APPLICATION OF AUGMENTED REALITY AND TANGIBLE INTERFACES TO MINIMIZE WORK ZONE EFFECTS ON MOBILITY THROUGH PARTICIPATORY PLANNING

FINAL PROJECT REPORT

by

Dr. Joseph Louis (Principal Investigator)
Adam Sunderman
McKenzie Gray
Jonah Spencer
Raja Petroff
Andrew Soltesz
Mark Sprouse
Oregon State University

Sponsorship:
PacTrans
Oregon State University

for
Pacific Northwest Transportation Consortium (PacTrans)
USDOT University Transportation Center for Federal Region 10
University of Washington
More Hall 112, Box 352700
Seattle, WA 98195-2700

In cooperation with US Department of Transportation- Office of the Assistant Secretary for Research and Technology (OST-R)

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The Pacific Northwest Transportation Consortium, the U.S. Government and matching sponsor assume no liability for the contents or use thereof.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
	01701494	

4. Title and Subtitle
Application of Augmented Reality and Tangible Interfaces to Minimize Work Zone Effects on Mobility through Participatory Planning

5. Report Date
12/6/2020

6. Performing Organization Code

7. Author(s)
Dr. Joseph Louis, ORCID# 0000-0002-2957-9146

8. Performing Organization Report No.
2018-S-OSU-2

9. Performing Organization Name and Address
PacTrans
Pacific Northwest Transportation Consortium
University Transportation Center for Region 10
University of Washington More Hall 112 Seattle, WA 98195-2700

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
69A355174110

12. Sponsoring Organization Name and Address
United States of America
Department of Transportation
Research and Innovative Technology Administration

13. Type of Report and Period Covered
Research Final

14. Sponsoring Agency Code

15. Supplementary Notes
Report uploaded at www.pacTrans.org

16. Abstract
Highway work zones account for about 10 percent of congestion but are an absolute necessity in ensuring that our transportation infrastructure can meet the mobility demands of growing populations. Unlike other causes of congestion, such as crashes, weather events, and rush-hour traffic, work zones are a result of planned actions by stakeholders. Furthermore, work zones require people to work in proximity to passing traffic, which creates a hazardous work environment. Therefore, the two issues of commuter mobility and worker safety through work zones motivated the development of a novel means of planning for work zones to ensure that all safety precautions are taken while commuters are not unduly affected. This research developed a novel interface that enables decision makers to obtain real-time feedback on work zone strategies through visualization of how they affect commuter mobility. The developed decision support system will consist of a tangible interface and will use augmented reality to enable different stakeholders to participate in planning the design of construction work zones. It is expected that the developed interface will enable participatory planning and will provide a means of interacting with collected geospatial traffic data and simulating traffic operations. This project also improved upon previous work on calculating mass-haul quantities for determining earthwork for roadways that are constructed over undulating terrains. Both these applications provide a novel means for multiple stakeholders to simultaneously interact with transportation and construction models.

17. Key Words
Work zone effects, Work zone planning, Augmented reality, Augmented reality sandbox

18. Distribution Statement
No restrictions.

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified.	Unclassified.		NA

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

Table of Contents

List of Figures

Abstract

Highway work zones account for about 10 percent of congestion but are an absolute necessity in ensuring that our transportation infrastructure can meet the mobility demands of growing populations. Unlike other causes of congestion, such as crashes, weather events, and rush-hour traffic, work zones are a result of planned actions by stakeholders. Furthermore, work zones require people to work in proximity to passing traffic, which creates a hazardous work environment. Therefore, the two issues of commuter mobility and worker safety through work zones motivated the development of a novel means of planning for work zones to ensure that all safety precautions are taken while commuters are not unduly affected. This research developed a novel interface that enables decision makers to obtain real-time feedback on work zone strategies through visualization of how they affect commuter mobility. The developed decision support system will consist of a tangible interface and will use augmented reality to enable different stakeholders to participate in planning the design of construction work zones. It is expected that the developed interface will enable participatory planning and will provide a means of interacting with collected geospatial traffic data and simulating traffic operations. This project also improved upon previous work on calculating mass-haul quantities for determining earthwork for roadways that are constructed over undulating terrains. Both these applications provide a novel means for multiple stakeholders to simultaneously interact with transportation and construction models.

Executive Summary

Highway work zones account for about 10 percent of congestion but are an absolute necessity in ensuring that our transportation infrastructure can meet the mobility demands of growing populations. Unlike other causes of congestion, such as crashes, weather events, and rush-hour traffic, work zones are a result of planned actions by stakeholders. Furthermore, work zones require people to work in proximity to passing traffic, which creates a hazardous work environment. Therefore, the two issues of commuter mobility and worker safety through work zones motivated the development of a novel means of planning for work zones by ensuring that all safety precautions are taken while commuters are not unduly affected. This research developed a novel interface that enables decision-makers to obtain real-time feedback on work zone strategies through visualization of how they affect commuter mobility. The developed decision-support system will consist of a tangible interface and will use augmented reality to enable different stakeholders to participate in planning the design of construction work zones. It is expected that the developed interface will enable participatory planning and will provide a means of interacting with collected geospatial traffic data and simulating traffic operations. This project also improved upon previous work on calculating mass-haul quantities for determining earthwork for roadways that are constructed over undulating terrains. Both these applications provide a novel means for multiple stakeholders to simultaneously interact with transportation and construction models.

Chapter 1: Introduction

The US Department of Transportation (USDOT) Strategic Plan underscores the importance of improving the mobility of people and goods through its strategic focus on infrastructure. Two types of congestion—recurring and non-recurring—typically compromise the mobility of roadway infrastructure. Recurring congestion occurs because of the incapacity of the roadway to meet traffic demand during peak travel periods, whereas non-recurring congestion is caused by incidents such as crashes, weather events, and highway work-zones that temporarily limit the capacity of the roadway and negatively affect the mobility of the traveling public.

This project focused on the mitigating the adverse effects of work zones on the mobility of commuters, as work zones account for 10 percent of all congestion. This choice was informed by the fact that work zones are necessary to ensure the safe and efficient functioning of transportation systems. Furthermore, of all the causes of non-recurring congestion, policymakers have most control over the presence, duration, and locations of work zones on highways, as opposed to crashes and weather events. Specifically, this research proposed a novel visualization system that enables stakeholders to simultaneously participate in planning exercises related to the scheduling, duration, and locations of work-zones based on typical local traffic patterns. Apart from answering the call to develop methods to improve the mobility of commuters, this research also contributed toward the US DOT's strategic goal of developing innovative practices and technologies that strengthen coordination across stakeholders and of developing systems that facilitate data-driven decision support systems.

The impacts of work zones on commuter mobility have been studied widely by both researchers and practitioners because of the need to undertake construction in a manner that causes the least disruption to the traveling public. Apart from using spot sensors for measuring traffic and traffic simulation methods to quantify the effects of work zones on commuter mobility, there have been efforts to utilize machine-learning techniques (Bae et al. 2017) and multi-objective optimization models to determine the tradeoffs between mobility and construction costs (Abdelmohsen and El-Rayes 2016).

While these techniques do provide stakeholders with data-driven decision-making capabilities about work zone mobility, they are limited in enabling interactivity with the data to run what-if scenarios. Even typical traffic simulations do not focus on enabling interactivity among multiple simultaneous users or on the intuitive modification of the spatial and physical properties of work zones.

The proposed system therefore combines the benefits of dynamic visualization with the collaborative nature of huddling around a table to enable users to interact intuitively with the running simulation and obtain real-time feedback. This research sought to develop a software system to be deployed on an augmented reality (AR) sandbox (UC Davis 2018) platform. This endeavor involved the following tasks:

1. *Review of state of the art and practice of work zone planning and effects on mobility:* A review of literature from academia and transportation agencies was conducted to establish the larger context for this research. Specifically, the configurable elements of work zone design during the planning stages was identified and was used to inform the development of the tangible interface in step 3.
2. *Simulation and visualization of traffic flow through work zones:* The SUMO traffic simulation suite (www.eclipse.org/sumo/) was utilized to create a reconfigurable 3-D representation of a work zone section on the highway. The flow of traffic through the

work zone was simulated, taking inputs relating to traffic patterns and volumes and outputting travel times, queue lengths, and delay caused by work zones.

