sickness. The application will thus minimize unnecessary movement where possible, by having a stationary user location and calm background scene. The user's head is in control of the camera at all times. This will prevent vestibular mismatch, the primary cause of VR motion sickness.

Interactions

The touchpad is the main input device for our image browsing VR application, but additionally we will make use of the gaze-tracking input provided by the tracked VR HMD to give context to touch interactions. This is motivated by the results of the preliminary study (appendix C). For example, tapping on the touchpad allows the user to perform an action on the interface element he/she is currently looking at.

Input Mapping

The touchpad provides several interactions to map to, specifically tapping, swiping, and multi-touch actions. Tapping can be used to trigger discrete actions similar to button presses. Swiping can be used for movement over an axis. Multi-touch actions can be used for complex actions such as pivoting one point around another. We will detail the exact mapping of interactions in section 4.4.

4.3.4. Metrics

For this evaluation we're mainly interested in the performance and usability of the touchpad within a VR application. We record data on performance by doing time recordings of specific tasks within the application that require the touchpad. For usability, we provide a questionnaire based on the NASA-TLX format, where users are asked to quantify experiences on a scale from 'best' to 'worst'.

4.4. Design concept of the VR application

In section 4.2 we described why we choose to build a VR image browser to evaluate the touchpad. The challenge here is that there is little in terms of established standards on which to base the design of the application, however there is some related work we can explore. We will describe and motivate the design concept of the VR image browser in the following subsections.

4.4.1. VR System

The VR system affects several design decisions we make for the VR application. We explored these constraints in section 4.3.1. To summarize: the tethered HMD has 6 DOF, but only when looking in the direction of the tracking camera. This means the user is expected to remain stationary.

4.4.2. Image Browser

The goal of our image browsing application isn't to be a fully featured product, but it should provide the user with functionality and tasks that are representative of a commercially developed image browser application. This means we need to design a basic graphical user interface for our application. We split it in two modes: the *image wall* in which the user navigates the current list of images, and the *image inspector* in which the user can examine a single image in detail.

Image Wall

The image wall is intended to present a large image dataset to the user. When compared to a traditional display, the user is not restricted by the size of the screen when in VR. We choose to represent the image dataset using a 2D grid that wraps around the user, similar to the inside of a cylinder. We expect the user to remain stationary and forward-facing. If the user wants to see content outside of his/her view, he/she can either look around or drag the image wall to rotate the correct images into view.

The layout of the image wall is set up with comfortable viewing angles in mind. A presentation from the Samsung Developer Conference on VR design [38] states that research showed humans can comfortably

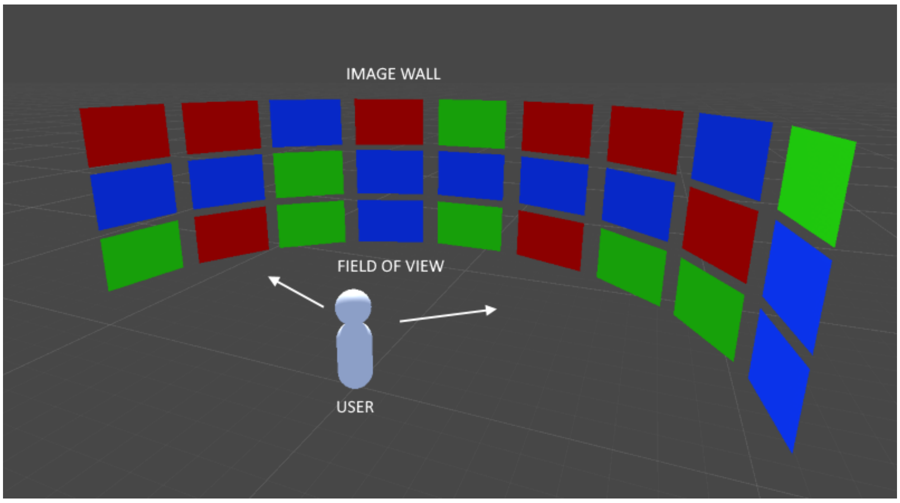

Figure 4.1: Concept of the image wall. The coloured panels represent images.

rotate their heads 30 degrees horizontally on either side, and 20 degrees vertically. In addition, we take some lessons from Mike Alger's research on VR interface design [39] for placing content within the virtual space relative to the user. With horizontal head rotation taken into account, the average user can rotate about 77 degrees to either side and see content comfortably and meaningfully. We keep content at least 0.5m away from the user to avoid the user straining his/her eyes too much.

The UI locomotion can still create discomfort even though the user itself isn't physically or virtually moving. UI / UX design research from Oculus [40] suggest that horizontal scrolling content is the most comfortable for the average user, hence we design the image grid so that it rotates horizontally. We also leave gaps between images instead of showing a solid wall, so the user can see the stationary background at all times.

Collections

An image browser commonly sorts images into different collections. We represent these collections with 'folder' objects. The user is able to open a collection and display its dataset on the image wall, or move images from the currently opened collection to a different one.

Image Inspector

The user is able to select an image from the image wall and focus on it in what we call the 'image inspector' context. In this context the user can inspect, rotate or scale the image. While these interactions don't serve a direct purpose for the user, they allow for more complex input actions to be tested with the touchpad.

4.4.3. Input Actions

Here we describe the available input actions that the image browser application supports.

Gaze Tracking

The user's gaze direction is derived from the rotation of the HMD. This is used to provide context to other input actions.

Touchpad Gestures

We describe a set of gestures that the application recognizes can be used to interact. All gestures use at most two pointers (i.e. two fingers), since established guidelines and practices show that any gesture that require 3 or more fingers are usually unintuitive or inaccurate.

The gestures are:

- Tapping (short press)
- Multi-tapping (two or more presses in quick succession)
- Tap and hold (long press)

- Swipe (press down a finger and quickly swipe in a direction)
- Pinch (press down two two fingers and move one towards the other in a straight line)
- Stretch (press down two fingers and move one away from the other in a straight line)
- Pivot (hold and press one finger, pivot a second finger around it in a circular motion)

4.4.4. INTERACTIONS AND MAPPING

This is an overview of all the interactions available to the user within the image browser, and what input actions map to them respectively:

- Target: move a crosshair to target images.
  - Mapped to: headtracking. Crosshair is always in center of the user's view.
- Select: selecting the targeted image or collection.
  - Mapped to: single tap.
- Multi-select: switching to multi-select mode and selecting multiple images.
  - Mapped to: tap and hold for one full second on an image. Image grid will show you're in 'multi-select' mode where you can select other images with single taps. Tapping an image that is part of your selection again deselects it. Tapping a collection will move all selected images to that collection, and exit multi-select mode. Alternatively, double tap to go out of multi-select mode (deselecting all images).
- Navigate: rotating the image grid to bring other images into your field of view.
  - Mapped to: swipe finger left or right when in the image grid.
- Manipulate: scaling and rotating selected images
  - Mapped to: when in the image inspector, drag left or right to rotate image around its pitch and yaw axis. Using the two-finger pivot gesture you can rotate around its roll axis. Using the pinch and stretch gestures you can scale the image up or down.
- Cancel: used to exit multi-select mode while in the image grid context, or to exit the image inspector context.
  - Mapped to: double tap.

The decisions for these mappings were mostly based on intuition, over the course of implementing the VR application. Comparing different mappings in our experiment would greatly increase the amount of variables, which is beyond the scope of this project.

4.5. IMPLEMENTATION DETAILS

Here we detail the implementation of the image browser application, along with the hardware and tools used.

4.5.1. HARDWARE

For our scenario we choose an Oculus Rift Development Kit 2 as our VR system, which supports low latency 6-DOF tracking as required by the design concept. The application runs on a desktop to which the VR HMD is connected. The main specifications of this desktop include a Intel Core i5 4570S processor, 8GB RAM and a AMD R9 280X graphics card.

The touchpads used in our evaluation come in the form of a Samsung Galaxy S4 Mini 4.3" smartphone and a Nvidia Shield 8" tablet, both using the Android operating system. Both devices run specialized apps that communicate touch input back to the desktop over wifi.

