Fall 12-2021

A Testing and Experimenting Environment for Microscopic Traffic Simulation Utilizing Virtual Reality and Augmented Reality

Mahmud Hasan
Old Dominion University

A TESTING AND EXPERIMENTING ENVIRONMENT FOR MICROSCOPIC TRAFFIC SIMULATION UTILIZING VIRTUAL REALITY AND AUGMENTED REALITY

by

Mahmud Hasan
B.S. February 2012, University of Dhaka, Bangladesh
M.S. June 2014, University of Dhaka, Bangladesh

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MODELING AND SIMULATION

OLD DOMINION UNIVERSITY
December 2021

Approved by:

Yuzhong Shen (Director)

Hong Yang (Member)

Duc T. Nguyen (Member)

ABSTRACT

A TESTING AND EXPERIMENTING ENVIRONMENT FOR MICROSCOPIC TRAFFIC SIMULATION UTILIZING VIRTUAL REALITY AND AUGMENTED REALITY

Mahmud Hasan
Old Dominion University, 2021
Director: Dr. Yuzhong Shen

Microscopic traffic simulation (MTS) is the emulation of real-world traffic movements in a virtual environment with various traffic entities. Typically, the movements of the vehicles in MTS follow some predefined algorithms, e.g., car-following models, lane changing models, etc. Moreover, existing MTS models only provide a limited capability of two- and/or three-dimensional displays that often restrict the user's viewpoint to a flat screen. Their downscaled scenes neither provide a realistic representation of the environment nor allow different users to simultaneously experience or interact with the simulation model from different perspectives. These limitations neither allow the traffic engineers to effectively disseminate their ideas to various stakeholders of different backgrounds nor allow the analysts to have realistic data about the vehicle or pedestrian movements. This dissertation intends to alleviate those issues by creating a framework and a prototype for a testing environment where MTS can have inputs from user-controlled vehicles and pedestrians to improve their traffic entity movement algorithms as well as have an immersive M^3 (multi-mode, multi-perspective, multi-user) visualization of the simulation using Virtual Reality (VR) and Augmented Reality (AR) technologies. VR environments are created using highly realistic 3D models and environments. With modern game engines and hardware available on the market, these VR applications can provide a highly realistic and immersive experience for a user. Different experiments performed by real users in this study prove that utilizing VR technology for different traffic related experiments generated much more

favorable results than the traditional displays. Moreover, using AR technologies for pedestrian studies is a novel approach that allows a user to walk in the real world and the simulation world at a one-to-one scale. This capability opens a whole new avenue of user experiment possibilities. On top of that, the in-environment communication chat system will allow researchers to perform different Advanced Driver Assistance System (ADAS) studies without ever needing to leave the simulation environment. Last but not least, the distributed nature of the framework enables users to participate from different geographic locations with their choice of display device (desktop, smartphone, VR, or AR). The prototype developed for this dissertation is readily available on a test webpage, and a user can easily download the prototype application without needing to install anything. The user also can run the remote MTS server and then connect their client application to the server.

This thesis is dedicated to the proposition

that a day without learning is a day wasted.

ACKNOWLEDGMENTS

There are many people who have contributed to the successful completion of this dissertation. A heartfelt thanks to Daniel Perez for his contribution, and support. I extend many, many thanks to my committee members for their patience and hours of guidance on my research and editing of this manuscript. The untiring efforts of my Ph.D. advisor, Dr. Yuzhong Shen, deserve special recognition. He never compromised with quality which has made me a better researcher and in turn a better human being. I would also like to thank Dr. Rick McKenzie (Late) who was a great human being, mentor and a friend to everyone who knew him.

NOMENCLATURE

MTS Microscopic Traffic Simulation

VR Virtual Reality

AR Augmented Reality

MR Mixed Reality

XR Extended Reality

TS Traffic Simulation

TCP Transmission Control Protocol

IP Internet Protocol

UDP User Datagram Protocol

CV Connected Vehicle

AV Autonomous/ Automated Vehicle

CAV Connected and Automated Vehicle

M&S Modeling and Simulation

CPU Central Processing Unit

GPU Graphics Processing Unit

DS Driving Simulator

API Application Programming Interface

GUI Graphical User Interface

M3 Multi-User, Multi-Mode, Multi-Perspective

HMD Head Mounted Display

V2V Vehicle to Vehicle

TABLE OF CONTENTS

Page

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This chapter briefly reviews the history of transportation, the history of Modeling and Simulation (M&S) and uses of M&S in the transportation field, among others. The motivation behind the work of this dissertation is also discussed here. Finally, this chapter describes the proposed work of creating a testing environment and enhancing microscopic traffic simulation using virtual reality, augmented reality, and a user chat system, along with its major contributions.

1.1 HISTORY OF TRANSPORTATION

Transportation is a cornerstone of human civilization with a very long history [1]. The main purpose of transportation was to save mankind time, cost, and effort. Humans domesticated animals around 4000 to 3000 BC to carry their loads [2]. Wheels were invented using wood in an area that is now known as Iraq around 3500 BC [3]. The earliest boats were made by lighting fire on big logs and then digging the burned wood into a canoe. The Egyptians invented the sailboat around 3100 BC. Around 2100 BC they created wooden ships for trading by sea [4]. The Romans were famous for building road networks across their empire so that their army could march from one place to another quickly [5]. They also built merchant ships with sails to carry a huge amount of cargo. After the Roman empire fell, the road transport reverted to the dirt track, but there was significant improvement in water transport in China and Europe [6].

Transportation improved during the 17th century. Stagecoaches were used to travel between cities, but they were still uncomfortable and expensive. The first turnpike road was opened in 1663 and required passengers to pay tolls to use the road. Transportation greatly improved in the 18th

century. The Watt steam engine was invented in 1769 [7]. Many more turnpike roads were opened. Canals were dug to carry goods. These canals contributed to the industrial revolution by making goods transportation cheaper [8]. In the mid-19th century transportation was revolutionized by railway [9]. The first cars were made in 1885 and 1886 [10]. Also, in the 1880s, the bicycle was invented, and it soon became quite popular [11]. Steamships became very popular in that century as well. In the 20th century after the Second World War, cars became much more affordable [12]. Speed limits were introduced on roads, electric traffic lights were installed, and highway acts were formulated, among other transportation regulations and inventions. The trams which used to be horse-drawn became electric. A new form of transportation started in that century, by air [13]. The first passenger jet service began in 1952.

Automobiles are by far the most utilized and useful mode of transportation in everyday life. From transporting people in cars, buses to carrying goods in cargo trucks, this mode of transportation has seen quite a development. The first steam powered vehicle capable of carrying people came in the 18th century [14]. Then in the 19th century more practical vehicles started to come into existence with hand brakes, steering and the multi-speed transmission. The invention of the internal combustion engine and better designs created a plethora of new developments in the automobile field. In the early 20th century electric vehicles became much more popular, but when mass production of gasoline-based vehicles reduced the price of the automobiles and more petroleum infrastructure started to develop, the popularity of electric vehicles started to decline. Almost a century later, due to environmental concerns, people started to think more about electric vehicles (EVs) and recent investments in this sector provided a strong indication that EVs are going to stay here for a long time now [15] until more cost efficient and environmentally friendly alternatives, like hydrogen fueled automobiles, take its place.

1.2 MODELING AND SIMULATION (M&S)

Modeling and Simulation (M&S) is the use of models (e.g., physical, mathematical, or logical representation of a system, entity, phenomenon, or process) as a basis for simulations to develop data utilized for managerial or technical decision making [16]. A model is a similar but simpler version of a system it represents. Depending on the objectives, models of the same system can vary significantly. The model should encompass most of the salient features that are desired to be tested but not be so complicated that it becomes too complex to experiment with. In short, models are the approximations of the real world. Then the models are utilized to run simulations, allowing multiple and different observations of the system being modeled. After one or multiple simulations using the model, analysis takes place, which then is used to verify and validate the research [17]. For example, if a restaurant manager needs to determine the number of cashiers required on a certain day and time, information regarding the restaurant must be researched first, such as number of customers over a period, number of cashiers used, customer waiting time, cashier serve time, peak hour, off peak hour loads, and location of the register. Based on the collected data, a model is created for this scenario with all the necessary variables like cashiers, customers, customer arrival rate, serving time, etc. There may be times when unexpected events may occur like a power outage, a broken register, or a sick cashier. These conditions also need to be considered. The mathematical model uses these data as inputs to run a simulation over a period of time and generates the output. Based on that output, suggestions can be made for the manager about what would be the optimum number of cashiers for different times. As the whole M&S system is based on collecting initial data, each model will be different for different restaurants. One scenario where simulation can be used is when the real system cannot be engaged. The real system may not be engaged because 1) it may not be easily accessible (e.g., satellite), 2) it may be dangerous to engage the system (e.g., nuclear power plant), or 3) it may be unacceptable to engage the system (e.g., data supercenter) [18].

In the First World War, M&S models were used to train pilots in a safer environment [19]. They are not like the sophisticated flight simulators that currently exist. Not many WWI flight simulators exist now. One of the earlier versions can be found at the RAF Museum in London. These earlier simulators were mostly made on top of a wooden platform with metal boxes with instruments to emulate cockpits and had guns placed above the cockpit in such a way that the bullets would not hit the propellers. In the late 1940s and 50s commercially available flight simulators started to become available. With advancements in computer technology, computational modeling and simulation started to become popular. Computational modeling and simulation utilize a mathematical model of the physical system. By altering the parameters of the mathematical model, different simulations are performed to understand different behaviors of the system. Many companies have created computer tools for specific purposes like MATLAB Simulink [20] for modeling, simulating and analyzing dynamic systems, NetLogo [21] for agent-based modeling and simulation, PTV Vissim [22] for microscopic traffic simulation, Vensim [23] for dynamic feedback models and Arena [24] for discrete-event simulation and so on. With faster CPUs and GPUs, virtual environments in the simulation are becoming more realistic. Nowadays, soldiers can delve into combat simulation wearing immersive headsets before engaging real enemies, and pilots train on realistic flight simulators with high-definition visual displays and motion feedback from a moving platform [25]. In the medical field, doctors perform virtual surgeries in the virtual world via haptic devices before performing the surgical procedure. From social science to medicine to road transport to space travel, modeling and simulation are being used in every field.

1.3 M&S IN TRANSPORTATION

Transportation is one of the fields where M&S is widely being used. From simulating real world traffic movements to driver behaviors to signal changes to connected vehicles, M&S is used for almost every facet of transportation nowadays. One area of transportation which heavily uses M&S is known as traffic simulation. Traffic simulation is the simulation of transportation systems using computer hardware, software, and mathematical models. Since most people travel by road, the modern world has seen the necessity of creating complex road networks to accommodate all the travelers [26]. It is not acceptable to create a road and later find out that there are some design flaws. To overcome or eliminate such potential design flaws, traffic engineers use modeling and simulation to model road networks in the simulated world with all the necessary traffic entities like vehicles, pedestrians, road signs, traffic lights, detectors, etc., using computer tools and then simulate them to see if anything goes wrong. Traffic simulation tools are not only used for road transport but also for railway and maritime transportation. The three major types of traffic simulation are microscopic, macroscopic, and mesoscopic traffic simulations.

1.3.1 Microscopic Traffic Simulation

Microscopic traffic simulation (MTS) is the analysis of various traffic entities (e.g., vehicles, pedestrians, etc.) traveling through a virtual road network consisting of many facilities such as highways, bus stops, bike routes, etc. The movements of these entities within the network are often governed by several behavior models (e.g., car-following, lane-changing, etc.) and rules (e.g., signal controls, gap acceptance, etc.) that attempt to mimic real-world conditions. Other than the representation of the physical network, the prominent role of MTS lies in its ability to assess networks as a system in a real-time environment. Significant contributions from the transportation community have been driving the evolution of the underlying models and rules over the years. Many developed MTS models can now be calibrated to simulate what occurs in reality with very high accuracy. Some notable microscopic traffic flow models are Safe-Distance Models [27], Stimulus-Response Models [28], Action Point Models [29], Cellular-Automata Models [30], etc. The earliest car-following model was a safe-distance model which was introduced in 1953 [31]. According to this model, vehicles adjust their speed depending on the safe distance to their leading vehicle. The basic idea behind the model is that the driver plans speed for the following instant (i.e., after a delay τ) such that s/he can safely stop even if the leading vehicle suddenly pressed the brake. The speed attained by a vehicle at a given time instant $(t + \tau)$ is given by

$$\dot{s}_f (t + \tau) = \min \{\dot{s}_{f,\,acc} (t + \tau), \dot{s}_{f,\,dec} (t + \tau)\} \qquad (1)$$

where,

$$\dot{s}_{f,\,acc} (t + \tau) = \dot{s}_f (t) + 2.5\, \ddot{S}_f\, \tau\, (1 - \dot{s}_f (t) / \dot{S}_f)\, \surd(0.025 + (\dot{s}_f (t) / \dot{S}_f)) \qquad (2)$$

$$\dot{s}_{f,\,dec} (t + \tau) = b_f\, \tau + \surd(\, b_f^2\, \tau^2 – b_f\, [2\, (s_l (t) – s_f (t) – (L_l + \Delta S^0)) - \dot{s}_f (t)\tau – (\dot{s}_l^2 (t) /b)])$$

$$(3)$$

where,

f, l = follower and leader, respectively;

s = space traveled by vehicle;

$\dot{s}$ = vehicle's speed;

$\dot{S}_f$, $\ddot{S}_f$ = follower's maximum desired speed and maximum acceleration, respectively;

b_f, b = most severe braking that driver of the following vehicle wishes to undertake and leader's most severe braking capability estimate by the follower;

L = leader's vehicle length; and

ΔS^0 = intervehicle spacing at a stop.

According to the stimulus-response model, vehicle acceleration is a reaction of three types of stimuli: 1) own current speed, 2) distance to the leading vehicle, and 3) relative speed with respect to the leading vehicle. An action-point model assumes that drivers only react if the change is large enough for them to be perceived. In a cellular-automata model, space and time are sometimes discretized. The road is discretized into cells and at each time step each vehicle is moved to zero, one or more cells following a certain algorithm [32]. Some popular MTS tools such as Quadstone Paramics [33], PTV Vissim [22] (Fig. 1), Cube Dynasim [34], Aimsun [35], and SUMO [36] are now either commercially or freely available. These tools allow traffic professionals to study innovative concepts and plans in a flexible, effective, and safe manner.

Fig. 1. An example of microscopic traffic simulation in PTV Vissim.

1.3.2 Macroscopic Traffic Simulation

Macroscopic traffic simulation describes traffic as if it were a continuum flow. In macroscopic traffic flow models, only aggregated variables such as density, average speed, and direction are considered. Macroscopic simulation models are based on the deterministic relationships of the flow, speed, and density of the traffic stream. The simulation in a macroscopic model takes place on a section-by-section basis rather than by tracking individual vehicles. Macroscopic simulation models were originally developed to model traffic in distinct transportation subnetworks, such as freeways, corridors (including freeways and parallel arterials), surface-street grid networks, and rural highways. Macroscopic traffic simulation has two major methodologies: kinematic wave models [37], and higher-order models [38]. Kinematic wave models are also known as LWR models. In LWR, space and time are discretized into cells usually 200 m long and with 0.5 s to several seconds of time steps. Densities are calculated using the previous densities coming in and out of the cells. A higher-order model includes an equation for acceleration and deceleration towards the equilibrium speed. Higher-order models are anisotropic since they say vehicles not only react to their leading vehicle but also to the following vehicle [32]. Some popular macroscopic traffic simulation tools are Cube Voyager [39], AIMSUN, PTV Visum [40] (Fig. 2), etc.

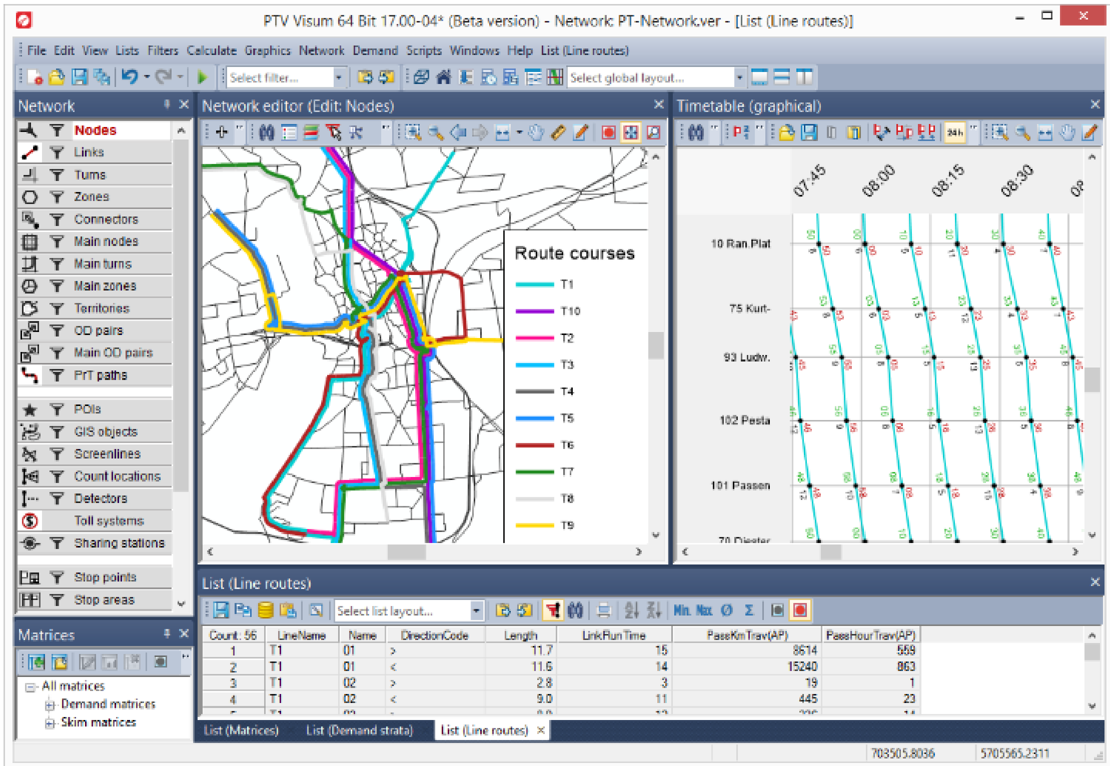


Fig. 2. An example of macroscopic traffic simulation in PTV Visum.

1.3.3 Mesoscopic Traffic Simulation

Mesoscopic models fill the gap between microscopic and macroscopic traffic simulation models. In mesoscopic traffic simulation, the behavior of individual vehicles is described as well as traffic as a continuum flow just like in fluid dynamics. It describes vehicle movements in aggregate terms of probability distribution. However, each vehicle has their own behavior. Mesoscopic models provide less fidelity than microsimulation tools but are superior to the typical planning analysis techniques. A mesoscopic traffic simulation family has several models like gas-kinetic models [41], headway distribution models [42], cluster models [43], etc. In a gas-kinetic model, traffic is described like how gas is flown. Just like the molecules of gas, the individual vehicle movements are not considered in this model; instead, distributions of speed and density are used [32]. Some popular mesoscopic traffic simulation tools are AIMSUN, PTV Vissim (Fig. 3), Mezzo, Cube Avenue [44], etc.

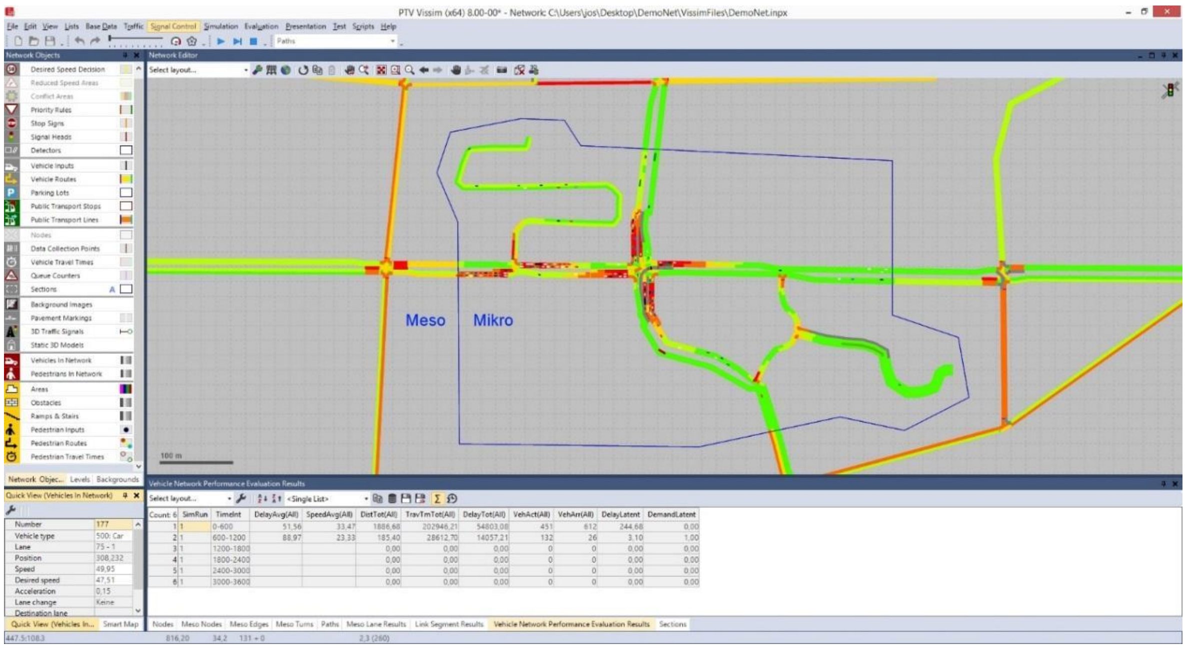


Fig. 3. An example of mesoscopic traffic simulation in PTV Vissim.

Another major application of M&S in transportation is Driving Simulation. Driving Simulation is where real users drive virtual vehicles in the simulated environment. Some virtual vehicles are connected to real world driving simulators with driving wheels, pedals, seats, monitors and sometimes motion platforms, to allow users to have a realistic experience. To make the user experience more realistic, these simulators often have feedback from the steering wheel like vibration. These driving simulators are used for driver training, vehicle research, user studies, and so on. Almost all transportation research institutions have one or two driving simulators set up in their laboratories. Similarly, vehicle manufacturers also use driving simulators to test their products.

1.4 MOTIVATION

The continued evolution of transportation systems has necessitated the extensive use of MTS models worldwide. The level of detail and accuracy required to address stakeholders' questions is often beyond the capabilities of many analytical models. Meanwhile, MTS models are warranted because of their high accuracy and fidelity to actual traffic conditions. Their versatility has been attested in many successful applications ranging from traditional operation problems [45], environmental issues [46], pedestrian evacuation and dynamics [47-50], and safety [51] to the latest connected and autonomous vehicle (CAV) technologies [52, 53]. In all applications, engineers often need to convey meaningful findings to various stakeholders. In practice, much effort has been devoted to statistical analysis of simulation output data. This is crucial in quantitatively comparing different scenarios for decision making [54]. The success of communication heavily relies on the ability of engineers to interpret and explain the stories behind various statistics. However, it is often challenging for stakeholders with different backgrounds (e.g., administrators, planners, public, etc.) to easily comprehend the stories and anticipate issues that may occur in each modeled scenario. Regarding such an issue, visualization of simulated scenarios is increasingly used to provide a common communication medium to facilitate deep understanding that crosses knowledge boundaries between engaged stakeholders.

In fact, the importance of simulation visualization in decision making and public outreach has been evidenced in many projects. According to a recent study [55] that examined the best practices in the use of MTS models, most surveyed transportation agencies (87 percent) used visual animations as a means to present simulation results to stakeholders. More than half of them believed that visualization is more effective at conveying simulation results than conventional charts or maps. Users reached a consensus that the main power of simulation visualization is that it enables users to review the animations and perceive any network view to confirm their observations of traffic conditions [55, 56]. This greatly helps initiate dialogues between different groups of stakeholders.

Most MTS models are now built with graphical user interfaces (GUIs) that provide displays of the simulation models in either 2-dimensional (2D) or 3-dimensional (3D) environments. These display modules, however, often confine users to viewing simulations via traditional 2D interfaces such as computer screens. Even with larger display devices (e.g., projector screens), the rendered simulation models have to be scaled down. The compressed scenes unavoidably lead to visually perceivable distortions. Moreover, although many simulation tools provide possible interactions through a keyboard and/or mouse, the options are often limited to adjusting the single-person viewpoint, zooming levels, etc. This restricts users from freely making real-time adjustments to the simulation scenarios. In addition, the built-in GUIs lack the ability to place users in an immersive 3D environment. Alternatively, animated video clips are often recorded by a camera navigating along predefined paths in the simulation scene. This imposes another problem that only a fixed number of views can be demonstrated. An entire audience can only view the same scene when playing the video. Furthermore, existing traffic visualization always runs on the same computer on which the traffic simulation is executed, severely limiting its capabilities to present the results to audiences at different geographical locations.

It is well documented that MTS can be used to find solutions to various issues related to the wide area of transportation. But, due to the complex and dynamic nature of transportation systems, MTS cannot fully address some important issues like traffic safety and efficiency when it involves human beings. In those cases, it is necessary to receive inputs from real users. MTS is not accurate enough to study those issues because the vehicles in MTS follow some predefined algorithms like a car following model or lane changing model, which do not fully describe the complex interactions between different vehicles and between vehicles and the environment. A developer can modify or override existing algorithms used by MTS through its application programming interface (API). Even so, MTS alone couldn't answer all the questions that arise in traffic and transportation studies. To address some of the issues, driving simulators (DS) have been used to perform various studies like evaluating novice drivers' attention span [57], how driving is affected by sickness [58] or drugs [59], and how texting while driving is a safety hazard [60], etc.

Although DSs make some studies possible, the traffic utilized by DSs is usually simple with only a few vehicles that do not reflect most real-world scenarios. Most DSs are standalone devices with only one driver, thus not allowing study of any driver-driver interactions like yielding to someone in a conflicting left turn, effects of forced lane changes and so on. Networked DSs [61] alleviate the issues in standalone DSs to some extent with multiple-user controlled vehicles. Integrating MTS and DS together provided some solutions to the problems in standalone MTS or DS [62-64]. The integration of the two simulators enhances the ability of both a traffic modeling and driving simulation, providing more types of controlled experiments and improving their results. For the DS, the integration can generate a more realistic traffic environment and present vehicle-vehicle interactions to strengthen the validity of the scene while allowing researchers to test special driving behaviors. For the MTS, the researchers or engineers cannot only calibrate the fundamental traffic flow models in the simulation by comparing the behaviors of the user-driven vehicles and the simulated vehicles but also can create and test new suitable models.

Lately, the continual advancements in computer graphics, visualization, simulation, and computing hardware have made it easier to emulate complex 3D environments in real time. Thus, the potential to directly link MTS to a truly immersive virtual reality (VR) environment and augmented reality (AR) environment is immense. This allows users to interactively operate the simulation while in process and opens promising avenues for exploring complex interactions between users and traffic scenarios being simulated. The rich possibilities of VR and AR are just now starting to be mature enough to innovate a number of transportation applications. It can provide the desired output for a traffic environment in true 3D mode. More importantly, users could “step into” the simulated environment, interact in novel ways and experience it firsthand.

