UniNet: A Mixed Reality Driving Simulator

by

David F. Arppe

A thesis submitted in partial fulfillment
of the requirements for the degree of

Masters of Science

in

Computer Science

University of Ontario Institute of Technology

Oshawa, Ontario, Canada

August 2019

THESIS EXAMINATION INFORMATION

Submitted by: **David F. Arppe**

Masters of Science in **Computer Science**

Thesis title: UniNet: A Mixed Reality Driving Simulator

An oral defense of this thesis took place on August 9, 2019 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Faisal Qureshi
Research Supervisor	Dr. Loutfouz Zaman
Research Co-Supervisors	Dr. Richard W. Pazzi
Examining Committee Member	Dr. Khalil El-Khatib
Thesis Examiner	Dr. Bill Kapralos, UOIT

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

The research work in this thesis that was performed in compliance with the regulations of Ontario Tech's Research Ethics Board under **REB Certificate number #15287.**

Signed: *David F. Arppe*

David F. Arppe

STATEMENT OF CONTRIBUTIONS

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

Abstract

Driving simulators play an important role in vehicle research. However, existing virtual reality simulators do not give users a true sense of presence. *UniNet* is our driving simulator, designed to allow users to interact with and visualize simulated traffic in mixed reality. It is powered by *SUMO* and *Unity*. *UniNet*'s modular architecture allows us to investigate interdisciplinary research topics such as vehicular ad-hoc networks, human-computer interaction, and traffic management. We accomplish this by giving users the ability to observe and interact with simulated traffic in a high fidelity driving simulator. We present a user study that subjectively measures user's sense of presence in *UniNet*. Our findings suggest that our novel mixed reality system does increase this sensation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many driving simulators have been developed, with most of them being used for driver training or research in the field of driver safety [43]. However, these simulators often present limited features in regards to traffic simulation, and user presence [10, 24, 25]. The need for high-quality Virtual Reality (VR) driving simulators with a focus on user presence is long overdue. In addition to this, a driving simulator with traffic simulation is a strong tool for Vehicular Ad-Hoc Network (VANET) research, which is made possible by our choice of traffic generation software. Network simulation is commonly used in networking research, to evaluate the performance of communication protocols and algorithms. Existing simulation tools for vehicular networks focus exclusively on network simulation. A driving simulator that combines network simulation, application prototyping, and testing would be beneficial to VANET researchers. For instance, one could evaluate the performance of a communication protocol or application by using a realistic virtual environment with thousands of vehicles and interacting with them before deploying their research in the real world, which is costly, and at times, unsafe.

In addition to a modular simulator with VANET capabilities, we introduce a

system for Mixed Reality (MR). Our system introduces the user as their own avatar in a virtual environment, by using stereoscopic cameras and passthrough VR technology. We designed the system to be compatible with existing VR systems, and most VR systems can easily upgrade to our proposed immersion configuration.

In this thesis, we study a driving simulator that combines realistic vehicle dynamics [41] with a high performance traffic flow simulation platform *Simulation of Urban MObility* (*SUMO*) [26]. We discuss the systems we have built within *Unity* [39], which connect external applications for a high quality driving experience.

1.1   Motivation

The entertainment industry strives to create a true sense of presence in virtual environments, and this provides several challenges for developers. The notion of a system that creates a sense of presence in VR, provides us with interesting challenges to consider. Our systems are specifically designed to solve these problems, while retaining a high quality, non-experimental feel akin to finished products. *UniNet* faithfully simulates a 2018 Ford Focus, for use in situations where a physical vehicle is unsafe or unreasonable. The gear ratios, horsepower, top speed, acceleration, and suspension match the target vehicle completely. Throughout the past 55 years, since its inception in 1964, virtual reality devices have not always been consumer products due to their high cost and low graphical fidelity [37]. The *Oculus Rift* [42] was one of the first commercial Head Mounted Displays (HMDs) to gain traction in the gaming industry. With VR hardware making strides in progress and improvements in quality, it is to be expected that VR-ready applications and supporting software follows close behind. Since it's introduction, variations on VR have been introduced. Augmented Reality (AR) and Augmented Virtuality (AV), are both a mix between reality and virtuality.

AR can be described as reality, augmented with aspects of a virtual environment. AV can be described as virtuality, augmented with aspects of reality. These modern forms of VR are relatively new and this thesis provides a contribution to AV, in the form of our MR technology.

1.2   Virtual Reality Driving Simulators

Virtual reality driving simulators have existed for as long as modern VR has existed [43]. Typically used for driver training, simulators have the advantage of being consistent. Simulators run real time simulations, in which all aspects of the virtual environment are controlled. The input to a driving simulator is designed as a realistic imitation of the target vehicle, and the underlying simulator model simulates the interaction between the user and the target vehicle. Visual, auditory, and motion output are common forms of feedback that the simulator can provide to the user, to complete the simulator model.

1.2.1   Issues with Presence in Simulators

An issue with current virtual reality driving simulators, is the lack of user presence. User presence is often confused for immersion, however this is not its definition [36]. Presence revolves around a user psychologically accepting a virtual environment, as opposed to just a user's senses being controlled by a virtual environment [34]. This distinction between the two terms is important for understanding that, although a VR simulator may be immersive, it is not necessarily giving the user a sense of presence.

1.3   Contributions

This thesis explores the benefits of MR technology, in enhancing user presence in virtual environments. We also developed an application pairing *Unity*, a commercial game engine, with *SUMO*, an industry-standard traffic simulator, to create a powerful visualisation tool for VANETs; capable of receiving real time user interactions. The technology was used during our study, which also confirmed our hypothesis that MR technology, leads to a heightened sense of user presence. The development of *UniNet* shows that consumer VR technology can be combined with modern cameras, to create a unique VR experience, falling elsewhere on the reality-virtuality continuum. The overall result of this work also provides the foundation for more immersive MR technology to be developed in future works, capable of the user a sense of presence. When listed, the main contributions of our work are the following:

1. Development of a Mixed Reality technology which uses stereo passthrough vision in Virtual Reality, and a green screen chamber.

2. Development of a driving simulator, which is connected in real-time to an industry standard traffic generator, and has two-way communication allowing for human interaction with the generated traffic.

3. A user study designed to measure the effectiveness of our Mixed Reality technology, by subjectively measuring user presence.

Minor work which supports our main contributions include: An algorithm which generates cities from *Open Street Maps* (*OSM*) data, a novel technique for rendering thousands of vehicles at once, and the construction of all of the hardware that supported the development of our MR technology.

1.4 Purpose of this Work

UniNet is the title of our driving simulator, with VANET and mixed reality capabilities. The driving force behind our work was to create a simulator, which can bridge a gap between vehicle network research. Future work can study the relationship between human interactions with simulated traffic, and a network simulator; however this thesis focuses on the development of *UniNet*, and a user study designed to measure user presence in *UniNet*.

1.4.1 Hypothesis

It is hypothesised in this thesis that if a user is introduced to a virtual environment using our custom passthrough mixed reality experience, they will feel a heightened sense of presence compared to a pure virtual reality experience.

1.4.2 Thesis Structure

Following this section of the thesis, Chapter 2 provides a background and literature review of immersion and presence. An overview of driving simulators, VANETs, and mobility models is presented. Chapter 3 introduces the architecture of *UniNet*, and its underlying systems. It also provides an in depth analysis of the technology developed and hardware built in preparation for the user study. Chapter 4 presents the study, which tests our hypothesis regarding immersion configurations, and their effect on user presence. Chapter 5 discusses the results of this work, and outlines future work. The overall result of this thesis is a driving simulator combining *Unity* with industry standard VANET simulations, capable of fully immersing a user in MR.

Chapter 2

Background and Related Work

2.1 Virtual Reality

The notion of Virtual Reality (VR) extends well before the advent of computers and technology as we know it. The earliest account of VR is presented in the book *"Le Théâtre et son Double"* (The Theater and its Double) by French playwright Antonin Artaud. The work is a collection of essays, written as an attack on 20th century theatrical convention. His work describes theatre as 'Virtual Reality', in 1938 [1]. A single year later in 1939 the *View-Master* was introduced, capable of presenting a pair of images to create a single stereoscopic image. It was not until the 1980s when the term *virtual reality* was popularized by Jaron Lanier, who is considered the father of VR [23]. Lanier discusses how VR goes beyond entertainment and gaming, and discusses its applications to treat war veterans; its use by doctors performing intricate surgeries; and as a means of prototyping most vehicles fabricated in the past 20 years [23].

Modern HMDs such as the Oculus *Rift* [42] bring VR to the consumer market, and the applications of VR are still being explored. The use of VR in driver training is

studied by Daniel J. Cox *et al.*, as they explored the effect of VR driver training with youth with autism spectrum disorder [10]. Their explorative study explicitly focused on youths with autism spectrum disorder, and how VR can be used to improve their overall driving outside of a VR simulator.

2.1.1 Mixed Reality
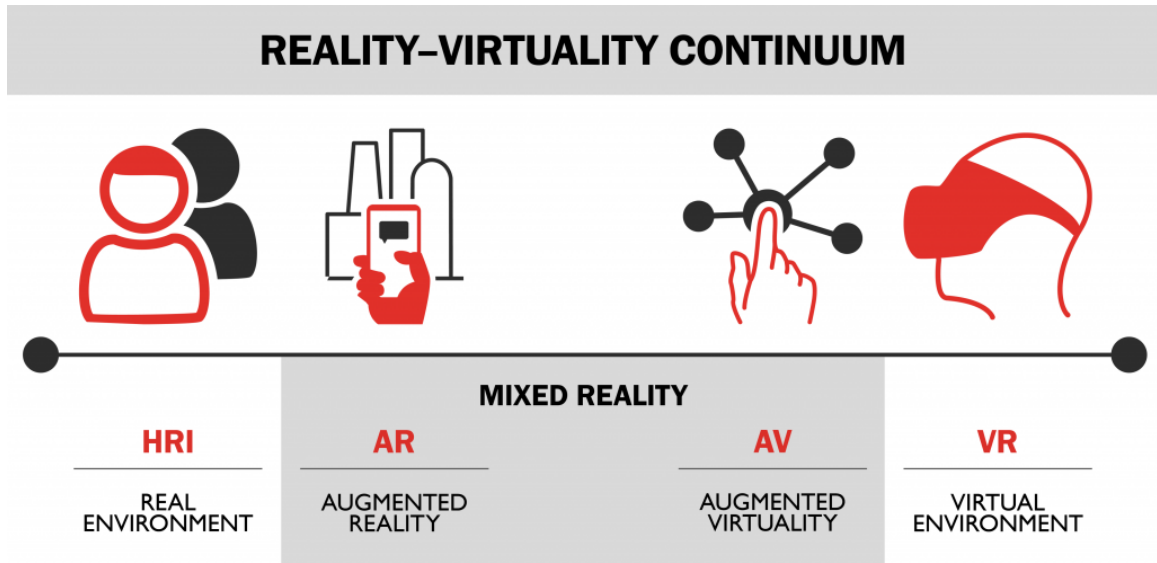
Figure 2.1: The Reality-Virtuality Continuum described by Milgram *et al.*

Mixed Reality (MR) visual displays, a subset of VR displays, is defined as merging the real and virtual worlds somewhere along the *"Reality-Virtuality Continuum"* (see Figure 2.1), a scale connecting real environments with virtual environments [31]. MR is a term used to describe a VR experience on the reality-virtuality continuum, and not a specific technology which achieves this experience. Augmented Reality (AR) technology is considered mixed reality on the reality-virtuality continuum, and can be seen used for a variety of applications, from educational displays at museums; to multiplayer smartphone games [2]. Augmented Virtuality (AV) is another form of MR, but less common than AR. Blissing *et al.* explored driving behaviours in

VR and a form of MR akin to AV. Their study was designed to understand how drivers' behaviours are affected by reality, VR, and MR. For their study, their MR configuration involved an Oculus *Rift DK2* HMD, with two cameras mounted onto the top, and a car. The cameras are designed to mimic the drivers' eyes, to give the user depth-perception. Their within-subjects study involved 22 participants experiencing each of their four configurations, while driving a real car. The four conditions were driving the car regularly, driving with the passthrough cameras in VR, driving with the passthrough cameras and traffic cones superimposed (MR), and full VR. The study required participants to drive a slalom course in these four configurations. The study concluded that the introduced HMD may affect driving behaviour, and that participants drove 35% slower when wearing the HMD. This particular MR configuration falls into the AR half of the Milgram *et al.* reality-virtuality continuum.

2.1.2 Immersion and Presence

Often confused or substituted for one another, an important distinction exists for the terms 'Immersion' and 'Presence'. For the purpose of this literature, we use the definition of immersion as the objective level of sensor fidelity a VR system or virtual environment provides; and presence as a user's subjective psychological response to a VR system [4, 36]. It is important to measure and quantify a user's sense of presence, in order to fully understand what affects user presence in a VR environment. Insko *et al.* discuss three methods for measuring user presence: Behavioural, Subjective, and Physiological [19].

Behavioural responses to events in VR is a form of measuring presence [12]. Freeman *et al.* designed a study to measure presence using postural responses to events. Their study used first-person footage of a rally race from the hood of the rally car. The variance in posture were compared with subjective measures of presence.

Due to presence being a subjective sensation, subjective measurements of presence are the most common form of measurement [18], having even been used in Freeman's behavioural responses study [12]. Their study used the subjective responses to confirm their behavioural responses. This is because presence is an emotional sensation, and is best measured subjectively. Hence, questionnaires are the preferred method of gathering subjective measures. The Bob G. Witmer presence questionnaire is used for the purpose of measuring presence [47,48]. A major issue with questionnaires as the primary form of measuring presence is that the user needs to take the questionnaire after the immersive experience, and the results depend on the user's memory [34]. However, the questionnaire approach to measuring presence is still preferred because questionnaires are easy to administer and analyze [19].

Physiological measurements have been used to measure a user's sense of presence. Heart Rate Monitors (HRMs) can be measured, and the change in heart rate can be affected by emotions, stress, fear, etc. [19]. Physiological measurements are very objective, but the disadvantage is that they can not be linked to the change in user presence easily [19]. Equipment required for physiological measurements can also create an unnatural environment, or suffer interference from electromagnetic fields or motion.

2.2   Driving Simulators

Driving simulators can be effective tools for researching due to their low cost and flexibility. Paired with a realistic traffic generator, a good driving simulator can make for an invaluable tool in traffic and VANET research, where human interaction is required. This section offers an overview of current driving simulators, VANET simulators, and traffic generators that were referenced while designing our simulator.

A driving simulator is an artificial environment, designed as a valid substitute of the actual driving experience [43]. Historically, simulators were designed for aircraft, primarily to train military pilots [21]. Unlike these early flight simulators, driving simulators today are used for much more than just driver training. They are used to assess driver safety [5], in VANETs [30] and HCI (Human-Computer Interaction) [6] research, and as an alternative to most other things that typically require a car. Most modern driving simulators are three-dimensional, with a high-quality physics simulation for the user-controlled vehicle [32]. The physics simulation is a key component of the driving simulator, and it converts user interaction with the system into signals captured by sensors through the steering wheel and pedals [21]. These signals are converted into inputs for the physics simulation, and the results from the simulation are presented back to the user in the form of computer graphics, sounds, force-feedback, and sometimes motion.