3. *Creation of AR markers to enable real-time interaction through a tangible interface:* The SUMO traffic simulation suite was used to extend default user interface (UI) capabilities to include the use of AR markers, which were themselves implemented by using Vuforia's AR platform (www.vuforia.com). This step enabled multiple users to interact simultaneously with the traffic simulation by changing the location and duration of the work zone and seeing their real-time effects on the mobility metrics of traffic.

This research proposed the development of a novel interactive visualization system that encourages multiple stakeholders to simultaneously interact with traffic simulations. The application is specifically geared toward allowing stakeholders to determine the optimum work zone configuration to minimize the zone's adverse effects on commuter mobility. This research provides a foundation upon which further exploration can be conducted into the use of novel visualization methods to handle the voluminous data generated by transportation systems.

To the authors' knowledge, this implementation of an AR sandbox is a first for traffic data visualization and could potentially pave the way for further application of the interface to other transportation engineering problems that are dependent on spatial attributes and existing terrain, such as highway alignment design, determination of no-passing zones, etc. It is anticipated that this interface will prove to be an engaging platform for use in educational settings to cater to kinesthetic and visual learning styles and will prove to be a novel participatory method for work-zone planning and analysis.

Chapter 2: Literature Review

A review of current literature in the field of mixed virtual reality and augmented reality sandboxes is provided to set the context for the research.

2.1 Augmented and Virtual Reality

Within the commonly used mixed reality continuum (figure 2-1), there are two poles, the real world (reality) and the virtual world (virtual reality), and either can be augmented by the other (Milgram and Kishino 1994). Augmented reality (AR) is the part of the spectrum in which virtual objects augment reality. Augmented virtuality (AV) is the part of the spectrum in which real objects augmented the virtual environment.

In an AR sandbox, both of these augmentations can occur. Sometimes the virtual representation of the sand is used to generate a virtual topographical map (AR); and sometimes the sand augments an understanding of data through exploration of the ways that changes in the sand affect a simulation (AV), for example of a cut and fill diagram. These theoretical differences have implications for AR sandbox design and implementation in educational settings. By understanding the types of tasks to be performed in the mixed reality sandbox environment, the software specifications for AR and AV tasks can be identified. These can help identify technologies for the investigation, design, and analysis tasks associated with spatial reasoning in the civil and construction engineering (CCE) domain.

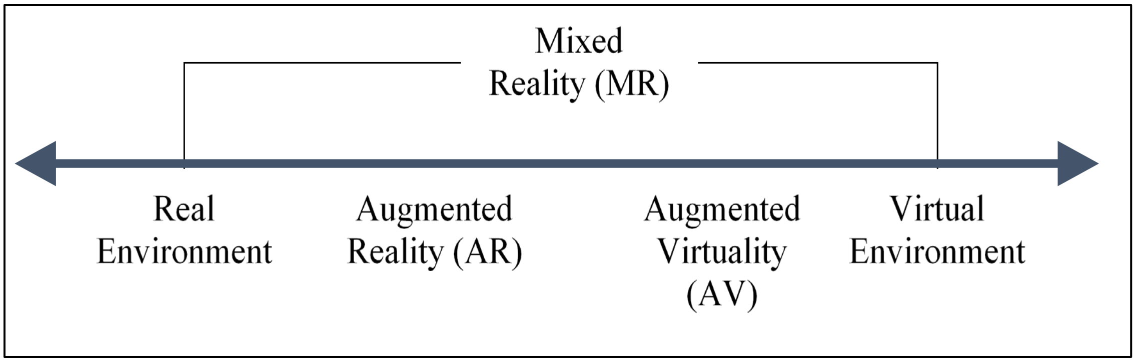

Figure 2-1. The mixed reality continuum (Milgram and Kishino 1994)

2.2 Augmented Reality Sandbox

The augmented reality (AR) sandbox is a device consisting of a depth sensor, projector, and a physical box filled with sand. It was initially developed at UC Davis (2018) as a tangible means of manipulating and visualizing spatial data. This device represents a novel application of AR wherein the virtual information is dependent on the physical shape of the real-world sand. Its primary applications thus far have been in providing students with an intuitive understanding of geographic concepts (Woods et al. 2016) such as topographic maps and hydrology (Zhang et al. 2020). Apart from concepts in these fields, other educational and visualization experiences have been created in the areas of natural sciences (Ables 2017), mathematics (Sanchez et al. 2016), and disaster response (Savova 2016). The common theme among all the AR experiences that have been implemented in an AR sandbox thus far has been the presence of a strong spatial component relating to vast areas of physical terrain. Furthermore, all the above experiences have allowed only one primary means of interacting with the underlying model and visualization, which has been to physically manipulate the sand. For the above applications, the primary concepts under study have been the natural terrain, which can be adequately represented and manipulated by the sand alone.

However, civil and construction engineering concepts involve built infrastructure to be considered in addition to the surrounding terrain. Therefore, this project built upon the current capabilities of the AR sandbox interface to provide a more general platform for the sandbox that will allow multiple modes of interactivity with the underlying terrain and physical infrastructure under consideration.

2.3 Point of Departure

The point of departure of this project from previous work lay at the intersection of the current state of the art for AR sandboxes and the need for an enhanced means of incorporating spatial reasoning in engineering educations, especially for civil and construction engineering. To date, implementations of the AR sandbox have not considered the interaction of built infrastructure with the terrain and rather have focused on the terrain itself, thereby limiting AR sandbox applications to areas of study relating to geology and hydrology. Therefore, the overarching goal of this project was to enhance the capabilities of the AR sandbox to enable users to create, visualize, and interact with digital representations of the built environment and infrastructure in addition to the physical terrain that is represented by the sand. To this end, this report describes the development of an enhanced sandbox and a case study of the development of an application for a specific construction engineering concept taught in the classroom.

Chapter 3: Development of the AR Sandbox

The AR sandbox used in this research was built from commonly available materials and implemented by the authors at Oregon State University. This section describes the hardware and software architecture that comprises the AR sandbox.

3.1 Hardware Design of the AR Sandbox

Figure 3-1 shows a schematic representation of the AR sandbox along with its three primary hardware components: depth sensor, projector, and computer terminal. As can be seen in the figure, the depth sensor and projector were mounted side-by-side on a metal bracket that extended over a box of sand. Both the depth sensor and the projector were connected to a computer terminal.

For portability, the sandbox was mounted on a wheeled double shelf cart, and the sand was placed on the top shelf along with the computer terminal. Half-inch (1.27-cm) thick clear acrylic board walls were affixed to the sides of the top shelf to accommodate the volume of sand. The finished sandbox is shown in figure 3-2.

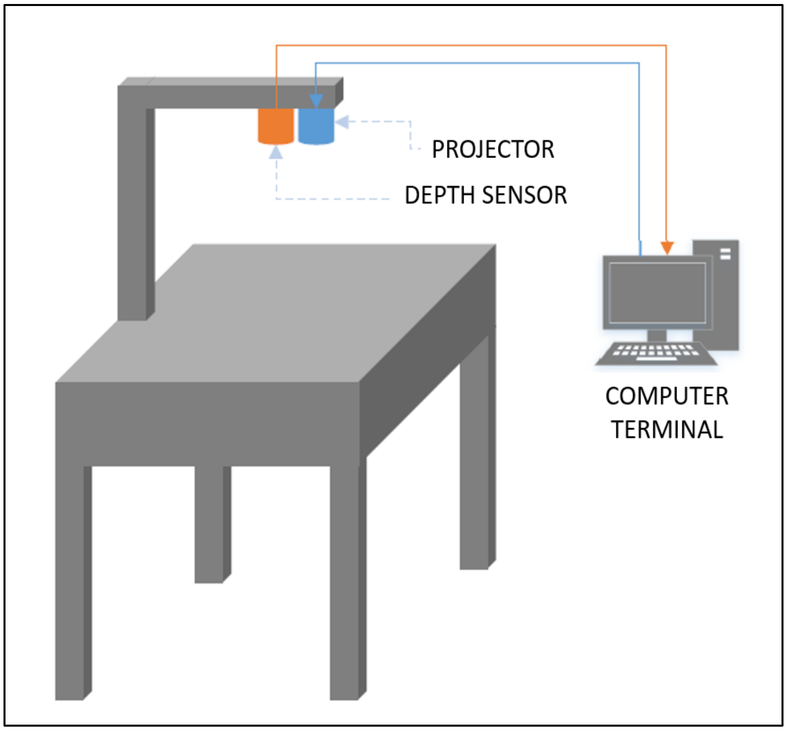

Figure 3-12. Schematic diagram of the AR sandbox

Figure 3-2. The AR sandbox mounted on a wheeled cart

The roles of the three primary components—the depth sensor, projector, and computer terminal—are described below.