1.5 GOALS AND ACCOMPLISHMENTS

This section discusses the goals of this dissertation and briefly discusses accomplishments. Currently, there are five goals to be discussed as follows.

Goal 1: To develop a framework for enhancing microscopic traffic simulation visualization using distributed network architecture and virtual reality technologies.

Accomplishment: The framework has been created using a client/server architecture where the MTS works as the server and the application module created by leveraging the latest gaming technologies works as the client. The MTS sends its traffic entity data to the client module where those received data are used to recreate the traffic entity trajectories. To prove the feasibility of the framework, a prototype was developed. PTV Vissim and Quadstone Paramics were used as MTS servers and a client application was developed using Unity, the game development platform which supports virtual reality. The virtual environment is a generic urban environment with a detailed road network, traffic lights and signs, high rise buildings, etc. This work has been published in a renowned journal [110].

Goal 2: To create a distributed microscopic traffic simulation with user-driven vehicles with VR capabilities.

Accomplishments: The original framework in Goal 1 allows multiple users to visualize the simulation in an immersive way from different perspectives, but the capability to freely control a vehicle was not present. To achieve this goal, the framework needed to be modified because previously there was only one-way communication from MTS to client module, but this time another communication channel was required, from the client module to the MTS so that the MTS can recognize a user-driven vehicle and update its status on its interface. Here, a driving simulator plugin of PTV Vissim was used to create this capability. This work has been presented as a poster at a prestigious conference and has been published in a journal [114].

Goal 3: To create a framework for developing an augmented reality based user-controlled pedestrian in the simulation.

Accomplishments: This framework leverages AR technologies and uses Microsoft HoloLens to have a user-controlled pedestrian in the simulation. The pedestrian moves at a one-on-one scale with the HoloLens movement. Due to the original M^3 architecture where one server can accept multiple clients, it is possible to connect multiple HoloLens pedestrians and experience the simulation simultaneously. The current implementation does not utilize any form of anchor sharing of the Holograms or objects of the environment, so the simulation environment viewed by each pedestrian will not be properly aligned but the prototype works as a proof of concept. This work has been published in a renowned journal [115].

Goal 4: To create a user communication system for the simulation environment.

Accomplishments: The distributed capability of this simulation means that users from different geographic locations can connect to the same simulation environment. When it happens, it is important that they have a means of communicating with each other and coordinating their efforts. A user chat system has been created using the Photon Chat to enable user-user communication.

Goal 5: To make the simulation accessible to the general public via the Internet and a browser.

Accomplishments: A XAMPP server has been used to test run and make the simulation prototype accessible via the INTERNET. Because it was only for testing, the server is not running all the time, but it is possible to run the server and download an app for the simulator via the INTERNET. Due to MTS dongle restrictions and budget constraints, currently the whole simulation server is hosted on a laptop computer.

1.6 STRUCTURE OF THE DISSERTATION

The dissertation is structured into six different chapters. Chapter 1 introduces the dissertation, explains the motivation behind it and discusses different goals and accomplishments. Chapter 2 presents in-depth discussion of relevant works and their pertinent shortcomings, if any. The first contribution of the dissertation is illustrated in Chapter 3 where a basic framework was created for an immersive multi-user, multi-modal and multi-perspective microscopic traffic simulation visualization. Chapter 4 describes the addition of the user-controlled vehicles into the framework along with improvements and experiments performed on it. Chapter 5 explains the addition of the user-controlled pedestrians into the framework in detail. Chapter 6 discusses the user chat system and the testing of running the simulation from an online server. Finally, chapter 7 presents the conclusion of the dissertation along with some limitations and possible improvements.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews previous relevant works, including visualization of microscopic traffic simulation and user interactions in the form of a driver or pedestrian.

2.1 VISUALIZATION OF MTS

The early generations of MTS models often display traffic entities in 2D computer graphics. For example, the NETSIM simulation model projects network components, vehicle movement, and operational measures in 2D graphics [65]. Likewise, TSIS/CORSIM is frequently used to simulate traffic operations of arterials and freeways [66]. Its output processor named TRAFVU permits users to animate simulation results in 2D [67]. A zoom command can be used to locate and exhibit individual facilities (e.g., intersections and interchanges). Synchro/SimTraffic performs 2D simulation like CORSIM for roadways and intersections. These simulation models typically require building a schematic representation of the roadways with a computer-aided drafting of the roadway network as the background. In 2D, vehicles are often modeled simply as flat rectangles [66]. Despite wide use in the design process, these models cannot create convincing visualizations to sufficiently convey to a layperson and therefore have limited success for public participation and dissemination.

In light of the limitations in 2D graphics environments, some studies have made efforts to enhance the display of MTS models by 3D visualizations. For example, Seneviratne [68] used OpenGL API to perform real-time 3D rendering of the vehicles in developing an intermodal terminal simulation model (ITSIM). Ayres and Mehmood [69] utilized the Java Monkey Engine to interact with OpenGL for 3D representation in developing an MTS platform. In fact, many off-the-shelf simulation models such as Vissim, Aimsun, and Paramics have presently incorporated a 3D viewer to visualize simulation [70, 71]. In addition, their 3D visualizations often can be enhanced by add-in programs. For instance, Nomden et al. [72] improved the visualization of Vissim by importing additional 3D models developed in 3ds Max for the City of Newcastle. With a more sophisticated configuration, Sun et al. [73] developed a control program using Vissim API to simulate pedestrian crossing behavior in a virtual environment. The virtual environment was assembled using three monitors aligned in a semicircle. Each monitor only shows part of the 3D scenes of three synchronized MTS models running on different computers. The control program allows participants to use a keyboard to control an avatar crossing a virtual crosswalk displayed on the middle screen. Despite the interaction via the avatar, these configurations limit participants to conducting experiments in a miniature virtual scene with inevitable visual distortions.

At present, few studies have attempted to leverage external VR toolkits and devices to visualize MTS models in more advanced virtual reality. As an example, Zhou et al. [74] extended the CORSIM simulation by implementing its 3D visualization with the EON Studio, a PC-based VR software. Their approach consists of two key components: (a) creation of 3D models (e.g., roadways and vehicles); and (b) specification of individual vehicle motion in 3D forms. Despite the reconstruction of the traffic flow in 3D, the study was limited in its ability to create realistic paths of vehicles on curves, smooth motion, and accurate representation of vehicle sizes for collision avoidance. Many important elements such as traffic signals and their control were not modeled in the visualization. Meanwhile, the low rendering speed prevented it from achieving smooth animation. Consequently, only Audio Video Interleaved (AVI) movies were generated in the final visualization. This, in turn, disallowed any interactivity the VR package offered. In [75], a visualization environment was created by exporting simulation results of Vissim to VR4MAX, a 3ds Max plug-in. Similarly, another study [76] discussed the use of Vissim and VR in railway station upgrade projects in South Africa. Pedestrian simulation models were developed in Vissim and their pedestrian protocol data was processed by a 3D VR Processor linked with the objects created and designated in 3ds Max. Linking the simulated output with 3D objects allows mirroring pedestrian behavior modeled in Vissim. However, the visualization of pedestrians in a virtual environment seems to be off-line. In other words, it was not possible to simultaneously run the MTS model and display the objects in parallel. In addition, users cannot become immersed in the constructed virtual scenes. Likewise, Hu [77] proposed the integration of pedestrian simulation in AnyLogic and VR to examine pedestrian flow and facility layout at subway stations. The proposed procedure used the spatial positions of simulated agents saved in an XML file as the input. Then it applied the micro-simulation plug-in (OpenMicroSim) of the UC-win/Road software to incorporate the XML files and created 3D visualization of the pedestrian movement. Essentially, the plug-in only served as a player to exhibit the imported simulation results. Despite the claimed photo-realistic 3D environment, the visual representation of pedestrian motion was not realistic because all the pedestrians must move along predefined navigation paths. Another issue is the efficacy of the developed procedure. Other than two screenshots of the simulation, there was no clue how many agents can be concurrently handled. Information about the rendering speed was also not reported. Thus, it was not clear whether the proposed procedure can smoothly visualize the crowds at the stations.

To sum up, a major issue of previous MTS 3D visualizations was the insufficient realism caused by the low fidelity of 3D environments, object movements (e.g., vehicles, pedestrians), or user interactions. In addition, existing MTS visualization tools only allow a single point of view at any time, no matter how many users are viewing the simulation. This poses a serious problem as many microscopic traffic issues cannot be easily discerned from a single viewpoint. Allowing multiple users to view the same traffic scenario from varied perspectives facilitates traffic problem solving, decision making, and public dissemination. Another drawback of existing MTS visualization is that it always runs on the same computer as the MTS itself does, which severely restricts the number of users who can view the MTS visualization at the same time. Although MTS visualizations are commonly recorded as videos, the lack of interactivity in recorded videos reduces the convincingness of the MTS. Alternatively, in the following section, this paper proposes implementable solutions to address these problems.

2.2 USER-CONTROLLED VEHICLES IN THE SIMULATION

There has been a lot of work where MTS was integrated with some other simulator or algorithm to enhance its capability. Vladisavljevic et al. [78] discussed the importance of integrating MTS and DS with a case study. However, because of the limitations in data transfer delays and import/export capabilities of the simulation programs, the authors only established a one-way communication. Punzo and Ciuffo discussed the issues of integrating driving and traffic simulation along with some solutions [63]. Due to the difficulties of integration, they only tested offline integration. Jin and Lam created an integrated driving-traffic simulation to study travel behavior of the drivers where the driving simulator display was traditional monitors with no immersion [79]. Lindorfer et al. created an enhanced driver behavior model which is different than the typical follow-the-leader model to overcome the shortcomings of the MTS driver model [80]. Olstam et al. used MTS to solve the issues of surrounding traffic in their DS by integrating them together [81]. The vehicles close to the driving vehicle was generated via a microscopic simulation model where the vehicle outside the viewpoint was created by a less time-consuming mesoscopic model. However, how the surrounding vehicles recognize the driving simulator vehicle was not discussed in the paper. Such interaction might not exist in this study.

Sun et al. constructed a multi-user driving simulator platform to evaluate the effectiveness of eco-driving under the speed guidance strategy [82]. This study didn't consider complex traffic flows. Sun et al. created a multi-DS and integrated with MTS to allow complex traffic flow into the simulation [83]. However, the driving simulators in their project were locally connected which restricts the user from accessing the simulation from any place in the world. Chun et al. exported traffic flow data from Vissim into 3DS MAX where they already had a detailed 3D model of the city and they used it to allow city council members to decide on a design choice [75]. The whole connection was offline and not in real time. Lorentzen et al. used a software tool called UC-win/Road to create a 3D virtual environment and allowed the user to see the traffic design and flow from a driver's perspective [84]. In all the works discussed above, the authors used the term virtual reality to represent the virtual 3D environment. None of the work used head mounted displays (HMD) to provide users an immersive and realistic experience.

More recent work used HMD VR devices or discussed the potential use of HMDs in traffic and driving simulations. Behzadan and Kamat investigated the application of 3D augmented reality (AR) for animating traffic simulation models in real time to support the planning and design of road construction projects [85]. A virtual reality HMD based driving simulator was created by Taheri et al. in [86]. The purpose of the study was to assess driver behavior in a more immersive and safer environment and measure their performance. A serious drawback of this study was that only one user's data were presented. Ropelato et al. created an HMD VR based driving simulator with adaptive tutoring [87]. The users drive in an immersive environment with a car wheel and motion platform, and an artificial intelligence-based system tracks user driving and provides suggestions and guidance to improve their driving. Other cars in the traffic were artificial intelligence-based cars with basic instructions such as stop at red light, give right of way, do not cross speed limit, etc. Without using MTS or real users in the traffic environment, they could not produce complex traffic scenarios. The potential use cases of VR, AR and Mixed Reality (MR) for traffic safety education such as ensuring parking security, efficient space utilization, reducing traffic accidents were discussed in [88]. Blissing et al. used Oculus Rift to study driver behavior in VR and MR [89]. The Oculus Rift had two additional cameras to capture the outside video and show it to the user as an MR experience. This comparative study found that a VR or MR driving simulator can be a good alternative to a regular driving simulator. Deb et al. discussed the efficacy of VR for pedestrian safety research [90]. The authors used HTC Vive and Unity game engine to create a pedestrian simulator. The crosswalk in their simulator has a length of about 5.5 m. The HTC Vive lighthouse sensors can be apart from each other at a maximum distance of approximately 4.98 m. It was not clear how the authors achieved a walkable crosswalk bigger than the maximum lighthouse distance. Sprotillo et al. created a semi-autonomous driving simulator with VR and 6 degrees of freedom controllers to study different interactive interfaces for a vehicle [91]. The purpose of this VR simulator was to train people to use different user interfaces regarding vehicular technologies. The emphasis of this study was the interaction between the driver and the vehicle's driving cabin and interactions with other vehicles were largely ignored. The potential of VR and procedural 3D modeling in transport planning was discussed in [92] where Erath et al. used ESRI CityEngine – a 3D modeling software tool, PTV Vissim for traffic micro-simulation and Unity3D to create VR visualization.

2.3 PEDESTRIAN IN THE SIMULATION

Many traffic simulation packages offer modules for pedestrian simulation in which crowds of artificial pedestrian agents are simulated in their virtual environment. While these simulations of artificial pedestrians are useful in traffic simulations, they lack the realism of pedestrian behavior. Simulating a person is a significantly challenging task and the realism of artificial pedestrian simulators is not enough in areas such as testing of autonomous vehicles, in which real behavioral responses are needed to assess and improve their system. Another mainstream example of pedestrian simulation is to use human-controlled pedestrian simulators. These pedestrian simulators are popular in the research community for diverse applications such as determining the effects of text messaging on pedestrians [93-95], improving child pedestrian safety [94, 96-98], cognitive monitoring and evaluation [99], testing of autonomous vehicles [100, 101] and improving Stated Preference (SP) experiments [102, 103]. Different combinations of virtual reality devices are used in each work. The following paragraphs offer a thorough examination of existing practices.

The simplest approach for VR pedestrian simulators consists of using desktop computers in which users typically move downscaled avatars in a 3D virtual world. This approach is frequently used in studies focused on working with child pedestrians because it is an inexpensive method that can be easily installed in most schools and child centers. For instance, Thomson et al. [97] developed a VR pedestrian simulator that was displayed on a single desktop monitor. In this study, the users observed the actions in a third-person point of view from an elevated semi-aerial position so that the intersections could be fully displayed on the screen. The pedestrians were able to cross the street with the press of a button. While this approach can be useful to understand where to cross a road, it does not allow the users to feel like they are actually crossing the street.

Alternatively, some approaches developed a first-person point of view and more immersive pedestrian simulator by using multiple screens that surround the users, offering a peripheral view of the scene. For example, McComas et al. [96] developed a simulator for teaching child pedestrian safety in which three 21” screens displayed a virtual environment consisting of different types of intersections. Additionally, they utilized a head tracking device to determine whether the children looked left-right-left before crossing. In their study, they did not mention how the children moved in the virtual environment, so it is assumed that they used a regular keyboard as the main input method. Using traditional input methods such as the keyboard or the mouse might lack realism and make the users feel like they are not actually moving in the simulated environment. To fix this, Schewbel et al. [98] proposed a three-screen desktop pedestrian simulator in which the users were on top of a plywood structure. Participants observed the traffic in the three screens from a first-person perspective. When they considered it safe to cross the intersection, they stepped down from the structure, which triggered a pressure plate connected to the virtual simulator. Once that signal was triggered, the view changed to a third-person perspective so that the participants could observe their avatar crossing the street. This study offered a more realistic input method to cross an intersection. However, users can only perform one action and they did not have the freedom of moving freely through the virtual environment.

Multi-screen pedestrian simulators offer the users a peripheral first-person view of the virtual environment, but they do not offer an immersive experience where the users may feel like they are in a real traffic scenario. To solve this, a few studies developed VR pedestrian simulators using a Cave Automatic Virtual Environment (CAVE) [104]. The CAVE consists of a set of three big screens which act as walls where images are projected, giving the users the feeling of being surrounded by a virtual world. The most significant limitation of this virtual reality device for pedestrian simulator is that it does not allow the users to move beyond the walls. The user may move in the limited area between the screens, but this space is typically not big enough to encompass a traffic scenario. For instance, Rahimian et al. [93] developed a VR pedestrian simulator using a CAVE in which the pedestrians could move in an area of 4.33x3.06 meters. While this area was big enough to conduct their study, the users were limited to moving in a single crossing inside of a significantly limited space. To solve this issue Banducci et al. [95] used a manual treadmill that was linked to the virtual environment so that when the users walked on it, the virtual world would move accordingly. While this upgrade allowed users to cross the street in a simulated environment, the pedestrians had the limitation of only moving in a single direction. A slightly different approach was followed by Dommes et al. [105], where they built a street-crossing simulator using a ten-screen image projection system that allowed the participants to walk in a two-lane intersection. In this case, users could freely move in an area that was 5.7 meters wide and 7 meters long. However, the space was still limited to a single intersection and many scenarios such as simulating a crosswalk spanning a two-way four-lane street are not feasible.

Lately, head-mounted display systems are a good VR alternative to the devices mentioned above. They provide a 360° view of the virtual environment that offers a highly immersive experience of the simulation. Additionally, it has been demonstrated that HMD devices provide more realistic VR experiences due to the capability of making users forget about their surroundings in the real environment [106]. Another advantage with respect to CAVE devices is their low cost and portability. There exist multiple studies that used this kind of devices to develop a pedestrian simulator. An early attempt to use HMD devices in pedestrian simulators is an investigation of road crossing realized in 2002 by Simpson et al. [107], in which they used an HMD with a resolution of 640x480 pixels that produced synoptic images with a horizontal and vertical field of view of 48° and 60° respectively. Although their HMD device had immersion problems due to the early state of the technology (low resolution, synoptic images, small field of view, etc.), they determined that using this type of device was a promising method for pedestrian simulators. The latest advances in head-mounted displays have produced high-resolution devices with stereoscopic images and greater fields of view that produce highly immersive VR environments. In an attempt to compare child pedestrian behavior with their parents' expectations, Morrongiello and Corbett [108] developed a pedestrian simulator using a Virtual Research systems HMD display with stereoscopic images and a resolution of 1280x1024 pixels that allowed the participants to move in a room of 8 m×5 m. The virtual scene consisted of a two-lane street with sidewalks. The same team used this device in a different study where they aimed to understand how injury risk arises [109]. Shuchisnigdha et al. [90] developed a pedestrian simulator using the HTC Vive and a virtual environment developed with Unity. Their experiment simulated a single crosswalk and users could freely move in an area of 4 by 7 meters approximately. Other studies have come up with a similar setup using the Oculus Rift headset [99, 100, 102], which allows the users to freely move in an area of maximally about 4 meters by 4 meters. While these devices offer a highly realistic environment, the boundaries of the scenario are still relatively small, which limit researchers to do their studies at a small intersection.

The development of a virtual environment that encompasses a significantly big area (e.g., a corridor with multiple intersections or a city) for pedestrian simulation has not been fully addressed. Hartmann et al. [100, 101] have mentioned in their studies the need of simulating a whole city with real pedestrians using HMD devices for the testing of autonomous vehicles. In [101], they discussed the possibility of using an augmented reality (AR) display, the Microsoft HoloLens, to simulate pedestrian behavior; and they provide a simple prototype in which the users wearing the headset can visualize the hologram of a car in the real world. To the best of our knowledge, there does not exist a virtual/augmented reality framework that can enable simulation with both user-controlled vehicles and pedestrians coexisting with other artificial vehicles in a large area.

To summarize the reviewed literature, Table 1 shows a comparison of the most relevant pedestrian simulators. The studies are ordered by the level of immersion (from low to high). The table shows some useful information for the development of our study.

Table 1. Comparison of Existing Pedestrian Simulators

Scope	Technology	Level of Immersion	Input Methods	Space Boundaries	Participants
Training child pedestrians	1-screen PC	Low	Designated Button	Users do not move	129 children of ages 7, 9 and 11
Training child pedestrians	3-screen PC	Low	Not mentioned	Users do not move	95 children from grades 4-6
Training child pedestrians	3-screen PC, Plywood structure	Low	Plywood structure	On/off the structure	102 children and 74 adults
Studying the impact of texting	CAVE	Medium	User's movement	4.33 m×3.06 m	48 undergraduate students
Studying the impact of texting	CAVE, Manual treadmill	Medium	Manual Treadmill	No limit walking forward, but unable to move sideways	37 university students
Studying age-related difficulties in crossing a street	10-screen image projector system	Medium	User's movement	5.7 m×7 m	58 healthy aged participants
Studying road crossing behavior	Early VR HMD	Medium-high	User's movement	6 m (forward)	24 participants between 5

					and 30 years old
Comparing child pedestrian behaviors with their parents' expectations	Virtual Research systems HMD	High	User's movement	8 m×5 m	139 children between 7 and 12 years old
Examining how risk of injury arises in child pedestrians	Virtual Research systems HMD	High	User's movement	8 m ×5 m	95 children between 7 and 11 years old
Pedestrian safety research	HTC Vive, Unity	High	User's movement	4 m ×7 m	26 university students
Cognitive monitoring and evaluation	Oculus Rift DK2, Unity, Leap Motion, Myo	High	Robotic arm (Myo), Hands (Leap Motion)	Users move according to arm input	No participants
Evaluation of VR State Preference experiments	Oculus Rift, Unity	High	User's movement	4.5 m ×2 m	42 university students

It is worth noting that most recent simulators that use HMDs as their VR technology develop their virtual environment with Unity. Besides, there is no existing pedestrian simulator without the limits of space constraints. To the best of the author's knowledge, the present study is the first to offer an AR-PED pedestrian simulator in which the users can freely move without any preset boundaries.

CHAPTER 3

BASIC FRAMEWORK DEVELOPMENT WITH A PROTOTYPE

The goal of this research study is to create a testing environment with enhanced microscopic traffic simulation by empowering it with the latest advances in computer hardware, software, and networking. To this end, this dissertation first developed an overarching hardware and software framework as the foundation for all the research tasks. The very first application of this framework was to alleviate the visualization limitations of regular MTS, discussed in section 1.4, by leveraging a game development platform and TCP/IP networking. After the successful connection between the MTS and the client application developed by a game development platform, new features were added to the visualization module. After addressing the visualization limitations of MTS this study proposes a novel framework for interactive visualization of microscopic traffic simulations. The proposed framework offers a truly multi-user, multi-perspective, and multi-mode (M^3) visualization architecture for MTS. M1: "Multi-user" means that different model users (e.g., engineers, analysts) that intend to explore the simulation models can access the simulation from geographically distributed locations. M2: "Multi-perspective" represents that these users can view the models in different roles (e.g., pedestrians and drivers). M3: "Multi-mode" denotes that the users can use different types of devices to view the simulation model like VR HMDs, desktop, etc. The highly interactive and immersive M^3 MTS visualization provides a powerful communication tool to important stakeholders of traffic simulation and facilitates decision making and public dissemination. The following sections detail the system design and key components of the proposed framework and discuss the prototype developed using the framework. The research work presented in this chapter has already been published in a prestigious journal as a journal publication [110].

3.1 SYSTEM ARCHITECTURE

Different from existing MTS visualizations that run on the same computer as the MTS itself, the proposed M^3 visualization framework employs a distributed architecture based on the client-server model where the MTS is the server and various visualizations are the clients. The key components of the proposed framework are illustrated in Fig. 1. The MTS server hosts the back-end microscopic traffic simulation engine, which can be any commercial, open source, or custom-made MTS platform. The clients are various computing devices that display the 3D visualizations of MTS, which can be traditional personal computers, smartphones, virtual reality devices, or augmented reality (AR) devices. Both the MTS server and clients are connected to the Internet and communicate over it. In the most common configuration, the server and clients reside on different computing devices. Although they can run on the same computer, the server and the clients still exchange information via network communications. Multiple clients can be connected to the server simultaneously, presenting the same or different views of the same MTS. The server executes simulation as a service (SaaS) and at each simulation time step, sends the vehicle information (ID, position, speed, and bearing) and other traffic information such as traffic signals (IDs, location, and signal states) to the clients.

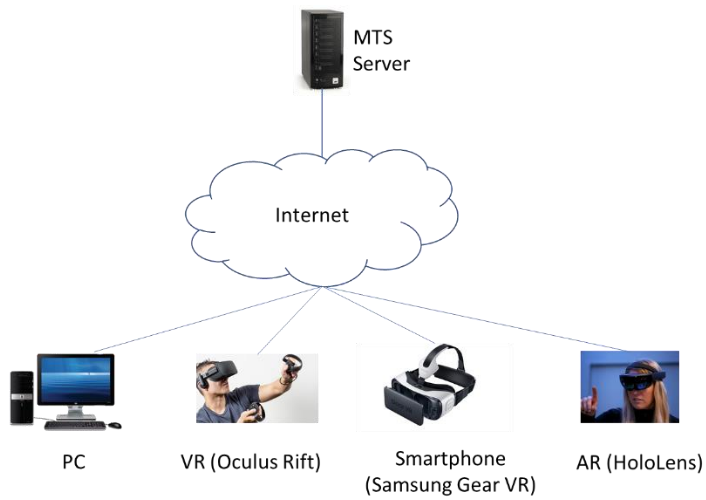


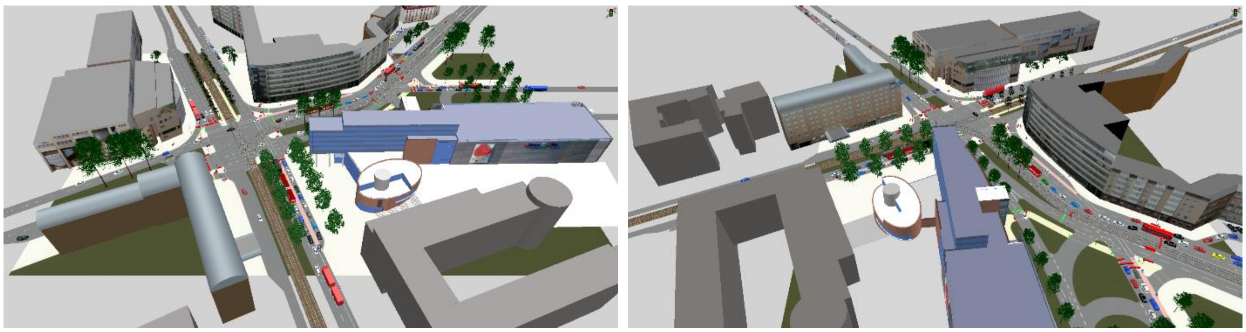
Fig. 4. M^3 Visualization system architecture.