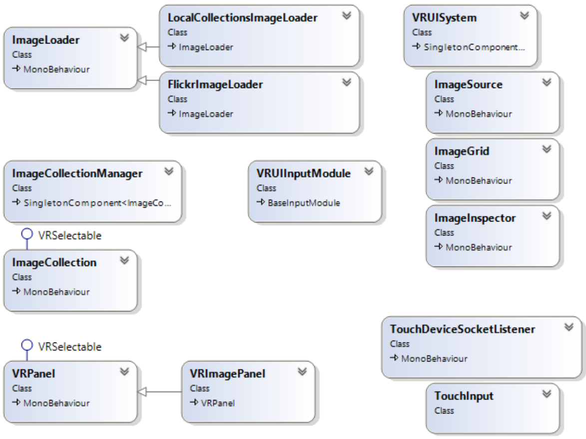

Figure 4.2: A simplified class diagram of the VR image browsing application.

4.5.2. TOOLS

The VR image browser application is build using the Unity game engine (version 5.4.2f1) [41]. Unity has in-build support for several VR systems, vastly simplifying the process of adding support for our hardware. The code is written in C# using Visual Studio 2015. The VR system also requires the Oculus Runtime to be installed on the desktop.

The specialized app created for the touchpad devices is also made using Unity. The minimum supported version of Android is 4.0.

4.5.3. APPLICATION STRUCTURE

There are two applications in our evaluation setup, the main VR image browser and the specialized app for the touchpad devices. However the latter is fairly simple: it records touch input and sends these events to the image browser application using UDP. Next we will detail the image browser application in more depth.

CLASSES

Figure 4.2 shows a simplified overview of the classes in the implementation of the VR image browser, leaving out various utility classes, and interfaces provided by Unity. Next are some descriptions of the roles each of these classes:

- **ImageLoader**: base class for image loading external image datasets into the application.
- **LocalCollectionsImageLoader**: image loader that loads images from a folder structure on the local disk.
- **FlickrImageLoader**: image loader pulls images from the online photo service Flickr and loads them into the application.
- **VRUISystem**: the core management class of the application. It manages the various contexts (image grid, inspector), preprocesses input events, and various application events.
- **ImageSource**: manages the image source context, in which the user can select a dataset source to load images from.
- **ImageGrid**: manages the image grid context, which displays the image wall and processes the various interactions you can perform with it.
- **ImageInspector**: manages the image inspector context, in which the user can manipulate a single image.
- **VRUIInputModule**: input module used by Unity's GUI system to add support for gaze tracking.
- **ImageCollectionManager**: creates and manages the interface elements for all loaded image collections.

Figure 4.3: View of the image browser application from the user's perspective.

Figure 4.4: Overview of the 3D scene in the Unity editor.

- **ImageCollection**: selectable interface element that opens an image collection on the image wall when triggered.
- **TouchDeviceSocketListener**: sets up a UDP client to listen on a specific network port for messages send from the Android app, and then passes it on to be processed.
- **TouchInput**: interface that converts the input data from the android app into simple input events for the rest of the application to use.

4.5.4. VISUALS

Figure 4.3 and 4.4 provide screenshots of the application used in our test setup. A calm stationary background provides a solid reference point for users, to increase comfort. Selectable items in the environment, such as images and the collection folders at the bottom, are highlighted when the user targets them. Due to the relatively low angular resolution of VR HMDs it's important to have text and other interface elements be large enough to be clearly readable.

4.5.5. DATA COLLECTION

The application has a central class, StudyTracker, used for collecting metrics during a user study. When a study is active, it records:

- Filenames of all images loaded into the application, and their associated collection.
- Durations of various tasks, both detail and summarized.
- Application events with timestamps.

This data is then exported into a text file in a human-readable format (Lua script) for our visualization tools.

4.5.6. CHALLENGES

The difficulties of developing the VR image browser were mostly in exploring the uncharted territory of VR UI design. Interactions that you can take for granted on a stationary screen do not always apply to VR, where it's best to avoid any static UI elements in the user's view. Other problems that had to be solved during the course of the implementation included:

- Loading image files from the disk (or web) into game engine dynamically, compressing them into textures, without creating noticeable drops in performance for the user.
- Communicating with an external device via UDP over local network, with a custom data protocol for touch input.
- Abstracting raw touch input into several relevant input actions.
- Matching manipulation of virtual objects with the user's expectations after specific input actions, usually based on the user's view orientation.

4.6. SUMMARY

In this chapter we described the scenario, using the tablet for a VR image browsing application. We then assessed four aspects of the scenario with our framework. Part of this assessment was to support the design of our own VR application, of which we provided the implementation details. This completed the experimental setup for testing the touchpad in VR. We will use this setup for a user study, which is further explained in chapter 5.

5

Evaluating the Touchpad in VR

In chapter 4 we designed our experimental setup, consisting of assessment of the scenario and the implementation of our own VR image browsing application,. In this chapter we will describe how this setup is utilized to run a user study for exploring and evaluating the use of the touchpad as an input device for VR. In the first section we will describe the main goal of this study, and how we aim to achieve that goal. After that we describe the user study setup, the exact metrics we aim to use, the participants, the results, and finally the evaluation of those results.

5.1. Main goal

The main goal of this particular study is to determine if the touchpad is a capable input method for a VR image browser. As stated in section 4.3.4, the metrics we aim to collect are related to performance and usability in this scenario. The following questions will serve as the basis:

- User experience: are the touch gestures intuitive and easy to use? Is it enjoyable for the user to use the touchpad?
- Performance: how fast do users complete tasks within the application? How does this compare between different setups?
- Setup: does the size of the touchpad or the way it's held have effect on user experience and performance? Which setup is preferred?

We will use the application designed in section 4.4, and do a user study that closely resembles the scenario laid out in section 4.2.

5.2. User study

To answer the questions for our study we take the following approach:

- User experience: ask users to quantify their experience by answering a questionnaire (based on the NASA-TLX format). Observe how they operate the input device during the study.
- Performance: the speed with which users perform gestures to complete specific tasks is recorded.
- Setup: have the user operate a small touchpad (smartphone) and a larger touchpad (tablet). While smartphones have been designed to always be held in hand, tablets are commonly positioned on a table. That's why for the larger touchpad we explore the positioning by letting the users operate it both while held in hand (supported on the user's forearm), or positioned on a stationary surface.

This means our independent variable is the setup (small touchpad, large touchpad held in hand, large touchpad on stationary surface). The dependent variables are the task performance speed, the way the input device is operated by the user, and the usability ratings provided by the questionnaire.

Figure 5.1: Photo of one of the participants during the user study.

5.3. Hypotheses

Given our variables, we formulate the following hypotheses:

- The large touchpad outperforms the small touchpad in usability and performance.
- There is a link between the way users operate the touchpad device and usability.
- The touchpad in general will receive favourable assessments for usability.

Our general prediction is that the touchpad provides a good match for the VR image browser in the given scenario. The touchpad is an easy to understand input device, relatively simple to bring along for potential users along with mobile VR systems during regular travel. We do not expect our input mapping in the VR image browser to be ideal yet, since that would require lot of iterative refinement. This could influence the difficulty with which users are able to perform the required gestures. In terms of user experience and performance we expect the smartphone setup to score worse than the tablet, due to its smaller surface.

5.4. Setup

Study participants were seated at a table while wearing the VR HMD. They were asked to perform a series of tasks in the VR image browser (from here on out called a 'run'), with each of the different touchpad setups. Setups were in semi-randomized order for even distribution over all tested user sessions. Each participant completed a 'practice run' with the first device in line, in order to get used to the application and the required tasks. The exact steps in each user session are shown in appendix B. After all tasks were completed, participants were asked to fill out the questionnaire shown in appendix A.

5.5. Participants

We collected data from 21 different participants in total. Given that we have a single independent variable this should provide us with statistically significant results. All participants were volunteering Computer Science students from the TU Delft, ages ranging from 18 to 23 years old. The participants were generally male (only one of female) and right handed (one of them left handed, one of them was mix-handed). All of them used touch devices on a daily basis. While this is not very representative of the wider demographic that would use an image browser, we expect this to work in our favour since it'll avoid possible skill differences between demographics affecting performance results. Nonetheless this should be kept in mind when making statements about the ratings that participants assign to usability in the questionnaire.