3.1.1 Depth Sensor

For the system to properly represent the height of the sand via projection, it had to capture the height of the sand as input. This could be done in several different ways: scales recording the weight of the sand atop it; a camera recording the surface of the sand and extrapolating the height from that; or a three-dimensional depth sensor taking in the height at various points. All of the potential choices had benefits and drawbacks. To make the best decision, the following were used as the primary criteria to assess the various options. These criteria included the following:

- Precision: How precise the measurements taken are.
- Cost: How much it will cost to implement.
- Data Manipulation: How much the raw data must be manipulated to achieve the desired information.

This section explores the three potential ways that the system could capture the height of the sand throughout the sandbox as input. The first of these choices was the installation of a grid of scales along the floor of the sandbox, measuring the amount of sand above that given point. The second choice was to utilize two cameras positioned above the sandbox, creating a stereoscopic image for the system to analyze. The final option was to utilize a three-dimensional, depth sensing camera placed above the sandbox.

1. Scales: The greatest drawback in using scales to determine the height of the sand would be the lack of precision. A scale will only ever yield the average height along the area that it is weighing. With smaller scales, a greater quantity would be needed to fill the area

of the sandbox, and a sharp increase in cost. With most scales costing around $100, the cost of creating a grid would get very expensive very quickly. In addition to the high costs and lack of precision, using scales would also require an increased amount of data manipulation. The data captured via the scales would be the weight of the sand placed atop it. To get the desired height data, that weight would have to be manipulated by using the density of sand and the area the scale covered to get the average height within the area. This would be far from efficient and would require a vast number of calculations.

2. Stereoscopic cameras: Utilizing two different cameras to capture a stereoscopic view would be more precise than using scales to measure the sand's weight. It would depend on the resolution of the camera, but instead of just providing the average height over larger areas, the cameras could provide a height for a given point along the surface of the sand. In addition to the increased precision, it would also be slightly more cost efficient. By having two cameras record the state of the sand, the system would behave much like the human eyes. By analyzing the disparity between a point from one camera to another, the height of that point could be derived. Because of the increased precision, more calculations would need to be executed to complete the data, and those calculation would be a lot more complex. Instead of an increase in cost for an increase in precision, using two cameras would increase the number of calculations required to manipulate the data to gain increased precision.
3. Depth sensors: Depth sensors would yield equal (if not more) precision than a stereoscopic set-up. They often utilize technologies similar to those of the systems above but take things a few steps further to improve on efficiency and precision. A depth sensor is typically a little more expensive than using a two-camera set-up, but makes up for that

in less data manipulation required. Depth sensors were designed specifically for the task at hand. As a result, they output the exact data needed, and no additional calculations would be required to obtain the various heights along the sand.

Of all the potential choices, it made sense that using a depth sensor would be the best choice. Depth sensors were designed specifically to provide the information that the system would need. They would have the greatest precision of all the options and would require the least number of system calculations, allowing the system to focus on its own calculations. Although they were not the cheapest option available, their strengths in the other categories far outweighed the negative. However, the conclusion that a depth sensor was the best way for the system to gather its input begged one question: Which depth sensor should be used? This sensor would need to be able to record the topography along the sand and send that information to the system for analysis. For the purposes of the AR sandbox for construction planning, the three best sensors were the Microsoft Kinect, the Intel RealSense SR300, and the Asus Xtion 2.

To ensure the best possible depth sensor for the sandbox, three criteria were used to compare the sensors. These components included the following:

- Quality of the camera: The quality of the camera included the resolution of the image taken.
- Ease of interface: How easily the information from the depth sensor could be sent to the computer.
- Cost: How much the device would cost to purchase.

Three depth sensors appeared to be the best options. The first of these was the Microsoft Kinect for Windows PC, a small sensor utilizing an infrared camera and emitter to capture depth. The second was the Intel RealSense Camera SR300. The successor to the Kinect, it utilized similar technology. The third and final option was the Asus Xtion 2. Also using infrared to determine depth, the Xtion was created to be a direct competitor to the Kinect and RealSense.

1. Microsoft Kinect: The Microsoft Kinect was one of the first devices to enter the market as an infrared-using depth sensor/camera. It emits a series of infrared beams and has a sensor to pick up the reflections and create a three-dimensional image. This technology and technique have been found to be very accurate and begin falling off only as the distance from the target increases. For the purposes of the sandbox, the depth sensor would be placed relatively close to the surface. Another benefit of utilizing the Kinect would be that Microsoft has already created an API for the sensor to interface with Windows computers. This framework was created to help developers trying to utilize the sensor get going as quickly and as easily as possible. In addition, given the age of the Kinect, there would be a vast amount of online assistant available to aid in the implementation of the device.
2. Intel RealSense: The Intel RealSense Camera was created to be the successor to the Kinect and is encouraged to be used in its stead. The Kinect is still used in abundance throughout the industry, but it is no longer being manufactured. As a result, the RealSense will slowly take its place. Being the next generation of the Kinect, the RealSense utilizes the same method for determining depth throughout a sensor's field. Designed to be in a depth range of 0.2 meters to 1.5 meters, the sensor fits into the exact range needed for this application. As with the Kinect's technology, the RealSense's precision has been found to be accurate when operating within a short to medium distance from the sensor. The RealSense builds off the Kinect's Windows interface and

interacts with a computer in much the same way with its own software development kit (SDK) and application programming interface (API).

3. Asus Xtion 2: The final depth sensor considered was the Asus Xtion 2. This sensor, like the others, uses the same technology to create its three-dimensional images. It uses the same technology; however, its hardware is not as strong as that of the Intel RealSense. As a result, the resolution of the image taken and the precision of the data are slightly less than those of the RealSense. Though there are drivers and ways for the Xtion to interface with a computer, they are not as efficient and not as well supported as those of the Kinect or RealSense. This can partially be attributed to the Kinect being created by Microsoft and the RealSense receiving its endorsement as a device. Given these factors together, the Xtion was found to be less impressive than either of the other two sensors. In comparing the abilities of the sensors and the costs, this sensor was out of the question.

Of all the depth sensors, the Intel RealSense would appear to be the best choice if considered in a vacuum. Both the Kinect and the Intel RealSense are known to work a little bit better than the Asus Xtion, and the RealSense takes all that is great about the Kinect and builds upon it. However, the Kinect was selected for the sandbox because of its use in a previous implementations of the AR sandbox.

3.1.2 Projector

For the height visualization to be presented on the surface of the sand, a projector would be the best choice. Projectors come in various strengths, refresh rates, and costs, but which would be the best for the sandbox's purposes? To determine the best projector for the system, several aspects were weighed. Those aspects included the following:

- Resolution: The resolution of the picture created by the projector. The higher the resolution, the clearer the picture projected will be.
- Brightness: How bright the projector's bulb will be. The brighter the bulb, the brighter the environment in which the projection can be.
- Cost: How much the device will cost to purchase.
- Size/Weight: The projector would need to be light enough to be mounted above the sandbox.

The following three projectors were determined to be the three best options to satisfy the needs of the sandbox system: ViewSonic PJD7720HD, BenQ TH670, and the AAXA M5 Mini Portable Business Projector.

1. ViewSonic: The ViewSonic projector is an all-around solid projector for its price tier. It offers a full 1080-p picture, allowing for data to be presented clearly to the user. Although the picture is in full-HD, the brightness is equally important. Without an adequate brightness, the image cannot be viewed at all. The ViewSonic has a brightness of 3200 lumens, meaning that the picture will be very clear in most environments. While not expensive, the ViewSonic is not very compact. It is approximately 12 in. x 9 in. x 4 in. in size and weighs 5.3 lbs.; a manageable size, although it would not be completely suitable.
2. BenQ: The next option, the BenQ TH670, is the strongest of the projectors. Like the ViewSonic, the BenQ projects a picture at full 1080-p HD quality. With 3000 lumens of brightness, the BenQ sits in the middle of the pack regarding brightness. Like the Epson, the BenQ allows for a screen of up to 300 inches in size, overkill for what the system would need to project. At $700, this is by far the most expensive of the projectors. For

this price, the BenQ offers a larger screen size and built-in speakers, two additions that would be far from necessary for the system. Along with the increased power and cost, the BenQ comes with increased size and mass. At approximately 13 in. x 19 in. x 15 in. and weighing 9 pounds, the BenQ is both larger and heavier than the ViewSonic. With these physical dimensions, it would be difficult to mount the projector in the desired position for the sandbox.