The clients present a realistic rendering of the 3D environment that contains the road network used by the MTS server and a graphical user interface (GUI) for users to interact with the visualization. The GUI allows users to freely change the viewpoint of the traffic simulation, e.g., Bird's-eye, Pedestrian, Driver, and Car Follower view. Different views facilitate comprehensive understandings of the traffic simulation from multiple perspectives. The Bird's-eye view provides a holistic observation of the overall traffic flow. The Pedestrian view is especially important for pedestrian safety study at various locations such as intersections. The Driver view perceives the traffic simulation as a driver inside a moving vehicle and offers better appreciation of the traffic condition that is otherwise impossible or difficult to comprehend, such as weather condition (e.g., raining), road condition (e.g., poorly maintained roadways), and lighting condition (e.g., sun glare). The Car Follower view allows users to observe the behavior of a vehicle from a third-person view behind the vehicle. Depending on the view selected, users can have further interaction options. For example, users can move as a pedestrian when the Pedestrian view is selected and then the view will be automatically updated based on the users' position and orientation. The proposed framework supports multiple display and input devices ranging from traditional desktop displays to smartphones, VR devices, and AR devices, as shown in Fig. 4. The names inside the parentheses are examples of computing device categories, e.g., Oculus Rift is one example of VR devices. A user can choose a display or input device that is most convenient or effective for her/him. M^3 visualization of MTS provides developers and users an unprecedented capability to visualize traffic simulations and to promote or disseminate simulation findings.

3.2 MTS SERVER

MTS models will be deployed as the server in the proposed framework. The server is a core component of the framework as it provides very high fidelity in both the representation of traffic networks, and dynamic interactions between different road-users and the traffic environment. Even complex traffic conditions can be modeled in an unprecedented level of detail providing realistic traffic models.

From a visualization perspective, most of the existing MTS models already have built-in 2D and/or 3D display functions, e.g., Vissim, Paramics, and Aimsun. Despite the differences in visualization procedures, fundamentally these MTS models need to import external 3D objects into the simulation network. For example, Fig. 5 shows the typical visualizations for a built-in demo in Vissim. For a non-professional user, the 3D representation helps understand the complex scenario. However, compared with professional 3D graphics tools, the manipulation of the imported 3D objects and their coordination with the entire simulation environment are very restricted. The integration of the 3D objects with the road network in Fig. 5 is still rather primitive, e.g., only basic rendering is utilized without realistic visual effects, such as advanced lighting, specular reflection, and shadows. In addition, existing MTS models restrict users to see a single view at a given time period on the same computer. Although few MTS models (e.g., Vissim) support driver view and pedestrian view, their sights are always fixed in the forward direction. Users cannot freely observe surroundings like a real driver or pedestrian.



(a) Viewpoint A (b) Viewpoint B

Fig. 5. Typical 3D visualizations of simulation models in Vissim.

To make the proposed framework more adaptable, the selection of MTS models should not be limited to a specific simulation package. Promisingly, our proposed visualization framework does not need to inherit any built-in visualization functions from existing MTS models. Thus, this provides users more flexibility to select a server from a variety of off-the-shelf simulators (e.g., PTV Vissim, Quadstone Paramics, Aimsun, SUMO, CORSIM, etc.). The minimum requirements for an MTS model to be able to function as a server in the proposed framework include: (a) the capability of generating vehicle trajectory profiles; and (b) an interface to access the traffic control information. The vehicle trajectory and coordinates will be used to recreate vehicle movements in the client application. The simulation model is not required to have very high resolutions (small time steps, i.e., 0.1s) as additional interpolation techniques can be applied to smooth vehicle movement if a small time step is not allowed in the model. Examples of interpolation could be Linear Interpolation for straight line movement and Spherical Linear Interpolation for turning.

3.3 INTERACTIVE M^3 VISUALIZATION FOR MTS

MTS models simulate traffic scenarios at the individual vehicle level. Realistic rendering of the environment and entities involved in the MTS is critical to effectively convey the findings to a wide range of stakeholders. Existing MTS visualizations do not fully and accurately depict the details of traffic scenarios, e.g., no existing MTS visualization provides a first-person view of the traffic situation as a driver. Some traffic situations can only be appreciated from a first-person perspective. For example, the strong sun glare when driving eastward in the morning is a safety hazard that can only be perceived from a first-person view, but not from a top view or Bird's-eye view. Other driving conditions such as sight distance and blind spots have similar complications. Visualization of MTS should take advantage of the latest development in computer graphics hardware and software, such as graphics processing units (GPUs) and various shaders and generate a virtual environment of the road network that is as realistic as possible, e.g., with shadows and reflection, in order to present the details of MTS and communicate findings.

Although MTS visualizations can be implemented using traditional display and user interaction technologies, e.g., LED monitors, and keyboard/mouse inputs, the newest development of VR and AR in the last few years provides more immersive and natural user experiences that are especially useful for MTS visualization. Several market-leading VR and AR devices are shown in Fig. 6. Oculus Rift, shown in Fig. 6(a) and developed by Oculus, was the first consumer VR device that generated broad market impact. Facebook acquired Oculus for $2 billion in 2014, starting the trend of latest investments in VR. Oculus Rift incorporates rotational and positional tracking. However, the positional tracking offered by Oculus is limited in terms of space. HTC Vive, shown in Fig. 6(b) and manufactured by HTC, tracks a user's position and orientation with a tracking space of 4.6 m × 4.6 m. Base stations (named "Lighthouse" and the two boxes in Fig. 6(b)) are used to track user movement with sub-millimeter precision. Both Oculus Rift and HTC Vive are driven by personal computers that meet specific minimum hardware requirements. The Gear VR, shown in Fig. 6(c), was developed by Samsung. It is different from Oculus Rift and HTC Vive in that a Samsung smartphone is used as the headset's display and processor, while the Gear VR controls the field of view and uses an inertial measurement unit (IMU) for rotational tracking. This device does not include positional tracking but is more accessible due to its low price

(a)

(b)

(c)

(d)

Fig. 6. Various VR and AR devices: (a) Oculus Rift, (b) HTC Vive, (c) Samsung Gear VR; and (d) Microsoft HoloLens.

compared with the rest of the devices. Microsoft HoloLens, shown in Fig. 6(d), is an AR device that produces an environment in which the real and virtual world are merged. This device allows the user to project 3D models in the real world to visualize and edit them. User interactions are performed through the users´ gestures and voice. Although this device does not offer an immersive view of the scene, it produces a hologram that can be positioned and scaled in the real world.

VR and AR devices offer a more immersive and natural user experience that facilitates better understanding of MTS. For example, devices with positional tracking such as HTC Vive and Microsoft HoloLens can track user movements without any manual inputs and, therefore, it is especially efficient for automatically updating Pedestrian View following user movements. VR and AR devices, together with the traditional PC displays, offer users multi-mode visualization of MTS. A user can select a mode that is most convenient for her/him and observe the MTS from different perspectives. Allowing multiple users at different physical locations to view the same MTS simultaneously via Internet connections greatly broadens the impact of MTS and facilitates discussion, collaboration, and decision making.

3.4 PROTOTYPE DEVELOPMENT AND DEMO

A prototype has been developed following the proposed M^3 framework. It was built upon a hypothetical city model covering an area of 2.0 miles $\times$ 2.0 miles, with approximately 1,000 buildings, road networks, and other facilities (e.g., bus stops, trees, signs, etc.) typically present in an actual city. It should be emphasized that users can freely replace it with any city models of their choice. The following sections provide a detailed description of the prototype development and demonstration. For the prototype, the MTS server currently runs on a laptop with the following configuration: (a) Intel® Core™ i7-7820HK CPU @ 2.90GHz; (b) 32.0 GB RAM; and (c)

Windows 10 Professional 64-bit Operating System. The modeled simulation network serves approximately 2,400 vehicles per hour. The simulation time step was set as 1/6 second. Users can adjust the demand and time step according to their own needs. The hypothetical city model can be replaced with any other city models.

3.4.1 Selection of MTS Server

In this prototype, Quadstone Paramics (Parallel Microscopic Traffic Simulator) and PTV Vissim has been used as the MTS server. Paramics is a suite of high-performance MTS tools originated from the Edinburgh Parallel Computing Center in Scotland [110]. Specifically, its "Modeller" tool is used in this study to generate the road network and respective entities. Both Paramics and Vissim have built-in car-following models and lane-changing models that imitate the psycho-physical feedback and gap acceptance behavior of various vehicles stochastically. Apart from the fundamental modeling and analysis features, the most important attribute for us to consider Paramics and Vissim in this study is their customizable Application Programming Interface (API). Since the default traffic models and functions may not always satisfy needs, their API allows modelers to access and develop customized components. More importantly, the API allows the simulation model to interface with many external programs and toolkits easily. This is critical for the present study because the simulation model, as a server, needs to communicate and exchange data with various visualization platforms in real time. The following figure shows the functional block diagram of the prototype developed.

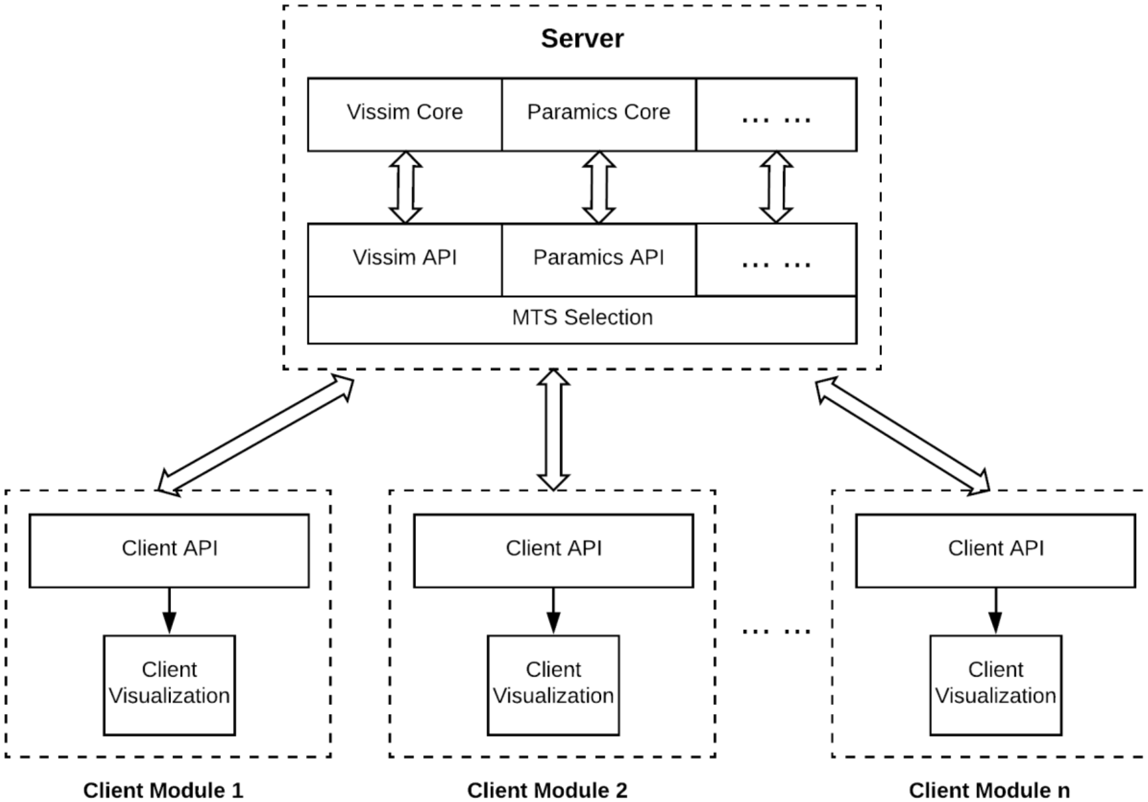


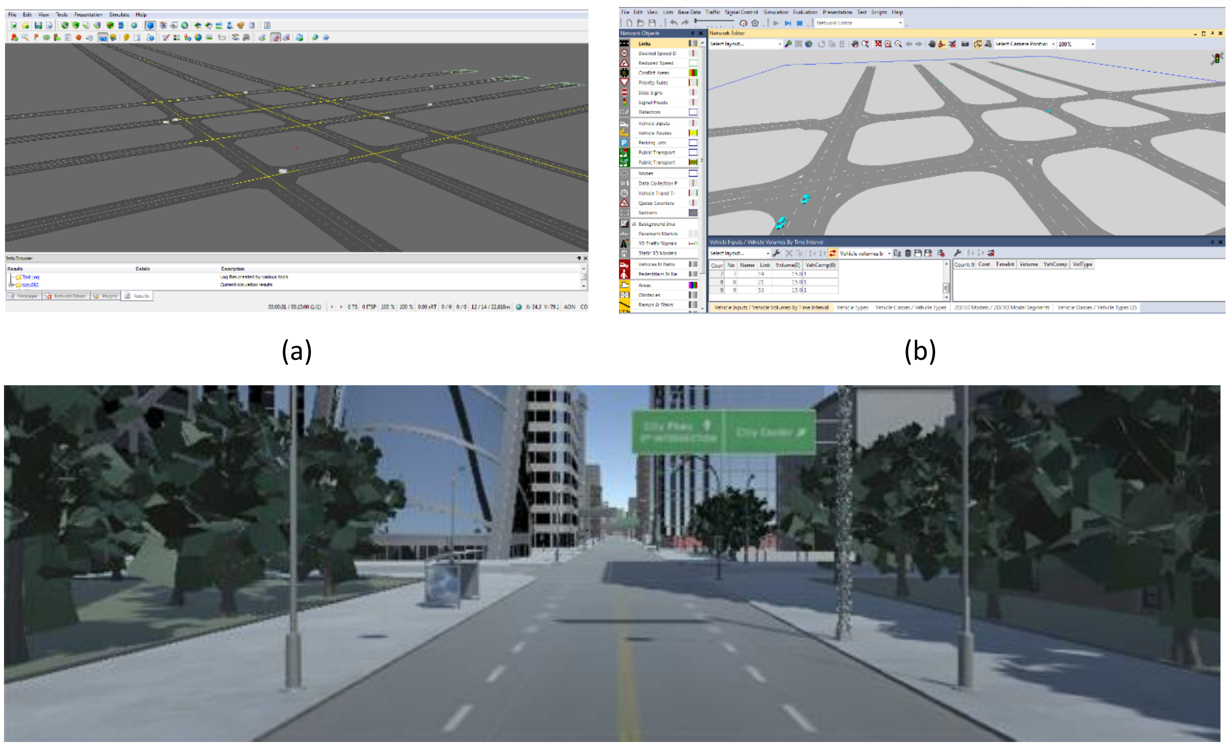
Fig. 7. Functional Diagram of the Basic Framework.

3.4.2 Virtual Environment Development

To create a virtual environment (VE) on the clients' end, Unity[1], one of the leading game engines, has been used. Unity is a fully integrated, cross-platform professional game engine, equipped with a wide array of capabilities for game development, such as advanced lighting, animation, game physics, and artificial intelligence. A set of advanced rendering techniques are utilized in the VE, e.g., shadow maps and environment maps. The target road network in the city model matches the node-link road network in the Paramics and Vissim model. It is imperative that

[1] For more details: https://www.unity3d.com/

the geometry of the road network in all models aligns appropriately. Fig. 8 demonstrates the development of the virtual environment. The process is reversible, depending on the initial model at disposal.



(a) (b)

(c)

Fig. 8. Virtual Environment in: (a) Paramics; (b) Vissim; (c) Unity.

Other than the static objects in the VE, creating accurate movement of the vehicles is quite indispensable. Unlike any post-simulation visualization, the movement of vehicles in the VE is synchronized with the real-time trajectory profiles of each vehicle. Paramics and Vissim shared the profiles with the VE through a customized API. Other than the position coordinates, the vehicle bearing information from the simulation model was also shared to determine the orientation of the vehicles in motion. In almost all MTS models, the vehicle turning is unrealistic because it snaps from one position to another when changing lanes or making a turn in each time step. To remove the snapping effect, Spherical Linear Interpolation (Slerp) was performed for vehicle rotation on the clients' end, achieving smoothed turning movement.

3.4.3 Distributed Multi-User Visualization System

To test the feasibility of the distributed client-server framework, both Vissim and Paramics was turned into the MTS server by incorporating networking communications using TCP/IP protocols. Each virtual environment visualization client connects to the server using the server's domain name or IP address. The following figure shows how the server is hosted in a distributed nature.

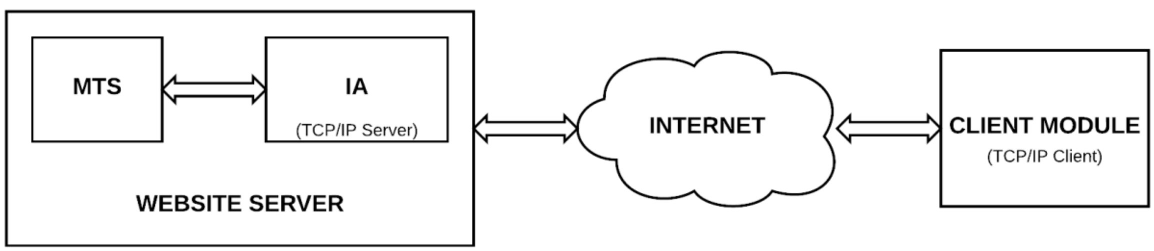


Fig. 9. Distributed Architecture of the Framework.

The server does not need to know the clients' IP. Up to five clients in different networks were connected to the server in this prototype. The following figure demonstrates one experimental configuration. Note that the three clients are connected to the MTS server through the wireless network. The road network in the MTS server shows the simulation model. Each client can freely choose to navigate through a scenario of their interest. For example, Client 1 follows the movement of a turning vehicle. Client 2 examines an intersection while standing near the waiting area of a crosswalk. These demonstrations suggest that users in different locations can simultaneously visit the server and “step into” the simulation visualization remotely, which confirms the success of the proposed distributed multi-user visualization framework for supporting project collaboration or public outreach.

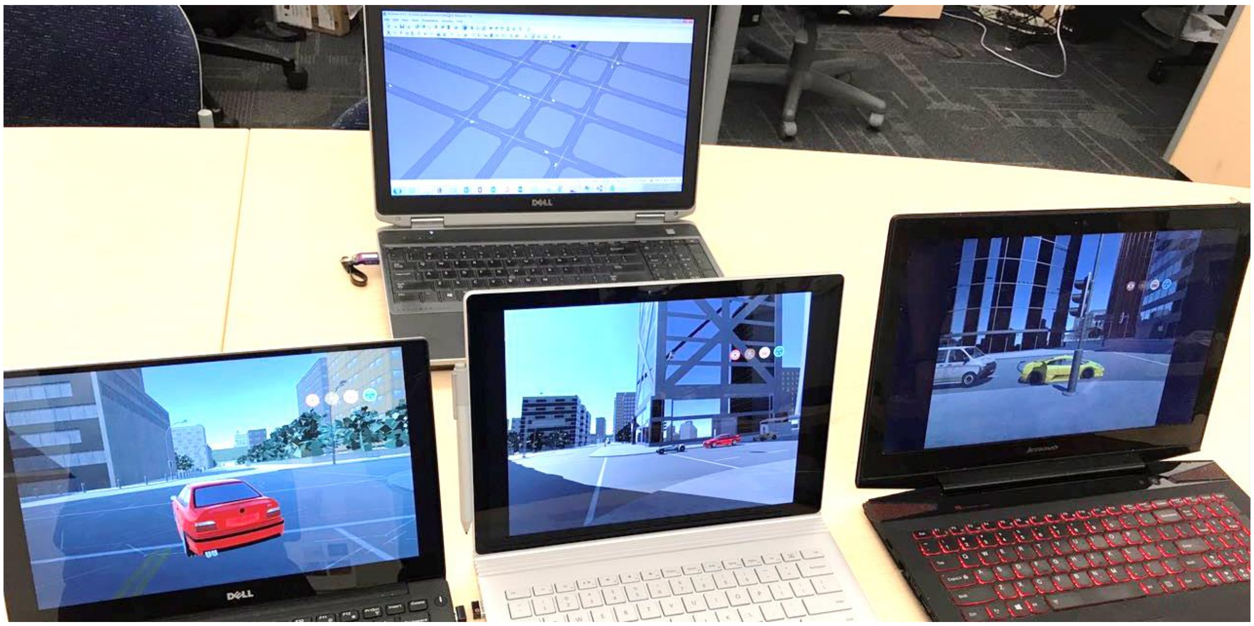
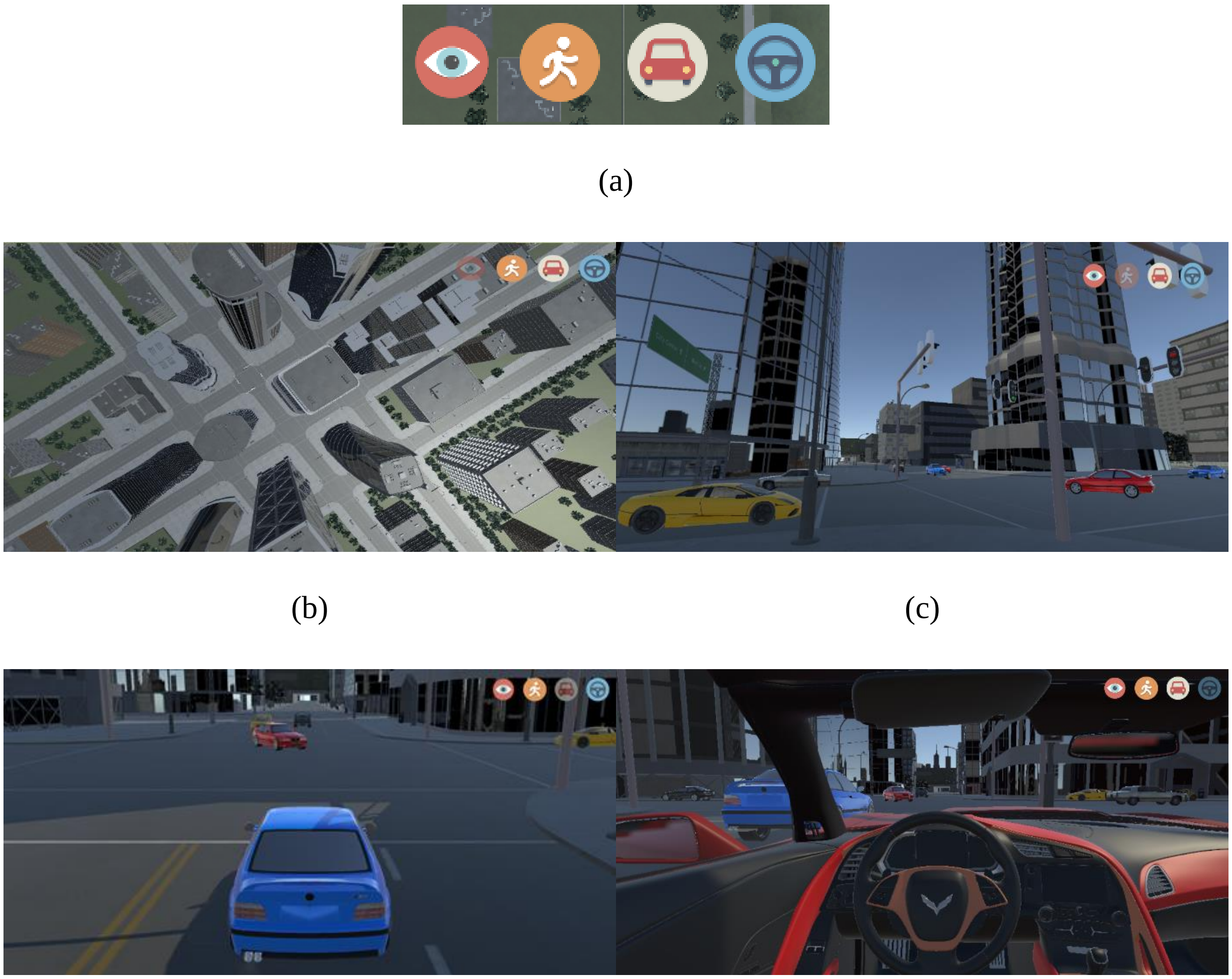


Fig. 10. Multi-user visualization system to support multiple distributed clients simultaneously.

3.4.4 Multiple Perspectives

Other than zooming and panning, each user is offered the capability to explore the visualized environment from different perspectives. For the current prototype, four different perspective modes are supported, including Bird’s-eye view, Pedestrian view, Car Follower view, and Driver view. A user can easily select a view using the menu as shown in Fig. 11(a). The buttons from left to right in the figure represent Bird’s-eye view, Pedestrian view, Car Follower view, and Driver view. In the desktop version of the visualization, the views can be selected by clicking one of the four buttons placed in the top-right corner of the screen. In the VR/AR version of the visualization, the users can pop up a user interface with the same four buttons for selection.



(a) (b) (c) (d) (e)

Fig. 11. Multiple perspectives of simulation: (a) View menu; (b) Bird’s-eye view; (c) Pedestrian view; (d) Car Follower view; and (e) Driver view.

By default, the visualization starts in Bird’s-eye View (Fig. 11(b)) where the user can see the overall network from an overhead position. The view can be in orthographic mode or in perspective mode. In this view, the user can zoom and pan. When the user selects Pedestrian view, a pedestrian is generated and the viewer can see through the pedestrian’s eyes (Fig. 11(c)), walk around, select vehicles, look up-down-around, etc. To be able to follow a car, the user selects Car Follower view and then clicks any one of the vehicles (Fig. 11(d)). The user’s view will start to move along with the vehicle from a distance and can look around freely. The same goes for the Driver’s view (Fig. 11(e)).

3.4.5 Multi-Mode Visualization

One of the goals of this project is to offer a multi-mode visualization so that users can visualize the MTS from user devices. To this end, we adapted the visualization to VR and AR environments, which offer immersive views of the scene. The implementation of the scene in virtual reality was deployed in an HTC Vive, an Oculus Rift and a Gear VR with a Samsung Galaxy Note 4, while the Microsoft HoloLens headset was used to implement the scene in augmented reality. Fig. 12 shows the final visualization experiments. Other than the conventional desktop view (see Fig. 12(a)), the proposed framework provides users an immersive environment using any of the VR and AR devices (see Fig. 12(b) to (e) for the examples).

Fig. 12. Demonstration of the multi-mode interactive visualization.

A crucial aspect in the adaptation of the scene to the VR and AR environments was the change of the user's interface and interactions with the scene. In the desktop applications with traditional displays, user interfaces are commonly implemented in the screen space, and a user UI is usually displayed at a fixed location on a screen, e.g., Fig. 11, and users interact with the UI with a mouse or keyboard. However, screen spaces no longer exist in VR/AR applications as there are no input devices to interact with such spaces. Instead, user interfaces are implemented directly in the world space in VR/AR applications. In this prototype, user interfaces are implemented in a panel that has a fixed distance from the users' eye and can be enabled or disabled from a button on the controller (see the left side of Fig. 12(d) for an example). The user can select a view using the controllers in the HTC Vive, the remote in the Oculus Rift, or the touchpad in the Gear VR. In the case of the Gear VR, the user must gaze with their head to the proper button in order to select it, since the device does not include a positional-tracking controller. Fig. 12(d) illustrates such a selection process, in which the user attempts to select the Pedestrian view. Another interaction with the scene is the movement of the users in the first-person view. In this case, we implemented a teleport system with the controllers in the HTC Vive so that the users can move to a point that they select (Fig. 12(b)). Users will then see a similar scene like the one on the TV monitor but in 3D mode through HTC Vive. Similarly, in the Oculus Rift, the user has the ability to move using the arrows on the remote controller (Fig. 12 (c)). In the case of the adaptation of the scene to Microsoft HoloLens, customized hand gestures have been designed for operations such as zooming, panning, selection, rotation, etc. The scene can be conveniently demonstrated in any environment using HoloLens, for example, a large auditorium in Fig. 12(e). More importantly, the user can move as a real pedestrian in a 1:1 AR environment without any manual input. None of the current microscopic simulation models have such visualization capability.