Due to network issues the data of the first user was discarded, because it negatively influenced the accuracy with which the VR application received the touchpad input. Additionally one of the run result files ended up corrupted, which is why it will also be ignored in the results. Thus overall the results are based on 20 different participants, with 59 different recorded runs (20 with tablet on table, 19 with tablet in hand, 20 with smartphone). Per run, the user had to do 30 subtasks in the form of marked images (10 in each of the 3 image collections), meaning 1770 marked images were cleared in total. The number of tasks per user was

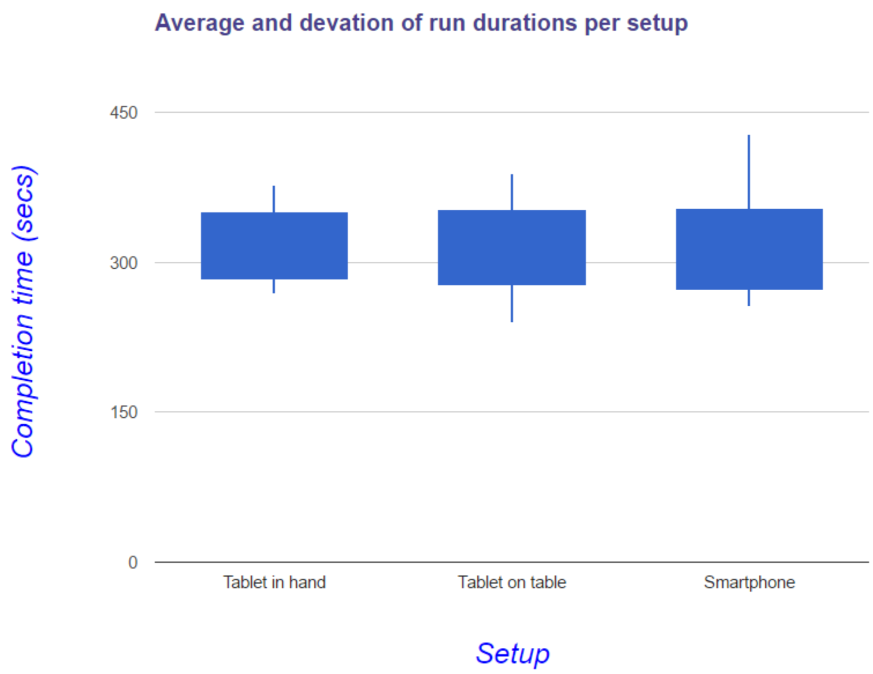

Figure 5.2

decided based on the amount of available images, expected duration of each task (we aimed to have each user clear all their runs within 30 minutes), and the amount needed to produce significant results.

5.6. RESULTS

We used candlestick graphs to summarize our data. The blocks represent the range from average minus standard deviation, to average plus standard deviation. The lines range from the minimum value to the maximum value recorded in our data.

The run completion times per device setup are shown in figure 5.2. The exact averages are 316.2 seconds for tablet in hand, 314.9 seconds for tablet on table, and 313.3 seconds for the smartphone. Standard deviations are respectively 33.9 seconds, 38.0 seconds, and 40.8 seconds.

We also grouped the run completion times in the sequence in which they were completed, independent of setup, to identify if users improved over time. This summary can be seen in figure 5.3.

The average completion times for the task associated with the marked images (rotating / scaling an image to the correct transform), per device setup, are summared in figure 5.4.

The average ratings that study participants gave to the various questions on the questionnaire are shown in figure 5.5, 5.6 and 5.7.

During the study we also collected data on how users physically handled touch devices, such as what fingers they preferred to use for multi-touch gestures, in order to explore the hypothesis about the link between user touchpad handling and performance. Several different 'finger poses' were noted:

- For the tablet in hand, 11 users used two thumbs, 4 users used their index finger + thumb, 3 users used their index + middle finger. One user used both index fingers.

- For tablet on table, 10 users used two index fingers, 6 users used index + middle, 2 used two thumbs, 1 index + thumb.

- For the smartphone, 17 users used two thumbs to interact, 1 index+thumb and 1 index+middle. Part of the overwhelming majority using two thumbs is probably explained by the fact the smartphone had to be held in landscape mode, a position in which the user usually interacts with the phone using both thumbs.