3. AAXA M5: The final projector under consideration was the AAXA M5 Mini Portable Business Projector. Once again, this projector offers 1080-p HD quality. However, the AAXA offers only 900 lumens of brightness. This brightness will work in most low-light situations but may become harder to see when the brightness of a room increases. At a price of $435, this is the cheapest of the options. What the AAXA lacks in brightness it makes up for in its compact size and low weight. It is 6 in. x 6 in. x 1.8 in. in size and weighs less than 2 pounds. Although its brightness is less than that of the others, 900 lumens is still one of the highest levels available at this size of projector.

All three of the projectors discussed had their pros and cons. They would all provide a full HD picture at 1080 p and adequate picture brightness for most environments, with the ViewSonic providing the greatest brightness at 3200 lumens. For BenQ projector, larger picture size would be provided at a higher cost and increased size. The ViewSonic PJD7720HD offered a better price and smaller dimensions and weight for an unimpactful reduction in picture size. However, its weight of 5 pounds would still hinder the mounting required. As a result, the AAXA was determined to be the best projector for the AR sandbox project. It was not as bright as the others, but it would be sufficient in most lighting environments, and its small, 2-pound body was too valuable to pass up.

3.1.3 Computer Terminal

To power the AR sandbox, a HP Z240 full tower desktop computer was used that had 16 GB of RAM, a 1-TB spinning disk, and an Intel core i7 processor with a base speed of 4.2 GHz. Furthermore, an Nvidia GTX 10XX series graphics card was implemented in the terminal to handle rendering. Regarding the computer terminal, it was necessary to use a full tower desktop because most graphics cards are too big to fit in a small form factor machine. Using an Nvidia GTX graphics card would provide the best results for rendering real-time graphics at the lowest cost in comparison to a similar workstation graphics card. A mouse and keyboard were connected to the computer terminal to enable additional interaction capabilities with the AR sandbox. The output of the terminal would be projected onto the sandbox, and therefore no additional monitor was provided with the sandbox, although one could be easily added for development and debugging purposes.

3.2 Software Architecture of the AR Sandbox

The software architecture of the AR sandbox was developed to provide future developers the opportunity to create applications related to core concepts in CCE that could be deployed through the AR sandbox interface without the need to interface with low-level details of the hardware components. Therefore, the fundamental functionality that was enabled involved obtaining depth image information from the depth sensor with the Kinect's API and storing that as a heightmap to provide heights in the sandbox at any given x and y location in the sensor region. Also, a geometry interpreter was implemented to convert heightmap data from the depth sensor into a three-dimensional mesh. Apart from dealing with the depth sensor, calibration was required to ensure that the sensor and projector would be correctly aligned with each other, as well as with the surface of the sand. These settings included adjustments to the area exposed to the depth sensor and to the area covered by the projection. This module would continually interact with the user interface (UI), prompting the user with the setting to be adjusted, and then would store those settings in the calibration settings. The four aspects of software design that were important to the AR sandbox were the rendering framework used, the height storage used, height representation, and depth visualization. These are discussed below.

3.2.1 Rendering Framework

At its lowest level, a rendering framework handles communication between an application and the graphics hardware. In this case, the term is expanded to also include game engines, which act as their own self-contained development environments and handle many other tasks in addition to rendering graphics. The options considered were the Unreal Engine, Unity Game Engine, and OpenGL.

1. *Unreal Engine 4*

Unreal Engine is an eighth-generation game engine produced by Epic Games (www.unrealengine.com). Notable features include real-time global illumination, full access to the engine's source code, and the ability to tweak code while the game is running to rapidly test features. Unreal Engine is free to download and use but requires 5 percent of the game developer's profits upon launching the title. The engine has a large userbase and an active forum, meaning that in-depth, user-created instructional content exists, and technical questions can be answered quickly. One of Unreal Engine's main strengths is its graphical fidelity. With very little effort a developer can create a realistic scene with convincing materials and lighting. For the scope of this project however, this was irrelevant as realistic rendering was not the desired result.

2. Unity Engine

Unity Engine is a game engine developed by Unity Technologies (https://www.unity.com/). It is free to use if the developer's total revenue or funding does not exceed $100,000. Two paid versions of Unity are available, with higher revenue caps and additional features. Unity's scripting language is C#, and user-created shaders are specified with the Graphics Library Shading Language (GLSL). One of Unity's biggest strengths is its ability to easily deploy on multiple platforms, such as Windows, MacOS, and Linux, as well as many mobile platforms and game consoles. This could be a boon to this project, as the ability to use a computer running Linux or MacOS to create an AR sandbox could vastly increase the accessibility of our final product, as well as decrease costs for the end user. Another strength is the relatively small learning curve. This makes it very easy for new developers to quickly begin creating their project in Unity, which would be beneficial to our project, given our short development period.

3. OpenGL

OpenGL is a cross-platform computer graphics API maintained by the Khronos Group. Unlike Unity or Unreal, OpenGL is not a game engine but instead an interface between an application and the graphics processing unit (GPU). This means that OpenGL does not provide a development environment or specify that a particular language be used to develop the application. The main benefit of using OpenGL is performance; because no additional features are provided, there is no overhead for features that go unused. Developers are free to specify exactly what they want the graphics hardware to do without interfacing with an additional layer of abstraction, such as a game engine. This can also be a drawback, as one of the main draws to using a game engine is the ease of development and the fact that the engine automatically takes care of much of the heavy lifting associated with rendering an object on the screen. For a project such as this, the majority of a game engine's features, such as physics, would go unused, and a majority of the rendering power built into the engine would be either turned off or hidden. This would make a much lower level system such as OpenGL more attractive.

Choice

While the more efficient OpenGL was certainly an attractive choice for this project, the best tool for the job was the Unity engine. Unity would provide a middle ground between Unreal 4, which is focused more on photorealistic rendering, and the simplicity of OpenGL. Unity would retain all the integrated build support of a game engine while still being easy to use and fully capable of supporting the requirements of this project. While additional performance might be gained from OpenGL, Unity's performance would be more than acceptable for the caliber of graphics that was required for this application.

3.2.2 Height Data Storage

To save the sandbox's topology, a digital representation would have to be created. This representation would need to contain enough information to correctly recreate the sand's surface. The options available for this purpose were the greyscale bitmap, FBX, and OBJ file formats.

1. *Greyscale Bitmap*

The most logical option for storing the terrain of the sandbox was with a greyscale bitmap image, since this was the format that the depth sensor would use to express this information. The benefit of this approach was that a bitmap would be relatively small in comparison to other options and would represent all the information required to recreate the sand's surface, without additional unnecessary metadata. The main downside of this approach was that if users wished to reconstruct the sand's surface in an easy to visualize manner, they would have to convert the height map to a mesh. An additional downside was the fact that a heightmap would be unable to represent undercuts (areas where the surface of the terrain folds back on itself, such as a cave) because it could represent only one height value per x-y coordinate.

2. FBX File Format

The FBX file format is a proprietary file format designed by Autodesk for storing three-dimensional scenes. The FBX format is one of the most widely used 3-D file formats because of its ability to also store additional scene data, such as camera position, skeletal meshes used for animation, lights, and materials, as well as its compact size. The main downside to this format is that Autodesk is not forthcoming about the structure of the FBX format, and while it does offer a C++ API that allows developers to export FBX files from their applications, the internals of this API are obfuscated. Individuals and organizations have made efforts to reverse engineer the format to integrate FBX import and export capabilities into a wider range of programs. Notably, the Blender Foundation has been able to successfully implement FBX support into the 3-D modeling package Blender without using the Autodesk-provided API.