3.5 SUMMARY

In this chapter the basic system architecture along with a developed prototype of the dissertation topic was presented. This basic framework established communication between the MTS server and the Unity client application. All the capabilities like participation of multiple users from geographically distributed locations, visualization from multiple perspectives, and interaction with multiple devices were presented. The next chapter discusses extending the basic framework by allowing users to control vehicles of their choosing from the simulation.

CHAPTER 4

MTS WITH USER-CONTROLLED VEHICLES

The previous chapter introduced the communication between the client application and the server. This chapter mainly discusses the extension of the framework by incorporating user-controlled vehicles into the system and methods to improve car movements using interpolation techniques. The key issues for including real time user input in the simulation are also addressed in this chapter. As most MTS can run faster than real time it is important to slow down the simulation to have real time user interaction capabilities. Finally, this chapter presents three experiments, performed by 5 participants, to validate the framework. The research presented in this chapter has been published in a prestigious journal called "Advances in Simulation Software Engineer" [114].

4.1 EXTENDED SYSTEM ARCHITECTURE

The core service provided by the server is microscopic traffic simulation (MTS). Any commercial off-the-shelf or free MTS software tools can be used by the server if it has an API for the developers. Currently, two of the most popular MTS tools are utilized: PTV Vissim and Quadstone Paramics. In addition to the MTS core, the server also contains a customized MTS API that is responsible for the client-server communication on the server side, such as establishing two-way TCP/IP communication and update relevant variables. The information exchanged between the server and client is minimal, including vehicle information (position and bearing), vehicle control (e.g., lane change), and signal states. Each client is a standalone application that contains a rich virtual environment representing the simulation scenario running on the server (including roads, buildings, and vehicles), receives information of vehicles controlled by the MTS on the server, and sends the inputs to the vehicle controlled by this client (each client controls one vehicle that is selected at the beginning of simulation), and renders the simulation scene based on updated information.

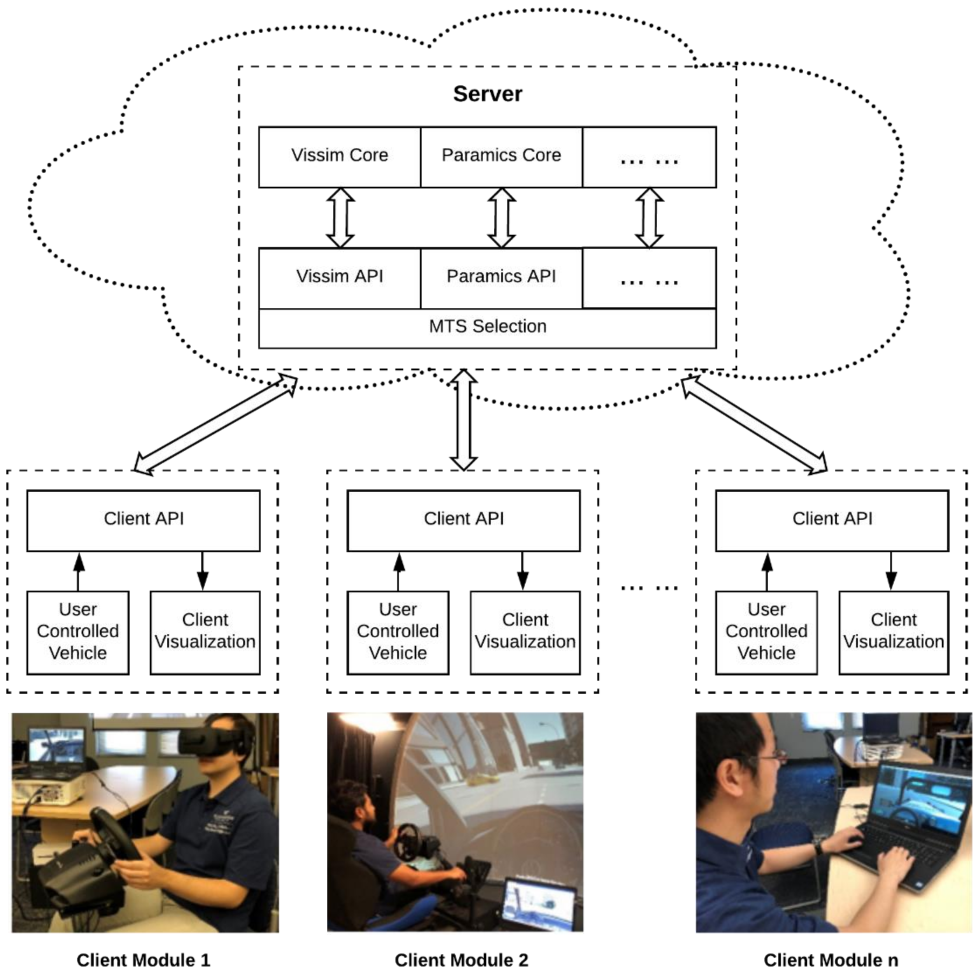


Fig. 13. Functional Diagram of the Extended Framework with User-Controlled Vehicles.

These functionalities are implemented by the several blocks inside the client in Fig. 13. Although the emphasis of this study is utilization of VR devices such as HMD devices and steering wheels, the client does support users without VR devices and in this case, the user simply uses traditional input/output devices, e.g., LED monitor, keyboard, or mouse, to interact with the simulation and control the vehicle. The server contains information of all vehicles (controlled by either the MTS or a client) and all clients update vehicle information accordingly based on the information sent by the server. Each client can select vehicles of different types and dynamics, e.g., sedan, SUV, truck, and exhibits different driving behavior based on the user (driver, e.g., teen driver, senior), road conditions (e.g., city streets, expressways), and so on. It is important to note that a client can also be a viewer of the simulation, e.g., a passenger in a vehicle controlled by the MTS or a pedestrian at an intersection. This powerful and flexible architecture enables transportation researchers to achieve better modeling and simulation results and allows them to convey their ideas and work more effectively.

4.2 CLIENT-SERVER SYNCHRONIZATION AND COMMUNICATION

Each client application is a standalone representation of the traffic simulation that is controlled by the MTS on the server. To include human-in-the-loop in traffic simulation, the simulation must run in real time, i.e., the passage of simulation time must be the same as that of real time, to allow realistic interactions between the user and the simulation. As the client application and server application are two separate applications running on different computing devices across the Internet, several technical issues must be addressed on the client and server to ensure proper client-server synchronization and communication, as required by the human-in-the-loop traffic simulation.

4.2.1 Road Network Correspondence between the Server and Client

The server and client serve different purposes. The major task of the server is to execute the MTS simulation, while the client presents the user a realistic virtual environment that contains the simulation results and allows the user to interact with the simulation (as a driver or a viewer of the simulation). To ensure consistency between the server and client, caution must be exercised when generating the models or data for the server and client. The most critical information for the server is the road network topology. On the other hand, much more detailed information for the same road network is needed on the client, such as its visual appearance and physical (surface) properties. The road network for the server can be extracted from the 3D scene on the client, or vice versa the 3D scene can be generated based on the road network on the server. In both methods, the two road networks must match in terms of geometrical shape and positions to allow most efficient communications between the client and server and minimal additional processing in the simulation (e.g., no need for additional transformations).

4.2.2 Simulation Time vs. Real Time

Time is an important element in many simulations, and particularly so for traffic simulations, e.g., travel time from origin to destination. Two types of time are involved when executing a simulation. The first is the simulation time, which is the time simulated in the simulation, e.g., a traffic simulation for the rush hours 6:00 am – 8:00 am. The second type is real time, which is the time needed to execute the simulation, e.g., 10 minutes needed to run the rush hour traffic simulation. For most simulations, simulation time and real time run at different speeds, depending on the type and complexity of the simulations. Simulation time usually runs faster than real time in microscopic traffic simulations to allow the traffic researcher to obtain results in an efficient way. However, to include human-in-the-loop capability, a simulation must be run in real-time, meaning that the simulation time and real time (wall clock) must run at the same speed, to allow realistic user interactions with the simulated system. Some MTS tools (e.g., Vissim) allow the simulation to run in real time directly, while other tools (e.g., Paramics) allow the simulation to pause for a certain amount of time at the end of each simulation step. The latter requires fine tuning of the pause time parameter for different simulations as they have different complexities, and thus varying run times.

4.2.3 Update Frequency Interpolation

Both the server and client perform calculations and update simulation entity information at a given frequency. However, most MTS tools can only support a limited number of updates per simulation second (e.g., Vissim: 10 timesteps/second, Paramics: 6 timesteps/second), while most interactive graphics applications such as the client application in this work require an update frequency of at least 30 Hz or 60 Hz to avoid flickering and other unpleasant visual artifacts. A satisfying and realistic user experience is especially important for the human-in-the-loop simulation to produce user experience and simulation results similar to those in the real world. To address the mismatch between the server and client update frequency, different interpolation techniques can be used to produce high-frequency updates based on low-frequency samples. The details of the update interpolation of this work will be discussed in Section 4.5. An ignorable drawback of this technique is that the client has a delay of 1 or 2 timesteps (0.1 s or 0.2 s) depending on the interpolation polynomial order.

4.2.4 Traffic Control Synchronization

An important aspect of transportation control is traffic signal, which usually is included in microscopic traffic simulations. While Vissim provides the information in its API, Paramics does not allow developers to access it from its API. To address this deficiency, the complete traffic control information is sent to the client before the simulation starts. The client then updates signaling control information on its own based on the simulation time, synchronized with the server. Although this does not solve all traffic control problems, it does provide a workaround for simple fixed signaling. Other information exchange between the server and client should be optimized and reduced to a minimum to improve simulation performance and scalability. For example, the client only sends its vehicle information to the server when its position changes.

4.3 PROTOTYPE

This section talks about the driving simulator's addition, improvement of car movements and creating an interactive offline visualization. The prototype was developed using Unity3D which can accept 3rd party driving simulator equipment, Virtual Reality and Augmented Reality headsets, and so on.

4.3.1 Driving Simulator

There are many driving simulators on the market ranging from low-cost to high-cost. The cost of the simulator is generally directly related to the level of complexity and realism the simulator has to offer. The following table (Table 2) shows different categories of driving simulators based on their cost.

Table 2. Driving Simulator Categories

Type	Price Range	Fidelity Level	Example
Extremely Low Cost	< $2000	Low	GTR Simulator
Low Cost	< $10000	Lower Medium	DS-575
Medium Cost	< $50000	Medium	CXC Motion Pro
High Cost	$50K~$100K	High	WTI Simulator
Extremely High Cost	> $100K	Very High	Formula I

In this project, we have built an extremely low-cost driving simulator by using a Logitech G920 Racing Wheel with pedals and shifters (Fig. 14 (a)), Open Wheelers gaming car seat (Fig. 14 (b)), and 3D car models for virtual cars. The car models have a very detailed interior to offer the user a realistic feel. The movement of the cars is strictly physics-based having friction, acceleration, braking and so on. When a collision occurs, the cars even deform. They also react to uneven road conditions.

(a)

(b)

(c)

(d)

Fig. 14. Driving Simulator: (a) Logitech G920 Gaming Racing Wheel, Shifter, Pedals; (b) Open Wheelers Gaming Car Seat; (c) Initial Set-up with a half-dome projection screen; and (d) Laptop Configuration.

The traditional understanding is that the higher the cost of the DS the higher the fidelity of the system. Here, fidelity means realism. The fidelity levels can be boiled down to two types: physical fidelity and behavioral fidelity. Physical fidelity relates to the degree to which the simulator replicates the physical properties of the driving situation, unlike behavioral fidelity, which is associated with the simulators' ability to replicate behavior observed in the world [111]. The problem is there is no one technique or metric against which the fidelity can be measured or compared. The best way to achieve better fidelity is to have better components individually which then makes a better simulator. The fidelity level of this driving simulator can be considered medium to high in this framework with VR. In Fig. 14(c) we can also see the setup with a half-dome projection screen and Fig. 14(d) shows a simplified laptop configuration.

For the half dome setup, the driving gears are set on top of a wooden box with a hollow compartment for the projector. The projector has expensive fish-eyed lenses to spread the image over the half-dome screen. For more immersive deployment, the driver can use the VR headsets.

While driving a car with the racing wheel, it is not realistic for the driver to use keypads, a mouse, or HMD controllers to interact with the scenes. That is why the buttons on the wheel were used to replicate those controls.

4.3.2 Establishing Communication between MTS and Client

In this framework, MTS works as the server and the client module consists of the driving simulators. There are different ways to establish communication between two different applications, but this framework employs TCP/IP communication protocol. The server and the clients generate window sockets in their respective ends and communicate over the internet.

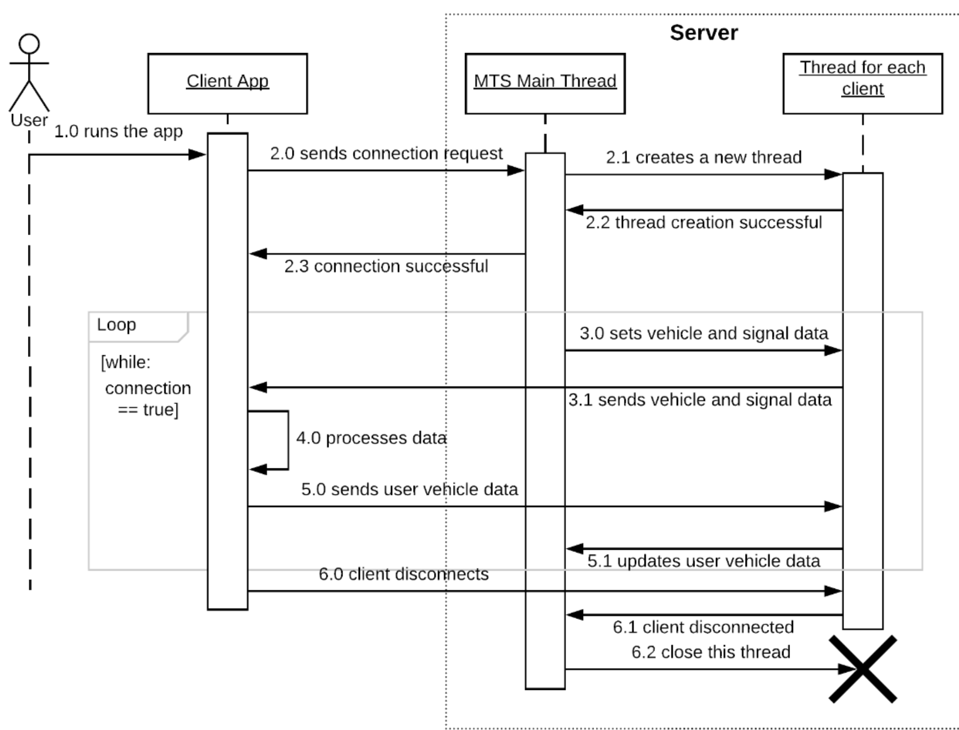


Fig. 15. Sequence Diagram for Events Handling.

A simplified version of the communication is shown in Fig. 15 via a sequence diagram. As we can see from the diagram above, a user initiates the program by running an executable application of the client module. The user manually chooses from the choices of the servers and sends a connection request. The server is executed and generates a thread for each client that connects to it. After the success of the connection, the server starts sending automated vehicle information to the clients. The user then processes those data and each user chooses a vehicle from a drop-down menu like the one in Fig. 16(a). When a vehicle is selected, the user gets control over that vehicle from the MTS meaning the MTS does not have control over that vehicle anymore. The user has the option to accelerate, decelerate, change lanes, and so on.

(a) (b)

Fig. 16. Two Different Views: (a) Car Selection in Top View; (b) Driver's View after Car Selection.

When the display devices are traditional ones like a pc or projection screens, the simulation in the client's end starts from the bird's eye view meaning a top view as shown in Fig. 16(a). After the user chooses a vehicle, the view immediately changes to the inside of the selected car (Fig. 16 (b)) and the user has control. The user can use the keyboard arrow keys or the driving gears to control the car.

4.3.3 Improving Car Movements

There is a mismatch in the update frequency between the MTS and the client application. The server sends vehicle information 10 times per second, but the client updates its view 50 times per second. Updating the cars' positions and orientations every 0.10 seconds does not look realistic enough from a user's perspective. To solve this issue, interpolation techniques have been used so that the cars are updated every 0.02 seconds. Fig. 17 illustrates the general idea of car interpolation.

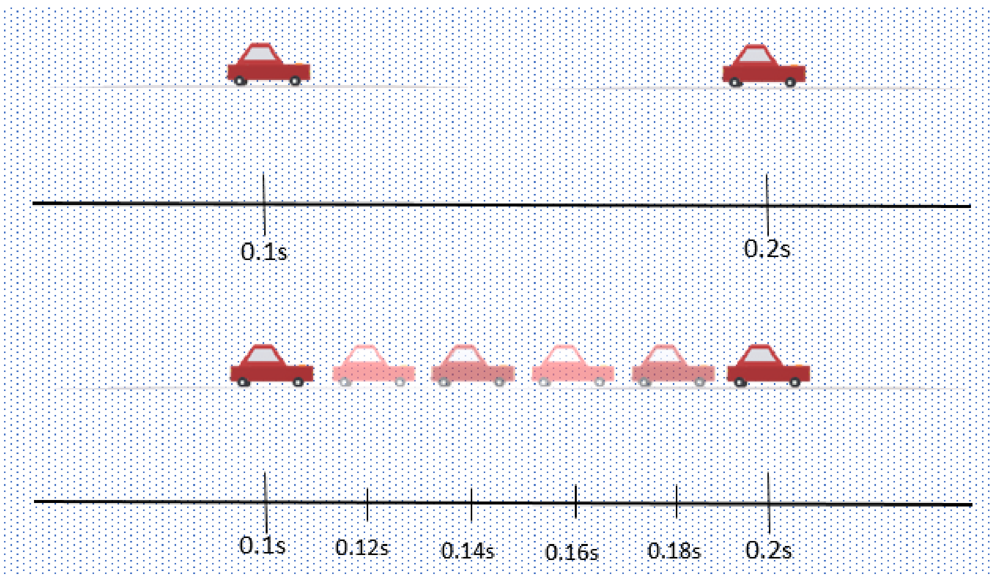


Fig. 17. Interpolating Vehicle Movement within a Time Interval.

Initially, linear interpolation is the simplest interpolation technique studied in this paper. It uses Eq. 4 to linearly interpolate the cars' positions and orientations, where P_0 and P_1 are the initial and final positions of a car, and P is the position calculated at time t. In this case, t is a percentage between the two positions, updated uniformly between the initial point ($t = 0$) and the final point ($t = 1$). The same equation is applied to the cars' orientations.

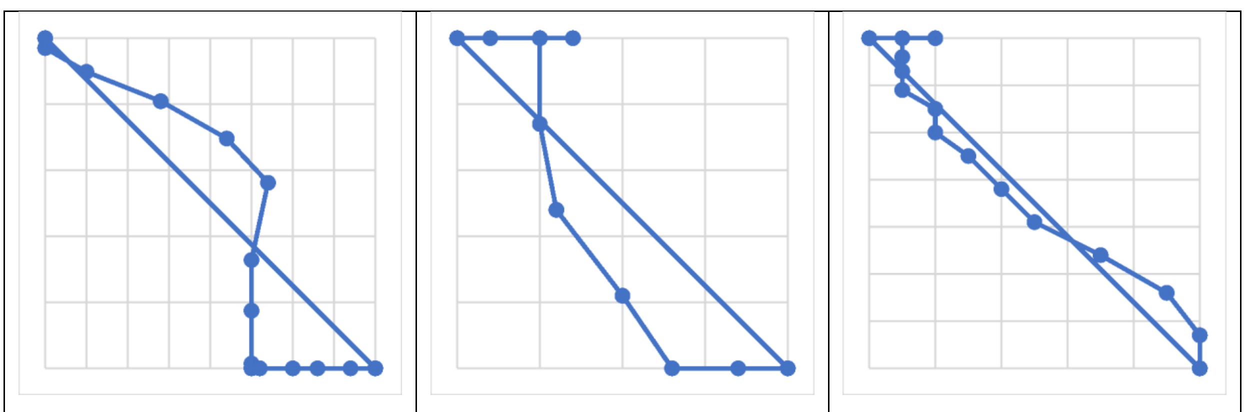
$$P(t) = (1 - t)P_0 + tP_1 \qquad (4)$$

Linear interpolation increases the update frequency of the cars using straight lines. A more realistic way is to use interpolation methods that generate curves between two positions. For this purpose, we use quadratic and cubic interpolation. Quadratic interpolation takes 3 points P_0, P_1 and P_2 and generates a curve using a quadratic Bezier curve, shown in Eq. 5. Once a curve has been generated, the next interpolated curve is computed using points P_2, P_3 and P_4.

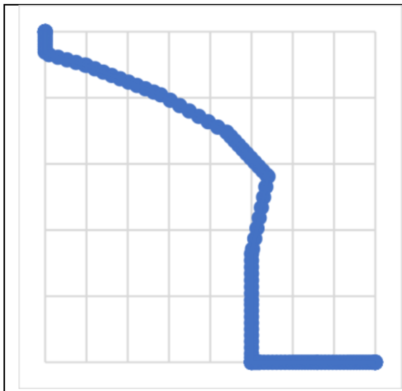
$$P(t) = (1 - t)^2 P_0 + 2 * (1 - t)tP_1 + t^2 P_2 \qquad (5)$$

Quadratic interpolation generates a point between 2 points using a middle point. The problem with this method is that the car will not go through the middle point. To solve this, we use cubic interpolation by computing a Catmull-Rom spline (Eq. 6). This equation uses 4 points (P_0-P_3) and generates the curve that goes through the 2 middle points P_1 and P_2.

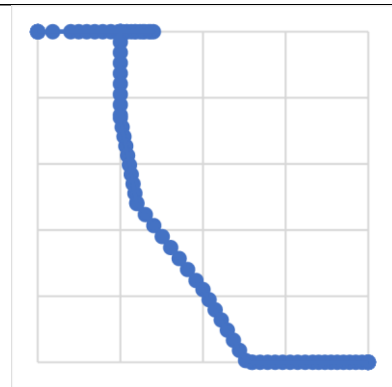
$$P(t) = (2t^3 - 3t^2 + 1)P_1 + (t^3 - 2t^2 + t)\frac{P_2 - P_0}{t_2 - t_0} + (-2t^3 + 3t^2)P_2 + (t^3 - t^2)\frac{P_3 - P_1}{t_3 - t_1} \qquad (6)$$



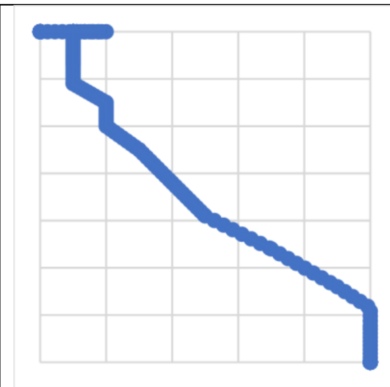
(a) No interpolation in right turn | (b) No interpolation in lane change | (c) No interpolation in left turn



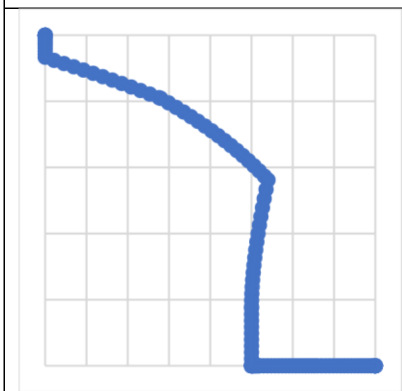
(d) Linear interpolation in right turn



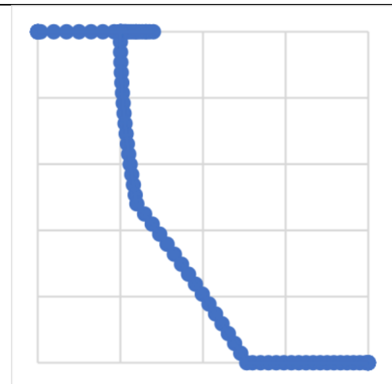
(e) Linear interpolation in lane change



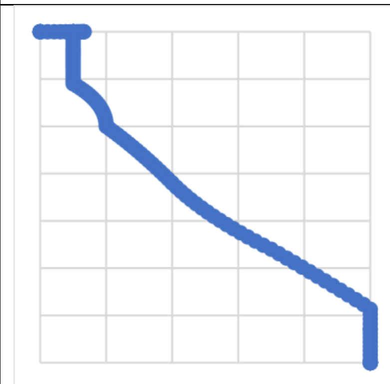
(f) Linear interpolation in left turn



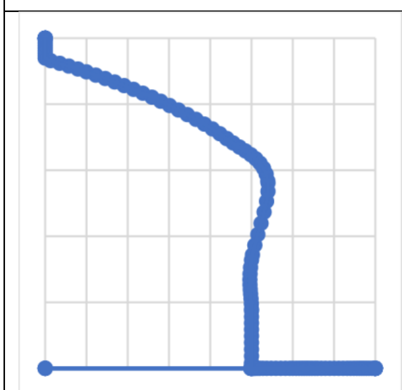
(g) Quadratic interpolation in right turn



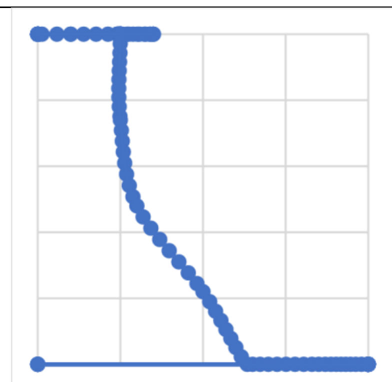
(h) Quadratic interpolation in lane change



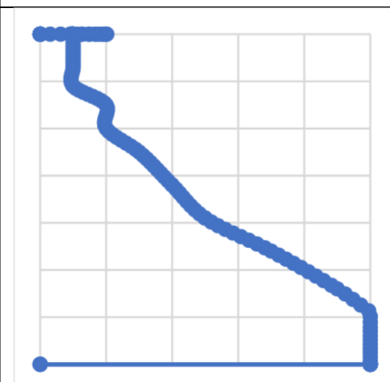
(i) Quadratic interpolation in left turn



(j) Cubic interpolation in right turn



(k) Cubic interpolation in lane change



(l) Cubic interpolation in left turn

Fig. 18. Results of Different Interpolation Techniques when Turning Right, Changing Lane, and Turning Left.