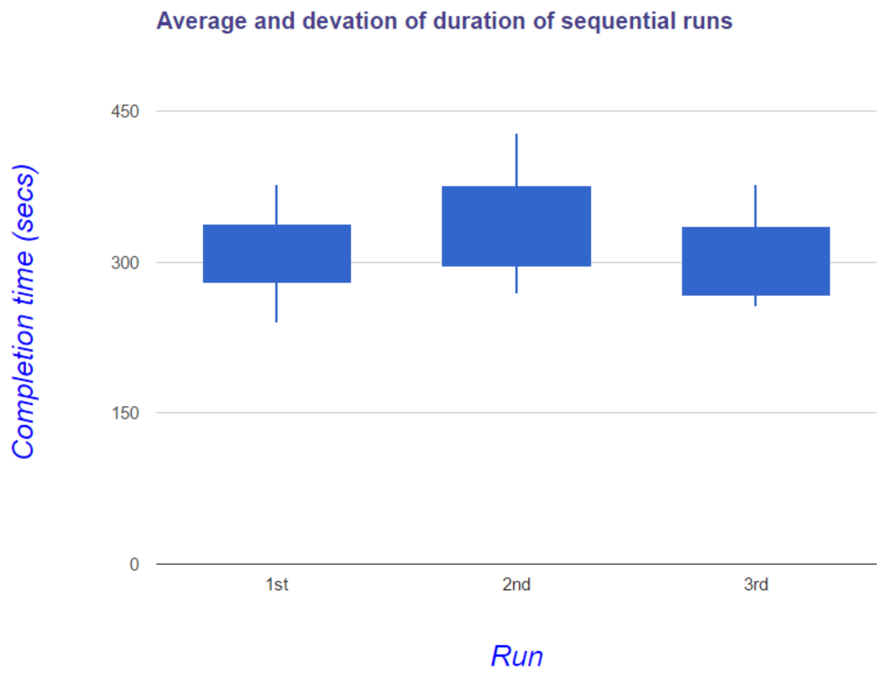

Figure 5.3

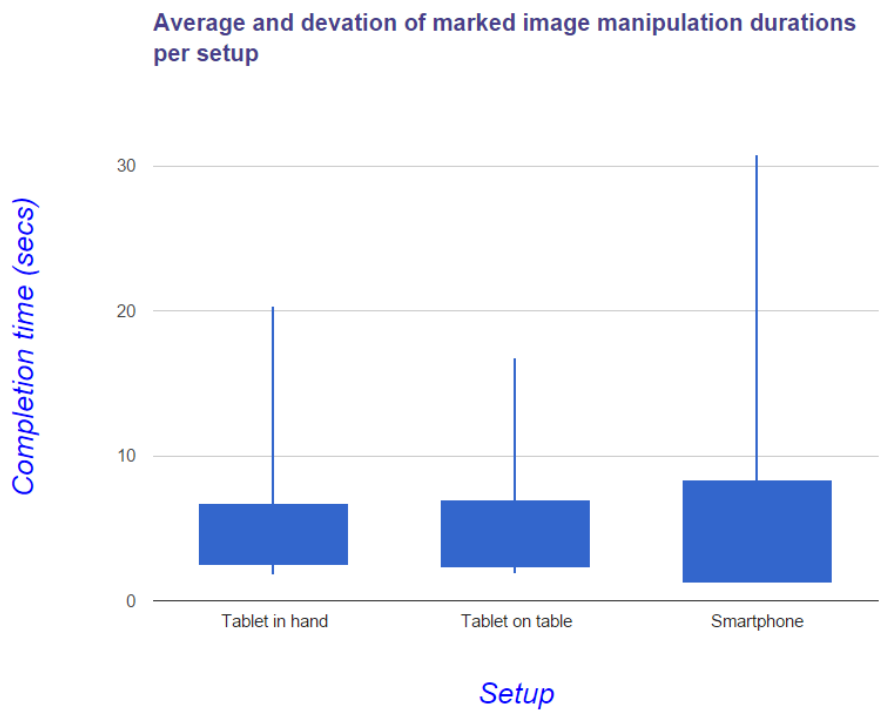

Figure 5.4

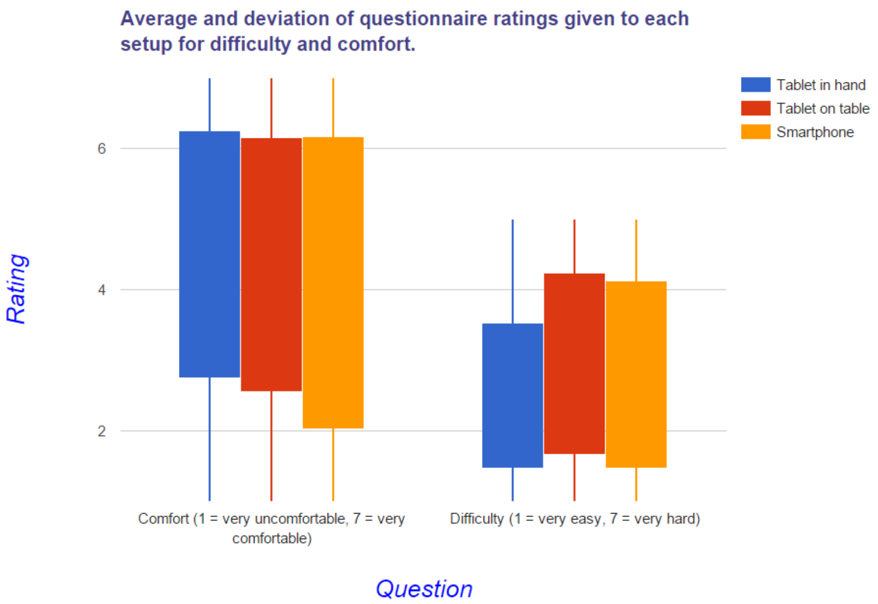

Figure 5.5

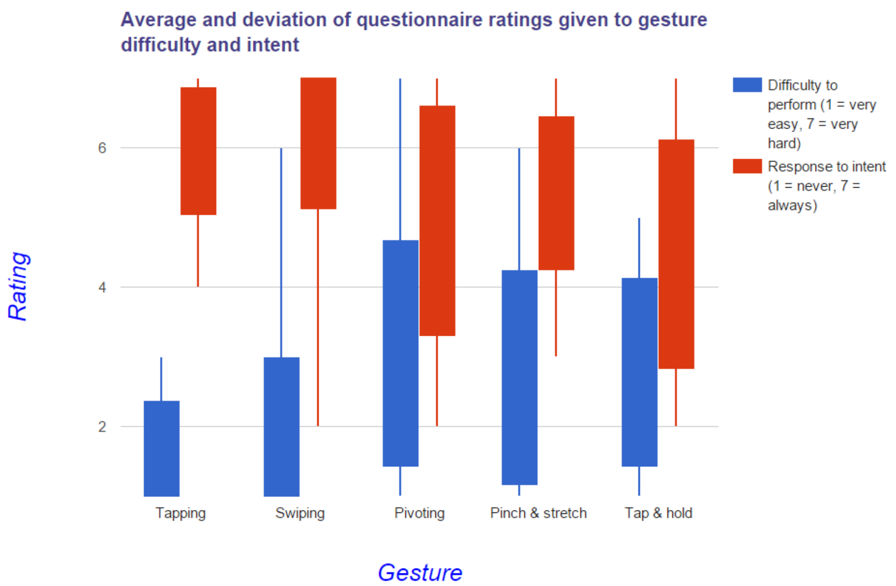

Figure 5.6

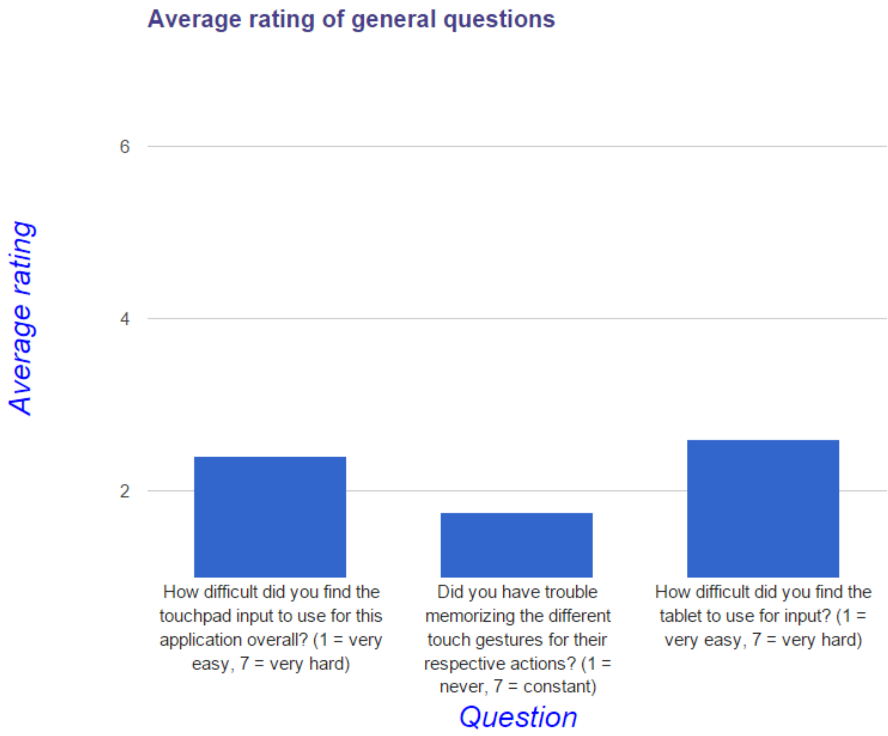

Figure 5.7

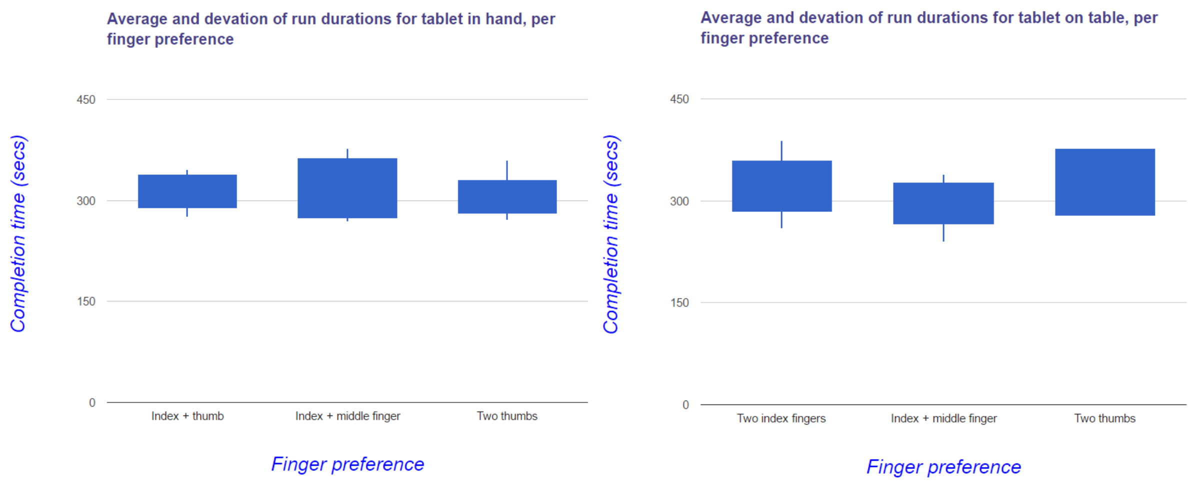

Figure 5.8

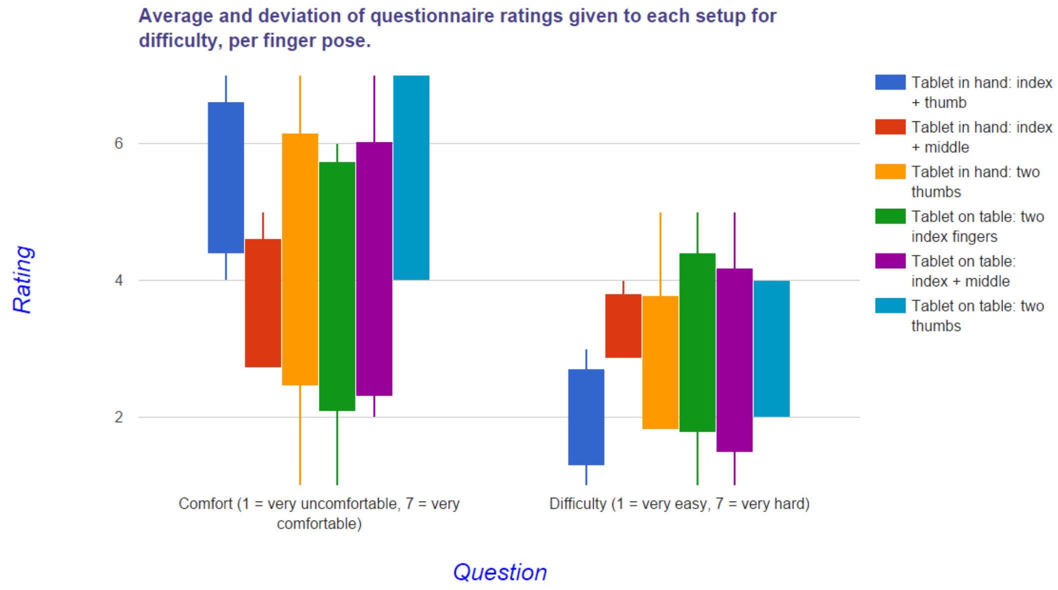

Figure 5.9

To see if this correlated with other results, we split our data for the two tablet setups over the different finger poses that have 2 or more data points. We excluded the smartphone setup since only 2 participants used different finger pose than two thumbs for that particular case. Figure 5.8 shows the difference in recorded completion times based on finger pose. Figure 5.9 shows the difference in ratings for the questionnaire questions about the comfort and difficulty.