3. OBJ File Format

The OBJ file format was originally developed by Wavefront Technologies for use with its motion capture program Advanced Visualizer. Unlike the FBX format, OBJ files are written in plaintext, meaning that they are relatively humanly readable. Another difference is that OBJ files are capable of specifying only geometry, and materials through a reference to an external MTL file. The benefit of the OBJ format is that it is supported across almost every 3-D graphics platform. Another benefit is the ease of creation because the files are written with ASCII characters. OBJ has two major drawbacks however: the first is that only geometry can be stored, which for this project would be a relatively minor concern. More pressing is the file size, especially in comparison to the FBX format. As an example, a 6,050-triangle mesh saved as an OBJ file would be 293 KB, whereas the same mesh saved as an FBX file would be only 127 KB, less than half the size. Scaled up, a 732,050-triangle mesh saved as an OBJ file would be 39.7 MB, and the equivalent FBX file would be only 11.9 MB.

Choice

We decided that the best approach for storing height data would be to use a greyscale heightmap, with the option of exporting an OBJ file should the user request one. An image file would be far more compact than any of the 3-D file formats investigated and would contain all the information necessary to recreate the sand's topology. The only downside was that it would be less meaningful than viewing an actual three-dimensional mesh of the terrain; a downside that could be mitigated by giving the user the option to export a file. The OBJ file format was preferable because of its ease of creation and universal support.

3.2.3 Height Data Representation

Raw data from the AR sandbox's depth sensor would take the form of a heightmap—a greyscale image whose pixels correspond to the height of the sand at a specific location. The heightmap would need to be interpreted by the application to display meaningful information on the sand.

1. *Two-Dimensional Representation*

One approach would be to use the value at each pixel in the heightmap to color pixels on a 2-D image to represent height. The benefit of this technique would be simplicity, as there would be only one step involved in converting raw height data to the final image. A drawback to this approach would be that it would offer little flexibility for future developers who might wish to implement additional features.

2. Three-Dimensional Representation

Another approach would be to use the height data to generate a three-dimensional mesh. A shader would then be used to apply color to the mesh to represent the height. A top-down view of this mesh would then be projected onto the sand. While this seems like unnecessary work and would add an extra step to the rendering process, it would give us a lot more flexibility when adding additional features. For instance, when simulating the effects of runoff on the terrain, the mesh could be used to calculate the flow of water. Additionally, if users wanted to save the state of the sandbox, having a three-dimensional representation of the topography at a specific time would give the saved data more meaning, as they would be able to view not only the path of the road but also the terrain through which the road was passing. One downside to this approach would be the additional computational cost of looking up the height of individual points. If using a mesh, it would first be necessary to determine on which triangle the point fells, then use the three vertices of the triangle to determine the height of the point geometrically. Another downside would be that, depending on the resolution of the heightmap, the resolution of the three-dimensional mesh might need to be down-sampled to efficiently render the mesh.

3. Coupled Two- and Three-Dimensional Representation

A third option would be to both retain the original greyscale heightmap and render a three-dimensional mesh. While this would be the most computationally expensive approach and would require more data to be stored in RAM, it would be very easy to obtain the height value at a given point. The point in three-dimensional space would need to be correlated to a pixel on the heightmap, which could then be sampled to get a height value. This approach would also mitigate the issue of having a lower resolution mesh than the original heightmap because the original heightmap could be maintained and referenced when exact heights were required.

Choice

Maintaining both a mesh and a heightmap was determined to be the best approach, as the additional computational overhead imposed would be more than made up for by the ease of acquiring height information. Additional RAM usage was not a major concern either, as the capacity of modern memory modules is so large. The ability to render a lower resolution mesh would also be helpful because even a low resolution heightmap would require many triangles to represent the heightmap at full resolution. As an example, a 480- by 480-pixel heightmap would require 4802 2 = 115; 200 triangles to be rendered. While not implausible with modern graphics hardware, this could be a concern for older computers.

3.2.4 Depth Visualization

Depth visualization consists of ways of interpreting the height or depth of a landscape or feature. It is a method of scientific visualization and, as an example, can be used to model or visualize the terrain of a geographic area. For our research, the feature being visualized would be the height or depth of sand in a sandbox. The graphics projected on the sand would need to conveniently communicate to users the lay of the land or, more specifically, the altitude or elevation of an area. This could be done effectively by using methods of representation such as a color map, a contour map, or a topographic map, which are described in the following sections.

1. Color Map

Color is a convenient and easy way of modeling data and allows people to quickly understand a complex system or environment. For example, altitude or elevation data can be used to make many different types of terrain displays and maps, such as color-scaled contour maps, conventional contour maps, and shaded relief maps. Therefore, the first choice for representing height would be a color map. This color map would use different colors to represent the height or altitude on the map, starting with blue for low altitude areas and progressively going up the color scale to shades of green, yellow, and finally red for high altitude areas. For example, land below sea level would start with a deep blue color while land above sea level would proceed from light shades of blue to light shades of green. Eventually, land at higher altitudes would continue from dark green to yellow to red indicating hills or mountains, respectively.

2. Contour Map

A classic method of representing terrain elevation is a contour map. The typical contour map uses successive contour lines to represent altitude. Features of the landscape, such as hills or mountains are outlined using these contour lines. These contours can be used to give an outline of the terrain.

3. Topographic Map

Topographic maps can be thought of as a combination of both color and contour maps. In fact, topographic maps can even be used to record a land area, containing the geographic positions and elevations of both natural and man-made features. In addition, topographic maps are also traditionally used in highway planning because they provide useful information about the features of the land, such as the approximate amount of cut and fill, drainage, where bridges may be needed, and degree of economic development of the area. This makes topographic maps an excellent choice for visualizing depth because they are in a format with which civil and construction engineers are familiar.

Choice

While both the color map and contour map could be used to represent the same thing, color is an easier indicator of height or altitude in comparison to contour maps because contours in a contour map need to be counted. Contrast this with easily looking at the color of an area and recognizing the elevation. However, topographic maps combine both advantages and therefore provide both means of reading the terrain. On top of this, topographic maps are a more traditional format with which civil and construction engineers are familiar. Therefore, a topographic map was the best choice because it could represent more information than a color or contour map. Additionally, topographic maps are already used in fields such as civil engineering and city planning.

Chapter 4: Earthmoving Visualization on the Sandbox

The following images were taken of the sandbox to demonstrate its capabilities for visualizing earthmoving operations for the construction of roadways. To this end, the sandbox provides three features: (1) visualization of terrain topography, (2) indication of cut and fill sections of the roadway, and (3) mass-haul table calculations. These topics are discussed below with visual examples.

4.1 Visualization of Terrain Topography

The depth sensor described in Section 3.1.1 was used to provide information regarding the terrain that was created by a user with the sandbox interface. These depth data were used to create color, contour, and topographic visualizations of depth (described in Section 3.2.4), which were then projected back onto the sandbox.

Figures 4-1 and 4-2 show the depth of the sandbox visualized with the color and topographic representations, respectively.