Lee *et al.* built a full motion driving simulator as a 'Virtual Reality' tool, without the use of VR technology as we know it today [25]. Their simulator recreated the visual, motion, audio and proprioceptive cues we associate with driving. At the time of its creation, the new level of immersion attained by their simulator inspired its title as a VR tool. In the past decade, driving simulators have become more accessible than ever. This is in part thanks to the video game industry, pushing driving physics and computer graphics to their full potential [32]. Our simulator is built around *Unity* [39], a high-performance game engine. The following subsections discuss some related literature which uses *Unity* as a base engine for a driving simulator. These works have inspired us to build our simulator in *Unity*.

2.2.1 Vehicular Ad-Hoc Networks

Unity is a powerful game engine on its own, but it can also be combined with *SUMO* for traffic generation, and discrete network event simulators for researching VANETs [3]. Biurrun-Quel *et al.* have developed a driver-centric traffic simulator by connecting *Unity* with *SUMO*. Their process involved establishing a connection between the two programs via Traffic Control Interface As a Service (TraCIAS), allowing remote control of *SUMO*. This established connection allowed the authors to poll vehicle position, and display it in *Unity*. In our simulator we approached a few things differently, namely synchronization between *Unity* and *SUMO*, Non Player Controlled (NPC) vehicle motion, and physics simulation.

Ropelato *et al.* [35] used *Unity* as the base for a virtual reality driving simulator. Their research into VR driver training builds on traditional driver training, using *Unity* as an engine to handle the vehicle physics calculations, render the virtual world into a HMD, and provide motion feedback with 6 Degrees Of Freedom (DOF). Their driving simulator took place in a virtual city generated by *CityEngine* [11], and featured AI traffic built in *Unity*.

Michaeler *et al.* propose in their work a system built entirely within *Unity* [30]. Having considered *SUMO* and discrete network event simulators, they chose to simulate Vehicle-To-Vehicle (V2V) communication within *Unity*. The justification for this was that *OMNet++* combined with *SUMO* would not run until the network calculation is finished, and was therefore unsuitable for combination with *Unity*. Their implementation relied on the Self-Organized Time Division Multiple Access (SOTDMA) protocol, and was able to simulate bad reception from distances, and building interference. Their simulation would parse road types from *OSM* [9], and generated traffic signs. This was based on both road data, and explicitly positioned signs.

An instance where *Unity* was used for visualization of data, can be seen in the

works of Guan *et al.* [17]. Their software for real-time 3D visualization of distributed simulations of VANETs uses *Unity*'s powerful rendering engine, to visualize a city generated by *ESRI City Engine* [11]. Their visualization software combines the affordances of a to-scale map, with the power of VANET simulations.

2.2.2 Mobility Models

SUMO [26] is an open-source traffic simulation application, along with supporting tools. *SUMO* is a microscopic traffic simulator, where vehicle 'types' defined by a file, are instantiated and given 'routes'. It performs a time-discrete simulation of traffic, for an arbitrary number of vehicles. Routes are generated externally, and assigned during run-time. Routes are paths along 'edges', which correspond in most circumstances to roads. Connections between edges can support traffic lights, and multiple edges can be assigned to a road to simulate multiple lanes.

Gonccalves *et al.* explored the use of *SUMO* in conjunction with a serious game driver simulator, to test Advanced Driver Assistance Systems (ADASs) [15]. Their work relies on *SUMO* not only for its multi-agent microscopic simulation, but as a 'center-server', providing all essential information to their other systems [16]. Their initial work explored researched the impact of mental workload and distractions on driver performance [14].
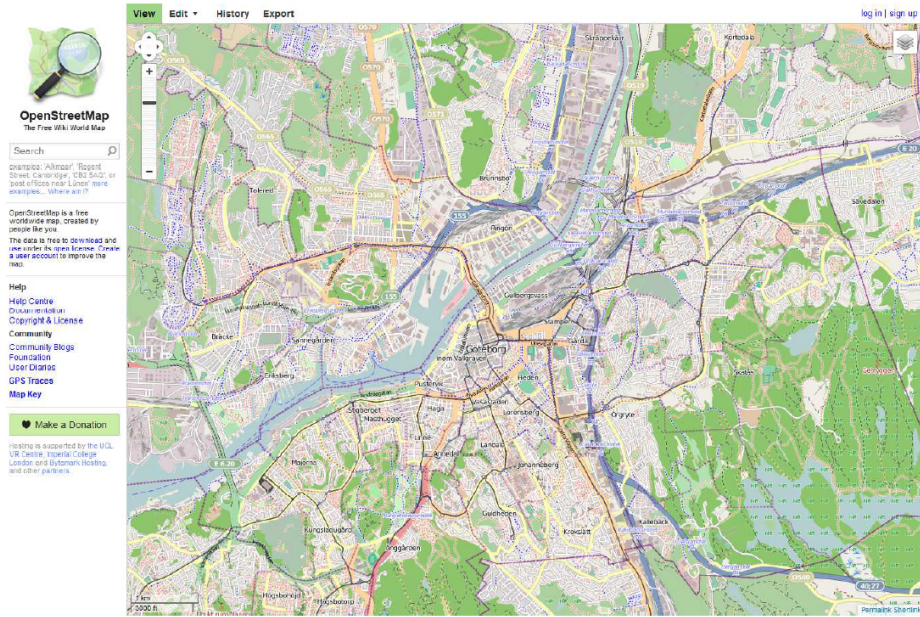
(a) *OSM* network of Gothenburg.
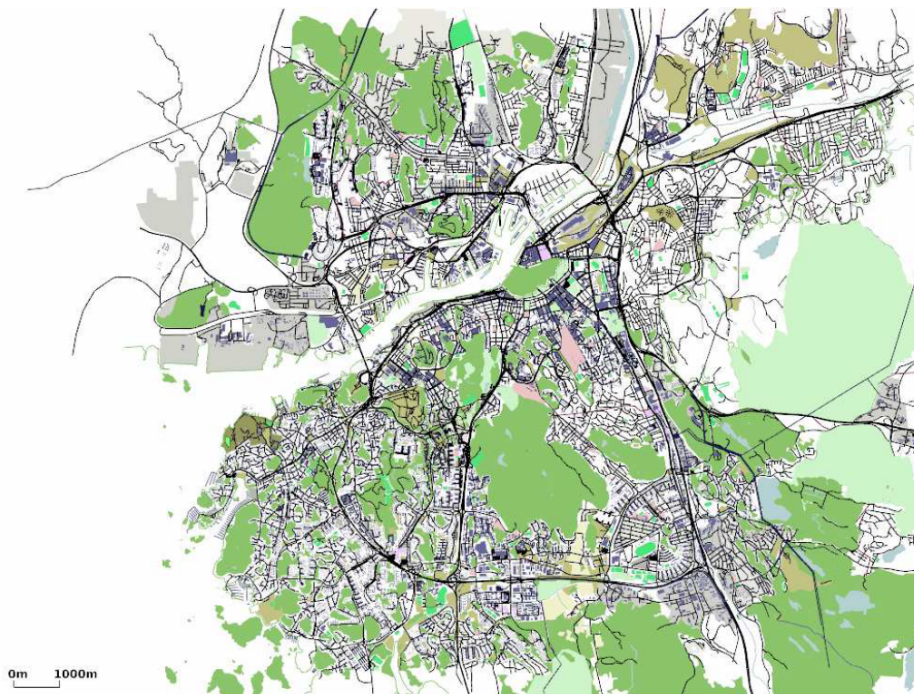
(b) Gothenburg network imported into *SUMO*.

Figure 2.2: Gothenburg taken from *Open Street Maps*, and loaded into *SUMO*, using *netconvert* from the *SUMO* suite.

To augment *SUMO*, supporting tools exist to generate routes, convert incompatible road networks into compatible road networks, and modify compatible networks. To perform these tasks in real-time requires a socket connection from an external application to *SUMO*. The Traffic Control Interface (TraCI) [46] API exists as a part of *SUMO*'s official release, and creates endless possibilities. For our research, we use TraCI to establish a connection between *SUMO* and *Unity*. It is not uncommon to find TraCI used to couple *SUMO* with communication simulators, such as *NS2* or *NS3* [8]. In the past decade, the TraCI protocol has been implemented in many programming languages. Our simulator makes use of a modern C# implementation of TraCI from CodingConnected [7]. *SUMO* supports multiple connections from different sources, and allows us to connect communication simulators in parallel with a visualization system.

2.3   UniNet compared to Related Works

Our proposed driving simulator was designed and implemented, to enhance immersion and user presence in VR driving simulators. Existing VR driving simulators used for driver training [10, 30] lack the benefits of this technology, as discussed in Section 4.4 of Chapter 4. We show, with significant results, that a user subjectively feels more 'present' in our MR configuration of *UniNet*.

Finally, we have also designed and implemented an improved architecture, for connecting *Unity* and *SUMO* (See Chapter 3), where each vehicle has a two-way communication with *SUMO* from *UniNet*. Our simulator allows for user interaction and involvement with the generated traffic. Current related works [3, 17] which connect *Unity* and *SUMO*, lack this two-way communication for human involvement in the traffic simulation.

Chapter 3

The UniNet Driving Simulator

The development of *UniNet* spanned several months from 2018 through 2019, and focused primarily on user presence in a physical driving simulator and real time traffic generation and simulation. Our approach was to develop a high quality simulator with respect to realism, through the use of renowned software libraries and external applications. The use of third party software allowed us to focus on the novel features of *UniNet*, such as our real-time mixed reality software. Section 3.1 discusses the architecture of *UniNet* from a critical point of view, where we discuss problems and challenges with existing software and how we initially aimed to solve the problems. Section 3.1.2 presents a high level description of *UniNet*, and the systems (both internal and external) that work together to create the resulting simulator. Finally, Section 3.2 contains a detailed breakdown of the development of *UniNet*, and in-depth descriptions of each system previously mentioned.

3.1 Architecture

In this chapter, we briefly introduce some the problems or limitations that exist with current driving simulators, and discuss our solutions to these problems. Section 3.2 further discusses the implementation of our solutions. Finally, we present the architecture of *UniNet*, and visually show how each component works together to create the end result.

3.1.1 Challenges with Existing Simulators

Human Interaction

The most common tools for traffic simulation often lack the built-in functionality for user interaction in the form of a driving simulator [26], and driving simulators often do not offer microscopic and continuous road traffic simulation [21]. This is due to the fact that most traffic research can be conducted without human interaction and pure simulation. We chose to address this issue by building a simulator combining an industry-standard traffic generator, with a high fidelity driving simulator. *UniNet* is the proposed solution to this problem. Our work is capable of running continuous traffic simulation, with real-time human interaction. The established system allows for two primary forms of human interaction:

1. Human interaction in the form of a user-controlled vehicle

2. Human interaction from outside of the traffic simulation, in the form of commands sent to the simulator

Each form of human interaction can have significant impact on the resulting traffic simulations, and enable new forms of VANET and Human-Computer Interaction (HCI) research.

Procedural City Generation

Tools for generating cities such as Esri *CityEngine* can be powerful when used for visualizing traffic flow in a 3D world, if the correct software is used to combine it with a traffic generator [11]. We designed and implemented the functionality of tools such as Esri *CityEngine* into *UniNet*, to generate cities procedurally. This type of procedural design simplifies the process of importing real-world data for research. *UniNet* is designed to generate textured buildings and roads from *OSM* data, and use real satellite imagery from *MapBox* for terrain textures. Real locations can be used to study traffic congestion and vehicle networks, when used with supported tools such as *SUMO*. Figure 3.1 demonstrates the procedural generation of Manhattan, the most densely populated and detailed borough of New York City. The real world data was downloaded from a community powered database.
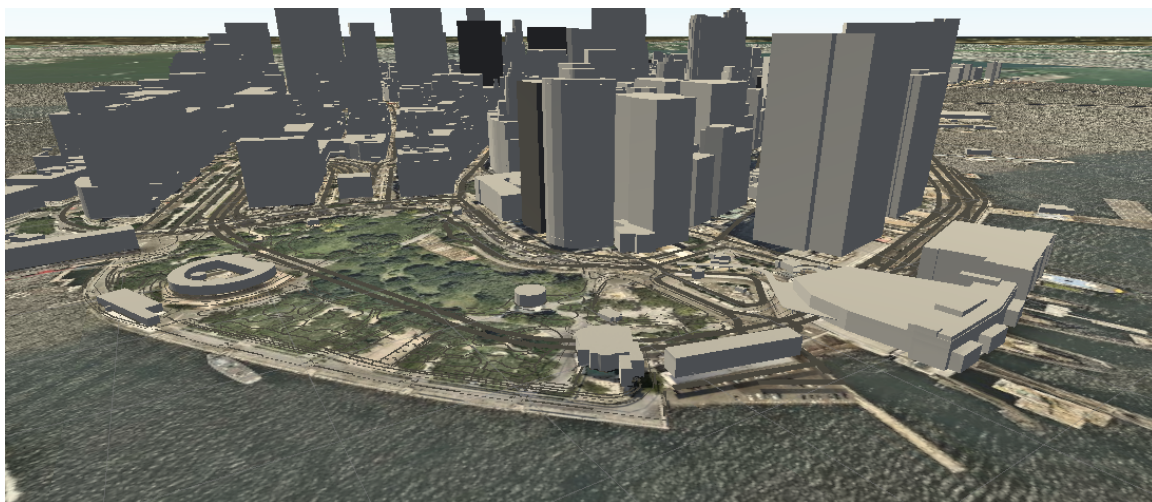

Figure 3.1: Procedural generation of Manhattan in *UniNet*. This image was taken after each building was blocked out, but before textures and details were added to the scene. Depending on the scale and detail of the scene, this process can take anywhere from 30 seconds to 10 minutes.

User Immersion

VR headsets encourage a new level of immersion, not often found with monitors or projectors. *UniNet*'s VR technology was designed to give the user a sense of presence in VR, or at least augment the immersion offered by consumer VR HMDs. The goal of *UniNet* is to replace traditional visual feedback with a personal VR experience, without introducing any compromises [22, 44].

Some high budget simulators have developed new ways of immersing the user, that aren't always practical for smaller research [24]. An example of a configuration that isn't feasible in most situations, is the use of an actual car as the cockpit for a driving simulator, designed to feature a familiar layout of controls (steering wheel, pedals) in order to not break immersion when operating the simulator [21]. Inside of a VR simulator, discrepancies between real-world controls and virtual controls may affect the user's immersion. Our novel solution is to use a stereoscopic passthrough camera, creating an MR system. Using this technology, we can superimpose the real world controls seen by the passthrough cameras onto the virtual car's dashboard. This technique is described further in this chapter, but is introduced in the following section.

UniNet also provides basic audio feedback from the user-controller vehicle, in the form of engine sounds. The sounds are controlled by the revolutions-per-minute of the engine, and the load factor on the engine. Ambient noise is provided to add realism to the simulated city and city traffic.