5.7. EVALUATION

The questions posed in section 5.1 about our three main factors, namely user experience, performance, and setup, are discussed in the following sections.

5.7.1. USER EXPERIENCE

Figure 5.5 shows that study participants rated each device setup about the same on average. However it also shows a big deviation in the ratings users gave to the comfort of each device. Every setup was rated from "very uncomfortable" up to "very comfortable" by different participants. Difficulty was rated relatively low (easy) on average for every setup, with tablet-in-hand rated slightly better than the other two device setups.

This result might indicate that the application itself was designed with a bias towards a specific handling of the touch devices, however it was still relatively easy to use for the study participants.

The questionnaire also presented questions about how difficult each touch gesture (described in section 1) was to perform and how well each gesture responded to the user's intent. The summarized results in figure 5.6 show a correlation between easier difficulty and better response to intent. In general, the multi-touch gestures such as pivoting and pinch and stretch were perceived to be more difficult, and didn't always respond to the user's intent. The tap-and-hold gesture was perceived to be the the least responsive.

Lastly, figure 5.7 shows the ratings of the remaining questions from the questionnaire. In general, users had little trouble memorizing the different gestures, and the difficulty of using the touchpad was rated fairly low on average.

5.7.2. PERFORMANCE

Based on figure 5.2, there does not seem to be a significant difference in average run completion times for each device setup. The standard deviations are between 30 to 40 seconds, which is relatively small considering it took the average study participant just over 5 minutes to complete a run. This suggests that the size difference between the tested devices and the way they were held, were not a big influence on overall user performance.

The graph in figure 5.3 shows how participants improved over the course of the three runs they completed.

However it does not show an upward curve as one would expect. Instead the 2nd run seems to take a bit longer on average than the 1st and 3rd. While the difference doesn't seem too significant, this could be explained by the fact that the user switches to a different device setup for the first time when they start the 2nd run. Prior to the 1st run, they have a practice run (which is not accounted for in the results) that uses the same device setup as the 1st run. In the end this does show that the user performance is relatively consistent throughout their tests, suggesting a relatively flat learning curve.

Figure 5.4 is a more detailed look at the average completion times of cleaning marked images (rotating / scaling an image to the correct transform), one of the repeating tasks that study participants perform during a run. The graph shows that each device setup has similar averages. The smartphone setup has a larger deviation than the tablet setups, suggesting that some users struggled more with this setup while others were faster compared to the tablet setups.

5.7.3. Setup

As discussed in the previous two sections, all three device setups score similarly in terms of both user ratings and measured performance. We further split the data to also look at subsets of participants grouped by the finger poses they used for touch gestures.

Figure 5.8 show that finger pose still does not influence user performance to any significant measure. When we split up the results from the questionnaire per finger pose, we get the graph shown in figure 5.9. Here there seems to be a clear difference between how participants rated the difficulty and comfort of the tablet-in-hand setup (the red and blue bars), correlating with the finger pose they used on the device. This again suggests that developing the application with the index + thumb pose in mind, meant it was more comfortable for users that used the application in a similar way. The other finger poses were still rated quite similarly on average. Using two thumbs for the tablet-on-table setup has a quite high average comfort rating, but it has to be noted that these were just two data points.

5.7.4. Summary

Given our evaluation we can now answer our original hypotheses:

- The large touchpad outperforms the small touchpad in usability and performance: **No significant difference in measured performance. Tablet has small advantage in usability.**
- There is a link between the way users operate the touchpad device and usability: **Participants show a significant difference in usability ratings when grouped by finger pose used.**
- The touchpad in general will receive favourable assessments for usability: **Usability ratings for difficulty were relatively low, but comfort ratings were spread out.**

6 CONCLUSION

6.1. SUMMARY

The objective of this thesis was to design a framework for evaluating input devices in VR applications, and use this to construct an experimental setup for exploring the touchpad in VR. Our framework assesses the VR system, to identify constraints. It assesses the input device characteristics, to identify issues and advantages with the proposed scenario. Next we assess the VR application, for considerations on comfort, interactions and input mapping. Finally we assess the metrics needed for the experimental setup.

To verify the utility of the framework we designed an experimental setup for evaluating the touchpad as an input device for VR. We targeted the scenario of using a VR image browser in a mobile environment. The framework allowed us to formulate an assessment of the scenario, and served as a basis for design concept of a custom VR image browsing application.

Using the VR image browser we conducted a user study to explore and evaluate the use of the touchpad in VR. In order to collect the desired data we formulated three questions relating to performance, usability and touchpad setup.

The study had 21 participants in total, generally students. Results of the study showed that the touchpad was easy to use within VR, but we identified an issue with how users physically handled the touchpad in various ways which were not considered during the implementation phase of the VR application. Multi-touch gestures were also considered harder to perform than the other interactions. Users did learn all gestures quickly and were able to use them without a noticeable learning curve. In terms of measured performance, the differences between the three tested touchpad setups was insignificant. We did identify a difference in performance scores when splitting up the data between different finger poses that were used by participants.

This thesis produced two results. Firstly, an experimental setup that showed that the touchpad is a capable input device for VR applications, being easy to learn and use. The exact touchpad size and setup is of little influence, the challenges mainly lie in the design of touch gestures. Secondly, a framework for aims to standardize future efforts in creating experimental setups for evaluating input devices in VR, in order to help solve the input problem that faces modern VR systems.

6.2. FUTURE WORK

The proposed framework is still in need of refinement. The VR industry as a whole is still in its infancy and new lessons on VR application design are coming out at a constant rate. We also believe our framework could expand to include evaluation of input devices for augmented reality (AR) applications in the future. Using our framework for the evaluation of novel input devices that the introductory chapter did not cover could also bring new insights. Another additional element that the framework could cover is the wide variety of physical spaces that VR could potentially be used in in the future, adding the available space and environmental characteristics to the assessment steps.

A

USER STUDY QUESTIONNAIRE

Before the start of a session, the user was asked to sign a consent form. Besides the disclaimer, the user will also be asked to fill in the following personal information:

1. Age (number)
2. Gender (open)
3. Handedness (left / right / mixed / ambidextrous)
4. Visual impairments (open)
5. Prior experience with virtual reality headsets (open)
6. How often do you interact with touch devices (smartphones / tablets)? (daily, sometimes, never)
7. How often do you interact with touchpads (laptop touchpad, etc)? (daily, sometimes, never)
8. How many touch devices and touchpads do you own? (1 / 2 / more than 2)

After the session was done, the user was asked to fill out the following questionnaire:

1. How difficult did you find the touchpad input to use for this application overall? (7-dot scale, very easy to very hard)
2. Did you have trouble memorizing the different touch gestures for their respective actions? (7-dot scale, never to constantly)
3. How difficult did you find the tablet to use for input? (7-dot scale, very easy to very hard)
4. How difficult was it to use the tablet when held in hand? (7-dot scale, very easy to very hard)
5. How comfortable was it to use the tablet when held in hand? (7-dot scale, very uncomfortable to very comfortable)
6. How difficult was it to use the tablet when it was set on the table? (7-dot scale, very easy to very hard)
7. How comfortable was it to use the tablet when it was set on the table? (7-dot scale, very uncomfortable to very comfortable)
8. How difficult did you find the smartphone to use for input? (7-dot scale, very easy to very hard)
9. How comfortable was it to use the smartphone for this application? (7-dot scale, very uncomfortable to very comfortable)
10. How difficult did you find the tapping touch gesture (for selecting images) to perform? (5-dot scale, very easy to very hard)

11. Did the tapping gesture always respond to your intent? (7-dot scale: never to always)
12. How difficult did you find the swiping touch gesture (for rotating the images) to perform? (7-dot scale, very easy to very hard)
13. Did the swiping gesture always respond to your intent? (7-dot scale: never to always)
14. How difficult did you find the pivoting touch gesture (for rolling the images) to perform? (7-dot scale, very easy to very hard)
15. Did the pivoting touch always respond to your intent? (7-dot scale: never to always)
16. How difficult did you find the pinch and stretch touch gestures (for scaling images) to perform? (7-dot scale, very easy to very hard)
17. Did the pinch and stretch gestures always respond to your intent? (7-dot scale: never to always)
18. How difficult did you find the tap-and-hold touch gesture (for multi-selecting images) to perform? (7-dot scale, very easy to very hard)
19. Did the tap-and-hold gesture always respond to your intent? (7-dot scale: never to always)
20. Any remarks about the touch controls specifically? (open)
21. Any final remarks about the application? (open)

B

USER STUDY STEPS

These were the steps for each user session in the user study, after the user put on the VR HMD and was given the touchpad device.