Fig. 18 above shows the comparative results of different interpolation approaches. It shows that the quadratic interpolation generates smoother trajectories in both turning and lane-changing scenarios. Thus, the quadratic interpolation was implemented in the modeling process.

4.3.4 Incorporating Virtual Reality

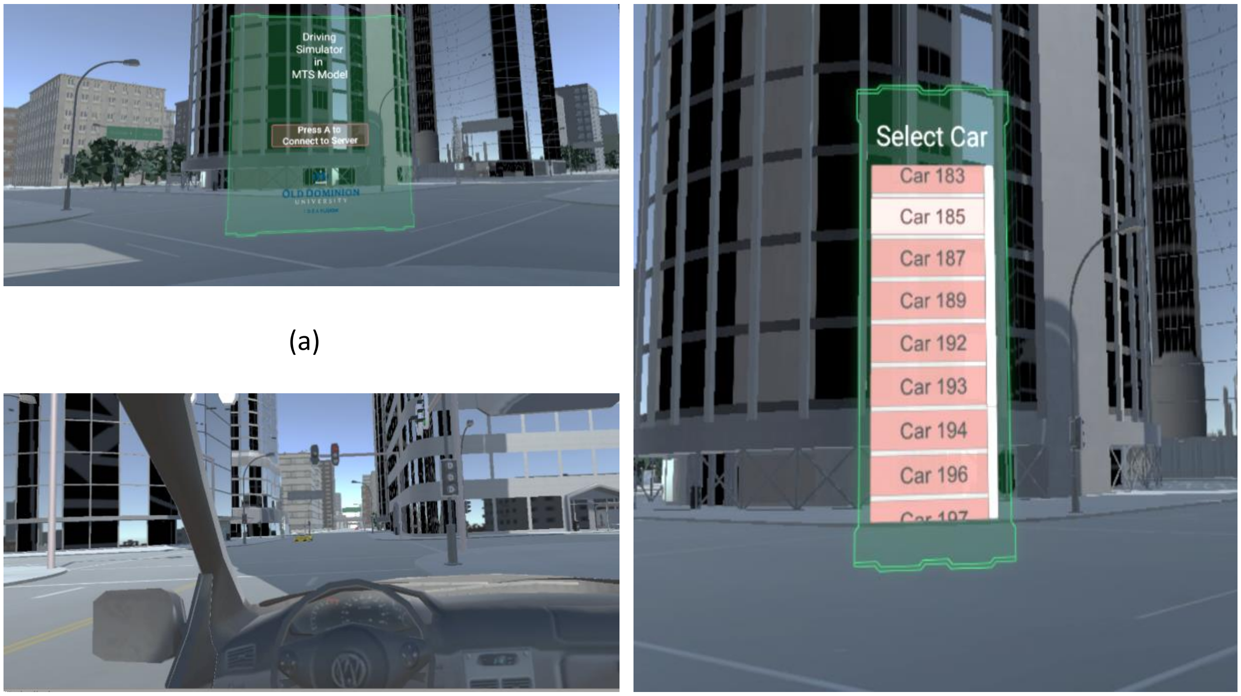
We have developed a working prototype with MTS as the server and driving simulator as the client. The user can simply visualize the scenario in traditional displays but with better graphics, and 3D models with higher resolution than MTS. However, to provide the user an immersive and realistic experience, it is imperative to incorporate virtual reality devices into this project. The first VR device to be used in this project is the HTC Vive. It has a wired headset with two handheld controllers and two lighthouse sensors to be mounted on tall stands diagonal to each other. The Vive provides an approximately $4m \times 4m$ play area where the sensors can track the headset movement. The user can walk within this area in a 1:1 scale. To travel further, the user moves the virtual camera rig with himself. The camera rig represents the play area in the virtual world.

The next VR headset that has been included in this project is the Oculus Rift. This headset has similar equipment as the HTC Vive. Instead of mounting its sensors on a stand, they are put on high ground like on a table. Both VR headsets provide full immersion to the user. In Fig. 19(a), we can see a user wearing an HMD to visualize and using a car wheel and pedals to drive and interact with the simulation. In Fig. 19(b), multiple users are driving in the same scenario, one using the wheel and pedals and others the traditional keyboard. The view on the projection screen is the one from the client 4 laptop which is the view that the user is getting while wearing the headset.

(a) (b)

Fig. 19. System Deployment Demonstration: (a) Driving Simulator with the VR Headset; and (b) Wirelessly Connected Users in the Same Environment.

Fig. 20 further demonstrates the first view of the user when he put on the VR headset. It starts on the pedestrian view with a GUI button asking to connect to the server. Once it is connected to the server the option for choosing a car menu starts to fill in with car ids and the user can freely select one as his target vehicle to drive.



(a)

(c) (b)

Fig. 20. User's View in the HMD: (a) Menu Option to Connect to the Server; (b) Menu Option to Select a Car from the List; and (c) View from Driver's Perspective.

Once a car is selected the user is transported to that car's driving seat with full control over that car. Now the user can speed up or slow down or change lanes by using the racing wheel and pedals. It should be mentioned that once a car is selected by a user, its selection option in the selection list is disabled so that other users cannot select the same car at the same time. For example, Car 185 in the list was selected and it is no longer available to other users. With the support of multi-user driving, the developed system will enable the simulation of multi-vehicle interactions with human input. It is also possible to enable each client application with one dedicated user-controlled vehicle from the very beginning. In this case, MTS will receive the dedicated vehicle's information as the client controlled car.

4.3.5 Offline Visualization

One special feature of the developed simulation framework is the offline visualization. The GUI allows a user to record a simulation and have an interactive visualization later. During the

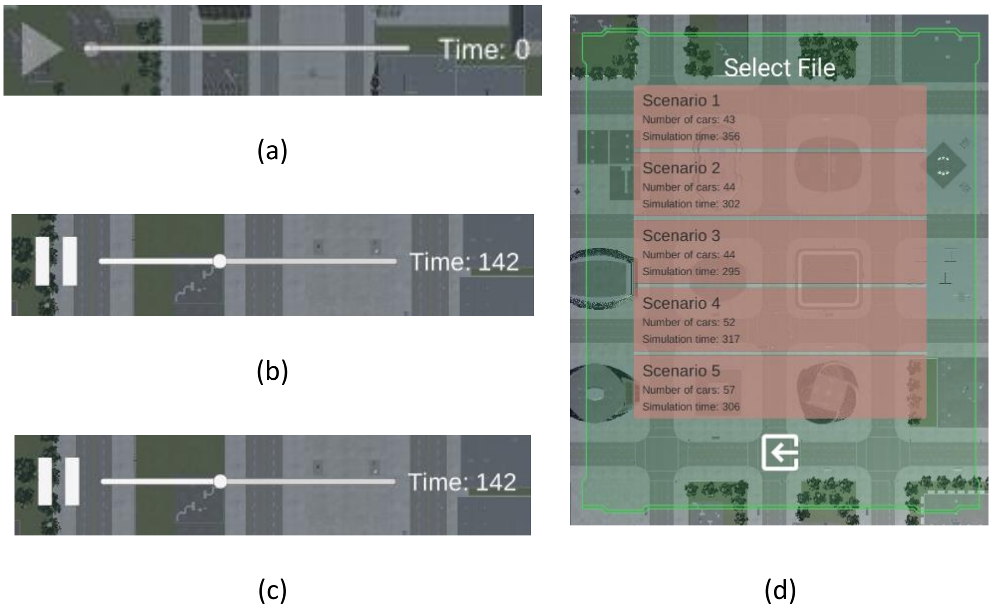


Fig. 21. Offline Visualization GUI: (a) Play Button; (b) Pause Button; (c) Time Slider; and (d) File Selection for Pre-recorded Simulation.

online visualization, the traffic in the client’s application is still run by the MTS and when any vehicle reaches its destination that vehicle is destroyed in both places. Thus, it is often difficult for an end user to experience every detail of the simulation during an online experience unless the simulation runs indefinitely. A better way is to record the visualization which will still have all the interactive options like bird’s eye view, car following, pedestrian view, driver view, zooming and panning and so on. In Fig. 21 we can see the GUI for the user from which s/he will be able to load a pre-recorded scenario.

4.4 EVALUATION OF THE FRAMEWORK

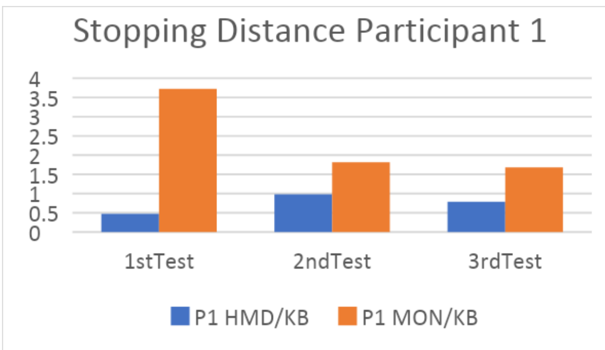
This framework supports multiple users from distributed locations and has the capability of incorporating virtual reality technologies for an immersive experience. To validate the framework, three different experiments were conducted with five different participants. Before and after the experiments the users were asked to fill in some pre-test and post-test questionnaires based on an iGroup presence questionnaire (IPQ) [112]. The pre-test questionnaire included the age, gender, driving license, previous VR HMD experience, previous gaming experience and whether they use computers on a regular basis. The participants' age ranged from 25 to 31; four of them were male and one female, none of them had a driving license, two of them had previous VR HMD experience, one of them never played any computer games and all of them use a computer on a regular basis.

4.4.1 Experiment I: Stopping Distance (Single User)

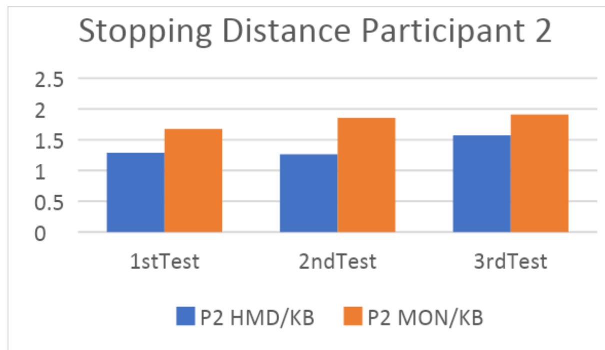
Purpose: The purpose of this experiment was to compare the user perception of stopping distance in the virtual world while driving a virtual car using a VR HMD (Resolution: 2160 x 1200) vs a 2D monitor (Resolution: 2560 x 1440).

Procedure: Each user participated first using a VR headset in this case the HTC Vive and then using the monitor. The car in the virtual world was put at three different positions marked as 1stTest, 2ndTest and 3rdTest in this experiment. The participants were asked to go toward an intersection and stop before the white stopping line. Then, the distance from the fixed point on the stopping line to the front of the car was recorded. If the car crosses the stopping line the distance is negative. All the participants used keyboard arrows to control the car.

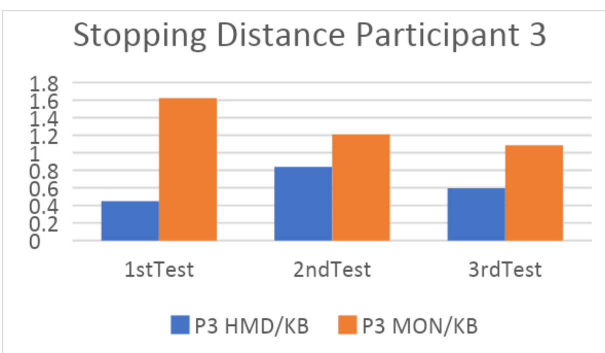
Results: Four out five participants performed better in VR than in the monitor. The following figure shows the plot for the stopping distance for each participant and the final one shows the average. The average shows that in VR people did about 8.5 to 40% better for different tests than in the monitor. Although the sample size is small, it proves the efficacy of using VR in such platforms.



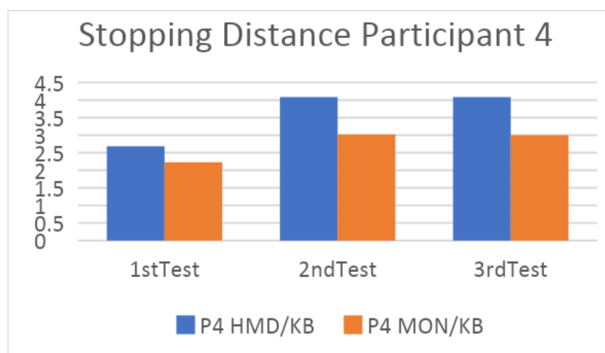
(a)



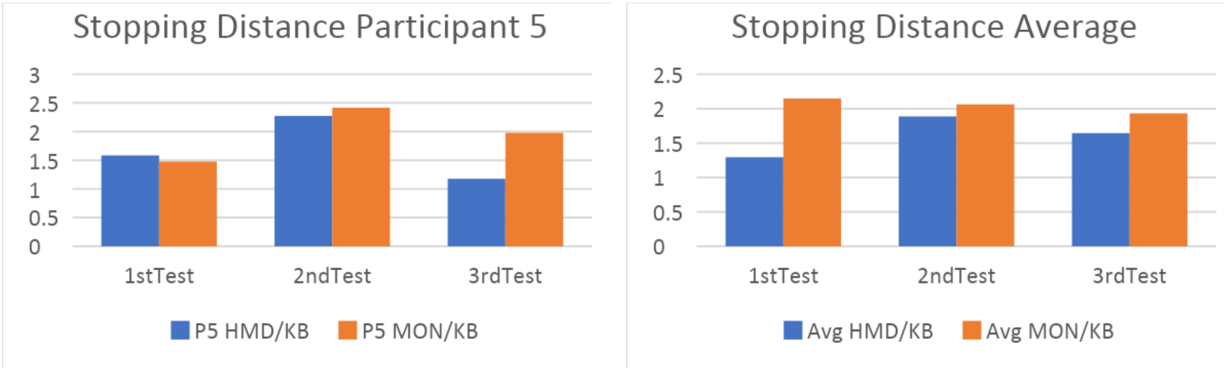
(b)



(c)



(d)



(e) (f)

Fig. 22. Stopping Distance for different participants.

4.4.2 Experiment II: Sun glare (Single User)

Purpose: The purpose of this experiment was to showcase the effectiveness of this framework with Unity3D which allows creation of realistic scenarios quite effortlessly and to test the efficacy of VR as well.

Procedure: The sun glare in the morning while driving eastward or in the afternoon while driving westward is quite problematic for any driver. A realistic sun was created in Unity3D (shown in the following figure) to mimic the sun glare effect and five participants were used to assess in which display, VR or monitor, did they feel the sun glare more realistic. There was an intersection with traffic lights and some signs billboards on the driver's way.

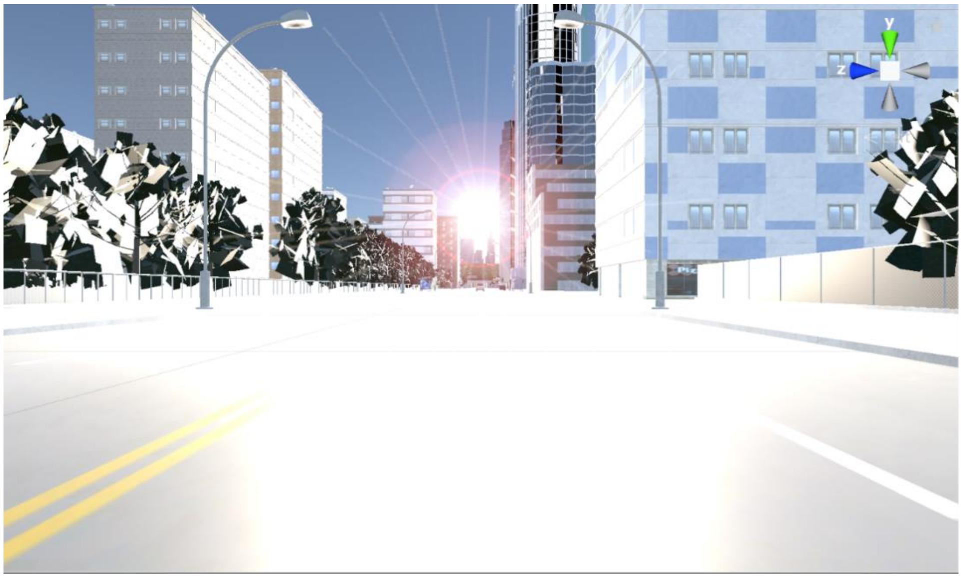


Fig. 23. Sun glare example scenario.

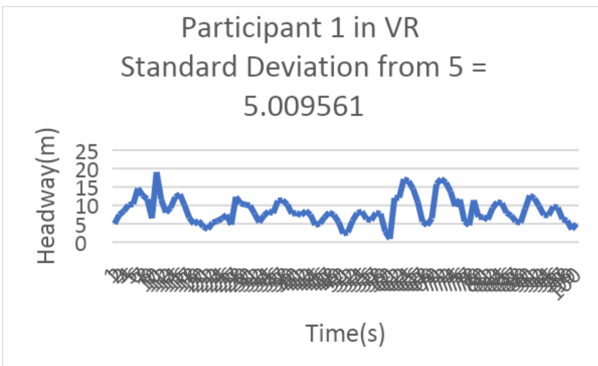
Results: It was a qualitative experiment. The participants were asked to compare the visibility of the traffic lights, signs and billboards in the sun glare while driving eastward both in VR and in the monitor. Four out of five participants said the sun glare effect felt more realistic in VR while the other participant remained neutral.

4.4.3 Experiment III: Maintaining Fixed Headway (Multiple Users)

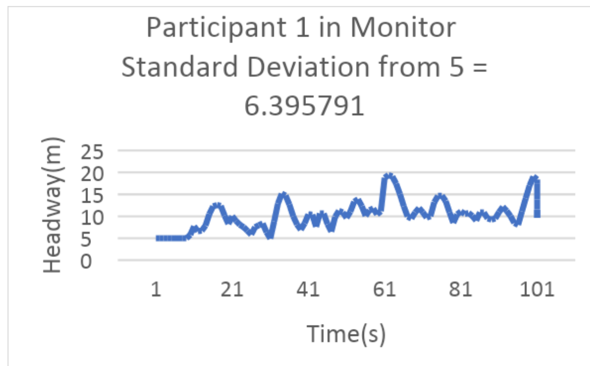
Purpose: The purpose of this experiment was to show the capability of multiple driver interaction.

Procedure: The participant would drive a following vehicle maintaining a fixed headway of about five meters from the leading vehicle controlled by the author from a different computer. All the participants were asked to try to keep a headway of five meters while the leading vehicle performs some simple maneuvers like speeding up, slowing down, stopping, etc.

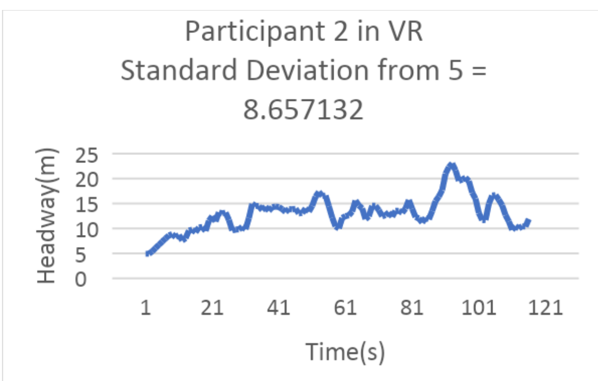
Results: This experiment had five participants. The headway for each participant was collected every second and the data were plotted, and the standard deviation were calculated. Four out of five showed better performance in VR than in the monitor. The following figure shows the headway plotted against time for each participant and their respective standard deviation from 5 as each participant was asked to maintain a 5-meter headway.



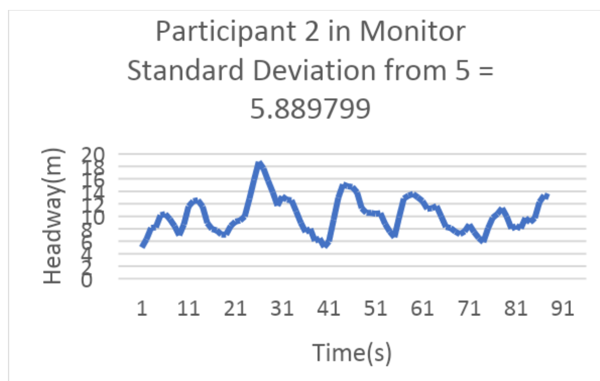
(a)



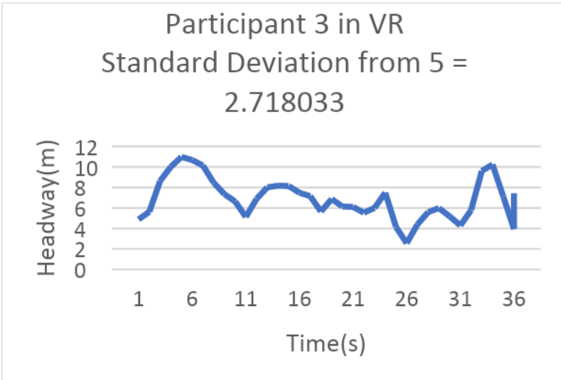
(b)



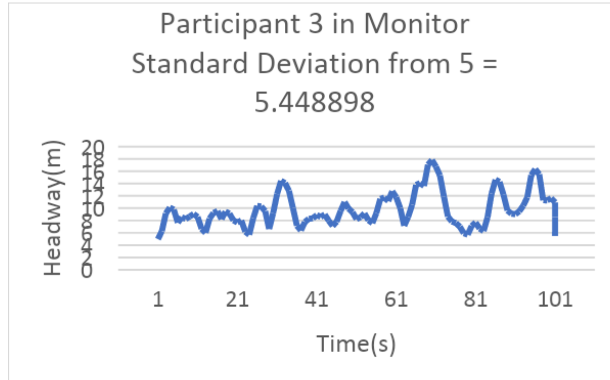
(c)



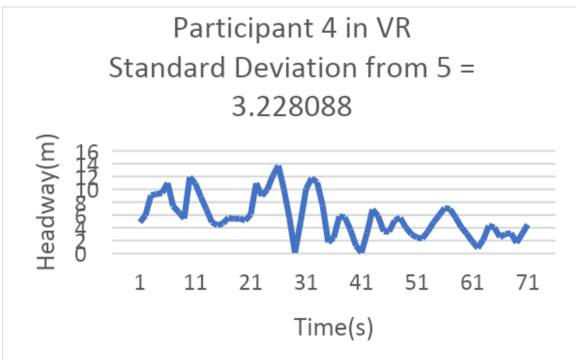
(d)



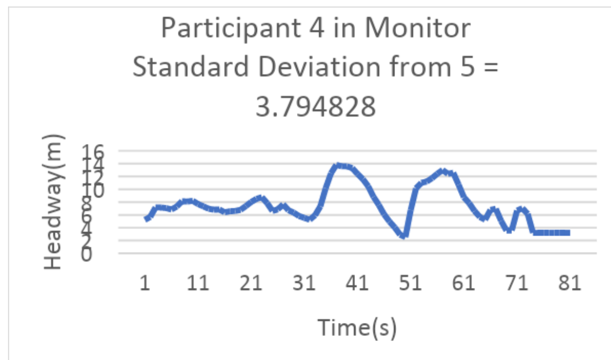
(e)



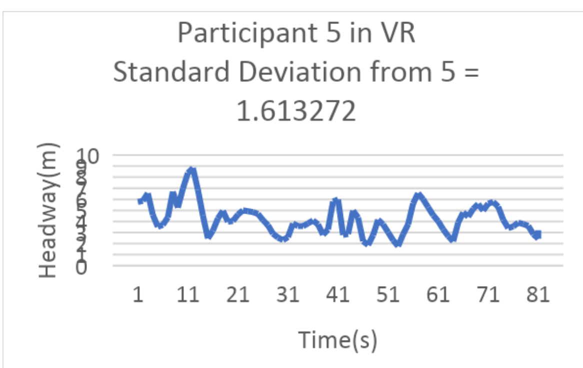
(f)



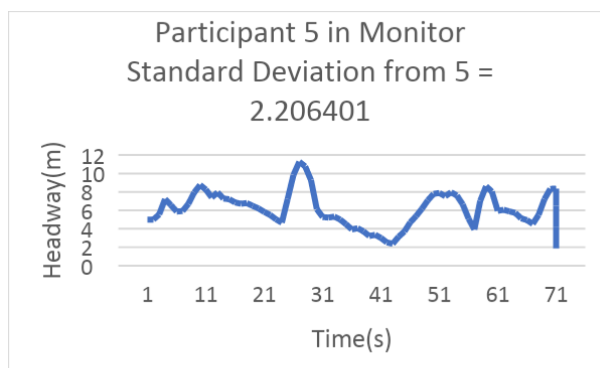
(g)



(h)



(i)



(j)

Fig. 24. Headway for different participants.

From the standard deviation from 5 we can see that four participants did better in the VR. Standard deviation from the mean was also calculated (not shown in the figure above) and it showed that three out of five participants did better in VR.

4.5 POTENTIAL APPLICATIONS

The following scenarios provide a brief discussion of some possible scenarios that can benefit from the use of the developed simulation framework. These are potential use cases, not developed in this dissertation work.

4.5.1 Traffic Operation Assessment

Assessment of traffic operation is one of the most important application areas of MTS models. Usually, the MTS models provide 2D graphics postprocessor to understand the output data. It is often not possible to analyze the operation from statistical output alone. The developed framework offers immersive 3D visualization that creates a more engaging experience for the user. Moreover, with various interactive options like viewing the simulation from different perspectives such as aerial view, pedestrian view, and driver view, the user can find out more about the operation than from some 2D view or statistical output.

4.5.2 Model Calibration

MTS models are quite useful for evaluating traffic engineering projects, for example, adding a lane to an existing road or adding a road to an existing network, changing the signal timing, installing variable message signs, and so on. However, the reliability of such models depends on how the parameter values in the simulation accurately capture the real-world system.

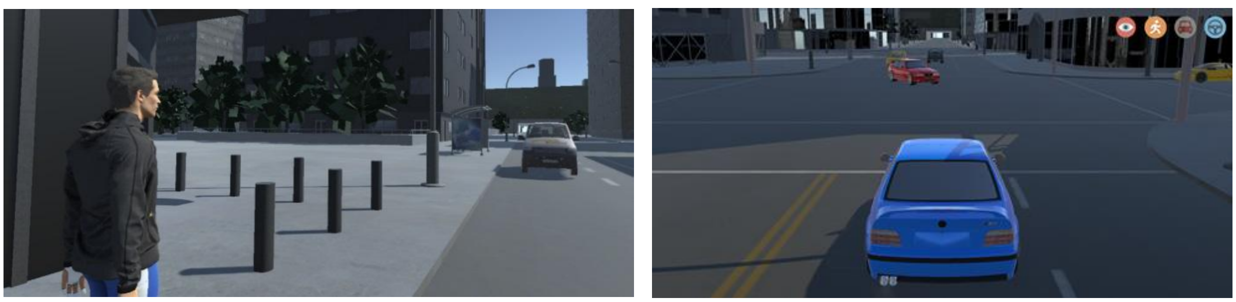
Calibration of the simulation model is required to achieve best reproducibility of the field condition. Calibration is the process by which the various components of the simulation model are set so that the model will accurately replicate observed traffic conditions. One important set of parameters that are often required for model calibration are driving behavior parameters such as headways, reaction time, acceleration decisions, lane changing decisions, etc. With our human-in-the-loop MTS, it is possible to collect driver behavior data from actual drivers with different ages, experiences, ethnicities, physical or mental disabilities; all running vehicles in the virtual environment. In particular, their interactions data can be of great value.