3.1.2 System Architecture

UniNet combines *Unity* and *SUMO* into a driving and traffic simulator, with many possible applications. Figure 3.3 offers a visual insight into how our simulator is
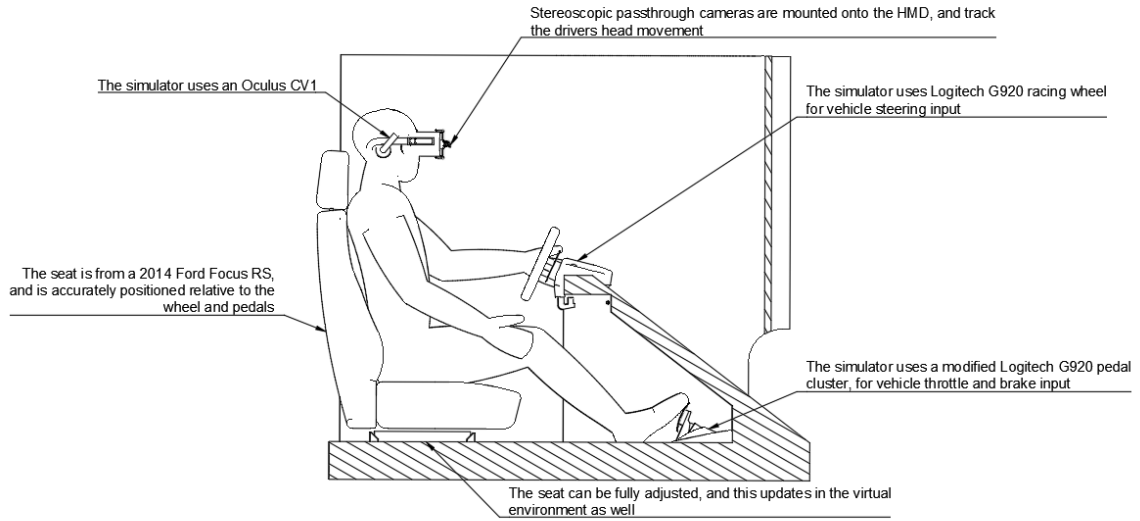
Figure 3.2: A cross-section side view, highlighting the specifications and scale of components in the simulator.

designed. In its current form, it can simulate and render hundreds of vehicles, with user input from a physical driving simulator controlling a virtual car. The user controlled car is driven from a custom built driving simulator featuring a green-screen chamber, where the user sees their own body in a virtual vehicle in real-time. This chamber is designed to surround the user such that they have 180° Field of View (FOV) coverage. Figure 3.2 provides a cross-section view of the simulator with a user for scale. At the beginning of the simulation, the user is given the option to procedurally generate a city, using real world locations as an input. The results are a full-scale copy of a real world city, that the user can drive in with virtual traffic. The traffic is generated by *Unity*, and sent to *SUMO* during the initialization phase. Each vehicle is updated by *SUMO* at a fixed time-step interval, and rendered by *Unity* to the virtual reality headset.

Figure 3.3: The *UniNet* architecture. Combining *SUMO*, *Unity*, and the supporting simulator hardware. This is a simplification of the architecture, meant to highlight the flow of data and execution of commands. The protocols interfacing *Unity* with the simulator hardware were omitted. *TraCI* is used for communication between *Unity* and *SUMO*.

3.1.3 System Specifications

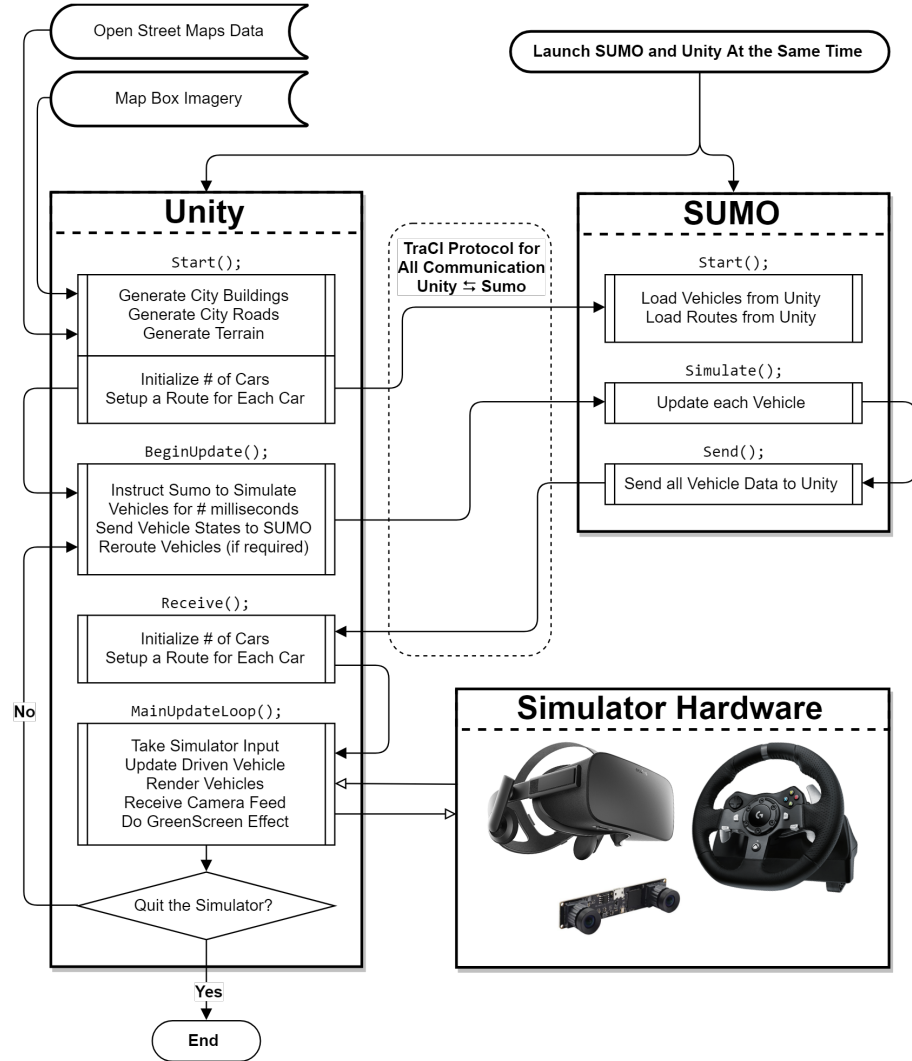|	Workstation #1	Workstation #2
Processor	Ryzen 5 1600x	Ryzen 5 2600x
Video Card	GeForce 970	(2x) GeForce 1080 Ti
RAM	16GB DDR3	32GB DDR4
Operating System	64-bit Windows 10	64-bit Windows 10

Table 3.1: System Specifications, for both workstations used throughout the development of *UniNet*

For the development of *UniNet*, both Workstation #1 and Workstation #2 were used (see Table 3.1). For the user study, we chose to use the more powerful Workstation #2, in order to drive the triple monitor configuration used as an immersion configuration.

3.2 Implementation

In addition to our use of *Unity* and *SUMO*, *UniNet* features high quality vehicle physics, with respect to realism. This section describes our implementation of TraCI [46] and *Vehicle Physics Pro* (*VPP*) [41], and their purposes. Each system comes together to create a high-fidelity driving simulator, which allows users to interact with traffic models in real time.

3.2.1 Vehicle Physics

Our initial simulator was designed and built around *Randomation Vehicle Physics* [20], an open source vehicle physics library. The appeal was its ease of integration into the *Unity* project. However, we later swapped to *VPP* [41] in favor of realism[1].

---

[1]We also considered *TORCS*, an open racing car simulator) [50], as an option for vehicle physics, but decided against it due to the appeal of *VPP*.

It is described as *"an advanced vehicle simulation kit for Unity 3D, providing fully realistic and accurate vehicle physics"* [41]. The integration of this physics library into our project was seamless, and we were able to focus on the technology for connecting *SUMO* and *Unity*.

The vehicle physics library was only used for the user-driven vehicle, and was not used for the visualization of traffic agents due to the complexity of the physics calculations. For this situation, we propose a *follow* technique with dead-reckoning. Each traffic agent updates their position to try and match the position and orientation of the cars simulated by *SUMO*. Due to the discrepancy in update rates, we use dead-reckoning to smooth this motion out. The *follow* algorithm follows a realistic steering model (Ackermann steering geometry) [41] to move, making for very convincing 3D movement.

3.2.2 Traffic Generation

SUMO [26] is an open source traffic simulator. It is capable of simulating thousands of agents traversing through a road network. It was our first choice for traffic simulation. The integration process was straightforward. For the pre-built city [38], we wrote a script to export the city map into a crude `*.net.xml` file, and used *NETEDIT* to clean it up [26]. *NETEDIT* was used to modify the direction of lanes, add traffic lights, and export the final `*.net.xml` file in the correct format for use in *SUMO*. We matched one-way streets and traffic lights with their visual counterparts in *Unity*.

SUMO is typically run from the console, but it could be run with the *SUMO* GUI (Graphical User Interface) option as well. We initialized *SUMO* so as to not simulate vehicle movement, unless instructed by an external process. We also set the duration of each simulated step to be 20 milliseconds. Vehicles are added and rerouted via TraCI [46]. So it is after doing these steps that we consider *SUMO* to

be fully configured. We designed *UniNet* to be the external process which commands *SUMO*
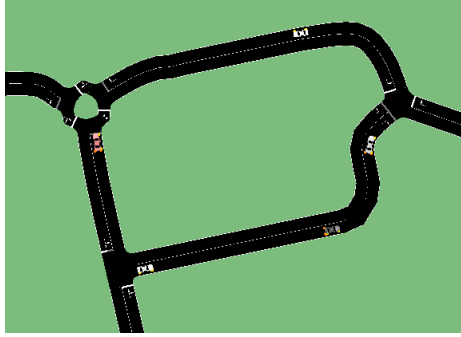
(a) City block as seen from SUMO

(b) City block as seen from the proposed *UniNet* simulator

Figure 3.4: *Unity* and SUMO are seen operating together. Our simulator in *Unity* provides enhanced graphics and user interaction.

Using an implementation of TraCI [46] in C# [7], we established a connection between *Unity* and *SUMO*. TraCI is used to populate the streets with cars from inside *Unity*, and connected each car with their agent in *SUMO*. When a user drives with the traffic simulation, a custom car is created, and labeled as an off-road vehicle type. This car is handled separately, and is mapped to the car powered by *VPP* inside of *Unity*. Its position is set each simulation update to match the position of the user car. In *SUMO* a vehicle can only occupy one lane at a time, so we also created a dummy car, and attached it to the rear bumper of the user controlled car. This prevents the simulated traffic agents from driving through the user's vehicle, when the rear bumper occupies a separate lane. Using *Unity*, we were able to add improved stop-sign behavior to *SUMO*. When the NPC vehicles enter a bounding box inside of *Unity*, their speed in *SUMO* is reduced to 0 for a specified amount of time. When their speed is restored, they continue as expected. Without this fix, NPC vehicles would roll through stop signs in a non-human like fashion, breaking the immersion for the driver. See Figure 3.4 for a side-by-side comparison of the same

city block, as seen in both applications in real time.

3.2.3 City Generation

We have developed an innovative automatic city generator, which uses real world map data as input. We also support traffic simulations in fictional cities, such as *WindRidge City* [38]. The advantages to a city procedurally generated from real roads includes: realistic road layouts, simple integration with map-based services, and real time generation of 3D models based on real building footprints.

Procedural City

To generate the city from real world map data, we found that a combination of data from *OSM* [9], and procedural mesh generation techniques implemented in *Unity* was our best option. The process of creating a city starts with specifying a region, using the longitude and latitude coordinate system. From here, the simulator can download the relevant satellite imagery, land use maps, building footprints, and roadways to create a representation of a real city. This process also works with rural and suburban areas. Algorithm 1 generates and textures 3D meshes for the roads, buildings, and terrain. All of this information is gathered from various services. *MapBox* [27] is a web service we used to download satellite imagery, heightmaps, and land-use maps. Satellite imagery is used to texture the terrain. Heightmaps are used to raise and lower the terrain, and the land-use maps have are used to control placement of vegetation and bodies of water.

The Unity game engine uses a Cartesian coordinate system, and all objects exist on a flat plane on the X and Z axis. Our city generator converts geographic coordinate system *longitude/latitude* pairs, into useable Cartesian coordinate system X/Z pairs. The method we use to convert the coordinates is called *Mercator projection.*

Algorithm 1: City Generation

    Data: $lonMin, lonMax, latMin, latMax$

    Result: Generates a city from OSM world data

1 $region \leftarrow Region(lonMin, latMin, lonMax, latMax)$;

2 $nodes \leftarrow openstreetmaps.DownloadNodes(region)$;

3 $ways \leftarrow openstreetmaps.DownloadWays(region)$;

    `// Generate 3D, textured terrain`

4 $texture \leftarrow mapbox.DownloadSatelliteImagery(region)$;

5 $heightmaps \leftarrow mapbox.DownloadHeighmaps(region)$;

6 $terrain \leftarrow GenerateTerrain(texture, heightmap)$;

    `// Extrude buildings from footprints`

7 foreach *Building b in ways* **do**

8 $buildingMesh \leftarrow ExtrudeBuilding(b)$;

9 $finishedBuilding \leftarrow TextureBuidling(buildingMesh)$;

10 $AddFinishedBuildingToCity(finishedBuilding)$;

11 end

    `// Generate roads from line segments`

12 foreach *Road r in ways* **do**

13 $roadMesh \leftarrow ExtrudeRoad(r)$;

14 $finishedRoad \leftarrow TextureRoad(roadMesh)$;

15 $AddFinishedRoadToCity(finishedRoad)$;

16 end

    `// Add details to the city`

17 $landuse \leftarrow mapbox.DownloadLanduseMap(region)$;

18 $Add3DPropsAndVegetation(terrain, landuse)$;

A drawback to the Mercator projection is that the distance from the equator will inflate distances coordinates, making distances between points inaccurate. A scalar multiplier θ is introduced and calculated based on the center of the downloaded city's bounding box. Its purpose is to scale coordinates further from the equator down, resulting in accurate distances. θ is multiplied into each of the incorrectly scaled X/Z pairs, and converted into a correctly scaled X/Z pair for use in *Unity*. We chose to scale all coordinates with the same θ value for simplicity, and as a speed optimization. We are aware that larger downloaded areas will become less accurate.

Due to floating point precision errors, we also needed a system to normalize the

bounds of the city around the origin inside *Unity* (0/0/0). This was simply done by computing the X/Z coordinate of the center of the downloaded region, and subtracted from each future coordinate processed by the city generator.

Heightmaps downloaded from *MapBox* [27] were sampled at each coordinate, and used to generate features such as hills, riverbeds, and mountains. The sampled height was also used when generating buildings and roads, giving a third dimension to our simulator.