B.1. ADDITIONAL INITIAL STEPS IN TRAINING SCENARIO

1. Application starts in image grid.
2. Instruct the user on how to operate the application using the touch device. Teach: target (highlight images by looking at them), navigate (rotate grid) and select interactions. First another collection is selected, so the user knows how to move to different collections. Then an image is selected, moving the application into the image inspector context.
3. Teach user manipulate (scale and rotate image) and cancel (move back to image grid) interactions.
4. Back in image grid, teach user multi-select (select three images, also deselect using single tap), cancel (stopping multi-select mode) and select (selecting a collection, causing the multi-selected images to move into that collection) interactions.
5. Explain how the test scenario operates. Users have to:
   (a) Move all pictures of subjects that do not correspond to the current collection, to the right collection (for example, pictures of buildings mixed in the collection of car pictures need to be moved to the collection of building pictures).
   (b) Select, scale and/or rotate all pictures in the grid marked with an icon.
   (c) Perform the above steps with each of the three collections.

B.2. STEPS PER SCENARIO

1. Application starts in image grid. The program will choose three subjects from the evaluation image set, and assign these to three different collections. Some images will be randomly mixed into different collections for the user to sort out (in step 3).
2. First collection of images is automatically opened.
3. User has to sort out specific images from a collection and add them to another collection by multi-selecting images and then selecting the appropriate collection. For example: remove all images of buildings from the collection of images that show cars. There are 10 images that need to be moved.
4. User has to rotate and/or scale specific images that are marked with an icon, by selecting them, aligning the picture with a transparent widget (that shows the target scale and rotation, it turns green when the picture is rotated correctly and at the right scale), and then returning to the image grid. There are 10 marked images, 5 of which have an incorrect rotation and 5 of which have an incorrect scale.

5. Once a collection has been ‘cleaned up’ (removed all marks and incorrectly sorted pictures, meaning at least 20 images were interacted with), move onto the next collection and perform steps 3 and 4 again. Do this for every collection. In total the user will interact with 60 pictures if no errors were made ((10 images that need moving + 10 that need to be scaled/rotated) x 3 collections).

6. Scenario ends if all collections are cleaned up. Move to next scenario.

C

Preliminary study: Input methods for a VR image browsing application

This preliminary user study was done early on in this thesis project, and served as one of the motivations for further exploring the touchpad as an input method for VR. Below is the report and evaluation of the user study.

C.1. Setup

To test the implementation of the VR image browsing application, a user study was set up to compare the different input methodologies and identify their strengths and weaknesses. The study was mostly aimed at getting performance data and surveying the users about their preferences. The input methods and devices used were:

- Mouse (generic)
- Gamepad (using Playstation 3 Dualshock controller)
- Touchpad (using an 7" Android tablet)
- Head-tracking (using the Oculus Rift DK2 HMD)

In order to compare each of these input methods, the study focussed on comparing user performance with several interactions within the VR image browsing application:

- Move-to-target: move a crosshair / selection box to target images in the grid.
- Selection: selecting the targeted image.
- Navigation: rotating the image grid to bring other images into your field of view.

The VR image browser itself was designed as a 2D grid of images that wraps around the user, like being inside a cylinder. The user itself is stationary, but the grid can be rotated in order to bring images from the far side of the grid into view without requiring the user to turn his / her head. Besides the image grid the application contained a simple static virtual environment to increase the user's comfort while wearing the head-mounted display.

During the study the user has to use the four following input configurations to complete a set of tasks:

- Mouse configuration
  - Move-to-target: moving the mouse moves the crosshair (which is projected on the same surface as the image grid).
  - Selection: left click selects images.
  - Navigation: scroll wheel rotates the image grid.

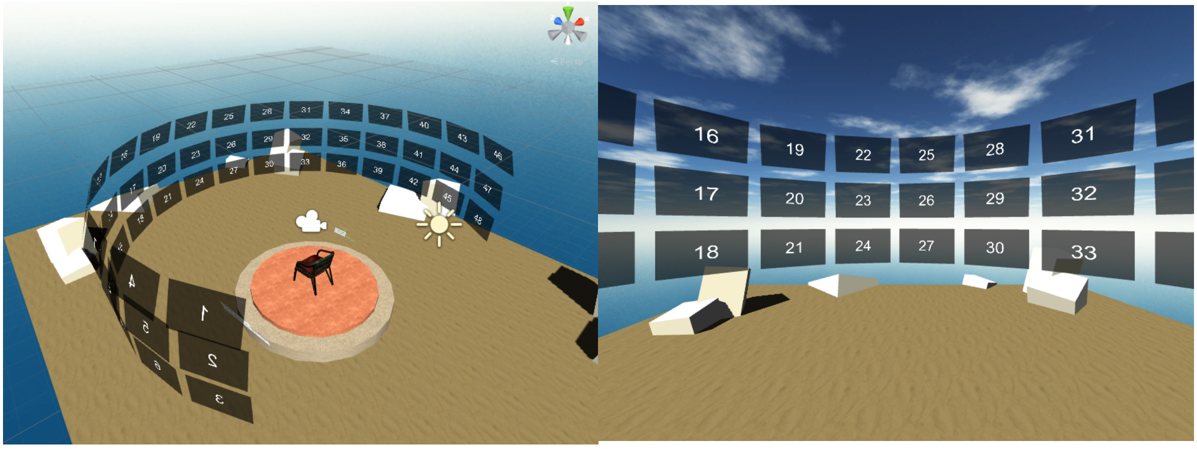

Figure C.1: Editor (left) and user (right) perspectives of the application

- Gamepad configuration
  - Move-to-target: instead of a crosshair, the user uses the left analog stick to let a selection box make discrete jumps between adjacent images.
  - Selection: press the bottommost action button (easiest accessible button for the right thumb) to select object in the selection box.
  - Navigation: by moving the selection further to the sides of the grid, the grid itself rotates to adjust and makes sure the selection box remains in view.
- Touchpad configuration
  - Move-to-target: uses head-tracking. The crosshair is located at the center of the screen and follows the user's head rotation.
  - Selection: the user taps the touch device screen to select objects that the crosshair is focussed on.
  - Navigation: swiping left and right on the device rotates the grid.
- Head-tracking configuration
  - Move-to-target: the crosshair is located at the center of the screen and follows the user's head rotation.
  - Selection: an image is selected by focussing the crosshair on it for a few seconds. A growing circle around the crosshair symbolizes the progress of the selection, providing feedback to the user.
  - Navigation: panels are visible on the sides of the grid. The user rotates the grid by moving the crosshair over these panels

C.1.1. IMAGE SET

For this study a dummy image set of 48 images with numbers displayed on them was used. These were ordered and displayed as seen in figure C.1. The numbers on the images had to be easy to identify from a distance while wearing the Oculus HMD, so that they are readable when positioned in the image grid. The intention here was that the user had to know where to find an image when told to find an a certain number. Even if they could not see the number in the images in front of their field of view, they had to be able to deduce in what direction to rotate the image grid to find the image with the number they were looking for. This was done so that the user's time-to-completion for each task was mainly influenced by the input method that was being tested, instead of the random amount of time spent searching.