4.5.3 Variable Speed Limits

Transportation agencies work tirelessly to improve traffic safety and efficiency. One of the attempts they have taken is called active traffic management (ATM), and variable speed limit (VSL) is one primary example of ATM. Traffic engineers implement VSL systems to add dynamic control to an ordinary static system called speed limits. Currently in most places around the globe static speed limit sign boards are set based on offline engineering studies. However, road traffic is dynamic by nature. People speed up or slow down depending on various conditions like congestion, weather, road conditions, etc. Depending on those conditions, VSL can be shown to people via electronic message boards. One major obstacle to successful VSL system deployment is how people behave when they see such systems. Researchers have performed behavior studies in response to the VSL systems in 2D flat screens. With our developed prototype, drivers will have an immersive view of the network so the response to the VSL system will be much more realistic and hence getting a much better understanding of the VSL deployment issues. Similarly, this framework can be used to test other traffic control plans like advance warnings in work zones.

4.5.4 Design Exploration

Traffic engineers, planners, and designers spend countless hours trying to figure out the best ways to convince the stakeholders about their project. They use animation, photomontage, statistical analyses to name a few methods to disseminate their ideas. For the first time the stakeholders will have the opportunity to drive in the virtual design and receive firsthand experience.



(a) (b)

Fig. 25. Different Views for Design Exploration: (a) Walking as a Pedestrian; and (b) Following a Car from a Distance.

In addition, they will also be able to walk around the virtual design, ride any running vehicle they want, and scale the model to get a better understanding of the proposed plan. Some of the possible views a user can have for design exploration are shown in Fig. 25. The figure on the left shows a pedestrian viewing a car driving towards himself and the figure on the right shows a viewer following a car from a certain distance. Other possible views and interactions could be top view, in-vehicle view, zooming, panning, capturing videos and pictures, etc.

4.5.5 Experience Autonomous Vehicle Systems

Autonomous Vehicle (AV) system is the latest technology that is highly expected to increase traffic safety and transportation efficiency. Autonomous vehicle or driverless vehicle systems are those where a vehicle senses its environment and drives accordingly without any person controlling it. There is some form of automation already in some of modern cars, for example: automatic parking. The vehicles generated by MTS are programmed based on specific algorithms. If modeled and programmed accurately, some vehicles can represent the AV. If included in the current framework, it generates a mixed traffic environment. As the users are already in the immersive environment, their reaction can be studied with more accuracy and fidelity. Further, if the human-controlled pedestrian avatar is involved, pedestrian-vehicle interactions can also be simulated.

4.6 SUMMARY

This chapter discussed the incorporation of user-controlled vehicles into the system. Some important aspects of the framework like client-server synchronization, improving car movements, creating offline visualization, and potential applications were also discussed. A prototype was presented along with three experimental user studies to validate the framework. The next chapter discusses the incorporation of user-controlled pedestrian into the simulation. The user will use Microsoft HoloLens, a mixed reality headset to visualize and interact with the simulation. The crucial point about the next chapter is that the user can walk around the virtual world in 1:1 scale without any restrictions.

CHAPTER 5

AUGMENTED REALITY-BASED USER-CONTROLLED PEDESTRIAN IN MTS

This chapter discusses the extension of the simulation framework, AR-PED, where AR-PED stands for Augmented Reality based PEDestrian. In this chapter, user-controlled pedestrians coexist with user-controlled vehicles as well as with artificial vehicles generated by an MTS server. A major portion of the work discussed here has been published in a prestigious research journal [115].

5.1 AR-PED SYSTEM ARCHITECTURE

The proposed AR-PED framework also employs a distributed architecture based on the client-server model where the MTS is the server and the clients can access the simulation to control a pedestrian, a vehicle or visualize the scene. The detailed structure of the proposed framework is shown in Fig. 26. Based on their purpose, there are three different types of clients: visualizers, drivers and pedestrians. The visualizer clients present the first type and their details were presented in [110]. The clients allow a visualization in which multiple users can see the same traffic simulation from multiple perspectives (bird's eye view, pedestrian view, driving view and car following view). The visualization can be experienced from different devices (also referred to as modes) such as desktop, laptops, mobile, and VR and AR devices. The second type of clients are the driver clients. Each of these clients can select one of the cars generated by the MTS server and control it using different IO devices. The development and deployment of this type of client has been presented in [114]. The third type corresponds to the pedestrian clients, which are the focus of this chapter. Pedestrian clients are controlled through the Microsoft HoloLens headset, which provides them with a view in AR of the virtual city. The pedestrian data is sent from these clients to the server and then, after it has been processed, the server is responsible for sending it back to the rest of the clients for visualization purposes.

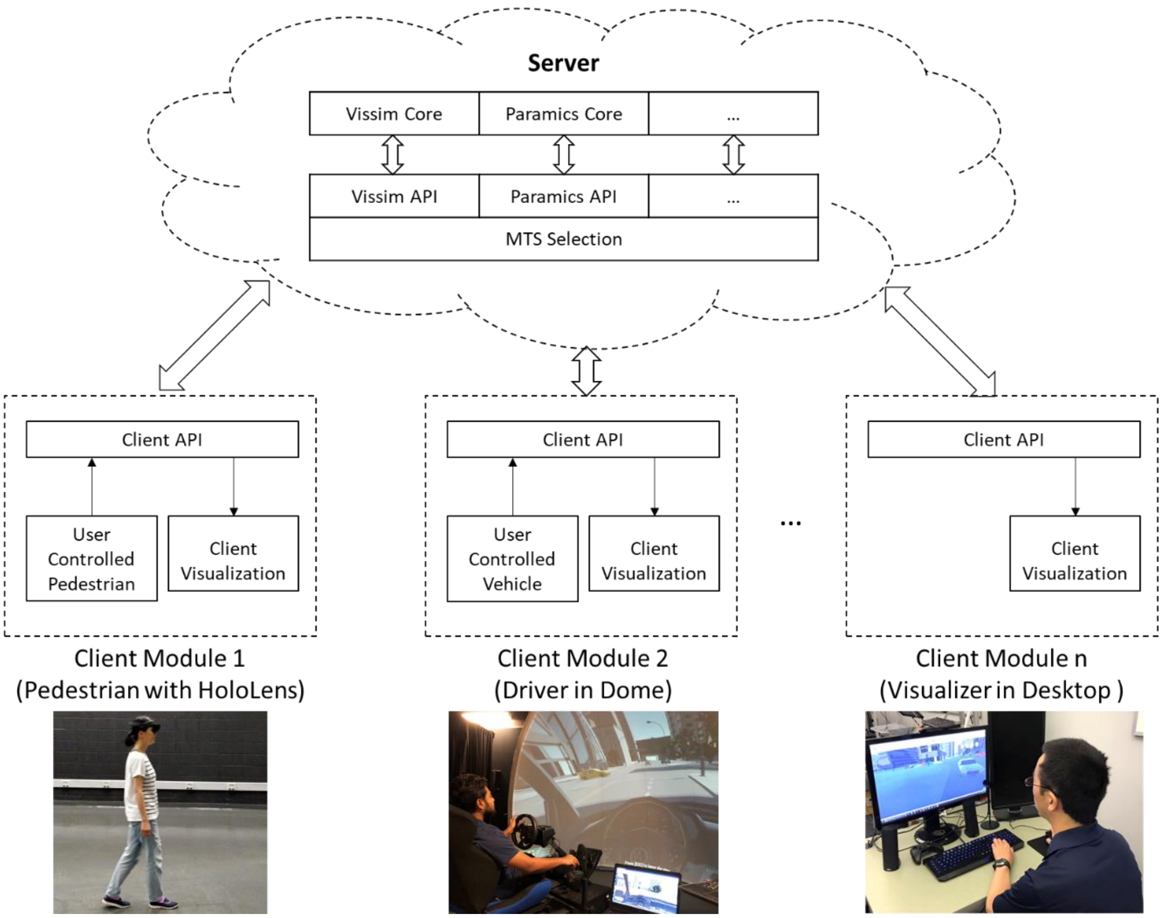


Fig. 26. AR-PED System Architecture.

The server processes all the data from the MTS engine and from the driver and pedestrian clients. This data is then sent back to the clients to ensure a uniform visualization across all the users. Visualization clients do not send any data to the server, since their purpose is solely to visualize the traffic simulation scene. The communication between server and clients is obtained by incorporating networking communications using the TCP/IP protocol. The connection between the server and each client is handled independently, and it is maintained until the client, or the server decide to disconnect it. In this case, the server is always running, and the clients can disconnect from the server, which closes their specific connection without affecting the communication with the other clients.

5.2 PEDESTRIAN CLIENTS

Pedestrian data is collected from the pedestrian clients and sent to the server, where it is processed and then sent back to all the clients so that they can visualize the pedestrians. Each pedestrian client can control one pedestrian, which will be visualized by the rest of the clients in the form of a virtual avatar with the proper dimensions. Once the connection with the server is established, pedestrians can move freely through the virtual city, and their corresponding avatars will be updated in the rest of the clients.

Pedestrian clients are controlled using Microsoft HoloLens, which is the most popular AR headset on the market. We believe that this device is ideal for developing a pedestrian simulator for multiple reasons. First, since the users are able to see the real world merged with the hologram of the virtual city, dizziness and other sickness induced by virtual reality [116] is not a problem. Although the experience of the user can be lessened by obstructions produced by objects from the real world, the device can be comfortably used in big and clear spaces such as basketball courts or parking lots, which are accessible by everyone. However, the most important advantage that this headset presents is the absence of any kind of cables. Most popular VR devices like HTC Vive and Oculus Rift need to be connected to a computer to work, which sets a limitation in the scene boundaries. Even in the case of using significantly long cables, the sensors in these devices need to be placed at a distance [117, 118], giving a maximum simulation area of about 4 m×4 m. In contrast, Microsoft HoloLens does not need a separate computer to work, and it includes the appropriate sensors and cameras inside the headsets, which guarantees users the ability to move through a virtual scenario without boundaries. On top of that, the users can be colocated, meaning be in the same space or remote or both.

The game engine allows us to integrate high resolution 3D models and incorporate real physics in the virtual world, which results in a more realistic simulation. Unity provides a collection of assets, scripts and examples that can be used as a basis for custom projects. This collection is called “Standard Assets” and, among other utilities, has the scripts necessary to start implementing a third-person character. The third-person character example contains a 3D model of a humanoid avatar, as well as a set of scripts and animations that allow the character to move realistically. The 3D model of their character, which they name “Ethan”, is shown in Fig. 27. The 3D model of Ethan can be replaced by any custom model as long as it is rigged in the form of a humanoid avatar. 3D models rigged in this form have a skeleton (shown in green in the figure) which controls the execution of the character’s animations.

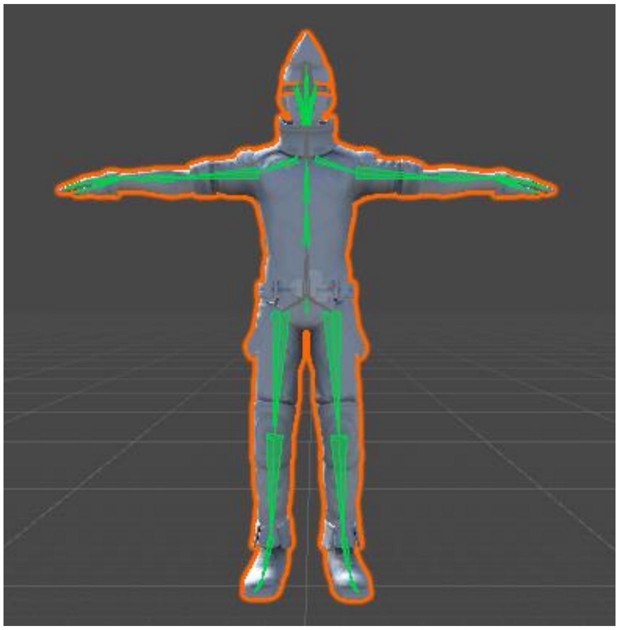


Fig. 27. 3D Model of Ethan, the Third-person Character Included in Unity's Standard Assets.

(The green skeleton shows the structure of the rigged Humanoid avatar.)

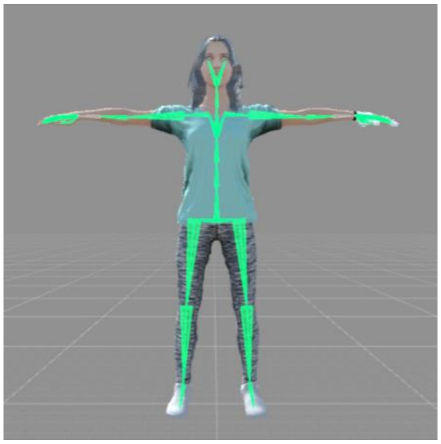
Unity's third-person character controller is initially designed so that players can control a human avatar from a third-person perspective using the keyboard or other input devices. However, in our case, we require the characters to move independently of the user's controls, since each pedestrian is controlled from its corresponding pedestrian client and not from the clients that visualize the avatar. For these cases, Unity provides a version of this asset that is called "AI third-person controller". This controller makes the character move according to a target position that can be set with a customized script. In this project, we use the AI third-person controller for the operation and visualization of the pedestrians. The main idea of the operation and visualization is to assign to each avatar the data obtained from its corresponding pedestrian client so that the human avatar moves according to the pedestrian client movement in 1:1 scale. The AI third-person controller script can be used with other 3D models as well.

5.3 PROTOTYPE

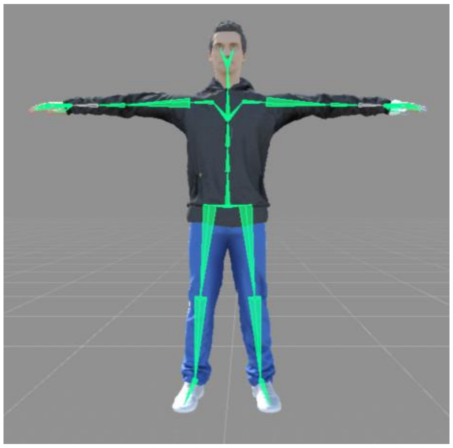
A prototype has been developed to show the efficacy of the proposed AR-PED framework. How the pedestrian avatars were created, how their movement animation is controlled, the way their GUI works, are all discussed in this section.

5.3.1 Pedestrian Avatars

The default 3D model proportionated by Unity for the creation of their third-person character is Ethan: a gender-neutral white/gray humanoid (Fig. 27) with the appropriate scripts and animations to resemble a person's movement. While the walking animations of the character are realistic enough to be used in our framework, the design of the 3D model does not resemble a realistic person. Because of this, we substituted the default 3D model with two models of more realistic female and male humans. The models are shown in Fig. 28, and they were obtained from two free assets in Unity's Asset Store offered by ISBIT Games [119]. The models are rigged in a humanoid structure, and they can inherit all the animations from the default model.



(a)



(b)

Fig. 28. Realistic 3D Models.

5.3.2 Demonstration of the Framework

To demonstrate our AR-PED simulation framework, we show different pictures of a user wearing the HoloLens headset with our pedestrian client framework, while the corresponding visualization of the virtual pedestrian is also displayed on a projector screen simultaneously. Readers can click on the picture to see a video of the demonstration. We used a measuring tape to measure a distance of 3 meters and placed 2 spheres on the crosswalk in the virtual scene that match the corresponding origin and destination points of our physical space in Fig. 29(a)-(c). The spheres were separated by 3 units in Unity, which corresponds to 3 meters in the physical world. Likewise, in Fig. 29(d)-(f), the distance was increased to 3.75 meters to exactly represent the width of a lane. Through these test scenarios, we can determine that the measurements of our virtual scene correspond with the measurements in the real world. Fig. 29(a)-(c) show a person using the pedestrian client framework and the corresponding visualization from top view, and Fig. 29(d)-(f) show the profile view during the walking process.

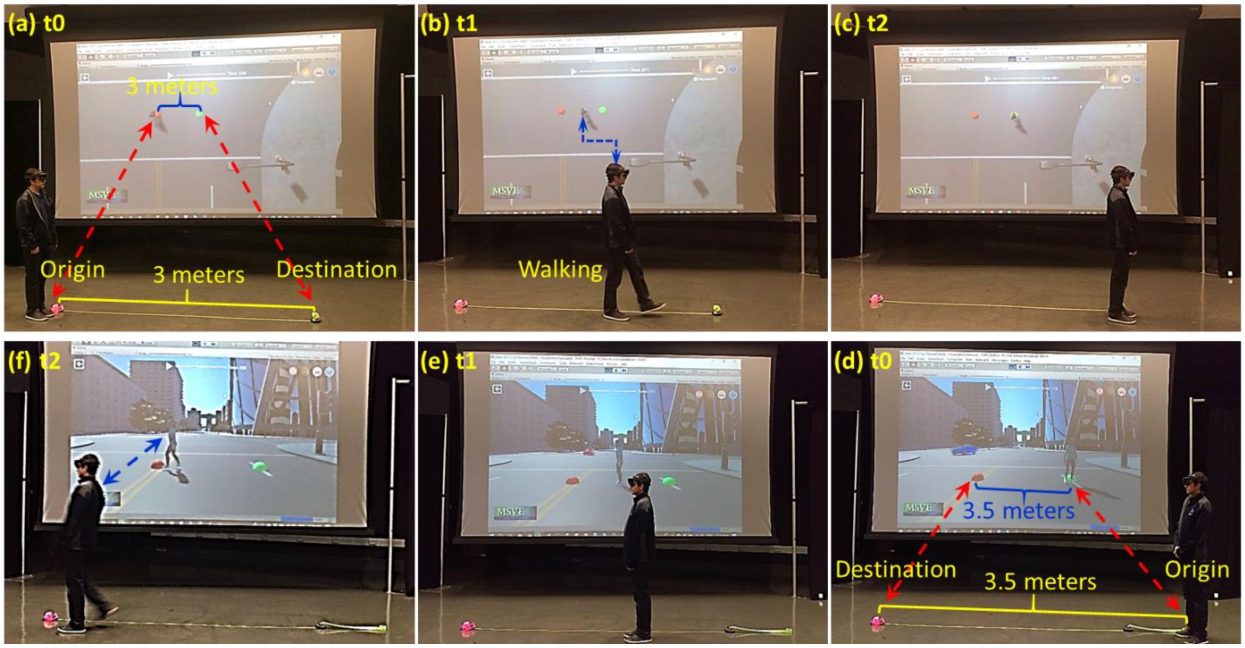


Fig. 29. A User Wearing the HoloLens Headset Walks Straight and the Corresponding Virtual Pedestrian is Visualized through a Projector Screen for Comparisons. (A top view of the virtual pedestrian is shown in (a) - (c), while (d) - (f) display the profile view of the virtual pedestrian.) [See the video demonstration of the framework: click here].

The above figure shows the virtual pedestrian in the projector screen from the perspective of a visualization client. Additionally, users can access the visualization from the driving clients. Fig. 30 shows a pedestrian and a driver client visualizing the simulation simultaneously. The pedestrian can see the car controlled by the driver, and the driver can observe the virtual avatar of the pedestrian. In this case, the virtual distance between the pedestrian and the driver was about 16 meters, while the physical distance was 1 meter approximately. Note that they are not required to be close to each other. The actual pedestrian and the driver can physically stay at different locations (e.g., different rooms). The virtual views from both clients are displayed in the figure, as well as a top-down view of the whole scene. Those two screens are not needed for conducting any real experiments. The pedestrian and driver will see all the scenes from the perspective of their own avatar through their headsets.

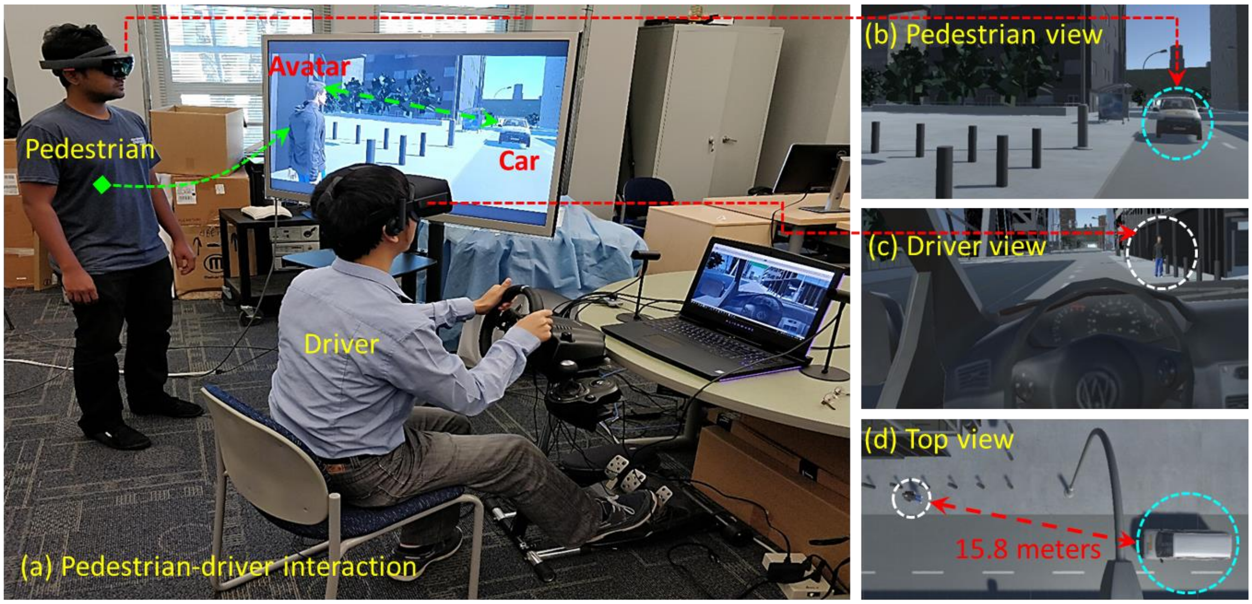


Fig. 30. A Driver and a Pedestrian Participate the Simulation Simultaneously: (a) Pedestrian-driver participation; (b) and (c) show the virtual views from the pedestrian and driver perspectives, respectively; and (d) shows a top view of the virtual scene.

5.3.3 Multiple User-Controlled Pedestrians in the Simulation

Just as multiple user-controlled vehicle clients can connect to the MTS server, multiple user-controlled pedestrian clients can also connect there. PTV Vissim has been used as the MTS server where it receives HoloLens pedestrian data from the intermediate application which in turn receives those data from the HoloLens Unity application. The following figure shows two user-controlled pedestrians inside the Vissim environment.

Fig. 31. Two User-Controlled Pedestrians in Vissim.

Since Vissim now knows about the pedestrians, the other entities like vehicles will also know about the pedestrians and behave appropriately around them.

Now that the pedestrian information is available to the intermediate application, we also want each HoloLens client to see other pedestrians in their environment. To make sure the pedestrians spawn near each other, we used the same Unity application build on two different HoloLens with just the user avatar in two different positions near each other. When these HoloLens clients connect to the Intermediate application, they start sending information regarding these avatar coordinates and they are then relayed back to all other clients like Driving Module, Pedestrian Module, and Visualization Module. The following figure (Fig. 32) shows two user-controlled pedestrians in the simulation environment.

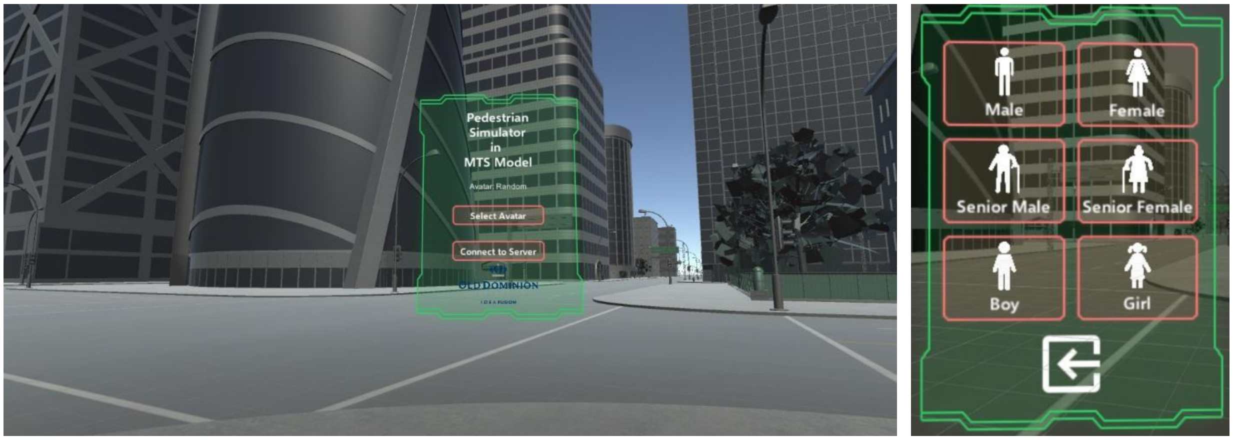
Fig. 32. User-Controlled Pedestrians in the HoloLens Environment (Video Link: https://youtu.be/EFtW7irUS28).

5.3.4 Pedestrian Avatar Movements and Animation

Pedestrians are an important part of this dissertation research and having realistic behavior of the pedestrians is vital. In order to achieve realistic pedestrian movements, animations have been used. These pedestrians receive their position and rotation data from HoloLens. When the position of the HoloLens changes, a vector is calculated to find out the direction of the movement and then the avatar is moved accordingly. When position does not change but the orientation data of the HoloLens changes, then the avatar only rotates it head. Due to avatar constraints, currently the head rotation is fixed along the Y (Up) axis and the rotation amount is clamped between -90^0 to $+90^0$.