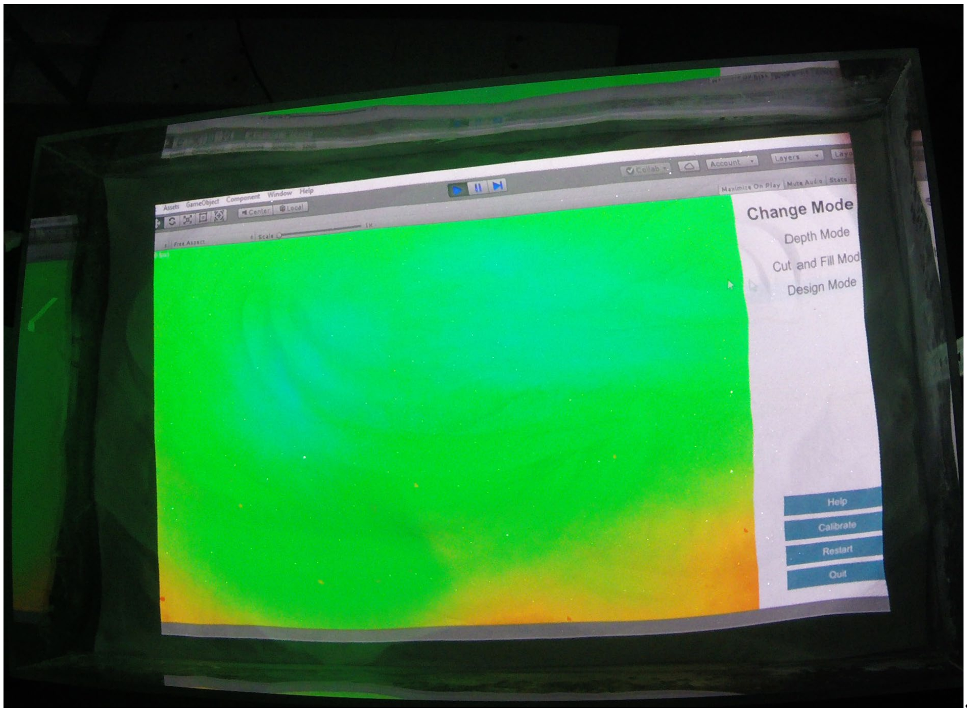

Figure 4-1. Color representation of terrain projected on the sandbox

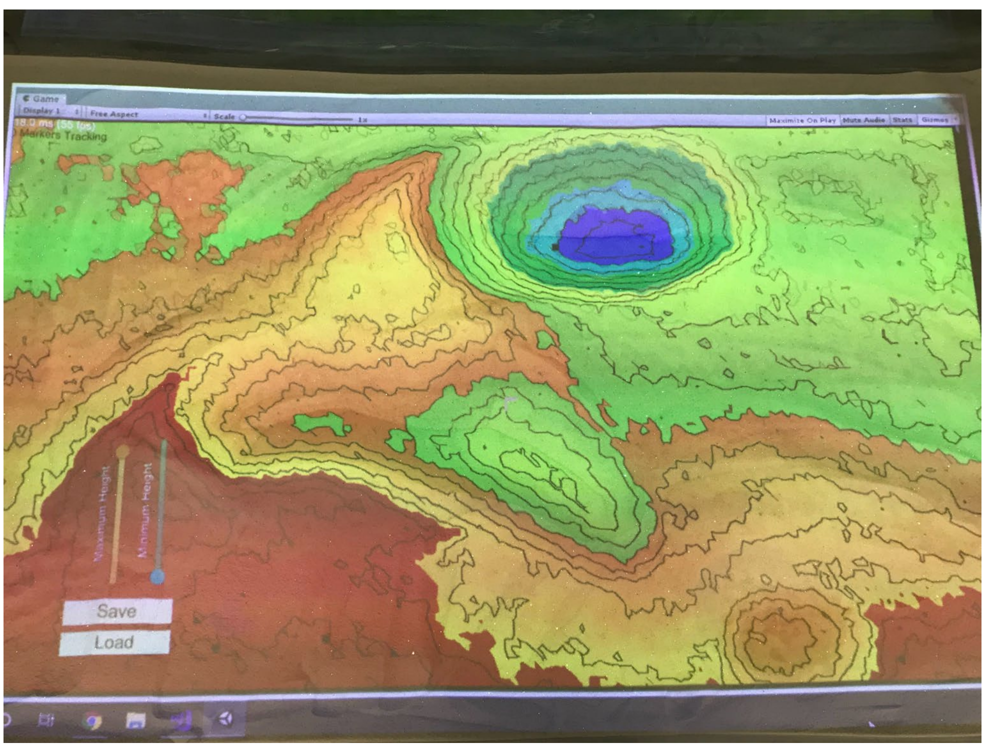

Figure 4-2. Topographic representation of terrain projected on the sandbox

The representations that were projected on the sandbox would enable users to tangibly create the type of terrain that they were interested in analyzing for their use case.

4.2 Visualization of Roadway Cut and Fill Sections

To identify the cut and fill sections of roadway to be constructed, users can define the horizontal and vertical alignment of roadway segments on the sandbox as shown in figure 4-3. The yellow points represent control points that could be manipulated with a mouse to change the horizontal or vertical alignment of the roadway.

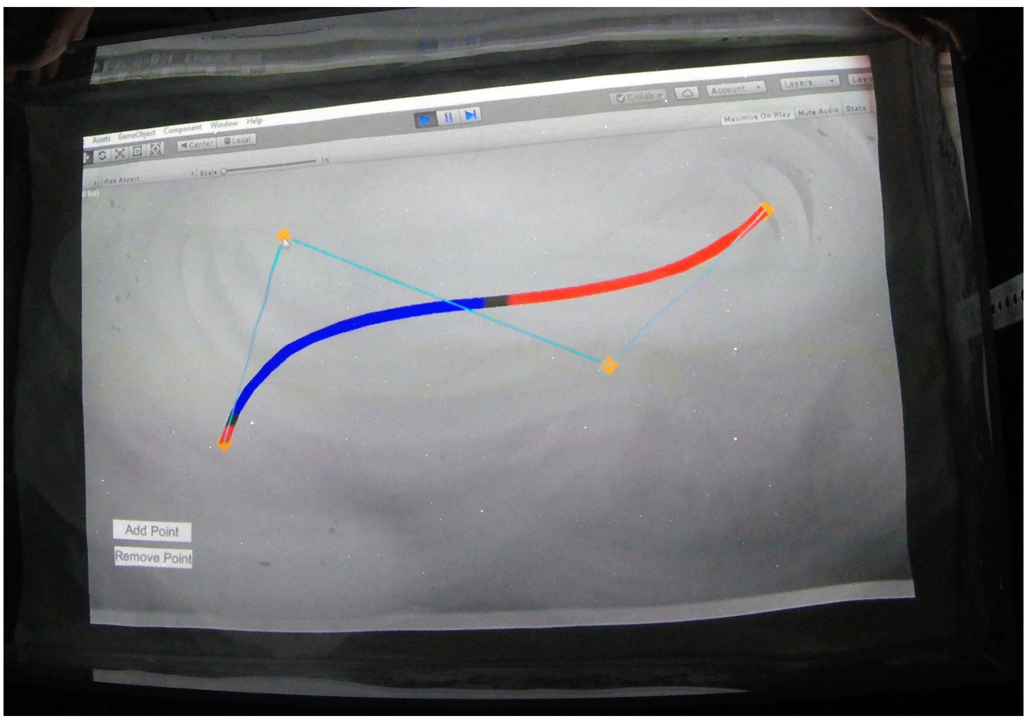

Figure 4-3. Defining roadway alignment for analysis

After the roadway alignment has been produced, the software automatically determines whether the section is cut (represented by red), fill (represented by blue), or on grade (represented by black). This is achieved by comparing the design elevation of the roadway segment along its centerline with the depth value of the terrain at that point. This can be seen in figure 4-3, and further examples of segments that are completely cut or fill are shown in figures 4-4 and 4-5, respectively. A further example in which the segment is partially cut and partially fill is shown in figure 4-6. The following figures also indicate the vertical alignment of the roadway (in yellow) and that of terrain along the roadway centerline (in yellow) at the bottom right of the sandbox.