C.1.2. TEST SCENARIO

Test scenario For each input configuration, the task-completion times were collected by recording the time it took a user to select a sequence of 4 numbered images in the image grid in quick succession. The first number is always "24" so that the user starts in the middle of the image grid, and serves as a start signal for the time

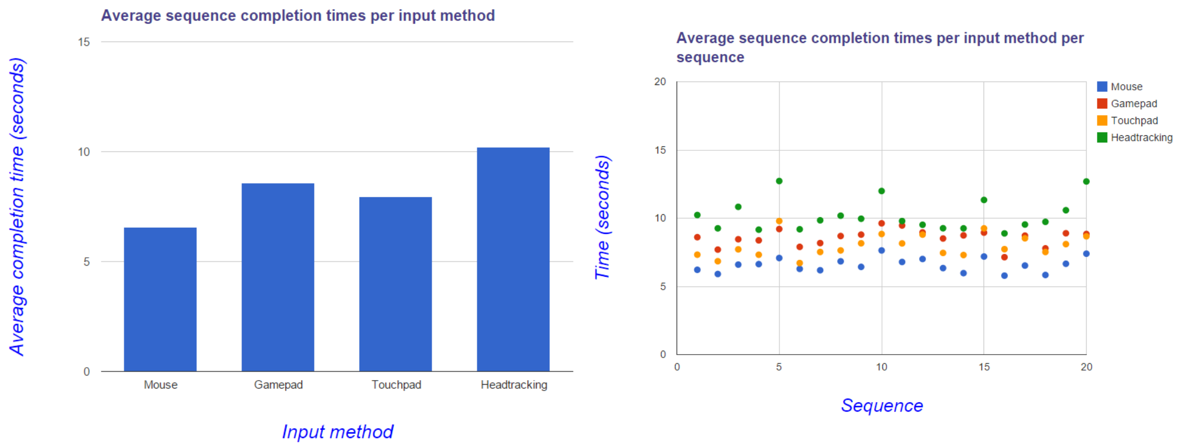

Figure C.2

recording. For example, the user would see displayed that he / she had to select the sequence "24" - "7" - "18" - "40". In this case the time spend between selecting "24" and "40" is considered the completion time. For each input configuration the study participant was given 5 of these sequences to practice and get used to the input device they were using. After this short 'tutorial' they had to complete 20 sequences. Each of these 20 sequences were constructed pseudo-randomly so that the user had to rotate the image grid and move around the crosshair / selection box about the same distance for each input configuration.

Once the user was done with the application tests, he / she was asked to fill out a questionnaire.

C.2. RESULTS

The left chart in figure C.2 shows the average completion times of all 18 study participants. The mouse was the fastest with an average of 6.57 seconds, followed by the touchpad with 7.97 seconds, then the gamepad with 8.58 seconds and finally head-tracking with 10.20 seconds. The right chart shows a more detailed scatter plot of the average completion time of each specific sequence, showing that each sequence's difference in distance between subsequent images affected each input configuration's completion time about the same.

In the questionnaire users were asked to rate (on a scale from 1 to 7) each input method on aspects such as ease-of-use, performance and enjoyment. Results are shown in figure C.3.

The user was also asked to rank each input method according to preference for subtasks such as navigating the menu and selecting images. See figure C.4.

And lastly users were asked to give a final rank for each input method based on what they would prefer to use. See figure C.5.

C.3. EVALUATION

Both the numbers and the user feedback show a big advantage for the mouse compared to the other input methods. For the other input methods it's a bit more mixed. The touchpad comes in second in terms of average completion time, but users enjoyed using the head-tracking itself more. Users disliked using the gamepad the most.

Before drawing conclusions from this data, the open user feedback also needs to be taken into account. The gamepad for example was considered to be suboptimal. Users often overshot their target when using the analog stick, suggesting that the accuracy of the implementation wasn't as good as it could be. The head-tracking input method was considered to be quite tiring for use over a longer period, requiring the user to move his head too much. A lot of the users commented on that the mouse also felt the most familiar of all the input methods.

We cannot draw any major conclusions about which input method is considered to be the best for use in a VR image browsing application, but it should provide further guidance when improving the application in the future.

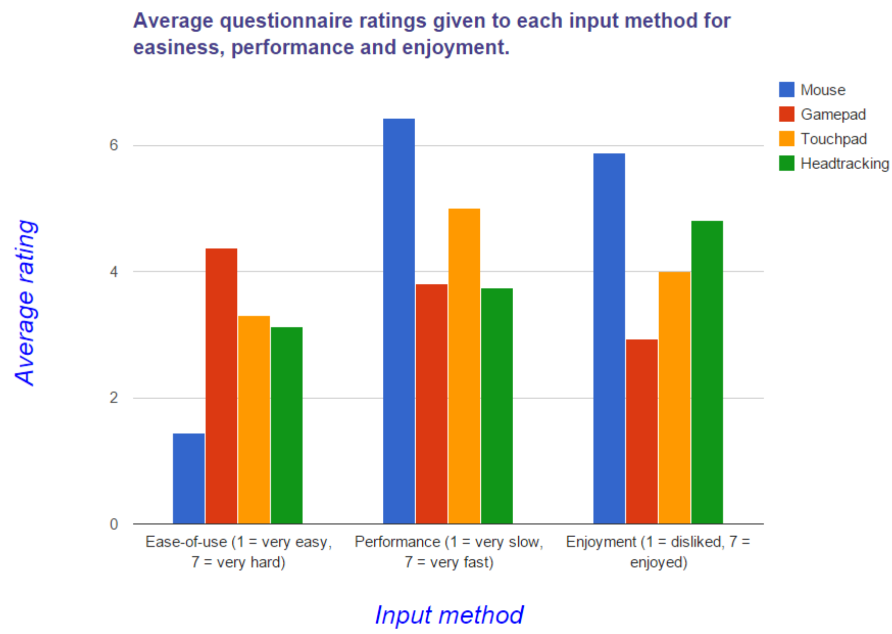

Figure C.3

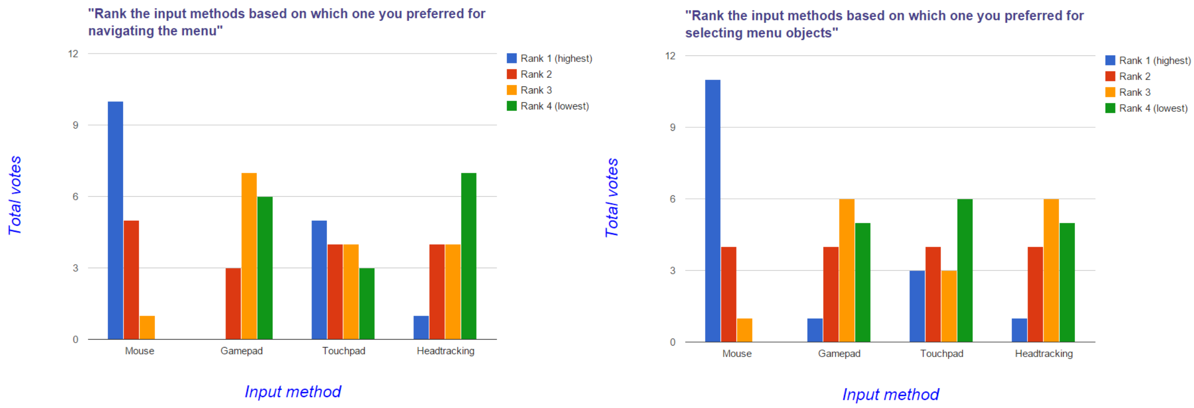

Figure C.4

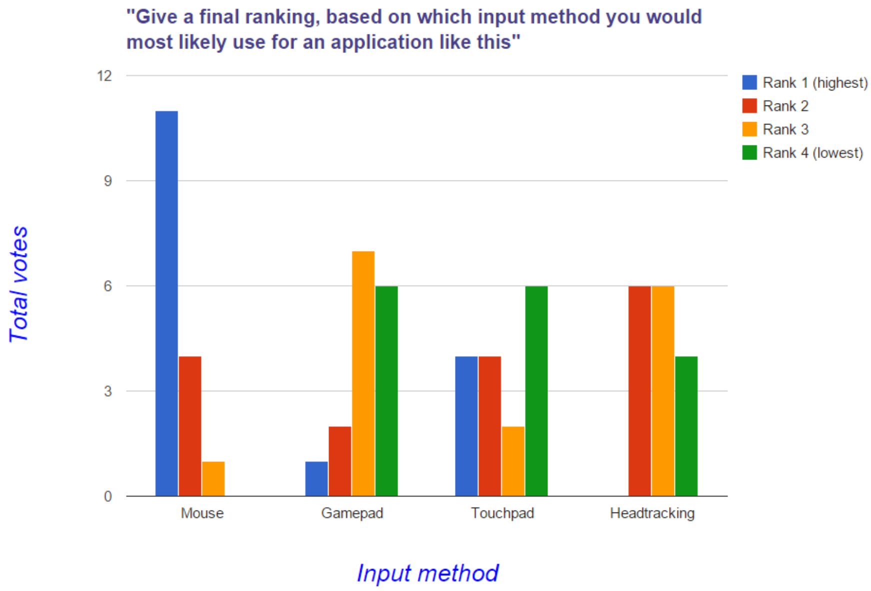

Figure C.5

C.3.1. SUMMARY

We summarize our findings in the following list of pros and cons for each input method. **Mouse**