5.3.5 GUI in Pedestrian Clients

A critical factor in the pedestrian clients' implementation on the HoloLens headset was the adaptation of the graphical user interface to an AR device. In the desktop applications with traditional displays, user interfaces are commonly implemented in the screen space, and a user UI is usually displayed at a fixed location on a screen. However, screen spaces no longer exist in AR applications as there are no input devices to interact with such spaces. Instead, user interfaces are implemented directly in the world space, and the users can interact with them using their voice or pointing with their finger. In our prototype, we developed a GUI for AR in which users can press buttons and toggles using the "gaze-and-commit" interaction, which is a combination of gazing at an element and then doing an air tap gesture to select it. Specifically, the GUI includes two buttons: one to select the type of avatar that will be visualized for the user and another one to connect to the server. Once the users press the "Connect to Server" button, they will not be able to change their avatar and the GUI will disappear from their environment, giving the users the freedom to walk through the virtual city without obstructions in their view. Fig. 33 shows the view from the pedestrian client once the system is initialized and the GUI panel that allows users to select their avatar. In the current prototype, the user can choose from six different preset avatars like Young Male or Female, Senior Male or Female, Kids as a boy or a girl. After the avatar is selected, the user then connects to the MTS server to experience the simulation through his/her HoloLens. The current prototype does not have an offline recorded version of the AR-PED session.



(a) (b)

Fig. 33. Pedestrian Client View upon Initialization of the System.

5.4 EVALUATION OF THE FRAMEWORK

Several experiments were performed to further evaluate the performance of the framework. All the experiments, except the first one, had five participant users. All of them were male, aged above 24 years old and all of them had previous experience using a HoloLens.

5.4.1 Latency of the Pedestrian Response

Purpose: The purpose of this experiment was to measure the latency between the user's response and the response in the virtual avatar.

Procedure: The latency is defined as the time elapsed between sending the pedestrian data from the user wearing the HoloLens and another client visualizing the corresponding virtual pedestrian.

Results: We measured a best-case latency of 0.06 seconds and a worst-case latency of 0.23 seconds. On average, we measured a latency of 0.15 seconds with a standard deviation of 0.05 seconds. We determined that this latency is small enough for the purposes of our framework. When demonstrating the system, users do not notice a delay between the pedestrian client and the visualization of the virtual pedestrian. This can be appreciated in the video demonstration that is linked to Fig. 29.

5.4.2 Behavior of the Pedestrian

Purpose: The purpose of this experiment was to determine if users behave differently when being in the framework as pedestrians from how they do in the real world.

Procedure: To do this, we measure the time that it takes a person to cross a 14-meter crosswalk both in the real world and while experiencing the virtual world. We performed this experiment with 5 different users. Each user crossed the road 4 times in the real world and 4 in the simulation.

Results: Fig. 34 shows the average time that it took each user to cross the intersection in the framework as well as in the real world. On average, pedestrians took 1.15 seconds more to cross the intersection in the framework than in the real world. This can be because users are wearing a headset to experience the simulation and, thus, they tend to walk relatively slower. However, we considered this difference to be low enough and we determined that users behave similar to the way they do in the real world.

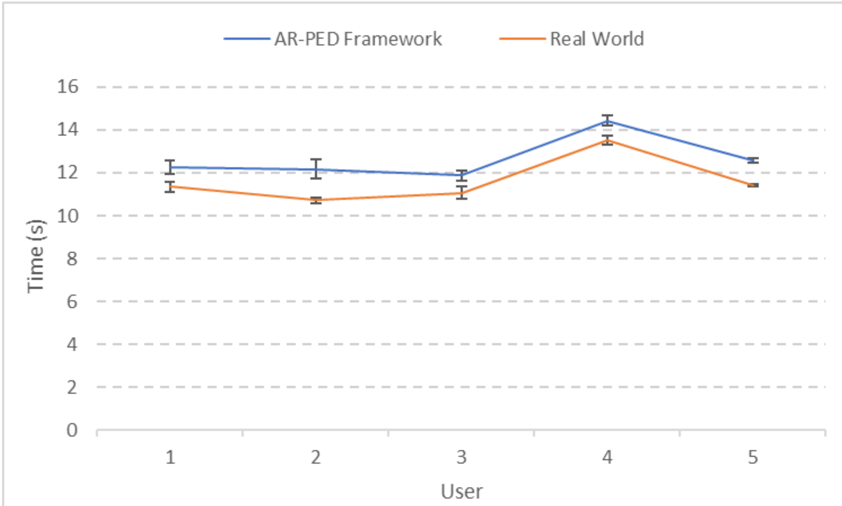


Fig. 34. Average Time (in seconds) that It Takes Each User to Cross a 14-meter Crosswalk in the Framework and in the Real World.

5.4.3 Accuracy of the Locations

Purpose: The purpose of this experiment is to evaluate the accuracy of the locations in the real world and in the virtual world.

Procedure: In the third experiment, we evaluate the accuracy of the virtual locations with respect to the actual location of the user wearing the HoloLens. To do this, we set a path in the real world that users can follow while wearing the headset. In a perfect case scenario, the pedestrian trajectory retrieved in the virtual world will match with our defined path. The path consists of five segments and four 90-degrees turns between them. Segments 1-4 are 3 meters long, while segment 5 is 2 meters. The following figure shows the setup for the experiment.

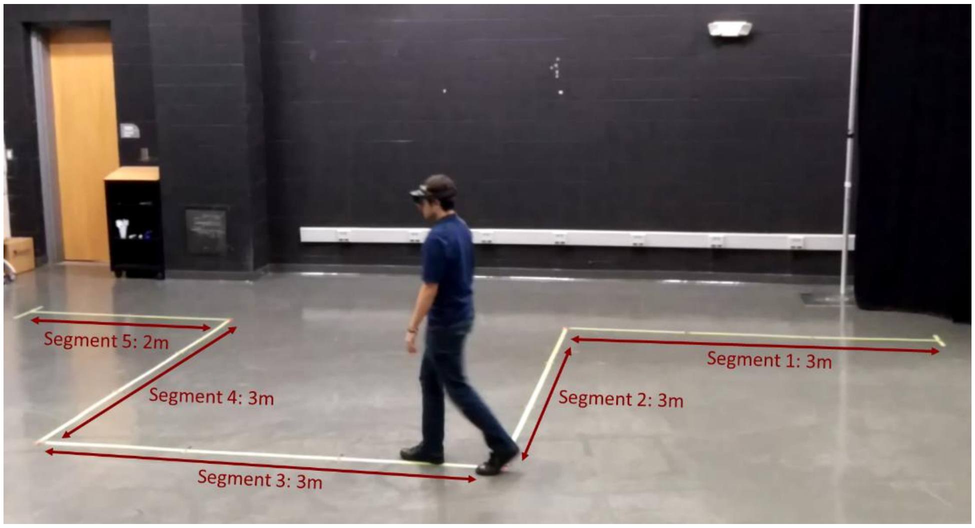


Fig. 35. The User Follows a Physical Path Composed of 5 Segments [The angle between adjacent segments is 90 degrees.].

Results: We performed the experiment with 4 different users, and each user follows the path thrice. Fig. 36 shows the virtual locations retrieved in each of the experiments and compares it with a reference line that corresponds with our physical path (as shown in Fig. 35). For each user, we use a graph showing their virtual locations in three different experiments, as well as a set of box plots showing the deviations per segment. When calculating the deviations, we omitted the small portions of the path in which users made turns, since a user might start performing the turn before reaching the end of a segment, and this behavior should not be counted as an error. Based on the experiment results, the virtual trajectories retrieved by the framework were not completely perfect, but they closely match the path defined in the real world. Additionally, it is observed that the deviation tends to increase in the latter segments of the path. This might be due to the calibration issues of the HoloLens headset. On average, we measured an absolute error of 0.11, 0.07, 0.13 and 0.15 meters for users 1- 4, respectively. Different factors can contribute to such errors. It has been reported that the location sensors in the HoloLens headset are more accurate at low movement speeds and in bright conditions [120]. The walking pattern of each user (speed, movement of their head, rotations, etc.) also play a big role in the estimation of the user location. On average we reported low deviations between the user's location and its virtual location. Since the final virtual paths closely match with the physical path in a large space, we determine that the virtual locations are accurate enough for the purpose of our framework.

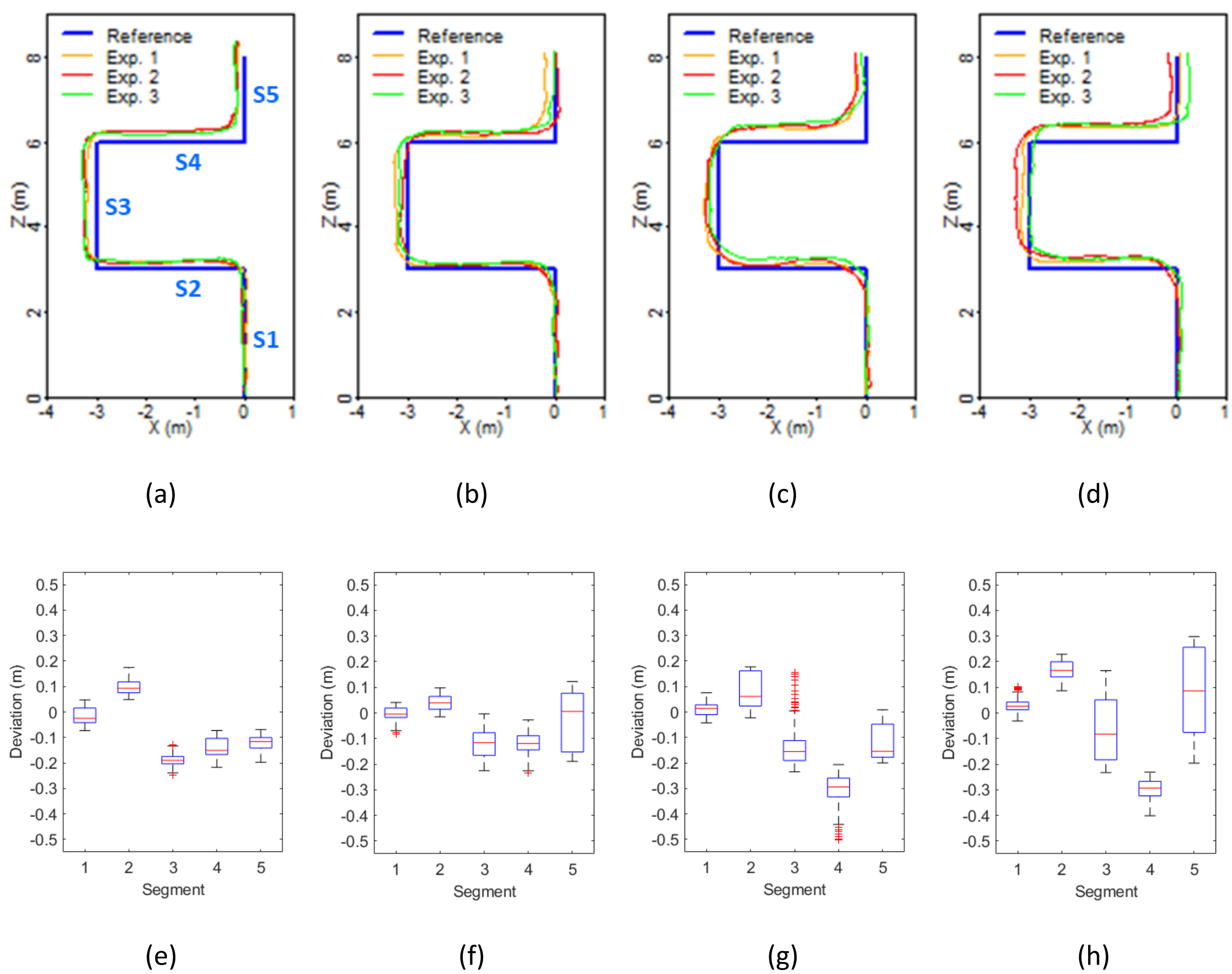


Fig. 36. Trajectories for Each User in Three Different Experiments (a-d) and Their Corresponding Box Plots of Deviations for Segments S1 to S5 (e-h).

5.5 SUMMARY

This chapter elaborated on the architecture and implementation of the user-controlled pedestrian in the simulation. The pedestrians can be controlled using the AR device Microsoft HoloLens or the keyboard arrow keys. The usefulness of such a platform is immense. Researchers can choreograph different scenarios in the simulation environment to emulate the desired behavior. Multiple HoloLens devices can connect to the same environment to experience the simulation together. The users can be in the same space or different or both. One limitation of this prototype still remains which is when users are in the same real-world space, their relative position and orientation in the real world might not reflect the same in the simulated world since the simulated worlds are not properly aligned. To ensure the alignment using anchor sharing is suggested by HoloLens but anchors sharing worked better for small Holographic objects, not the for the large environment like this protype has. Because of that, remote users were mostly tested in this framework. The next chapter describes how the simulation can be accessed via online browser and implementation of a user chat system in the simulation environment.

CHAPTER 6

USER CHAT SYSTEM & ONLINE SERVER

This chapter discusses two additional features added to the simulation: a web-based server to allow access from the web browser, and a user chat system.

6.1 ONLINE SERVER SYSTEM

The framework and the prototype developed in this dissertation can be used for various purposes. The support of Virtual Reality and Augmented Reality in the simulation makes the framework quite useful and powerful. To disseminate the power of the developed simulation and to provide accessibility to everyone, it is important to have the system available online. This whole system has two major components: one is the server and the other is the client. One server can handle multiple clients. The clients connect to the server using the Internet. There are also two components of the server: the MTS application and the intermediate application. Currently, the server is hosted on a laptop computer. When the intermediate application is running, the clients can connect to the server. This system is enough for the proof of concept. Now, to make this simulation platform accessible to people from a different local network, we need to create a website and a dedicated server to host all the necessary components. The components required for the online system are as follows:

Web Server: This would be an online server that hosts the webpage for a user to access via an Internet browser. This server hosts MTS packages, client modules created from Unity, intermediate application that accepts connections from MTS packages and the client modules and so on.

HTML and PHP Scripts: HTML script was used to create the webpage. A PHP script allowed executing the intermediate application on the server from a browser. It also allows the user to download a client module from the server to the user's computer.

A test web server was created using an open-source cross-platform web server solution stack called XAMPP, developed by Apache Friends. It has an Apache server and MySQL database which uses personal computers to make it a web server and turn it to a host. Using the local IP address 127.0.0.1 from the web browser of the same computer, connections can be made to the XAMPP server. To access the server from a remote machine, a port forwarding and tunneling tool called ngrok has been used. This tool bypasses the current firewall of the computer and allows access to the port the XAMPP server is running. The following figure (Fig. 37) shows a webpage on a remote machine that was created for this project.

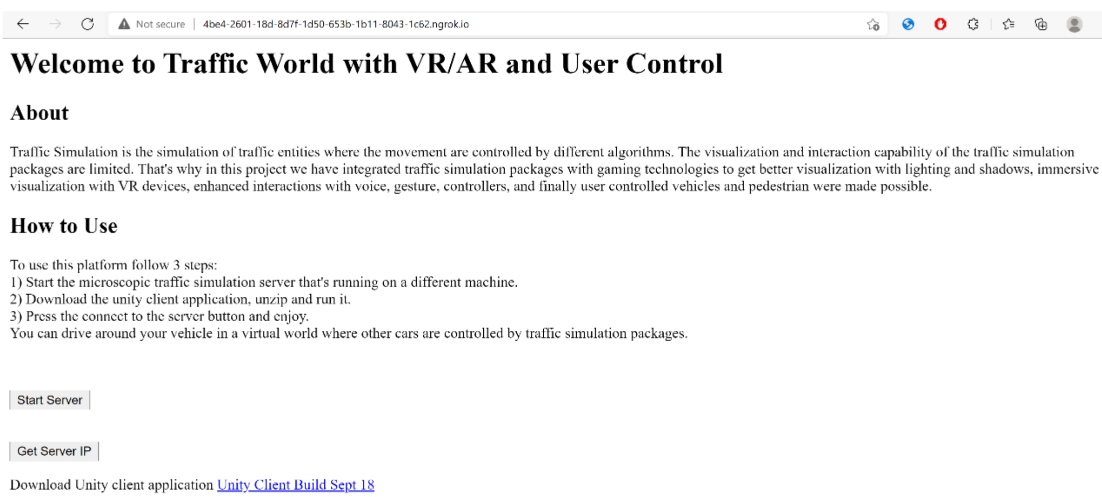


Fig. 37. A Webpage allowing access to the Simulation from the Browser.

This webpage is still under development and the number of features available are still limited. Currently, a brief introduction about the simulation is provided. Then, instructions on how to use this page are given. The user can download an .exe file of the Unity build to run the simulation client. The user also can start the MTS server from this webpage. Once the MTS server is running and the user gets the Server IP address, that address can be used on the client application to connect to the MTS server and have a live experience of the simulation.

6.2 CREATING A USER CHAT SYSTEM

This simulation framework also supports a user chat system. The chat system is a way of communicating with each other in the simulation environment via the internet. The benefits of having a chat system are manifold.

- First, participating users will be able to communicate with each other without getting out of the simulation.
- Second, one person that has an administrative role can dictate the interactions that the participants have to perform via this system.
- Third, people from different locations will be able to coordinate their actions in the simulation and so on.

6.2.1 Chat System Architecture:

The user chat system does not use the built-in TCP/IP network of our framework. Instead, a separate Unity asset called Photon Chat [121] is being used. Unity's previous multiplayer networking system called Unity Networking or UNet has been deprecated. Currently, a work of a new solution for networked games is in progress. This new system uses Multiplay [122], a dedicated game server enterprise to host the games instead of Unity Servers. This new system also supports Vivox [123], a dedicated hosted service for voice and text communication in games. The

Photon Chat used in this simulation has its own cloud server. Once the user starts the chat system, they get the option to input their username. After the “Connect Button” as shown in Fig. 38 has been pressed, the application subscribes to a default channel.

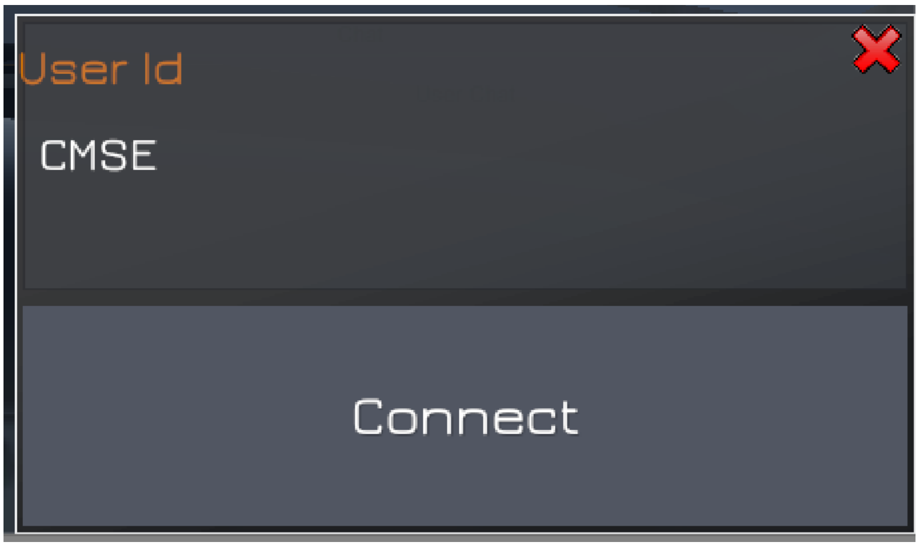


Fig. 38. Chat User Input Field.

In this prototype, there is only one public channel, and all users connect to the same one. The following figure shows the system architecture of the Photon Chat system.

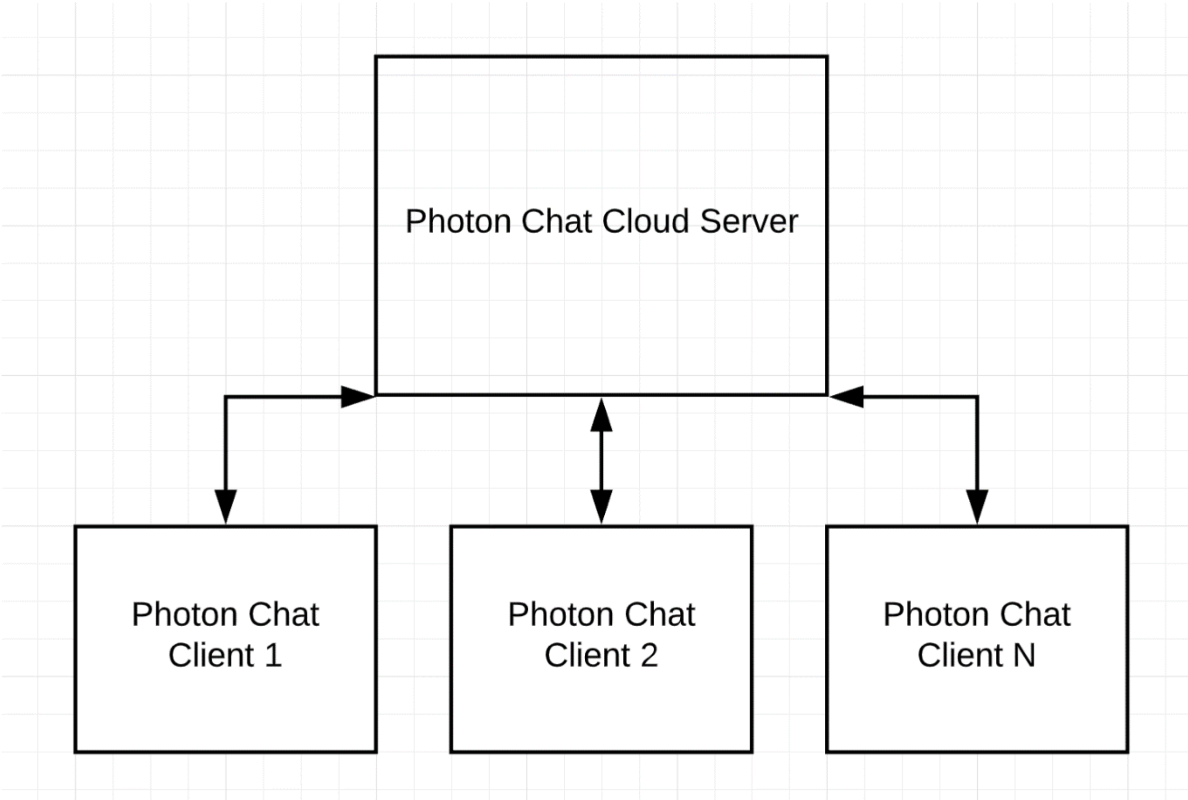


Fig. 39. Photon Chat System Architecture.

As Fig. 39 shows above, one cloud server handles all communication between the chat clients. The server creates the channel to which all clients subscribe and publish messages. Once a client publishes any messages to the channel, the server broadcasts those messages to all the clients that subscribed to that channel. If another client connects later, the server will broadcast all the previous messages to the new client if the server was properly configured to do so. Fig. 40 shows the chat system UI in this simulation.

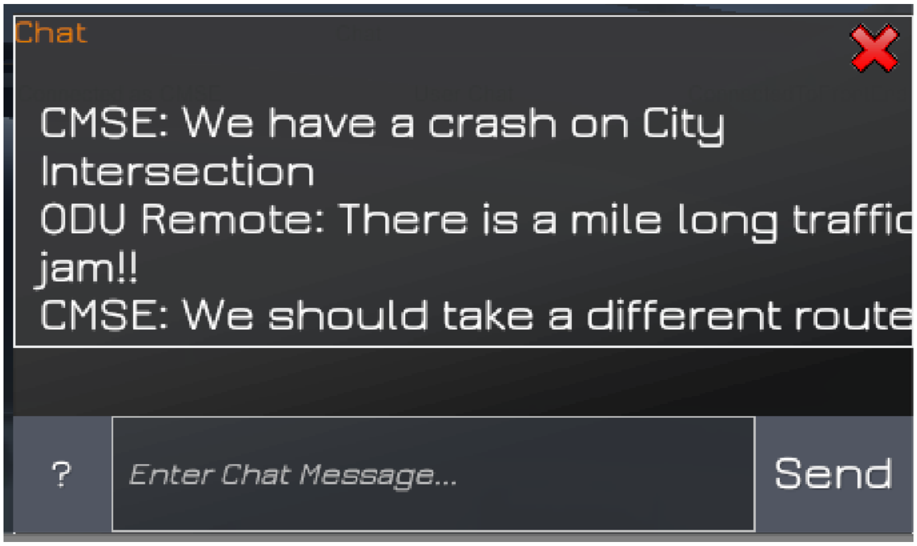


Fig. 40. A User Chat System (Video Link: https://youtu.be/1flack7oehI).

The chat panel shows all the messages published to the channel. The right “Cross/ X” is used to cancel/ close the chat system. The “?” is the help button. The input field in the middle is for typing messages. After finishing typing, the user can publish messages by pressing the “Send” button on the UI or pressing the ENTER/ RETURN key on the keyboard.

6.2.2 Chat System Use Cases:

The use cases for such systems would be many like:

Studying Driver Distraction: Many reports and research show that texting while driving is a major cause of accidents now a days. Cellphones have become a common personal item and people are too used to driving with them. People send and respond to text messages even while they are behind the wheel. The split-second distraction can often lead to an accident. The user chat system in this simulation environment can be used as a tool to perform user studies on focus groups to measure the effects of texting and driving.

Studying Pedestrian Distraction: Texting can cause pedestrians to walk differently than usual. This simulation environment can be a good platform to perform some user studies to see how a pedestrian reacts to texting and walking.

Chatting/ Texting Effect on Different Age Groups: It is also known that different people can react to similar events in different ways. Age is a large factor when it comes to split-second reactions, so it is also important to find out how age can affect texting and driving. We can perform a study on different age groups and find out how texting affects their driving or walking as a pedestrian.

Merits/ Demerits of Automated Help Services: Modern vehicles are often equipped with automated services that provide help or ques for the driver. These vehicles often have smart displays on the dashboard or on the windshield glass. The chat UI of this simulation can be placed on different positions of the vehicles to find out how they affect the driver.

CHAPTER 7

CONCLUSION

The simulation framework developed in this dissertation opens the door for newer and safer means to perform various forms of research studies. The integrated MTS and multiple driving simulators in virtual reality not only enable researchers to perform a user study in an immersive manner but also allows engineers to effectively disseminate their ideas to stakeholders and the public. The distributed nature of the framework provides flexibility for participation from geographically dispersed locations. The inclusion of VR headsets in this framework allows users to experience immersive simulation with low-cost consumer level VR devices. The inclusion of AR-based pedestrians in the simulation is also a novel and unique approach. Having input from real pedestrians at a one-to-one scale in the simulation increase the realism of the simulation. Above all, having the capability of VR drivers and AR pedestrians in the same simulation environment where the majority of the simulated vehicles are controlled by traffic simulation packages is a significant advancement of the state-of-the-art microscopic traffic simulation.

It should be noted that the prototype developed in this dissertation is still a proof of concept. Several user studies performed in this dissertation showed that using VR and AR in conjunction with MTS packages is a viable and useful option for researchers and engineers anywhere. Utilizing the distributed configuration, the projects developed using this framework are accessible from anywhere in the world. Any non-technical personnel can get a better understanding of the project because of the immersive nature of the simulation. In addition to that, the intuitive user interface developed in this prototype has a lot of additional features for ease of use.