City Model in the User Study

For the user study, we used *WindRidge city*. This city was designed by game developers and researchers for autonomous simulation [38]. One of the advantages to using this city, is its size. It contains all of the important features of a city in a relatively small footprint. In order to use this city with *SUMO*, we created a tool to map the roads inside of the *Unity* editor. This map is then exported as a `*.net.xml` file, and imported into *NETEDIT* as seen in Figure 3.5. It is then cleaned up, and used with *SUMO*. As a final step in preparing the city, we also swapped road signs to match the local road signs.
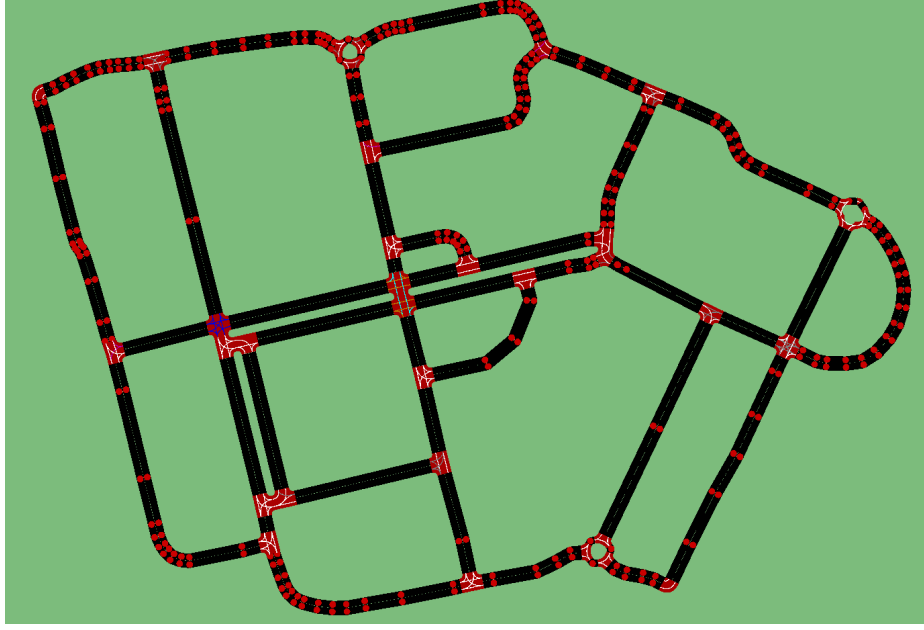
Figure 3.5: *WindRidge City* as seen in *NETEDIT*, showing lane directions and junctions for the simulation in SUMO.

3.2.4 Hardware

Passthrough Cameras

The term *passthrough virtual reality* refers to a configuration where the user can see the real world while inside a VR headset, via cameras built into or mounted on it. For our simulator, we use two HD (High-definition) cameras, to give the user a stereoscopic passthrough experience. The Stereoscopic camera pair are mounted to the front of an *Oculus Rift* [42], as seen in Figure 3.6. Properties of the camera mount are:

1. The interpupillary distance (IPD) is fixed at 60mm. This distance should closely match distance between the pupils in the users left and right eye, and 60 mm matches the human average IPD [49].

2. The downwards angle of the cameras is fixed at 5°. This is to compensate for

a mismatch between the physical cameras, and the virtual cameras inside of the *Oculus Rift*, where the vertical FOV does not match the *Oculus*. Since our camera mount is designed specifically for a driving simulator, objects in the lower FOV (steering wheel, hands, legs) are considered more important, justifying the fixed angle of the cameras. Without this fix, the user will not see his/her arms when looking straight ahead.

3. Both cameras are parallel. Typically with stereoscopic cameras or our eyes, the stereo convergence is adjusted based on the focal point. Due to hardware limitations, we implemented a software-based solution to stereo convergence. Our left and right cameras are offset in 2D to force objects in and out of focus. This focus is then adjusted to match the stereo convergence of the virtual cameras in the headset.

The stereoscopic camera pair is a 60 fps, 2560×960 USB camera, with a real time connection to *Unity*. The camera we are using has a latency of approximately 170 ms, which is compensated for inside of the game engine using a technique where the world space rotation of the headset in the virtual environment is recorded each frame. The cameras use the rotation information from the timestamp when it was captured, to choose their 3D orientation relative to the head. This allows the *Oculus Rift* and the virtual passthrough camera canvas to be synchronized. Simulator sickness was reduced by compensating for the latency of the cameras using this technique. The latency comes from the processor used on the camera's circuit board (see Figure 3.6b). A faster processor could encode and compress the video stream quicker, reducing the latency of the system.
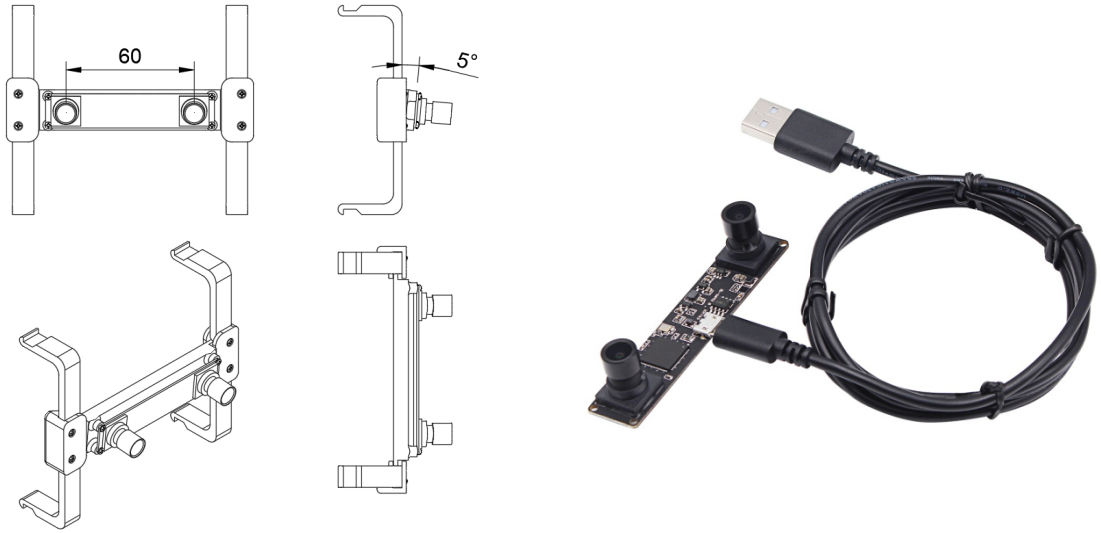
(a) The mounts for the front face of the *Oculus Rift*.

(b) The stereoscopic camera, with a no-distortion lens.

Figure 3.6: The 3D printed mount, for attaching the synchronized stereoscopic camera onto an *Oculus Rift*. It was designed to cover the least amount of constellation tracking LEDs possible.

Green Screen

In order to see the virtual world through the aforementioned passthrough VR system, we developed a green screen chamber, which surrounds the driving simulator completely. We use a real time green screen algorithm run on the Graphics Processing Unit (GPU), to present the virtual world to the user in MR. For the driving simulator, this has the unique advantage that the user will see their arms and a real vehicle dashboard, while driving in a virtual city and keeping the benefits of virtual reality. Figure 3.7 shows a third-person view of the simulator composited with the virtual scene, and Figure 3.8 shows what the end-result looks like, when the video-feed and virtual world are composited together.

The algorithm for the green screen is a form of chroma key compositing, to layer

29

Figure 3.7: A third-person view of a user driving the simulator in front of a green screen, composited with the virtual vehicle.

the captured camera feed onto the virtual world. Because our algorithm is run in parallel, we chose to use difference keying instead of chroma keying. Keying is a term used when two images are composited, based on chroma ranges (color hues). Difference keying is a similar algorithm, which uses the difference between red and green pixels in the source image to composite it onto the destination image. This has the disadvantage of limiting us to using only the color green, however it is more efficient.

Vehicle Input

UniNet is a standard driving simulator in terms of input. Our simulator uses an off-the-shelf Logitech G920 racing wheel, with force feedback. The clutch pedal was removed from the pedal cluster to avoid any confusion, as the vehicle we chose for the user study was an automatic transmission. Research into whether controllers affect

(a) Before: The passthrough cameras capture the user in the green screen chamber.

(b) After: The user is composited onto a virtual environment.

Figure 3.8: The simulator user can see their hands in virtual reality. This feature was added to help the user become immersed.

immersion, supports our choice in a racing wheel with high quality force feedback [28, 29].

Chapter 4

User Study

We designed a within-subjects experimental user study to test if *UniNet*'s MR system improved the user's sense of presence in the virtual environment. We compared our MR system with two VR systems, and one non-VR control. Section 4.1 describes our criteria for participants, and some demographic information about the recruited participants. Section 4.2 provides a technical description of the immersion configurations used in the study, and the physical hardware. Section 4.3 discusses the procedure of our sftudy, the experimental design, and the trials each participant was presented with. Section 4.4 presents all of our findings from the questionnaires, quantitative data gathered from *UniNet* itself, and qualitative results from a semi-structured interview. Finally, Section 4.5 summarizes our results. We discuss the findings, and how these complement or disprove related work. We also provide our interpretation of these findings.

4.1 Participants

27 unpaid participants were recruited for the study, and 24 were able to complete the study (15 male, 9 female). Our criteria for participants was a person with VR experience, or driving experience. The participants' ages ranged from 18-57 years old ($M = 27.75$, $SD = 9.821$), with driving experience ranging from 0-41 years ($M = 9.146$, $SD = 10.417$). Of the 24 participants, 13 required corrective lenses during the virtual reality experience. 10 of our participants had used VR 1-10 times in the past, with three participants having used VR 50+ times and four participants having never experienced VR. Three of our recruited participants were unable to complete the study, due to simulator sickness during their first VR trial.

4.2 Apparatus

The experiment was conducted with a custom driving simulator, built for the purpose of mixed reality simulations. The specifications of our workstations used throughout development can be found in Table 3.1.3. The MR simulator can be broken down into three core components: The VR headset, the green screen chamber, and the stereoscopic cameras. Section 4.2.1 discusses the VR headset used, Section 4.2.2 discusses the green screen chamber constructed, and Section 4.2.3 discusses our choice in cameras for pass-through mixed reality. Finally, Section 4.2.4 discusses the triple monitor configuration, used as a non-VR control.

4.2.1 Virtual Reality Headset

The VR headset is an Oculus *Rift CV1*, and features a 1080 × 1200 Organic Light-Emitting Diode (OLED) panel for each eye, running at 90 Hz. The diagonal FOV of
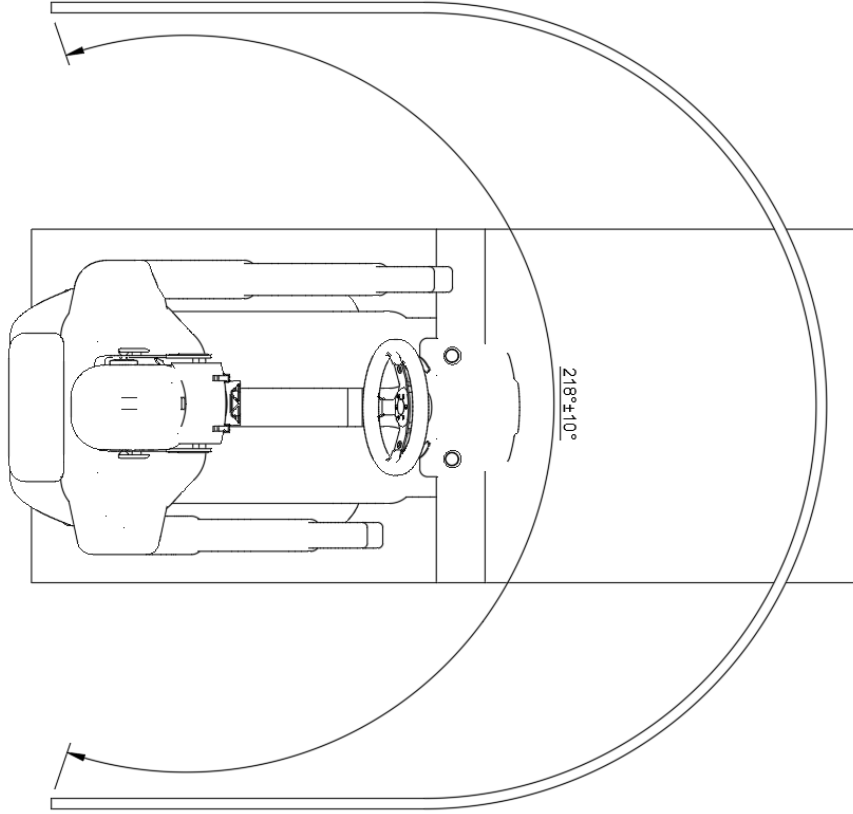
Figure 4.1: Field of View of the user in the green screen chamber.

each eye is 110°, and 94° horizontally. The Oculus *Rift CV1* features constellation tracking, which is an outside-in style of tracking where infrared LEDs cover the front and sides of the headset. The accompanying constellation sensor can track the position and rotation of the Oculus HMD with sub-millimeter accuracy and near zero latency [42].

4.2.2   Green Screen Chamber

The green screen chamber was custom built to surround the front of the user. It surrounds $\approx 220°$ of the user's FOV (see Figure 4.1). This configuration does not cover the upper FOV of the user, however it is compensated for in code by adding a virtual green screen to the scene using the HMD rotational information. The chamber

Figure 4.2: Flood lights mounted to the green screen chamber. They are angled to light all sides of the chamber.

is designed to roll forward and backward on rigid casters, allowing the user easy access in and out of the simulator (See Figure 4.3). LED flood lights are mounted onto the top brace of the green screen chamber. The lighting is mounted directly to the chamber, so that the orientation of the bulbs relative to the screen never changes (See Figure 4.2). The screen is curved to prevent shadows in corners of the fabric. This is crucial, because the real-time GPU implementation of the green screen algorithm can not compensate for incorrect lighting in real time.

4.2.3 Stereoscopic Cameras

The stereoscopic camera chosen is a synchronized pair of 960×1280 (960p) 60 Frames Per Second (FPS) cameras. Each camera is 1.3 megapixels, and capable of capturing 90° FOV without distortion. Due to their low operating voltage and working current (DC 5V, 160 mA~20 mA), the cameras are both powered with a single 15 ft micro USB cable, with no noticeable signal degradation. The cable length matches the length of the *Oculus Rift*'s HDMI/USB cable. The cameras are mounted strategically,

Figure 4.3: Rigid casters allow the green screen chamber to move forward and backward linearly, for accessing the driving simulator.

in order to minimize coverage of the constellation tracking infrared LEDs on the *Oculus Rift*. The mount was 3D printed using standard black polylactic acid (PLA) filament, and conform to the front of the *Oculus Rift*. The stereoscopic camera is tilted downward 10°, in order to compensate for the lower FOV that the cameras have, compared to the *Oculus Rift*. We chose to tilt the cameras down, so that the user's legs are in their FOV while driving, because in most cases nothing is presented vertically above the user's line of sight. Figure 4.4 shows our 3D printed mount. The stereoscopic camera is mounted in the centre of the *Oculus Rift*, matching the height of the users eyes.

(a) The Oculus *Rift CV1* with our mount
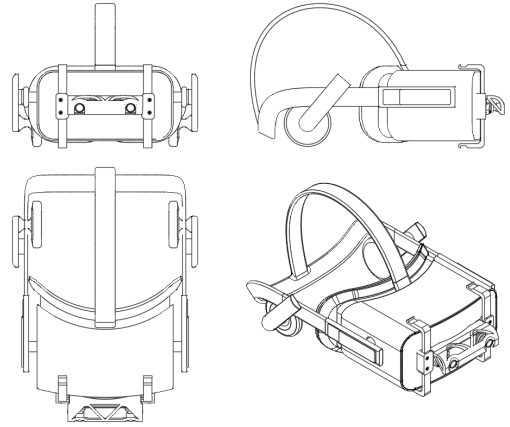
(b) Technical drawings for the camera mount, before 3D printing

Figure 4.4: 3D printed mounts for the cameras, allowed them to be mounted to the front face of an Oculus *Rift*.

4.2.4 Triple Monitor Setup

One of the trials in our user study used a non-VR configuration (see Figure 4.5). For this setup, we constructed a custom triple monitor rig, which can be wheeled in and out of position. Each monitor is 1920×1080 (1080p), with a combined resolution of 5760×1080. The rig is mounted onto a frame which can straddle the simulator. This was a requirement, in order to properly conduct our study. The experiment was counterbalanced using a 4×4 Balanced Latin square, therefore the non-VR and VR dependant conditions were constantly swapped.