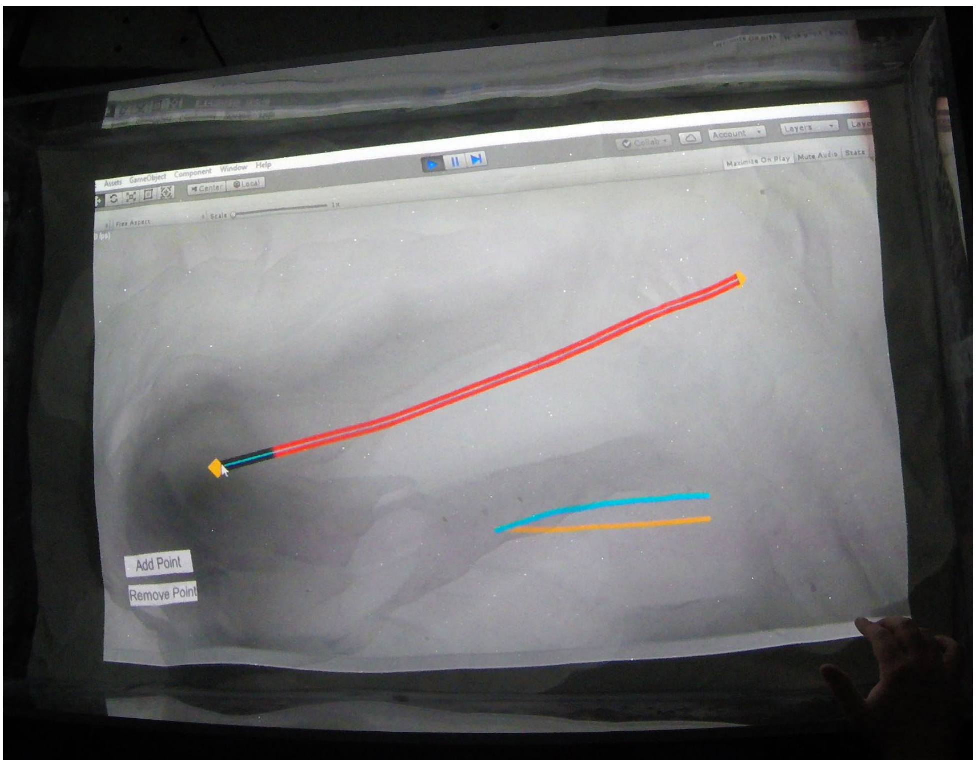

Figure 4-43. Mostly cut segment of a roadway

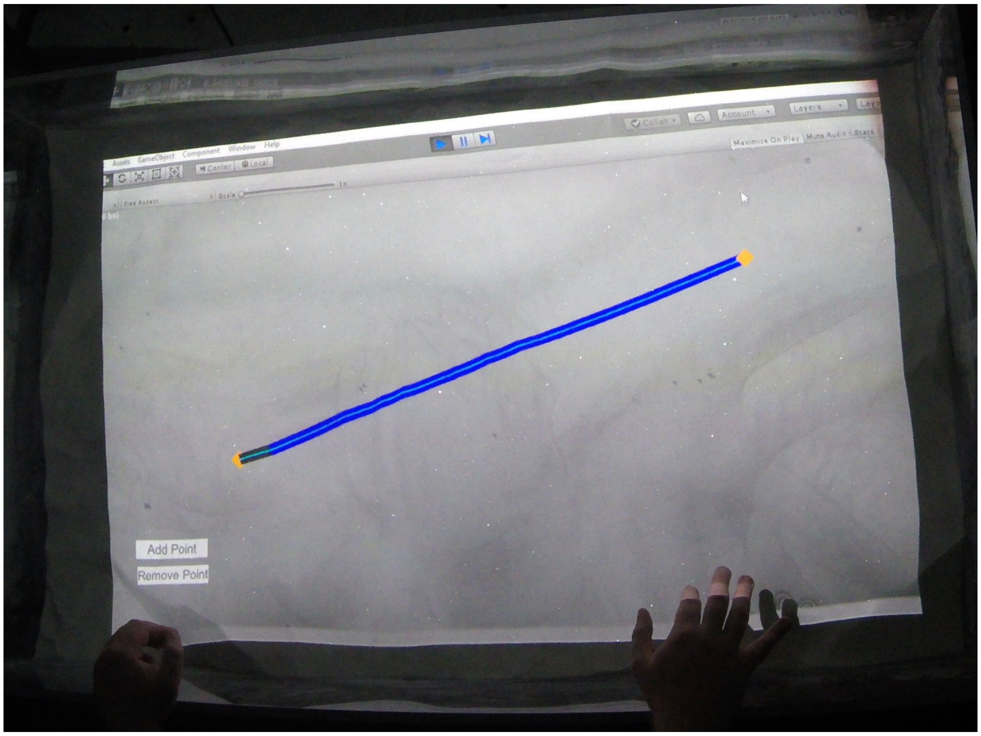

Figure 4-5. Mostly fill segment of a roadway

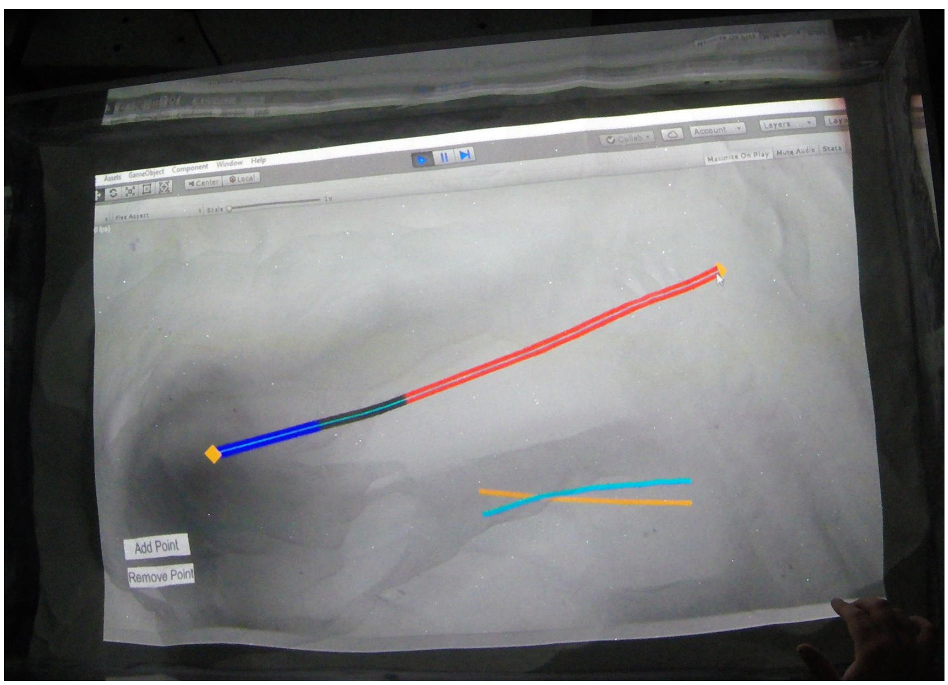

Figure 4-6. Partial cut and fill segment of a roadway

Visualizing the cut and fill segments of the roadway would serve the purpose of educating students as to the quantity of earthmoving and direction of haul that would be required. In the implemented example, these values are updated in real time on the basis of depth sensor readings. Therefore, students could move the sand from the cut to the fill sections and see in real time the changes to the cut and fill sections of the roadway. This would serve as a valuable tangible educational tool to augment lessons related to the construction of roadways.

4.3 Mass-Haul Table Calculations

The final capability that was implemented for earthmoving was the creation of tables to aid in the calculation of cumulative earthmoving quantities along the centerline of the roadway. To this end, the entire roadway was divided into segments representing 100’ each, and the earthmoving for each segment was calculated for each of those segments and analyzed for mass-haul calculation as described by Peurifoy and Ledbetter (2003). Currently, the spreadsheet can be displayed on the sandbox, although future implementations will seek to provide visualizations of earthmoving quantities, haul distances, and haul grades that can be obtained from further analysis of the mass-haul spreadsheet. Figure 4-7 shows the mass-haul table projected on the sandbox.

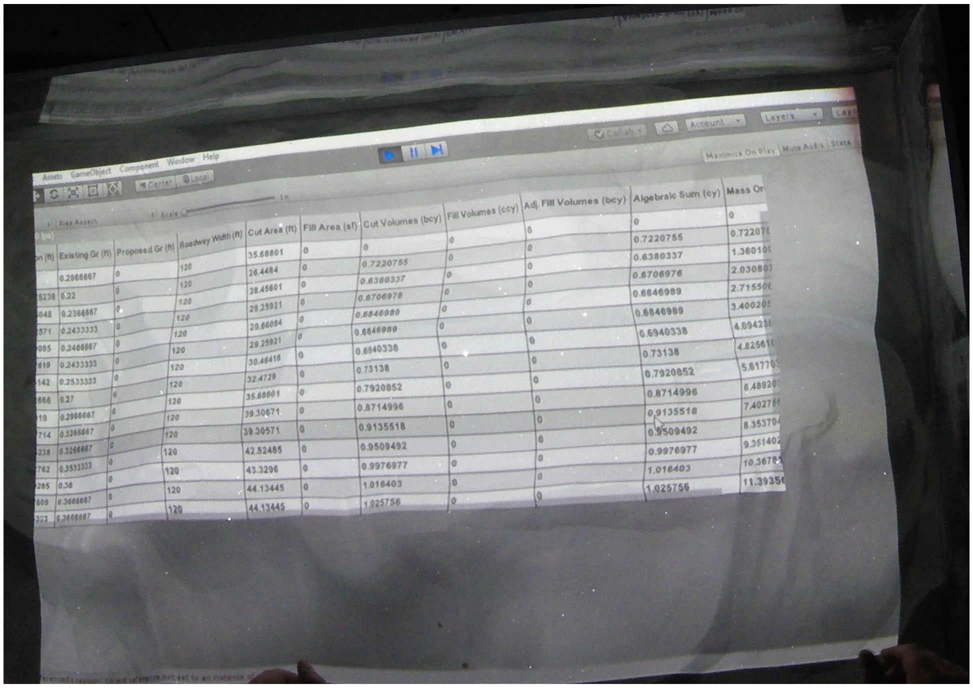

Figure 4-7. Displaying a mass-haul table on the sandbox

Chapter 5: Traffic Simulation for the AR Sandbox

Augmented reality sandboxes (AR sandbox) have become increasingly popular in recent years for topographical image mapping and water flow simulations. The basic concept behind an AR sandbox is simple. The unit comprises a physical box of sand, video projector, depth sensor, and computer. As users interact with the sandbox and shape the sand within, they also manipulate in real time an image that is projected onto the sand. Despite their increasing popularity, AR sandboxes still have a great deal of untapped potential. This document outlines a plan to use the AR sandbox at Oregon State University to improve teaching and planning methods in the fields of civil and construction engineering. The changes were intended to add new functionality in the form of a marker-based system for adding various interactive elements to a sandbox. These elements could be used for traffic simulations and the general construction of an image to be displayed on the sand.