One concern of the VR HMD based system is the simulator sickness. This can be mitigated by using a well calibrated motion platform which will provide users the motion that their body expects when they drive a car. In the current project, the developers tested the prototype without experiencing any sort of sickness, but extensive study with user participation will be required to test the validity of this system. On the other hand, Microsoft HoloLens does not cause any motion sickness for the user, but the weight of the original HoloLens, which was used in this dissertation research, could cause some discomfort for the user. A lightweight alternative like HoloLens 2 could be a good solution to that problem. Moreover, due to the online nature of the framework there could be some delay of transmission depending on the network speed, and computer configuration. These issues could also be mitigated with better devices and networks.

There are some aspects of the presented dissertation research that can be improved and/or modified in the future. The current framework uses TCP/IP network communication protocol. Other communication protocols like UDP could be explored for improving the performance. The benefit of using VR in the simulation has already been proven, but the realism of the driver experience could be further improved by using a motion platform with real-time feedback. This dissertation also introduced AR-based pedestrians in the simulation and explained the potential benefits of having multiple user-controlled pedestrians. One area that is still unexplored is the AR world alignment when two or more HoloLens devices launch the same simulation. This could potentially be solved by reloading the environment of the rest of the HoloLens devices based on the first HoloLens that connected to the server and aligning their world.

REFERENCES

[1] Wikipedia. "History of Transport." https://en.wikipedia.org/wiki/History_of_transport.

[2] "History of Domesticated Animals." [Online]. Available: https://storymaps.arcgis.com/stories/893c422c13424a089b781564e9f69735.

[3] G. Scholtz, "Scarab beetles at the interface of wheel invention in nature and culture?," *Contributions to Zoology*, vol. 77, no. 3, pp. 139-148, 2008.

[4] W. F. Edgerton, "Ancient Egyptian ships and shipping," *The American Journal of Semitic Languages and Literatures*, vol. 39, no. 2, pp. 109-135, 1923.

[5] M. Avi-Yonah, "The development of the Roman road system in Palestine," *Israel Exploration Journal*, vol. 1, no. 1, pp. 54-60, 1950.

[6] T. Lambert. "A Brief History of Transportation." http://www.localhistories.org/transport.html.

[7] H. W. Dickinson, *A short history of the steam engine*. Cambridge University Press, 2011.

[8] R. L. Ransom, "Canals and development: A discussion of the issues," *The American Economic Review*, vol. 54, no. 3, pp. 365-376, 1964.

[9] K. Knothe and F. Böhm, "History of stability of railway and road vehicles," *Vehicle System Dynamics*, vol. 31, no. 5-6, pp. 283-323, 1999.

[10] K. Williams, *Cars: Analysis, history, cases*. Berghahn Books, 1994.

[11] D. V. Herlihy, *Bicycle: the history*. Yale University Press, 2004.

[12] E. K. Chatterton, *Steamships and Their Story*. Cassell, Limited, 1910.

[13] G. Carter, R. Couch, and D. J. O'Brien, "The evolution of air transport systems: a pictorial review," *The Journal of emergency medicine*, vol. 6, no. 6, pp. 499-504, 1988.

[14] K.-H. Dietsche and D. Kuhlgatz, "History of the automobile," in *Fundamentals of Automotive and Engine Technology*: Springer, 2014, pp. 1-7.

[15] A. S. O. Yu, L. L. C. Silva, C. L. Chu, P. T. S. Nascimento, and A. S. Camargo, "Electric vehicles: Struggles in creating a market," in *2011 Proceedings of PICMET'11: Technology Management in the Energy Smart World (PICMET)*, 2011: IEEE, pp. 1-13.

[16] Wikipedia. "Modeling and simulation." https://en.wikipedia.org/wiki/Modeling_and_simulation.

[17] J. A. Sokolowski and C. M. Banks, *Modeling and simulation fundamentals: theoretical underpinnings and practical domains*. John Wiley & Sons, 2010.

[18] J. A. Sokolowski and C. M. Banks, *Principles of modeling and simulation: a multidisciplinary approach*. John Wiley & Sons, 2011.

[19] A. B. Hill, "A pilot in the first world war," *British Medical Journal (Clinical research ed.)*, vol. 287, no. 6409, p. 1947, 1983.

[20] Mathworks. "Simulink." https://www.mathworks.com/products/simulink.html.

[21] U. Wilensky. " NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL." http://ccl.northwestern.edu/netlogo/.

[22] "PTV Vissim." http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/.

[23] V. Systems. "Vensim." https://vensim.com/.

[24] W. D. Kelton, *Simulation with ARENA*. McGraw-hill, 2002.

[25] W. Bles and R. Hosman, "Desdemona-Advanced disorientation trainer and (sustained-G) flight simulator," in *Modeling and Simulation Technologies Conference*, 2000, p. 4176.

[26] J. Barceló, *Fundamentals of traffic simulation*. Springer, 2010.

[27] B. Ciuffo, V. Punzo, and M. Montanino, "Thirty years of Gipps' car-following model: Applications, developments, and new features," *Transportation research record*, vol. 2315, no. 1, pp. 89-99, 2012.

[28] S. Siuhi and M. S. Kaseko, "Parametric study of stimulus-response behavior for car-following models," 2010.

[29] L. Pariota and G. N. Bifulco, "Experimental evidence supporting simpler Action Point paradigms for car-following," *Transportation research part F: traffic psychology and behaviour*, vol. 35, pp. 1-15, 2015.

[30] G. H. Bham and R. F. Benekohal, "A high fidelity traffic simulation model based on cellular automata and car-following concepts," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 1, pp. 1-32, 2004.

[31] B. Cao and Z. Yang, "Car-following models study progress," in *2009 Second International Symposium on Knowledge Acquisition and Modeling*, 2009, vol. 3: IEEE, pp. 190-193.

[32] S. P. H. Femke van Wageningen-Kessels, Kees Vuik, and Hans van Lint, "Traffic and Transportation Simulation: Looking back and looking ahead:Celebrating 50 years of traffic flow theory, a workshop," *Transportation Research Circulars*, pp. 1-16, 2015.

[33] Portrait Software International Ltd. "Quadstone Paramics." http://www.paramics-online.com/.

[34] CITILABS. "Cube Dynasim." https://www.citilabs.com/software/cube/cube-dynasim/.

[35] "aimsun." https://www.aimsun.com/.

[36] E. Foundation. "SUMO- Simulation of Urban MObility." https://sumo.dlr.de/userdoc/index.html.

[37] J. E. Miller, "Basic concepts of kinematic-wave models," US Geological Survey, 2330-7102, 1984.

[38] Y.-q. Jiang, P. Zhang, S. Wong, and R.-x. Liu, "A higher-order macroscopic model for pedestrian flows," *Physica A: Statistical Mechanics and its Applications,* vol. 389, no. 21, pp. 4623-4635, 2010.

[39] CITILABS. "Cube Voyager." https://www.citilabs.com/software/cube/cube-voyager/.

[40] "PTV Visum." http://vision-traffic.ptvgroup.com/en-us/products/ptv-visum/.

[41] S. P. Hoogendoorn and P. H. Bovy, "Gas–Kinetic Model for Multilane Heterogeneous Traffic Flow," *Transportation research record,* vol. 1678, no. 1, pp. 150-159, 1999.

[42] X. Chen, L. Li, and Y. Zhang, "A Markov model for headway/spacing distribution of road traffic," *IEEE Transactions on Intelligent Transportation Systems,* vol. 11, no. 4, pp. 773-785, 2010.

[43] A. Ferrara, S. Sacone, and S. Siri, "Microscopic and mesoscopic traffic models," in *Freeway traffic modelling and control*: Springer, 2018, pp. 113-143.

[44] CITILABS. "Cube Avenue." https://www.citilabs.com/software/cube/cube-avenue/.

[45] S. I. J. Chien, D. G. Goulias, S. Yahalom, and S. M. Chowdhury, "Simulation-based estimates of delays at freeway work zones," *Journal of Advanced Transportation*, vol. 36, no. 2, pp. 131-156, 2002.

[46] M. Barth and K. Boriboonsomsin, "Energy and emissions impacts of a freeway-based dynamic eco-driving system," *Transportation Research Part D: Transport and Environment,* vol. 14, no. 6, pp. 400-410, 2009.

[47] Z. Yue, W. Jiangyan, D. Huang, and S. Shengyang, "Pedestrian simulation modeling for World Expo 2010 shanghai," *Journal of Transportation Systems Engineering and Information Technology,* vol. 9, no. 2, pp. 141-146, 2009.

[48] M. Monteleone, N. Veraart, S. Chough, and W. Aviles, "Pedestrian Simulation Modeling Study for World Trade Center Memorial," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2073, pp. 49-57, 2008.

[49] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, "Simulation of pedestrian dynamics using a two-dimensional cellular automaton," *Physica A: Statistical Mechanics and its Applications,* vol. 295, no. 3, pp. 507-525, 2001.

[50] J. Dijkstra, H. J. Timmermans, and A. Jessurun, "A multi-agent cellular automata system for visualising simulated pedestrian activity," in *Theory and Practical Issues on Cellular Automata*: Springer, 2001, pp. 29-36.

[51] C. Lee, B. Hellinga, and F. Saccomanno, "Evaluation of variable speed limits to improve traffic safety," *Transportation research part C: emerging technologies,* vol. 14, no. 3, pp. 213-228, 2006.

[52] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Transactions on Intelligent Transportation Systems,* vol. 13, no. 1, pp. 81-90, 2012.

[53] S. Le Vine, A. Zolfaghari, and J. Polak, "Autonomous cars: The tension between occupant experience and intersection capacity," *Transportation Research Part C: Emerging Technologies,* vol. 52, pp. 1-14, 2015.

[54] L. Bloomberg and J. Dale, "Comparison of VISSIM and CORSIM traffic simulation models on a congested network," *Transportation Research Record: Journal of the Transportation Research Board,* no. 1727, pp. 52-60, 2000.

[55] H. Sbayti and D. Roden, "Best practices in the use of micro simulation models," *Prepared for: American Association of State Highway and Transportation Officials (AASHTO) Standing Committee on Planning. AECOM, Arlington, VA,* 2010.

[56] P. Hidas, "Modelling lane changing and merging in microscopic traffic simulation," *Transportation Research Part C: Emerging Technologies,* vol. 10, no. 5-6, pp. 351-371, 2002.

[57] E. Chan, A. K. Pradhan, A. Pollatsek, M. A. Knodler, and D. L. Fisher, "Are driving simulators effective tools for evaluating novice drivers' hazard anticipation, speed management, and attention maintenance skills?," *Transportation research part F: traffic psychology and behaviour,* vol. 13, no. 5, pp. 343-353, 2010.

[58] L. J. Findley and J. Fabrizio, "Driving Simulator Performance in Patients with Sleep Apnea1-3," *Am Rev Respir Dis,* vol. 140, pp. 529-530, 1989.

[59] J. M. Weiler *et al.*, "Effects of fexofenadine, diphenhydramine, and alcohol on driving performance: a randomized, placebo-controlled trial in the Iowa driving simulator," *Annals of Internal Medicine,* vol. 132, no. 5, pp. 354-363, 2000.

[60] G. Rumschlag, T. Palumbo, A. Martin, D. Head, R. George, and R. L. Commissaris, "The effects of texting on driving performance in a driving simulator: The influence of driver age," *Accident Analysis & Prevention,* vol. 74, pp. 145-149, 2015.

[61] H. Cai, Y. Lin, and R. R. Mourant, "Study on driver emotion in driver-vehicle-environment systems using multiple networked driving simulators," in *Proceedings of the Driving Simulation Conference North America (DSC'07)*, 2007.

[62] Y. Hou *et al.*, "An integrated traffic-driving simulation framework: Design, implementation, and validation," *Transportation Research Part C: Emerging Technologies,* vol. 45, pp. 138-153, 2014.

[63] V. Punzo and B. Ciuffo, "Integration of driving and traffic simulation: Issues and first solutions," *IEEE Transactions on Intelligent Transportation Systems,* vol. 12, no. 2, pp. 354-363, 2011.

[64] T. N. That and J. Casas, "An integrated framework combining a traffic simulator and a driving simulator," *Procedia-Social and Behavioral Sciences,* vol. 20, pp. 648-655, 2011.

[65] E. B. Lieberman and B. Andrews, "The role of interactive graphics when applying traffic simulation models," in *1990 Winter Simulation Conference Proceedings*, 1990: IEEE, pp. 753-758.

[66] N. W. Garrick, P. Miniutti, M. Westa, J. Luo, and M. Bishop, "Effective visualization techniques for the public presentation of transportation projects," in "Report No.: NETC 48," University of Connecticut, 2005.

[67] L. E. Owen, Y. Zhang, L. Rao, and G. McHale, "Street and traffic simulation: traffic flow simulation using CORSIM," in *Proceedings of the 32nd Conference on Winter Simulation*, 2000: Society for Computer Simulation International, pp. 1143-1147.

[68] P. N. Seneviratne, "Simulation and Animation Model for Planning and Designing Transit Terminals," IDEA Programs, Transportation Research Board, 2005.

[69] G. Ayres and R. Mehmood, "On discovering road traffic information using virtual reality simulations," in *Computer Modelling and Simulation, 2009. UKSIM'09. 11th International Conference on*, 2009: IEEE, pp. 411-416.

[70] S. A. Boxill, "An evaluation of 3-d traffic simulation modeling capabilities," in "Report No.: SWUTC/07/167621-1," Texas Southern University, Houston, Texas, 2007.

[71] G. Papageorgiou, P. Damianou, A. Pitsillides, T. Aphamis, D. Charalambous, and P. Ioannou, "Modelling and Simulation of Transportation Systems: a Scenario Planning Approach," *Automatika: Journal for Control, Measurement, Electronics, Computing & Communications,* vol. 50, no. 1-2, pp. 39-50, 2009.

[72] H. Nomden, D. Greenwood, M. Horne, T. Muizelaar, and E. M. Thompson, "Traffic simulation in 3D world," *Northumbria Working Paper Series: Interdisciplinary Studies in the Built and Virtual Environment,* vol. 2, no. 1, pp. 26-38, 2009.

[73] J. Sun, Y. Yang, and H. Wang, "Development and application of a simulation-enhanced platform for pedestrian crossing behaviors experiment," presented at the 90th Annual Meeting of the Transportation Research Board, Washington, DC, 2011.

[74] B. Zhou, J. Blum, and A. Eskandarian, "Virtual reality visualization of microscopic traffic simulations," *Transportation Research Record: Journal of the Transportation Research Board,* no. 1937, pp. 159-166, 2005.

[75] W. Chun, C. Ge, L. Yanyan, and M. Horne, "Virtual-reality based integrated traffic simulation for urban planning," in *Computer Science and Software Engineering, 2008 International Conference on*, 2008, vol. 2: IEEE, pp. 1137-1140.

[76] L. Hermant and R. Herrmann, "Bridge City Railway Station Project: ensuring optimum levels of service through pedestrian simulation incorporating Virtual Reality (VR): urban and rural engineering," *Civil Engineering= Siviele Ingenieurswese,* vol. 19, no. 9, pp. 10-15, 1977.

[77] M. Hu, "A high-fidelity three-dimensional simulation method for evaluating passenger flow organization and facility layout at metro stations," *simulation,* p. 0037549717715107, 2017.

[78] I. Vladisavljevic, J. M. Cooper, P. T. Martin, and D. L. Strayer, "Importance of integrating driving and traffic simulations: case study of impact of cell phone drivers on traffic flow," 2009.

[79] M. Jin and S.-H. Lam, "A virtual-reality based integrated driving-traffic simulation system to study the impacts of intelligent transportation system (ITS)," in *Cyberworlds, 2003. Proceedings. 2003 International Conference on*, 2003: IEEE, pp. 158-165.

[80] M. Lindorfer, C. F. Mecklenbräuker, and G. Ostermayer, "Modeling the Imperfect Driver: Incorporating Human Factors in a Microscopic Traffic Model," *IEEE Transactions on Intelligent Transportation Systems,* 2017.

[81] J. J. Olstam, J. Lundgren, M. Adlers, and P. Matstoms, "A framework for simulation of surrounding vehicles in driving simulators," *ACM Transactions on Modeling and Computer Simulation (TOMACS),* vol. 18, no. 3, p. 9, 2008.

[82] J. Sun, D. Niu, S. Chen, and K. Li, "Development and investigation of a dynamic eco-driving speed guidance strategy for signalized highway traffic," 2013.

[83] J. Sun, Z. Ma, T. Li, and D. Niu, "Development and application of an integrated traffic simulation and multi-driving simulators," *Simulation Modelling Practice and Theory,* vol. 59, pp. 1-17, 2015.

[84] T. Lorentzen, Y. Kobayashi, and Y. Ito, "Virtual reality driving simulation: Integrating infrastructure plans traffic models and driving behaviors," in *2009 Annual Meeting and Exposition of the Intelligent Transportation Society of America*, 2009.

[85] A. H. Behzadan and V. R. Kamat, "Simulation and visualization of traffic operations in augmented reality for improved planning and design of road construction projects," in *Simulation Conference, 2008. WSC 2008. Winter*, 2008: IEEE, pp. 2447-2454.

[86] S. M. Taheri, K. Matsushita, and M. Sasaki, "Virtual reality driving simulation for measuring driver behavior and characteristics," *Journal of transportation technologies,* vol. 7, no. 2, pp. e123-e123, 2017.

[87] S. Ropelato, F. Zünd, S. Magnenat, M. Menozzi, and R. Sumner, "Adaptive Tutoring on a Virtual Reality Driving Simulator," in *1st Workshop on Artificial Intelligence Meets Virtual and Augmented Worlds (AIVRAR) in conjunction with SIGGRAPH Asia 2017*, 2017: ETH Zurich.

[88] T. Pan, Y. Zhang, and T. T. Yang, "Inquiring the Application of Mixed Reality in the Traffic Safety Education," 2017.

[89] B. Blissing, F. Bruzelius, and O. Eriksson, "Driver behavior in mixed and virtual reality–a comparative study," *Transportation Research Part F: Traffic Psychology and Behaviour,* 2017.

[90] S. Deb, D. W. Carruth, R. Sween, L. Strawderman, and T. M. Garrison, "Efficacy of virtual reality in pedestrian safety research," *Applied ergonomics,* vol. 65, pp. 449-460, 2017.

[91] D. Sportillo, A. Paljic, M. Boukhris, P. Fuchs, L. Ojeda, and V. Roussarie, "An immersive Virtual Reality system for semi-autonomous driving simulation: a comparison between realistic and 6-DoF controller-based interaction," in *Proceedings of the 9th International Conference on Computer and Automation Engineering*, 2017: ACM, pp. 6-10.

[92] A. L. Erath, T. Maheshwari, M. Joos, J. Kupferschmid, and M. A. van Eggermond, "Visualizing Transport Futures: the potential of integrating procedural 3d modelling and traffic micro-simulation in Virtual Reality applications," in *96th Transportation Research Board Annual Meeting*, 2017: FCL, Singapore ETH Centre.

[93] P. Rahimian *et al.*, "Using a virtual environment to study the impact of sending traffic alerts to texting pedestrians," in *Virtual Reality (VR), 2016 IEEE*, 2016: IEEE, pp. 141-149.

[94] D. C. Schwebel, J. Severson, Y. He, and L. A. McClure, "Virtual reality by mobile smartphone: improving child pedestrian safety," *Inj Prev*, vol. 23, no. 5, p. 357, Oct 2017, doi: 10.1136/injuryprev-2016-042168.

[95] S. E. Banducci *et al.*, "The effects of cell phone and text message conversations on simulated street crossing," *Human factors*, vol. 58, no. 1, pp. 150-162, 2016. [Online]. Available: http://journals.sagepub.com/doi/pdf/10.1177/0018720815609501.

[96] J. McComas, M. MacKay, and J. Pivik, "Effectiveness of virtual reality for teaching pedestrian safety," *CyberPsychology & Behavior*, vol. 5, no. 3, pp. 185-190, 2002.

[97] J. A. Thomson, A. K. Tolmie, H. C. Foot, K. M. Whelan, P. Sarvary, and S. Morrison, "Influence of virtual reality training on the roadside crossing judgments of child pedestrians," *Journal of Experimental Psychology: Applied*, vol. 11, no. 3, p. 175, 2005.

[98] D. C. Schwebel, J. Gaines, and J. Severson, "Validation of virtual reality as a tool to understand and prevent child pedestrian injury," *Accident Analysis & Prevention*, vol. 40, no. 4, pp. 1394-1400, 2008.

[99] J. Orlosky, M. Weber, Y. Gu, D. Sonntag, and S. Sosnovsky, "An interactive pedestrian environment simulator for cognitive monitoring and evaluation," in *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*, 2015: ACM, pp. 57-60.

[100] M. Hartmann, M. Viehweger, W. Desmet, M. Stolz, and D. Watzenig, "“Pedestrian in the Loop”: An approach using virtual reality," 2017.

[101] M. Hartmann, M. Viehweger, W. Desmet, M. Stolz, and D. Watzenig, "" Pedestrian in the Loop": An approach using augmented reality," 2018.

[102] B. Farooq, E. Cherchi, and A. Sobhani, "Virtual Immersive Reality for Stated Preference Travel Behavior Experiments: A Case Study of Autonomous Vehicles on Urban Roads," *Transportation Research Record*, p. 0361198118776810, 2018.

[103] Z. Patterson, J. M. Darbani, A. Rezaei, J. Zacharias, and A. Yazdizadeh, "Comparing text-only and virtual reality discrete choice experiments of neighbourhood choice," *Landscape and Urban Planning*, vol. 157, pp. 63-74, 2017.

[104] H. Creagh, "Cave automatic virtual environment," in *Electrical Insulation Conference and Electrical Manufacturing & Coil Winding Technology Conference, 2003. Proceedings*, 2003: IEEE, pp. 499-504.

[105] A. Dommès, T. Le Lay, F. Vienne, N.-T. Dang, A. P. Beaudoin, and M. C. Do, "Towards an explanation of age-related difficulties in crossing a two-way street," *Accident Analysis & Prevention*, vol. 85, pp. 229-238, 2015.

[106] B. G. Witmer and M. J. Singer, "Measuring presence in virtual environments: A presence questionnaire," *Presence*, vol. 7, no. 3, pp. 225-240, 1998.

[107] G. Simpson, L. Johnston, and M. Richardson, "An investigation of road crossing in a virtual environment," *Accident Analysis & Prevention*, vol. 35, no. 5, pp. 787-796, 2003.

[108] B. A. Morrongiello and M. Corbett, "Using a virtual environment to study child pedestrian behaviours: a comparison of parents' expectations and children's street crossing behaviour," *Injury prevention*, vol. 21, no. 5, pp. 291-295, 2015. [Online]. Available: http://injuryprevention.bmj.com/content/injuryprev/21/5/291.full.pdf.

[109] B. A. Morrongiello, M. Corbett, M. Milanovic, and J. Beer, "Using a virtual environment to examine how children cross streets: Advancing our understanding of how injury risk arises," *Journal of pediatric psychology*, vol. 41, no. 2, pp. 265-275, 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4884905/pdf/jsv078.pdf.

[110] H. Yang, Y. Shen, M. Hassan, D. Perez, and J. Shull, "A Framework for Interactive M3 Visualization of Microscopic Traffic Simulation," *Transportation Research Record: Journal of the Transportation Research Board*, 2018 2018, doi: https://doi.org/10.1177/0361198118787088.

[111] G. D. Cameron and G. I. Duncan, "PARAMICS—Parallel microscopic simulation of road traffic," *The Journal of Supercomputing*, vol. 10, no. 1, pp. 25-53, 1996.

[112] B. Philips and T. Morton, "Making Driving Simulators More Useful for Behavioral Research-Simulator Characteristics Comparison and Model-Based Transformation: Summary Report," 2015.

[113] "iGroup (2008). Igroup presence questionnaire (IPQ). Website: http://www.igroup.org/pq/ipq/download.php#English.".

[114] M. Hassan, D. Perez, H. Yang, and Y. Shen, "Distributed Microscopic Traffic Simulation with Human in the Loop and Virtual Reality Technologies," Advances in Engineering Software, Vol 154, 2021.

[115] D. Perez, M. Hasan, Y. Shen, and H. Yang, "AR-PED: An Augmented Reality Enabled Pedestrian-in-the-Loop Simulation.", *Simulation Modeling Practice and Theory*, Vol 94, p237-249, 2019.

[116] S. Nichols and H. Patel, "Health and safety implications of virtual reality: a review of empirical evidence," *Applied ergonomics*, vol. 33, no. 3, pp. 251-271, 2002.

[117] Oculus VR. "Oculus Roomscale—Tips for Setting Up a Killer VR Room." https://www.oculus.com/blog/oculus-roomscale-tips-for-setting-up-a-killer-vr-room/ (accessed 06/21/2018).

[118] Steam VR. "Steam VR Support - What are the space requirements of the Vive?" https://support.steampowered.com/kb_article.php?ref=7770-WRUP-5951 (accessed 06/21/2018).

[119] ISBIT Games. "First Person Lover - Female Character and Clothes Pack." https://assetstore.unity.com/packages/3d/characters/humanoids/first-person-lover-female-character-and-clothes-pack-41056.

[120] Y. Liu, H. Dong, L. Zhang, and A. El Saddik, "Technical evaluation of HoloLens for multimedia: a first look," *IEEE MultiMedia*, vol. 25, no. 4, pp. 8-18, 2018.

[121] "Photon Chat by Exit Games." https://www.photonengine.com/en-US/Chat.

[122] Unity Corporation. "multiplay." https://multiplay.com/.

[123] Mercer Road Corp. "Vivox." https://www.vivox.com/.

VITA

Mahmud Hasan
Computational Modeling and Simulation Engineering
1300 Engineering & Computational Sciences Building, Norfolk, VA-23529

Summary

An Engineer and Researcher with a focus on Unity Game and Simulation development for Virtual Reality, Augmented Reality and Autonomous Vehicles.

Education

Doctor of Philosophy (Ph.D.), Graduation: December 2021
Computational Modeling and Simulation Engineering, Old Dominion University, Norfolk, VA

Master of Science (M.S.), Graduation: June 2014
Applied Physics, Electronics and Communication Engineering, University of Dhaka, Bangladesh

Bachelor of Science (B.S.), Graduation: February 2012
Applied Physics, Electronics and Communication Engineering, University of Dhaka, Bangladesh

Experience

Simulation Software Engineer, April 2021 – Present
DEKA Research & Development Corporation, Manchester, NH

Graduate Research Assistant, January 2015 – December 2021
Computational Modeling and Simulation Engineering, Old Dominion University, Norfolk, VA

XR Developer Intern, May 2020 – August 2020
VARtech LLC., Suffolk, VA

Publications

Distributed Microscopic Traffic Simulation with Human-in-the-Loop Enabled by Virtual Reality Technologies, Advances in Engineering Software, 154 (2021): 102985

AR-PED: A Framework of Augmented Reality Enabled Pedestrian-in-the-Loop Simulation, Journal: Simulation Modelling Practice and Theory, 2019

A Framework for Interactive M^3 Visualization of Microscopic Traffic Simulation, Transportation Research Record 2672, no. 44 (2018): 62-71