4.3 Procedure

Participants began by completing a questionnaire about their driving experience, virtual reality experience, and demographic information. Upon completion, each user was presented a Positive and Negative Affect Schedule (PANAS) [45] questionnaire. The PANAS questionnaire is a 20 question self-report questionnaire, consisting of

Figure 4.5: The triple monitor rig used for the non-VR trial.

a 10-question positive scale, and 10-quesiton negative scale. Each item is rated on a 5-point Likert scale, and was administered to measure the positive and negative affect before the trials began. When finished with the questionnaires, participants began the study. The participants were seated in the driver's seat of *UniNet*, and the trial was briefly explained to the participant. See Section 4.3.2 for a description of the trials. After each trial was completed, the participant was administered three questionnaires:

- **Bob G. Witmer PQ:** We administered this questionnaire first, as the trial was fresh in the participants mind. The questionnaire has 21 questions, taken from the Witmer presence questionnaire v3.0. The questions were chosen in order to correctly analyze four factors from the 6-factor model discussed in the original paper. The factors analyzed were Involvement (Inv), Adaptation/Immersion (AI), Consistent with Expectations (CE), and Interface Quality (IQ). The fac-

tors we excluded were Audio Fidelity, and Haptic/Visual Fidelity, because the questions were either not relevant to our research, or constant between each configuration.

- **NASA Task Load Index (NASA-TLX):** The perceived workload of each configuration was evaluated using NASA-TLX, which is a multidimensional assessment tool, and widely used to assess tasks. Total workload is divided into six subscales. Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration. A lower score on each of these subscales represents a low perceived workload for a given task.

- **PANAS:** We administered the PANAS questionnaire after each trial, and once at the beginning of the study. PANAS is used as a mood measure, in the assessment of positive and negative affect. Affectivity is a term in psychology, describing when a person is influenced by their emotions.

After all trials and questionnaires were completed, a semi-structured interview was conducted.

4.3.1 Experimental Design

The study was a 4×4 mixed factorial design (4 orders $\times$ 4 configurations). Order was the between-subject independent variable and was counterbalanced using a balanced 4×4 Balanced Latin square. The within-subject independent variable was configuration. Four configurations were tested as follows: A MR configuration, where the user was seated in a green-screen chamber with our passthrough VR system; A VR configuration, where the user sees the interior of the vehicle with no virtual avatar; A VR configuration, where the user sees a virtual avatar in place of themselves, interacting with the vehicle; and a triple monitor non-VR control configuration, where

the user is seated in front of three HD gaming monitors. The dependent variables were reaction time (in seconds), Presence Questionnaire score, NASA-TLX score, and PANAS score.

(a) The MR configuration

(b) The VR with fake hands configuration
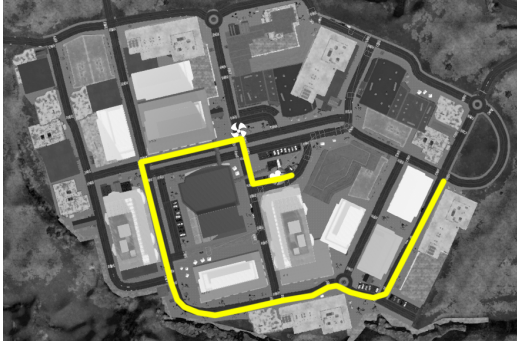
(c) The VR without hands configuration

(d) The triple monitor configuration

Figure 4.6: Each of the four configurations, as seen from the user's point of view.

4.3.2   Trials

We designed one trial for participants to complete for each configuration: four in total. Each trial was a similar scenario in *UniNet* where the user was presented with auditory and visual navigation cues from a virtual Global Positioning System (GPS) inside of the virtual car. The GPS was mounted to the dashboard. Each trial had

a unique route, and each of the four routes given by the GPS took approximately one minute. The lengths of each trial can be found in Figure 4.7. The trials were completed inside of a virtual Ford Focus. Each aspect of the simulated car has been recreated to match its physical counterpart. Throughout the duration of each trial, the user would encounter artificial traffic. The interactions were two-way, and the user influenced traffic congestion as well as navigating through any traffic. Near the end of each trial's route, an event was spawned to instigate a reaction from the participant. The events for the MR route and the triple monitor configurations, were car crashes. An NPC car would crash directly in front of the user. For the remaining two routes, the event was a jump-scare. An NPC car would leave a street-side parking spot as the participant was passing the parked vehicle. Both types of events instigated a reaction, either in the form of swerving or braking. The reaction times were measured by analyzing inputs to the vehicle, or collisions. A time-stamp when the event was spawned, was saved as an output, and the difference in time between the event and participant's input would be their reaction time in seconds. The events were designed to mimic traffic collisions, to encourage a more visceral reaction when the user was more immersed.

(a) The MR route (520 m)
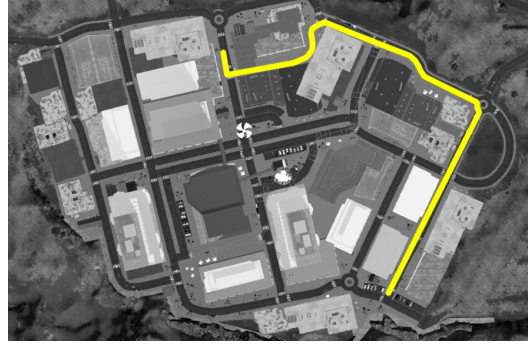
(b) The VR with fake hands route (490 m)
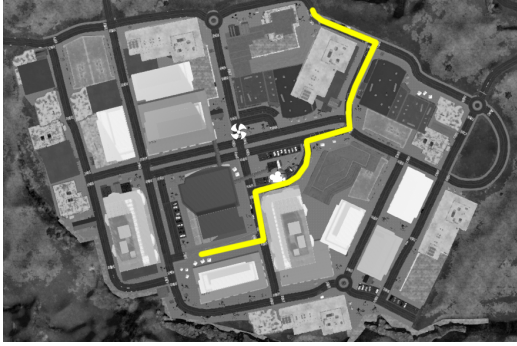
(c) The VR without hands route (430 m)

(d) The triple monitor route (510 m)

Figure 4.7: Each of the routes that participants followed during the corresponding immersion configuration.

4.4 Results

Our findings come from analysis of reaction times to the spawned reaction time events (Section 4.4.1), NASA-TLX responses (Section 4.4.2), responses from the Bob G. Witmer Presence Questionnaire V3.0 (Section 4.4.3), PANAS responses (Section 4.4.4), and an analysis of interview responses from a semi-structured interview (Section 4.4.5).

4.4.1 Reaction Times

Shapiro-Wilk normality tests revealed no significant deviations from normality for reaction times in all pairs of order and configuration. Similarly, Levene's test also did not reveal a signification deviation from homoscedasticity. The main effect of order was not significant, $F(3, 20) = 0.83$, ns. This suggests no ordering effect was found and, as a result, counterbalancing was successful. A Mauchly's test indicated that the assumption of sphericity had been violated for Configuration, $W = 0.46$, $p < 0.05$, therefore, the degrees of freedom were corrected using Greenhouse-Geisser estimates of sphericity ($\varepsilon = 0.725$). The results show the significant main effect of Configuration, $F(2.17, 43.5) = 10.66$, $p < 0.0005$, $\eta_G^2 = 0.24$. Post-hoc pair-wise t-tests with Bonferroni adjustment were performed. Significant differences were found as follows: MR ($M = 1.08, SD = 0.51$) and triple monitor ($M = 0.619, SD = 0.41$), $p = 0.00348$; Triple Monitor and VR without hands ($M = .749, SD = 0.347$) and MR, $p = 0.00034$; VR with fake hands and VR without hands ($M = 1.09, SD = 0.27$), $p = 0.000088$. Figure 4.8 shows the box plots of the reaction times measured for each trial.

```
              Pairwise comparisons using paired t tests


                              Mixd.R.   Trip.Mn.  VRw/FakeH
Triple Monitor                0.00348   -         -
Virtual Reality with Fake Hands 0.11392 0.76989   -
Virtual Reality without Hands 1.00000   0.00034   8.8e-05


             P value adjustment method: bonferroni
  Configuration                     count   mean    sd
  <fct>                             <int> <dbl> <dbl>
1 Mixed Reality                        24  1.08  0.515
2 Triple Monitor                       24  0.619 0.416
3 Virtual Reality with Fake Hands      24  0.749 0.347
```
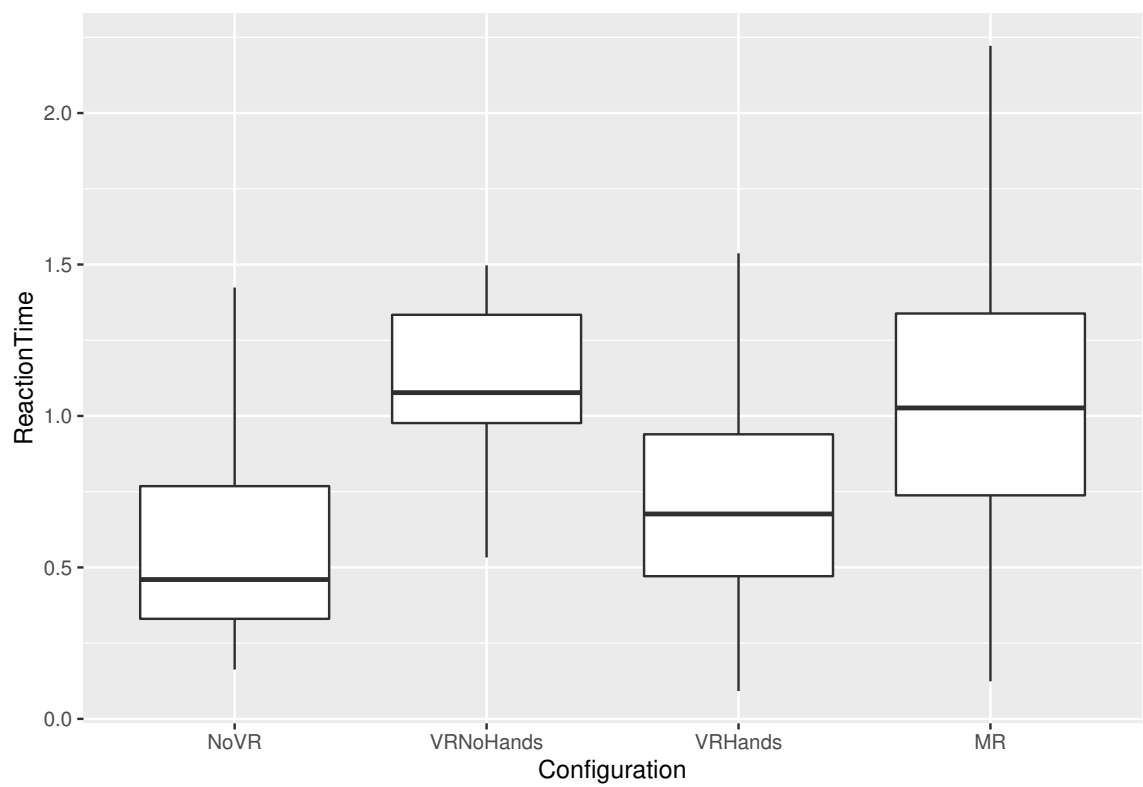
Figure 4.8: The box plots of the reaction times measured for each trial.

4.4.2   NASA-TLX

Shapiro-Wilk normality tests revealed multiple violations of normality for NASA-TLX score. As a result, a non-parametric alternative to repeated-measures Analysis of Variance (ANOVA) was used to analyze the data. A Friedman's test was carried out to compare the NASA-TLX scores for the four configurations of the setup. A significant difference was found, χ^2 (3) = 13.946, $p = 0.00298$, $W = 0.19$. A Conover post-hoc test (multiple comparisons of rank sums for unreplicated blocked data) with Benjamini & Hochberg adjustment revealed a significant difference between triple monitor ($M = 31.2$, $SD = 14.2$), and MR ($M = 27.8$, $SD = 21$), $p = 0.05$; Triple monitor and VR with fake hands ($M = 29.7$, $SD = 19.6$), $p = 0.033$. Figure 4.9 shows the box plots for the overall weighted NASA-TLX scores from each trial.
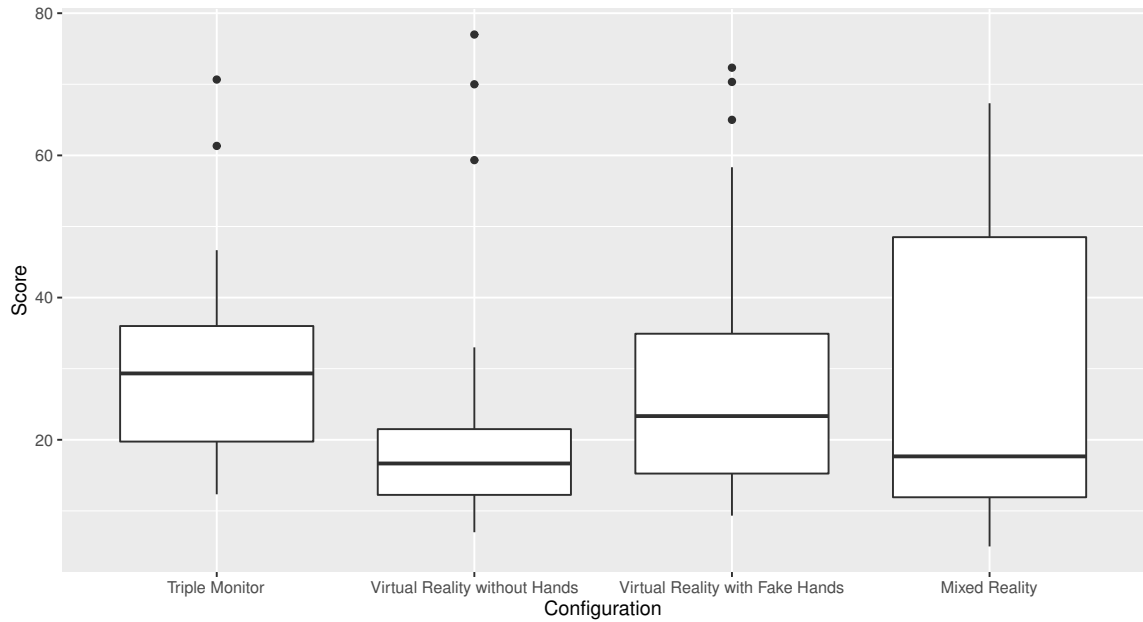

Figure 4.9: The box plots for the weighted NASA-TLX scores.

```
                           Mixd.R.   Trip.Mn.  VRw/FakeH
Triple Monitor                0.0503    -         -
Virtual Reality with Fake Hands 0.2952   0.0033    -
Virtual Reality without Hands   0.4713   0.1329    0.1247


  Configuration                     count   mean     sd
  <fct>                             <int> <dbl> <dbl>
1 Mixed Reality                      24   27.8   21.0
2 Triple Monitor                     24   31.2   14.2
3 Virtual Reality with Fake Hands    24   29.7   19.6
4 Virtual Reality without Hands      24   22.6   18.9
```

Galy *et al.* propose a method of analyzing the gathered NASA-TLX data, which is to analyze the individual subscales [13]. Similarly to the overall score, Shapiro-Wilk normality tests revealed multiple violations of normality for the raw NASA-TLX scores of individual subscales. As a result, just like with the overall score, Friedman tests were carried out to compare the raw NASA-TLX subscale scores for the four configurations of the setup.

```
                           Mixd.R.   Trip.Mn.  VRw/FakeH
Triple Monitor                0.059     -         -
Virtual Reality with Fake Hands  0.774    0.062     -
Virtual Reality without Hands    0.344    0.307     0.430
```

A significant difference was found for 'Performance', $\chi^2(3) = 8.6502$, $p = 0.03432$, $W = 0.12$. A Conover post-hoc test with Benjamini & Hochberg adjustment did not reveal significant differences at $\alpha = 0.5$. Differences were found at slightly above thresholds as follows: Triple Monitor ($M = 40$, $SD = 22.6$) and VR with fake hands ($M = 30.8$, $SD = 22.7$), $p = 0.062$, MR ($M = 22.9$, $SD = 21.6$) and and triple monitor, $p = 0.059$.