- Pros:
  - Fastest input method in application
  - Users are familiar with it
- Cons:
  - Limited to seated-VR setups, requires a surface to rest on

Gamepad

- Pros:
  - Not tied to a location, can be used while moving around
  - Lot of potential buttons and triggers that can be mapped to actions
- Cons:
  - Not great for navigating image datasets compared to alternatives
  - Current implementation not very accurate
  - Could be hard to use for non-gamers

Touchpad

- Pros:
  - Touch actions quite intuitive for some interactions in the VR environment
  - Touch device could potentially be carried around with the user
- Cons:
  - Touch actions not too useful for moving a crosshair or selection box around, needs to work in conjunctions with head-tracking

Head-tracking

- Pros:
  - Intuitive when combined with the right visual feedback for navigating the image grid and selecting objects
  - Requires no other input devices besides the HMD, very suited for mobile VR
- Cons:
  - Lots of head movement causes fatigue quite fast

Bibliography

[1] B. Edwards, *Unraveling the enigma of nintendo's virtual boy, 20 years later,* (2015).

[2] Visbox, *Cave automatic virtual environment,* (2016).

[3] E. V. Laboratory, *Cave2: Next-generation virtual-reality and visualization hybrid environment for immersive simulation and information analysis,* (2016).

[4] Oculus, *Oculus rift kickstarter,* (2012).

[5] A. Wawro, *Carmack reveals the challenges of mobile vr game development,* (2014).

[6] Kinect, *Website,* (2017).

[7] L. Motion, *Website,* (2017).

[8] U. Schultheis, J. Jerald, F. Toledo, A. Yoganandan, and P. Mlyniec, *Comparison of a two-handed interface to a wand interface and a mouse interface for fundamental 3d tasks,* in *3D User Interfaces (3DUI), 2012 IEEE Symposium on* (IEEE, 2012) pp. 117–124.

[9] J. Steuer, *Defining virtual reality: Dimensions determining telepresence,* Journal of communication **42**, 73 (1992).

[10] G. Burdea, P. Richard, and P. Coiffet, *Multimodal virtual reality: Input-output devices, system integration, and human factors,* International Journal of Human-Computer Interaction **8**, 5 (1996).

[11] J. Ku, R. Mraz, N. Baker, K. K. Zakzanis, J. H. Lee, I. Y. Kim, S. I. Kim, and S. J. Graham, *A data glove with tactile feedback for fmri of virtual reality experiments,* Cyberpsychology & behavior **6**, 497 (2003).

[12] F. P. Brooks, *What's real about virtual reality?* IEEE Computer graphics and applications **19**, 16 (1999).

[13] O. Van der Meijden and M. Schijven, *The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review,* Surgical endoscopy **23**, 1180 (2009).

[14] A.-S. Augurelle, A. M. Smith, T. Lejeune, and J.-L. Thonnard, *Importance of cutaneous feedback in maintaining a secure grip during manipulation of hand-held objects,* Journal of Neurophysiology **89**, 665 (2003).

[15] M. Mine, A. Yoganandan, and D. Coffey, *Making vr work: building a real-world immersive modeling application in the virtual world,* in *Proceedings of the 2nd ACM symposium on Spatial user interaction* (ACM, 2014) pp. 80–89.

[16] P. Bristol, *Designing touch,* (2016).

[17] N. T. Dang, M. Tavanti, I. Rankin, and M. Cooper, *A comparison of different input devices for a 3d environment,* International Journal of Industrial Ergonomics **39**, 554 (2009).

[18] M. Kavakli and J. R. Thorne, *A usability study of input devices on measuring user performance in computer games,* in *Proceedings of First International Conference on Information Technology and Applications* (2002) pp. 291–295.

[19] S. Zabramski, *Careless touch: a comparative evaluation of mouse, pen, and touch input in shape tracing task,* in *Proceedings of the 23rd Australian Computer-Human Interaction Conference* (ACM, 2011) pp. 329–332.

[20] V. Woods, S. Hastings, P. Buckle, and R. Haslam, *Development of non-keyboard input device checklists through assessments,* Applied ergonomics **34**, 511 (2003).

[21] G. Turpin, J. Armstrong, P. Frost, B. Fine, C. Ward, and L. Pinnington, *Evaluation of alternative computer input devices used by people with disabilities,* Journal of medical engineering & technology **29**, 119 (2005).

[22] B. Thomas, *Evaluation of three input techniques for selection and annotation of physical objects through an augmented reality view,* in *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality* (IEEE Computer Society, 2006) pp. 33–36.

[23] J. Accot and S. Zhai, *Performance evaluation of input devices in trajectory-based tasks: an application of the steering law,* in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (ACM, 1999) pp. 466–472.

[24] P. M. Fitts, *The information capacity of the human motor system in controlling the amplitude of movement.* Journal of experimental psychology **47**, 381 (1954).

[25] R. Vertegaal, *A fitts law comparison of eye tracking and manual input in the selection of visual targets,* in *Proceedings of the 10th international conference on Multimodal interfaces* (ACM, 2008) pp. 241–248.

[26] R. J. Beaton, R. J. DeHoff, N. Weiman, and P. W. Hildebrandt, *An evaluation of input devices for 3-d computer display workstations,* in *OE LASE'87 and EO Imaging Symp (January 1987, Los Angeles)* (International Society for Optics and Photonics, 1987) pp. 94–101.

[27] J. Sykes and S. Brown, *Affective gaming: measuring emotion through the gamepad,* in *CHI'03 extended abstracts on Human factors in computing systems* (ACM, 2003) pp. 732–733.

[28] M. Mahmud and H. Kurniawan, *Involving psychometric tests for input device evaluation with older people,* in *Proceedings of the 17th Australia conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future* (Computer-Human Interaction Special Interest Group (CHISIG) of Australia, 2005) pp. 1–10.

[29] D. A. Bowman, J. L. Gabbard, and D. Hix, *A survey of usability evaluation in virtual environments: classification and comparison of methods,* Presence **11**, 404 (2002).

[30] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster, *A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment,* Personal Technologies **1**, 208 (1997).

[31] D. Medeiros, L. Teixeira, F. Carvalho, I. Santos, and A. Raposo, *A tablet-based 3d interaction tool for virtual engineering environments,* in *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry* (ACM, 2013) pp. 211–218.

[32] J. Wang and R. Lindeman, *Coordinated 3d interaction in tablet-and hmd-based hybrid virtual environments,* in *Proceedings of the 2nd ACM symposium on Spatial user interaction* (ACM, 2014) pp. 70–79.

[33] R. Nagashirna, N. Sakata, and S. Nishida, *Study of the touchpad interface to manipulate ar objects,* in *Artificial Reality and Telexistence (ICAT), 2013 23rd International Conference on* (IEEE, 2013) pp. 78–84.

[34] S. Knoedel and M. Hachet, *Multi-touch rst in 2d and 3d spaces: Studying the impact of directness on user performance,* in *3D User Interfaces (3DUI), 2011 IEEE Symposium on* (IEEE, 2011) pp. 75–78.

[35] J. Feltham, *Abrash: Vr needs to hit 16k to match 'retinal resolution',* (2015).

[36] H. Akiduki, S. Nishiike, H. Watanabe, K. Matsuoka, T. Kubo, and N. Takeda, *Visual-vestibular conflict induced by virtual reality in humans,* Neuroscience letters **340**, 197 (2003).

[37] B. Lewis-Evans, *Designing to minimize simulation sickness in vr games,* (2015).

[38] A. Chu, *Vr design: Transitioning from a 2d to a 3d design paradigm,* (2014).

[39] M. Alger, *Vr interface design pre-visualisation methods,* (2015).

[40] K. Brady and R. Emms, *Oculus connect 2: Navigating new worlds: Designing ui and ux in vr,* (2015).

[41] Unity, *Website,* (2017).