For 'Frustation', at $\alpha = 0.5$, a Conover post-hoc test with Benjamini & Hochberg

adjustment revealed a significant difference between MR ($M = 15.6$, $SD = 17.1$) and Triple Monitor ($M = 35$, $SD = 21.5$) , $p = 0.0039$. At thresholds slightly above, differences were also found between MR and VR with fake hands ($M = 26.5$, $SD = 21$), $p = 0.0617$, and MR and VR without fake hands, (14.5), $p = 0.0617$.

```
                                 Mixd.R.  Trip.Mn. VRw/FakeH

Triple Monitor                   0.0039   -        -

Virtual Reality with Fake Hands  0.0617   0.2581   -

Virtual Reality without Hands    0.0617   0.2581   0.9055
```

4.4.3 Bob G. Witmer PQ

Normality tests revealed no significant deviations from normality for the scores in all of the four factors: Adaptation/Immersion (AI), Consistent with Expectations (CE), Interface Quality (IQ) and Involvement (Inv). However, significant outliers were discovered for AI and IQ. See Figure 4.10. As a result, we performed mixed ANOVA tests on CE and Inv, and Friedman test (a non-parametric alternative to repeated-measures ANOVA) on AI and IQ scores.
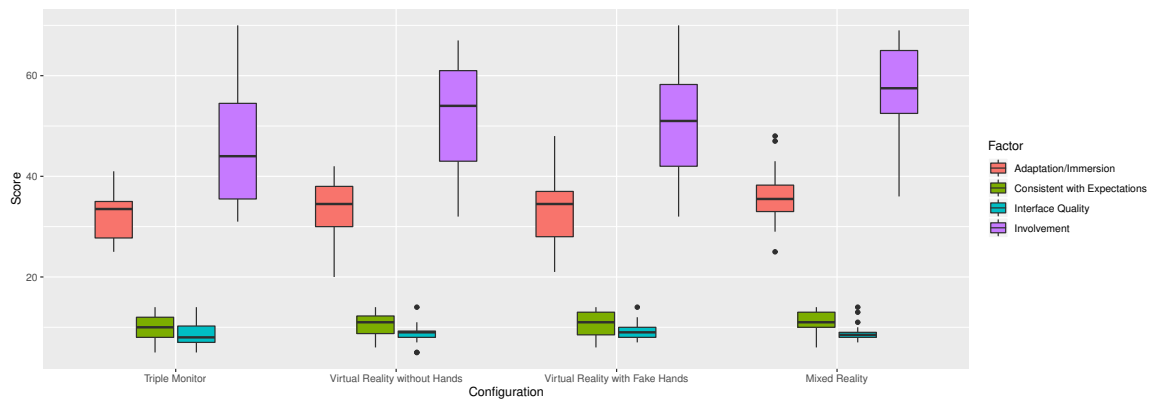

Figure 4.10: Scoring for four of the original six factors from the Bob G. Witmer PQ questionnaire. Due to the number of questions determining each factor, Involvement is scored from 0 to 70, Adaptation/Immersion is scored from 0 to 49, and each other factor is scaled from 0 to 14.

AI: A significant difference was found, $\chi^2(3) = 10.92$, $p = 0.01217$, $W = 0.15$. Despite this, a Conover post-hoc test with Benjamini & Hochberg adjustment failed to reveal any significant difference between the configurations.

IQ: No significant differences were found between the configurations, $\chi^2(3) = 5.1659$, $p = 0.16$, $W = 0.07$.

CE: The main effect of configuration was not significant, $F(3, 60) = 1.93$, $p > 0.05$.

Inv: The main effect of order was not significant, $F(3, 20) = 0.65$, ns. This suggests no ordering effect was found and, as a result, counterbalancing was successful. The main effect of configuration was significant, $F(3, 60) = 10.15$, $p < 0.0001$, $\eta^2_G = 0.14$. Post-hoc pair-wise tests with Benjamini & Hochberg adjustment were performed. Significant differences were found as follows: Triple monitor ($M = 45.5$, $SD = 11.3$) and MR ($M = 56.8$, $SD = 9.1$), $p = 0.00056$; VR with fake hands ($M = 50.5$, $SD = 11.6$) and MR, p $= 0.019$; MR and VR without fake hands, $p = 0.02687$; Triple monitor and VR with fake hands, $p = 0.009$; Triple monitor and VR without hands ($M = 52.2, SD = 10.3$), $p = 0.00912$. The results for each individual question from the questionnaire are summarized and shown in the diverging stacked charts in Figure 4.11, 4.12 and 4.13.

```
                              Mixd.R.    Trip.Mn.   VRw/FakeH
Triple Monitor                0.00056    -          -
Virtual Reality with Fake Hands 0.01909  0.00895    -
Virtual Reality without Hands   0.02687  0.00912    0.31955


  Configuration                 count   mean     sd
  <fct>                         <int>  <dbl>  <dbl>
1 Mixed Reality                    24   56.8   9.10
2 Triple Monitor                   24   45.5   11.3
```

```
3 Virtual Reality with Fake Hands     24  50.5 11.6

4 Virtual Reality without Hands       24  52.2 10.3
```
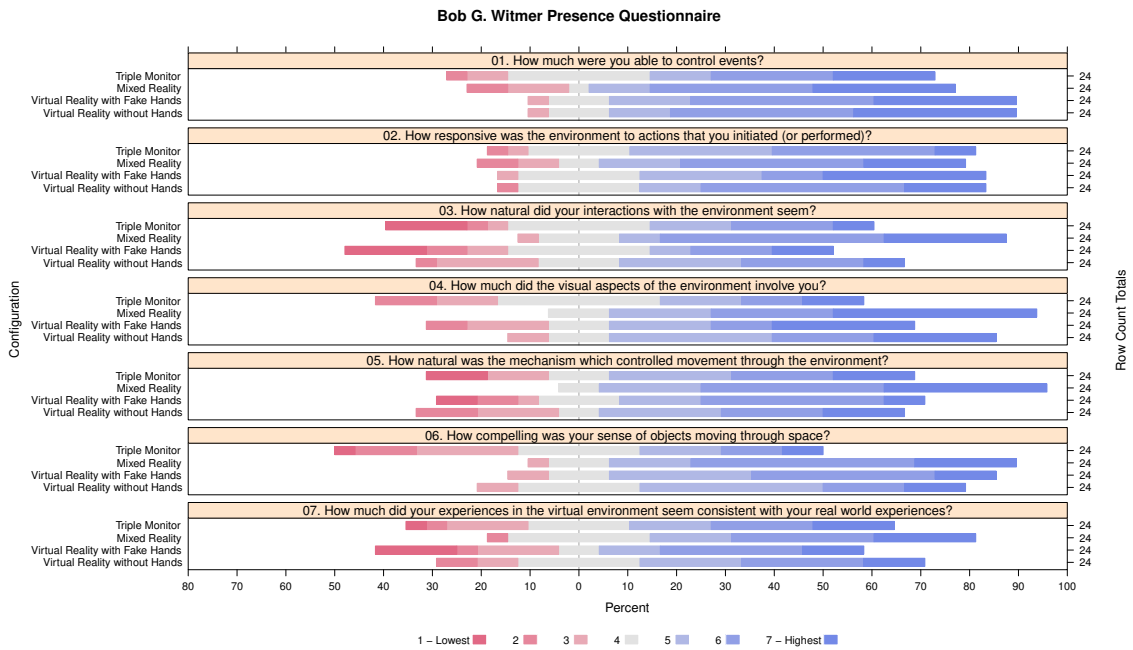

Figure 4.11: The Bob G. Witmer Presence Questionnaire was administered after each trial for each participant (Q.1 - Q.7).
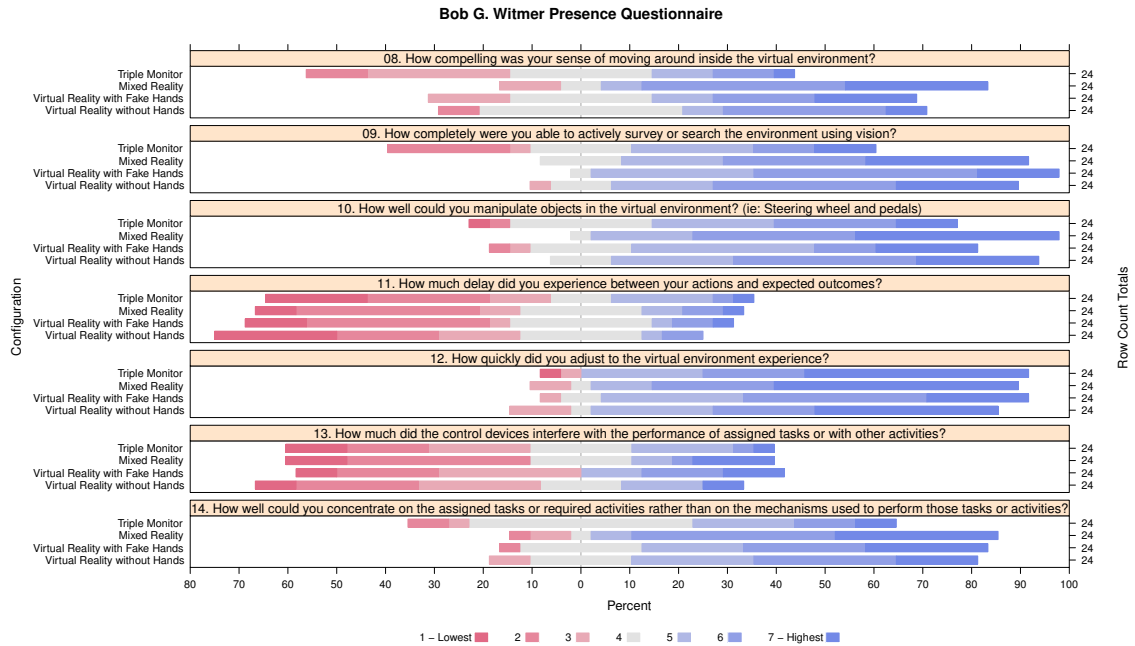
Figure 4.12: The Bob G. Witmer Presence Questionnaire was administered after each trial for each participant (Q.8 - Q.14).
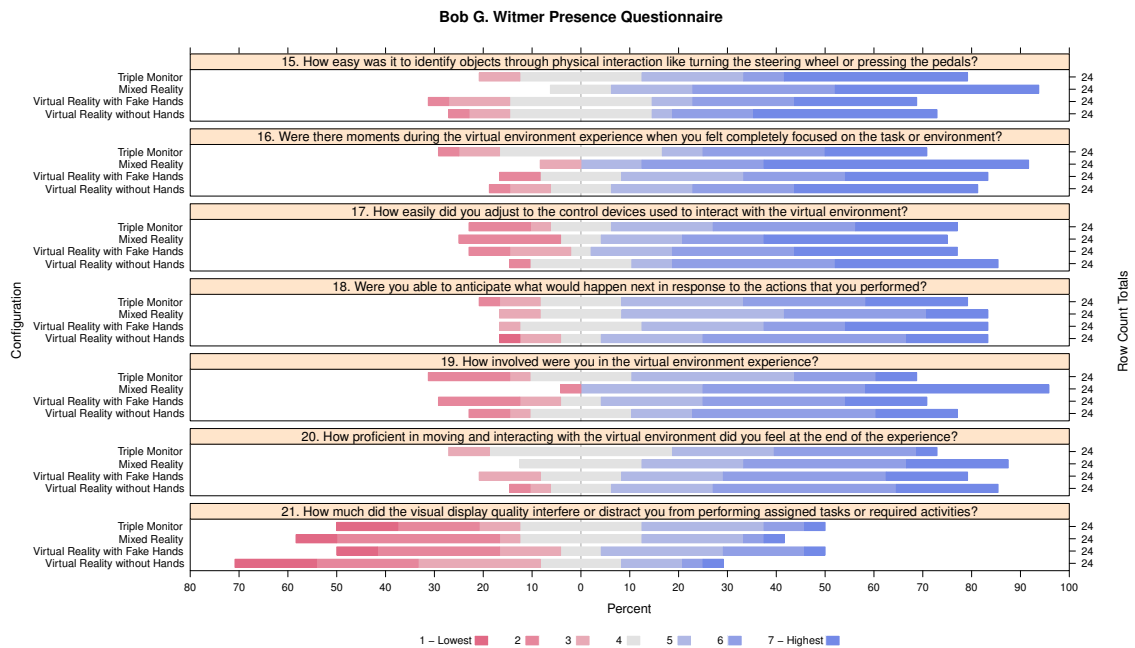

Figure 4.13: The Bob G. Witmer Presence Questionnaire was administered after each trial for each participant (Q.15 - Q.21).

4.4.4 PANAS

Shapiro-Wilk normality tests revealed multiple violations of normality for the PANAS scores in both: positive and negative affect. As a result, a non-parametric alternative to repeated-measures ANOVA was used to analyze the data. A Friedman's test was carried out to compare the affect scores for the five administrations of the PANAS questionnaires. No significant difference was found for the negative affect scores, χ^2 $(4) = 7.6532$, $p = 0.1$. However, a significant difference was found for the positive affect scores, χ^2 $(4) = 12.787$, $p = 0.012$, $W = 0.13$. A Conover post-hoc test with Benjamini & Hochberg adjustment revealed a significant difference between the start of the experiment $(M = 32.1, SD = 7.24)$, and administration after the triple monitor configuration $(M = 30.2, SD = 9.51)$, $p = 0.0074$.
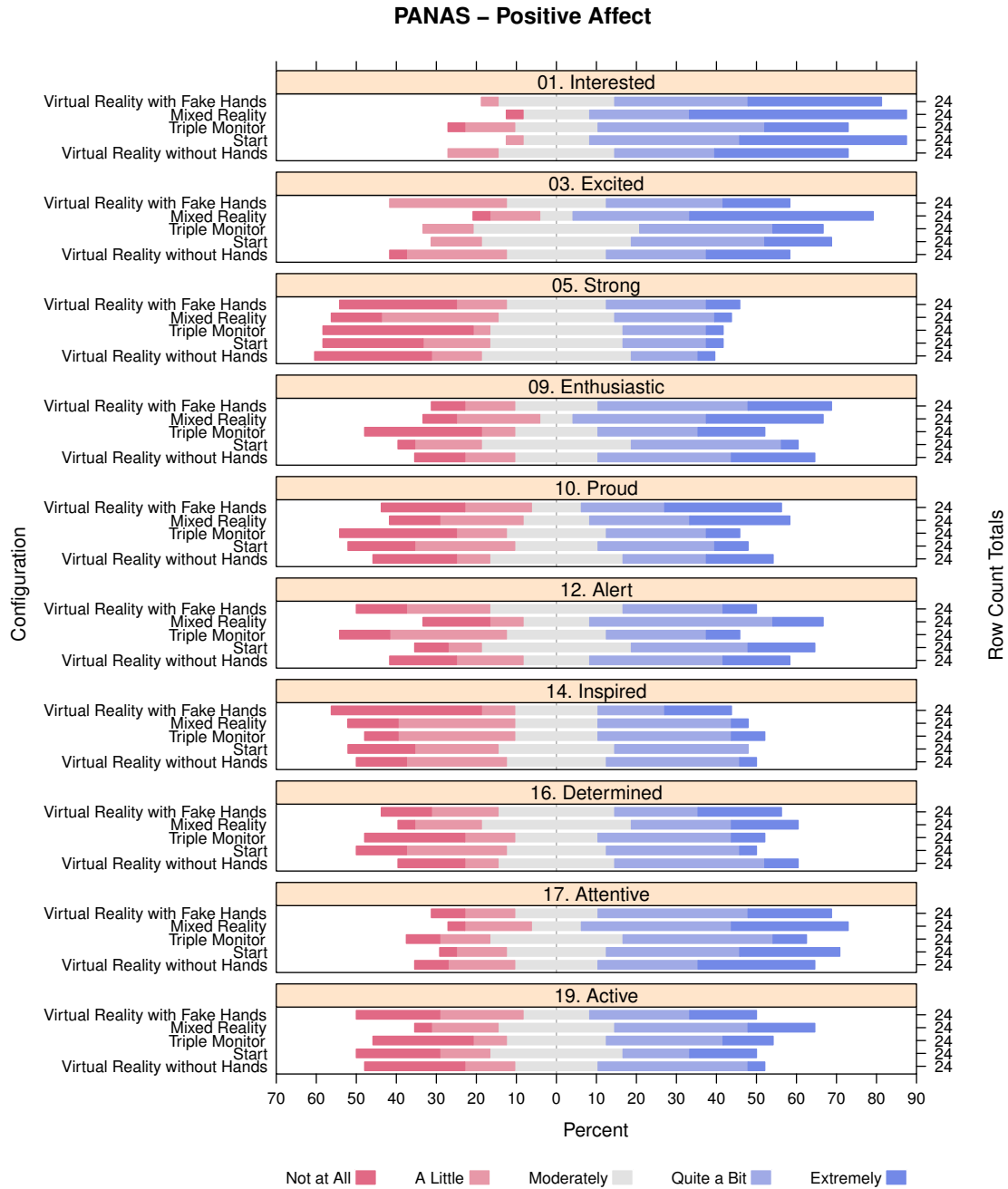
Figure 4.14: PANAS Positive results.

Figure 4.14 shows that participants felt interested and excited towards the MR configuration, and at the beginning of the study. Less relevant emotions such as strength and inspiration do not see much variation between the configurations.
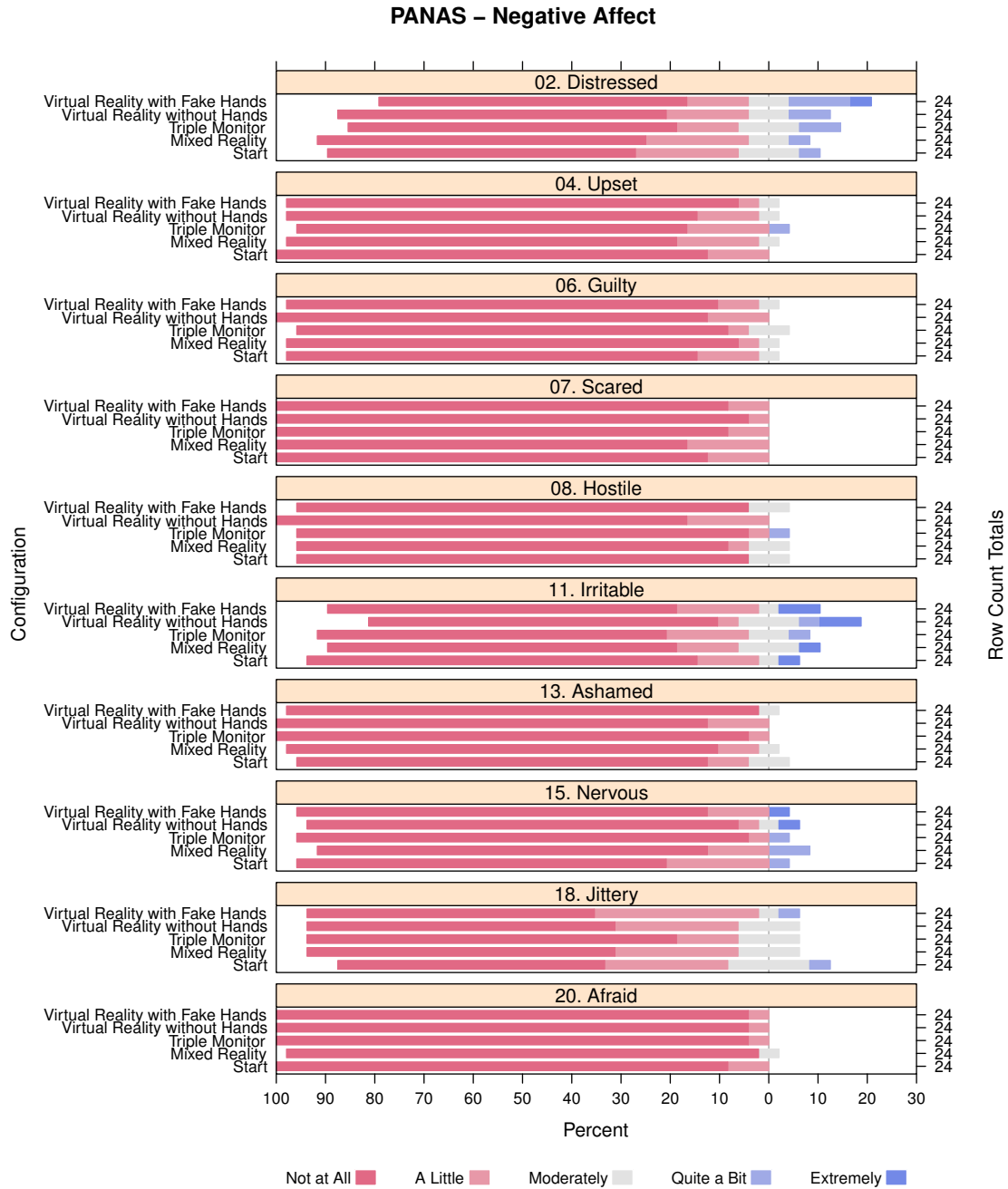
Figure 4.15: PANAS Negative results.

Figure 4.15 shows that most negative emotions are neutral, and similar for participants among configurations. At most, only 20% of participants felt negative emotions during the study, as seen with the Distressed and Irritable questions.

```
                                Mixd.R.   Start     Trip.Mn. VRw/
   FakeH
Start                           0.1211    -         -        -
Triple Monitor                  0.2905    0.0074    -        -
Virtual Reality with Fake Hands 0.9250    0.1211    0.2938   -
Virtual Reality without Hands   0.8640    0.1721    0.2038   0.8640


   Configuration                    count   mean     sd
   <fct>                            <int> <dbl> <dbl>
1 Start                               24   32.1   7.24
2 Mixed Reality                       24   34.5   8.62
3 Triple Monitor                      24   30.2   9.51
4 Virtual Reality with Fake Hands     24   32     9.93
5 Virtual Reality without Hands       24   31.6   9.29
```

4.4.5   Qualitative findings

At the end of our study, we conducted a semi-structured interview. The interview questions were designed to gather user opinions about their preferred configuration, and their reasoning. Some questions were taken modified from a presence questionnaire, which we chose to not administer because it required heavy modification. We also asked questions to investigate how immersed the user felt throughout the study. For all 24 users, MR was the preferred configuration, when answering Q1 (See Table 4.1). For some, it was due to the novelty of the new technology, however participants explained that their inputs seemed to have more effect on the vehicle in the MR configuration. This was not the case, and the inputs to the vehicle remained the same among all configurations. This could be attributed to a heightened sense of presence and a better connection to the vehicle, compared to the VR trials. We anticipated critical feedback regarding the low pixel density of the MR cameras and

measured 150–180 ms of latency, however no users mentioned this during the study. No additional simulator sickness was experienced in MR among our participants. The sound-cancelling headphones were mentioned by 8/24 users, when asked Q4. They were described as a major contributing factor, to the reason they felt immersed, as oppose to being consciously aware of the room the study took place in. An interesting finding was that most users felt this way even with the triple monitor configuration. The lab where the study took place was very calm, and this might not have been the case if the environment was crowded with people or other distractions.

	Question
Q1	Can you elaborate about which immersion configuration you liked more and why?
Q2	Can you elaborate which immersion configuration you disliked and why?
Q3	To what extent did the simulation hold your attention?
Q4	To what extent did you feel consciously aware of being in the real world whilst driving?
Q5	To what extent were you aware of yourself in the virtual environment?
Q6	To what extent did you feel that the simulation was something you were experiencing, rather than something you were just doing?
Q7	Would you like to drive the simulator again? If so, which immersion configuration?

Table 4.1: The verbal script for the semi-structured interview administered after the study.

In response to Q7, all users mentioned they would drive the simulator again if given the opportunity, and 23/24 participants mentioned they would choose the MR immersion configuration if given the choice. A single participant mentioned they would only try the triple monitor configuration again if given the choice. This participant experienced mild simulator sickness, which was their reasoning. However despite the motion sickness, the MR immersion configuration was their response to

Q1 due to its novelty.

The most recurring feedback we received in regards to *UniNet* which was not a question during the interview, was that motion feedback or haptics in the seat/pedals would improve the overall experience. This is something we plan on investigating in future works.

4.5 Discussion

In our user study, we followed a within-subjects experimental design, to test if *UniNet*'s MR immersion system improved the user's sense of presence in the virtual environment. We are able to show that the MR configuration is more immersive, however the results are primarily subjective, and come from the questionnaires we chose to include in our study.

Our study produced two types of results for each participant. Subjective results, and behavioural results based on reactions to vehicle collision events. Our analysis of the subjective results supported our hypothesis, however we could not draw any conclusions from the behavioural results. Insko writes that, due to presence being a subjective sensation, subjective means of measuring presence have become the most popular [19]. Therefore our inability to corroborate the results from our questionnaires with the behavioural measurements taken, does not disprove our hypothesis.

The in-simulator portion of the study contained four trials, designed to compare four configurations of *UniNet*, and instigate visceral reactions which we measured as reaction times. The reason we chose to compare four configurations, was to compare common existing options for VR simulations, and a non-VR control with our technology. In summary, here is a brief description of why we chose these four immersion configurations.

1. **Mixed Reality**: This configuration is the most unique, and features our technology combined with existing VR technology. Each user is presented with a unique experience, featuring their own body as a virtual avatar in the virtual environment.

2. **Triple Monitor**: This is the configuration most people are familiar with, and acted as a control for our study because it does not use VR technology. Instead, it relies on three monitors, with the outer monitors angled in to give the user a higher field of view.

3. **Virtual Reality with Fake Hands**: Providing the user with a virtual avatar is a common configuration in many VR simulators, and can help with the logic of a scene, for instance: In our configuration, the wheel of the car is turned by a virtual avatar instead of turning by itself.

4. **Virtual Reality without Hands**: VR without hands is another existing option featured in many VR simulators, and provides the user with an experience that is not interrupted by a virtual avatar.

None of the trials presented were timed, and users were allowed to take as much time as needed to finish the trials. Given an average speed of 40 km/h, each trial takes approximately 1 minute. However we noticed that users drove faster than this limit. Due to the varying speed participants were driving, the reaction time events were not as urgent for each participant. Figure 4.8 in Section 4.4.1 shows an average reaction time above 1 second for the MR trial, and an average reaction time less than 0.5 seconds for the triple monitor trial. We have concluded that the difference in reaction times is attributed to the events presented to the users. In order to prevent learning bias from participants, we crafted four unique trials, however the trials are unique in more than one sense. They do not guarantee a reaction time relative to the

instant they are created. For instance, the mixed reality trial's reaction time event is a car crashing into a fire hydrant in front of the path that the participant takes (See Figure 4.16).

Figure 4.16: The reaction time event in the Mixed Reality trial.

The Bob G. Witmer presence questionnaire was the first questionnaire participants completed after each trial. Figure 4.10 shows a high level of involvement with the MR immersion configuration. The involvement questionnaire featured questions such as *"How natural did your interactions with the environment seem?"* and *"How compelling was your sense of objects moving through space?"* and *"How involved were you in the virtual environment experience?"*. These results are significant in direct comparison with our non-VR control, as per the post-hoc pairwise tests. The responses with the Involvement factor for MR ($M = 56.8$, $SD = 9.1$) and triple monitor ($M = 45.5$, $SD = 11.3$) had a p-value of 0.00056. The triple monitor immersion configuration has similar, but less significant differences paired with VR with fake hands ($p = 0.00895$) and VR without hands ($p = 0.00912$). This shows that as a control, the triple monitor was the least immersive as per the PQ questionnaire's in-

volvement factor, and MR is the most immersive configuration. Another interesting result is the difference in involvement between our two VR configurations with hands and without hands ($p = 0.31955$). It is significantly less significant.

The NASA-TLX questionnaire was the second questionnaire participants completed after each trial. The purpose of the NASA-TLX questionnaire is to assess the perceived workload of a task or system. We observed significant differences between the task load index of the triple monitor immersion configuration ($M = 31.2$, $SD = 14.2$) and the VR with fake hands immersion configuration ($M = 29.7$, $SD = 19.6$), with a p-value of 0.0033. This could be due to the fact that the 'Performance' scale on the NASA-TLX questionnaire may have been biased by the reaction time events that were spawned. Due to the differences in these events, the user's self-perceived performance could be viewed as unsuccessful (producing a higher score), as seen in the case of the triple monitor configuration. The VR without hands immersion configuration, may have had a simpler driving scenario, which would result in a lower score. This is due to the fact that the task load index of each trial is similar enough, that performance and frustration may be the only significant factors.

We analyzed the performance and frustration factors individually [13], and found significant differences between Triple Monitor and MR immersion configurations. For the 'Performance' factor, the Triple Monitor ($M = 40$, $SD = 22.6$) and MR ($M = 30.8$, $SD = 22.7$) configuration showed significant difference with a p value of 0.059. This could be attributed to the lower FOV with the Triple Monitor immersion configuration, as we noticed worse performance among participants when turning at intersections and junctions. Users' self-perceived performance was also highest in the MR configuration. For the 'Frustration' factor, the Triple Monitor ($M = 35$, $SD = 21.5$) and MR ($M = 15.6$, $SD = 17.1$) configuration showed sig-

nificant difference with a p value of 0.0617. This could be due to the same reasons as the performance factor. Overall, performance and frustration could be signs of a heightened sense of presence in the MR configuration.

The PANAS questionnaire was the final questionnaire participants filled out, before either completing the study, or beginning the next trial. It was also administered after the general information questionnaire used to gather participant information at the beginning of the study. The purpose of this questionnaire is to gauge the emotions, or mood, of the participants. The questionnaire was originally designed by taking terms with a strong connection to one dimension (positive or negative), and a weak connection with the opposite dimension. We found that the positive mood of the participants at the start of the study ($M = 32.1$, $SD = 7.24$) was significantly higher than their positive mood during the triple monitor immersion condition ($M = 30.2$, $SD = 9.51$), with a p-value of 0.0074 (See Figure 4.10). The balanced design of the study means that this measured difference is likely not due to participants mood changing over the course of the study itself. The PANAS questionnaire uses a 5-point Likert scale, and we noticed high 'Interested' emotions (positive) after the start and MR immersion configuration. We also observed the highest level of 'Excitement' (positive) after the MR immersion configuration. The triple monitor configuration yielded the lowest overall 'Enthusiastic' (positive) emotion. The 'Distressed' emotion (negative) was significantly higher during the VR with fake hands trial than it was during the MR trial. This result could be due to the uncanny appearance of the virtual avatar used during the VR with fake hands immersion configuration.

Our results show a heightened sense of immersion was experienced by users, in *UniNet*'s MR immersion configuration. These conclusions were drawn from the results of the Involvement factor of the Bob G. Witmer presence questionnaire, individual questions from the PANAS questionnaire, and our qualitative findings from the semi-

structured interview.

Chapter 5

Conclusion

As the market for VR continues to grow, the development of MR technology should grow with it. The reality-virtuality continuum is defined by the mixed-reality area between reality and virtuality, and *UniNet* was designed to fit within this range. This thesis focused on the effect of user presence in a MR driving simulator, and the construction of a physical product.

The study outlined in Chapter 4 investigated the effect of our MR immersion configuration, on user presence. The study hypothesized that our MR configuration would increase the user's sense of presence in the virtual environment, when compared to traditional VR and non-VR configurations. Participants were presented with four trials to complete in *UniNet*, and each trial finished with a vehicle collision event to create a behavioural response from participants. The subjective results were significant, and in favor of our study's hypothesis.

Prior to the study, we designed and tested the hardware and software for *UniNet*. *Unity* and *SUMO* are the primary systems controlling player vehicles and NPC vehicles respectively. Our technology is built to work with the *Oculus Rift*, using commercially available stereoscopic cameras mounted to the front face of the HMD. Our soft-

ware creates a passthrough VR experience with this hardware configuration. When combined with the green screen chamber constructed for *UniNet*, our technology fits on the reality-virtuality continuum as a unique mixed reality experience.

5.1   Lessons

To create a system like *UniNet*'s presence system, requires problem solving. The following are the steps taken to resolve issues that were encountered.

Tracking and anchoring the camera stream onto the virtual world

To reduce the latency problem, we projected the camera feed in the direction that the user's head was facing at the instant the image was captured. With our configuration we had an average latency of 170 ms, and using this amount of latency as an example, we projected the camera feed relative to the virtual camera with the orientation the user's head had 170 ms prior. The result is an image that is correctly anchored to the virtual world, however is 170 ms behind.

Lighting inconsistencies with the green screen

To improve the difference keying algorithm, our green screen was curved around the user. We chose a cloth material, and tensioned it to remove wrinkles. The green screen chamber has the ability to roll forward and backward, but to keep consistent lighting, we fixed LED flood lamps to the chamber. The lights retained their position relative to the green screen with this configuration.

Matching the camera properties with the virtual camera properties

The FOV of the virtual cameras and Oculus HMD cameras are all known values, and we chose the stereoscopic camera to closely match these values. The cameras already had minimal distortion, but we still removed the distortion. Using a chessboard tracking pattern and OpenCV, we were able to remove the remaining distortion. The calibration variables received in OpenCV were used with a GPU version of the algorithm, and we prevented further Central Processing Unit (CPU) bottleneck.

5.2 Limitations and Future Work

Further research must be put into measuring presence quantitatively. The user study would benefit from a revisit with more focused subjective measurements, and better controlled behavioural measurements. The behavioural measurements we took could not be interpreted to their fullest potential, and similar work shows that these types of measurements are still viable if implemented correctly [33,40]. The behavioural results from our study did not show significant results, and our collection of behavioural data could be improved greatly. The hardware for *UniNet* could be improved with time, and simplified. The current iteration of the hardware has a limited FOV and camera resolution, which can be improved upon with better hardware.

It is also important to further research the impact of latency and camera properties on user presence in MR. Throughout our user study, users experienced camera latency of over 150 ms, with no negative side effects or additional simulator sickness. Furthermore, our green screen had a sub-par lighting configuration, and shadows caused artifacts near the bottom of the user's peripherals.

UniNet has the potential to be paired with VANET specific applications for networking research, which was tested but not fully explored. Future work could explore

the use of *UniNet* in this academic context, and how user interaction in real time can affect V2V networks.

Future work can study methods of enhancing the green screen algorithm, via disparity mapping from the stereoscopic camera rig used for the passthrough VR. This would solve the current problem of lighting issues, as both depth mapping technology and green screen technology could create a better key for compositing the user onto the virtual environment.

Future work could also explore the use of motion feedback presented to the user, in the form of a full motion simulator. This would require a complete rebuild of *UniNet* from the ground up, with a different purpose in mind. The motion feedback was the most common feedback received from participants of the user study.

Bibliography

[1] ARTAUD, A. *Theatre and its Double*. Alma Books, 2018.

[2] BILLINGHURST, M., CLARK, A., LEE, G., ET AL. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction 8*, 2-3 (2015), 73–272.

[3] BIURRUN-QUEL, C., SERRANO-ARRIEZU, L., AND OLAVERRI-MONREAL, C. Microscopic driver-centric simulator: Linking unity3d and sumo. In *World Conference on Information Systems and Technologies* (2017), Springer, pp. 851–860.

[4] BOWMAN, D. A., AND MCMAHAN, R. P. Virtual reality: how much immersion is enough? *Computer 40*, 7 (2007), 36–43.

[5] BOYLE, L. N., AND LEE, J. D. Using driving simulators to assess driving safety, 2010.

[6] CHARISSIS, V., PAPANASTASIOU, S., CHAN, W., AND PEYTCHEV, E. Evolution of a full-windshield hud designed for current vanet communication standards. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)* (2013), IEEE, pp. 1637–1643.

[7] CODINGCONNECTED. Codingconnected traci. https://github.com/CodingConnected/CodingConnected.Traci.

[8] CONTRIBUTORS, N. The network simulator - ns-3. https://www.nsnam.org/, 2006.

[9] CONTRIBUTORS, O. Planet dump retrieved from https://planet.osm.org. https://www.openstreetmap.org, 2017.

[10] COX, D. J., BROWN, T., ROSS, V., MONCRIEF, M., SCHMITT, R., GAFFNEY, G., AND REEVE, R. Can youth with autism spectrum disorder use virtual reality driving simulation training to evaluate and improve driving performance? an exploratory study. *Journal of autism and developmental disorders 47*, 8 (2017), 2544–2555.

[11] ESRI. Esri cityengine. https://www.esri.com/en-us/arcgis/products/esri-cityengine/overview, 2008.

[12] FREEMAN, J., AVONS, S. E., MEDDIS, R., PEARSON, D. E., AND IJSSELSTEIJN, W. Using behavioral realism to estimate presence: A study of the utility of postural responses to motion stimuli. *Presence: Teleoperators & Virtual Environments 9*, 2 (2000), 149–164.

[13] GALY, E., PAXION, J., AND BERTHELON, C. Measuring mental workload with the nasa-tlx needs to examine each dimension rather than relying on the global score: an example with driving. *Ergonomics 61*, 4 (2018), 517–527.

[14] GONÇALVES, J., ROSSETTI, R. J., AND OLAVERRI-MONREAL, C. Ic-deep: A serious games based application to assess the ergonomics of in-vehicle information systems. In *2012 15th International IEEE Conference on Intelligent Transportation Systems* (2012), IEEE, pp. 1809–1814.

[15] Gonçalves, J. S., Jacob, J., Rossetti, R. J., Coelho, A., and Rodrigues, R. An integrated framework for mobile-based adas simulation. In *Modeling Mobility with Open Data*. Springer, 2015, pp. 171–186.

[16] Gonçalves, J. S., Rossetti, R. J., Jacob, J., Gonçalves, J., Olaverri-Monreal, C., Coelho, A., and Rodrigues, R. Testing advanced driver assistance systems with a serious-game-based human factors analysis suite. In *2014 IEEE Intelligent Vehicles Symposium Proceedings* (2014), IEEE, pp. 13–18.

[17] Guan, S., Grande, R. E. D., and Boukerche, A. Real-time 3d visualization for distributed simulations of vanets. In *Proceedings of the 2014 IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications* (2014), IEEE Computer Society, pp. 138–146.

[18] IJsselsteijn, W. A., De Ridder, H., Freeman, J., and Avons, S. E. Presence: concept, determinants, and measurement. In *Human vision and electronic imaging V* (2000), vol. 3959, International Society for Optics and Photonics, pp. 520–529.

[19] Insko, B. E. Measuring presence: Subjective, behavioral and physiological methods.

[20] JustInvoke. Randomation vehicle physics 2.0. https://github.com/JustInvoke/Randomation-Vehicle-Physics.

[21] Kaptein, N. A., Theeuwes, J., and Van Der Horst, R. Driving simulator validity: Some considerations. *Transportation research record 1550*, 1 (1996), 30–36.

[22] Kemeny, A. From driving simulation to virtual reality. In *Proceedings of the 2014 Virtual Reality International Conference* (2014), ACM, p. 32.

[23] LANIER, J. *Dawn of the new everything: A journey through virtual reality.* Random House, 2017.

[24] LÊ, H., PHAM, T. L., AND MEIXNER, G. A concept for a virtual reality driving simulation in combination with a real car. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications Adjunct* (2017), ACM, pp. 77–82.

[25] LEE, W.-S., KIM, J.-H., AND CHO, J.-H. A driving simulator as a virtual reality tool. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)* (1998), vol. 1, IEEE, pp. 71–76.

[26] LLC, M. Simulation of urban mobility, 2000.

[27] MAPBOX. Mapbox. https://www.mapbox.com/.

[28] MCEWAN, M., JOHNSON, D., WYETH, P., AND BLACKLER, A. Videogame control device impact on the play experience. In *Proceedings of The 8th Australasian Conference on Interactive Entertainment: Playing the System* (2012), ACM, p. 18.

[29] MCGLOIN, R., FARRAR, K. M., AND KRCMAR, M. The impact of controller naturalness on spatial presence, gamer enjoyment, and perceived realism in a tennis simulation video game. *Presence: Teleoperators and Virtual Environments 20*, 4 (2011), 309–324.

[30] MICHAELER, F., AND OLAVERRI-MONREAL, C. 3d driving simulator with vanet capabilities to assess cooperative systems: 3dsimvanet. In *2017 IEEE Intelligent Vehicles Symposium (IV)* (2017), IEEE, pp. 999–1004.

[31] Milgram, P., and Kishino, F. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems 77*, 12 (1994), 1321–1329.

[32] Narayanasamy, V., Wong, K. W., Fung, C. C., and Rai, S. Distinguishing games and simulation games from simulators. *Computers in Entertainment (CIE) 4*, 2 (2006), 9.

[33] Panerai, F., Droulez, J., Kelada, J., Kemeny, A., Balligand, E., and Favre, B. Speed and safety distance control in truck driving: comparison of simulation and real-world environment. In *Proceedings of the Driving Simulation Conference* (2001), pp. 91–108.

[34] Rigby, S., and Ryan, R. M. *Glued to games: How video games draw us in and hold us spellbound.* AbC-CLIo, 2011.

[35] Ropelato, S., Zünd, F., Magnenat, S., Menozzi, M., and Sumner, R. Adaptive tutoring on a virtual reality driving simulator. *International SERIES on Information Systems and Management in Creative EMedia (CreMedia) 2017*, 2 (2018), 12–17.

[36] Slater, M. A note on presence terminology. *Presence connect 3*, 3 (2003), 1–5.

[37] Sutherland, I. E. Sketchpad a man-machine graphical communication system. *Simulation 2*, 5 (1964), R–3.

[38] Technologies, U. Windridge city asset. https://assetstore.unity.com/packages/3d/environments/roadways/windridge-city-132222.

[39] Technologies, U. Unity 2018.3. https://unity.com/, 2018.

[40] VAN DER HEIDEN, R. M., JANSSEN, C. P., DONKER, S. F., AND MERKX, C. L. Visual in-car warnings: How fast do drivers respond? *Transportation Research Part F: Traffic Psychology and Behaviour* (2018).

[41] VOCES, A. G. Vehicle physics pro. https://vehiclephysics.com/, 2019.

[42] VR, O. Oculus rift. https://www.oculus.com/, 2012.

[43] WACHTEL, J. Brief history of driving simulators. *Tr News*, 179 (1995).

[44] WALCH, M., FROMMEL, J., ROGERS, K., SCHÜSSEL, F., HOCK, P., DOBBELSTEIN, D., AND WEBER, M. Evaluating vr driving simulation from a player experience perspective. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (2017), ACM, pp. 2982–2989.

[45] WATSON, D., CLARK, L. A., AND TELLEGEN, A. Development and validation of brief measures of positive and negative affect: The PANAS scales. *Journal of Personality and Social Psychology 54*, 6 (1988), 1063–1070.

[46] WEGENER, A. Traci. https://sumo.dlr.de/wiki/TraCI, 2008.

[47] WITMER, B. G., JEROME, C. J., AND SINGER, M. J. The factor structure of the presence questionnaire. *Presence: Teleoperators & Virtual Environments 14*, 3 (2005), 298–312.

[48] WITMER, B. G., AND SINGER, M. J. Measuring presence in virtual environments: A presence questionnaire. *Presence 7*, 3 (1998), 225–240.

[49] WOLDEGIORGIS, B. H., LIN, C. J., AND LIANG, W.-Z. Impact of parallax and interpupillary distance on size judgment performances of virtual objects in stereoscopic displays. *Ergonomics 62*, 1 (2019), 76–87.

[50] WYMANN, B. Torcs. https://sourceforge.net/projects/torcs/.

Appendix

User Study Questionnaires

Figure 5.1: Demographic Data Questionnaire

	Not at all	A Little	Moderately	Quite a Bit	Extremely
Interested					
Distressed					
Excited					
Upset					
Strong					
Guilty					
Scared					
Hostile					
Enthusiastic					
Proud					
Irritable					
Alert					
Ashamed					
Inspired					
Nervous					
Determined					
Attentive					
Jittery					
Active					
Afraid					

Figure 5.2: PANAS Questionnaire

Mental Demand

Low | | | | | | | | | | | | | | | High

How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Physical Demand

Low | | | | | | | | | | | | | | | High

How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

Temporal Demand

Low | | | | | | | | | | | | | | | High

How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?

Performance

Successful | | | | | | | | | | | | | | | Unsuccessful

How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?

Effort

Low | | | | | | | | | | | | | | | High

How hard did you have to work to accomplish your level of performance? (mentally and physically)

Frustration

Low | | | | | | | | | | | | | | | High

How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Figure 5.3: NASA TLX Questionnaire

Physical Demand

How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

or

Performance

How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?

Figure 5.4: Example NASA-TLX Verification Question

Figure 5.5: Witmer Presence Questionnaire Item Stems (Modified Version 3.0)