Computer simulations are an effective tool in planning and teaching. By using simulations, engineers and scientists can test scenarios in simulated environments without any real-world risk. This technology also provides an opportunity for students to learn in a practical, hands-on way without having to leave a classroom. However, the problem with most computer simulations is that they require a software engineer to design and set up. Our goal was to reduce this dependency by creating a model for building working computer simulations using physical objects and augmented reality. Currently, the college of Civil and Construction Engineering's (CCE) Augmented Reality Sandbox (AR Sandbox) is incomplete. Civil engineering professors want to be able to use the AR Sandbox as a traffic simulation environment that can be built and altered with marker-based object placement. Markers would be physical objects placed in the sandbox that were uniquely identifiable by current or additional hardware. Markers would signal the system to project images of arbitrary objects, such as buildings or streetlights, into the sandbox. These objects would then be added to a digital scene on the control computer on which the simulation would run its logic. This feature was the main purpose for this project, but problems arose with the current implementation of the software. The software was hard to set up, the calibration for the depth sensor was nearly non-existent, and the projected elevation images did not display contour lines.

The goal of creating a traffic simulation for use in the classroom was achieved by expanding upon a terrain mesh generated by the AR Sandbox with the Unity Game Engine. This mesh is formed with the current sand height information the AR Sandbox's depth camera reads in real time. On this mesh, pre-made road patterns would be imposed, and traffic simulation would be performed by using a pre-built traffic simulator. Scenes would be built, and simulations would be run by using Unity and then projecting a top-down view of this simulation onto the sandbox. Unity makes it simple to create 3-D scenes and has a built-in physics engine that would be useful for simulations. To implement the AR marker functionality, the software development kit Vuforia was used. Vuforia uses computer vision to track and identify real-world objects. It integrates well with Unity and provides an easy, powerful interface for digitally re-creating a 3-D object through object recognition. Objects, when placed in the sandbox and recognized, would catch a pre-defined Unity asset that would be drawn to a scene where the marker was placed.

5.1 System Requirements for Traffic Simulation

The current AR Sandbox system contained several modules for doing various tasks. These modules included a cut-and-fill mode for calculating earth moving requirements for large transportation projects, a topographical mode for displaying current sand height information within the sandbox, and a design mode for projecting and editing sections of roadway. The proposed upgrade would add modules giving users the ability to create full road networks and to place scene items without the need for a keyboard or mouse. These scenes would create the imagery projected in the AR Sandbox and form a digital infrastructure on which traffic simulations could be run. The AR Sandbox would need the following functionality:

- Object and road projections: The AR Sandbox should project a digital representation of user-created road networks and the objects around them that form a scene.
- Object recognition: The AR Sandbox should have a way to convert real-world objects that a user places in the sandbox into the appropriate model for a scene.
- Traffic simulation: The AR Sandbox should allow users to create scenes with road networks for the purpose of visualizing traffic simulations. Scene items should include streets, buildings, and traffic lighting but could be any arbitrary item. Traffic simulations should run in real time on a created road network and should be interactive to allow for changes at run time. The system should be able to perform traffic simulations with pre-loaded and user-built road maps. This simulation would need to be realistic enough to be used in a classroom setting. Traffic simulations should be able to be viewed mesoscopically, where colored sections of roadway indicate how well traffic is moving, or microscopically, where the behavior of individual cars can be observed.

5.2 System Design

The following components enable the traffic simulator on the AR Sandbox, as shown in figure 5-1:

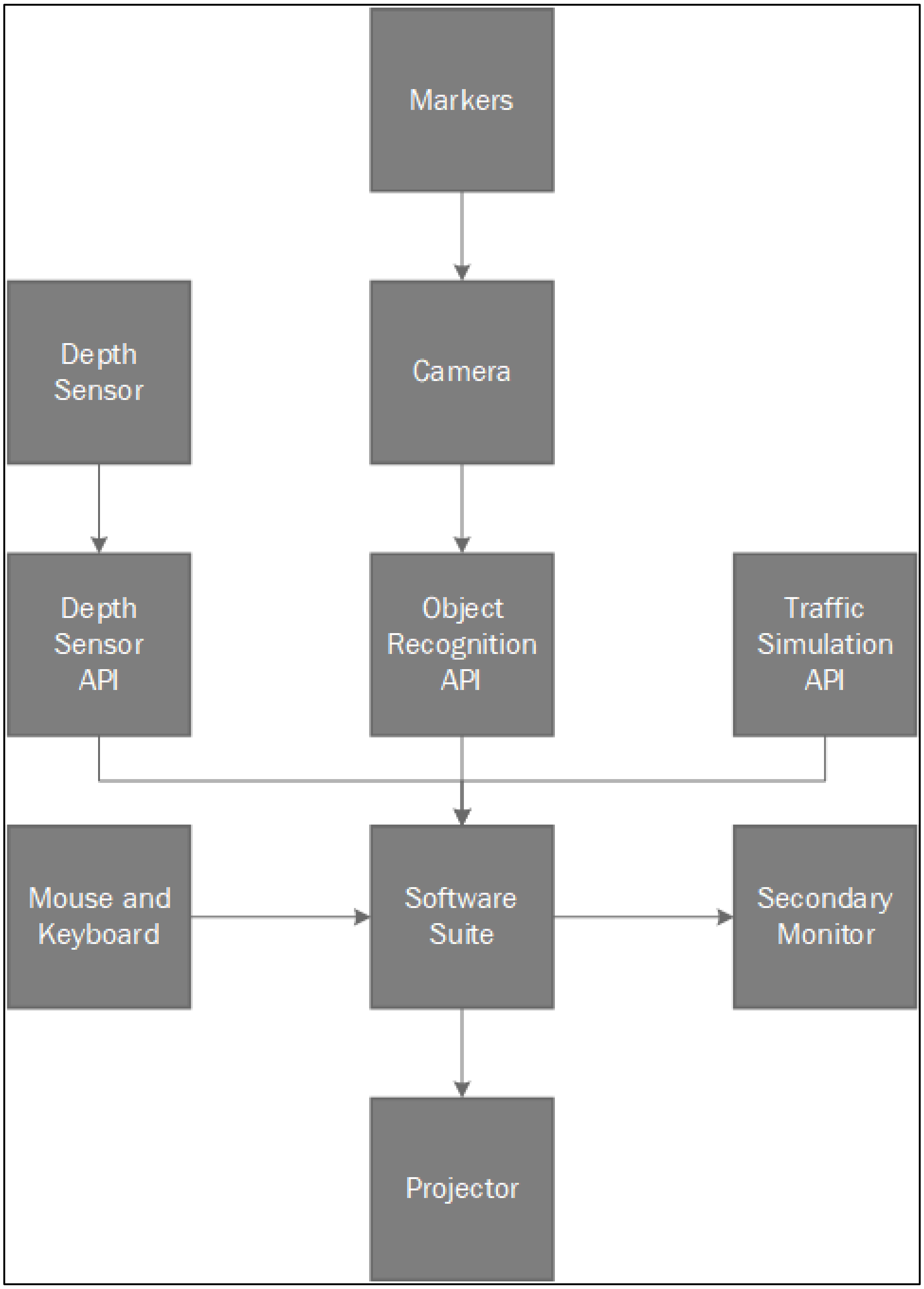

Figure 5-1. Overview of system architecture

1. Geometry Interpreter

The geometry interpreter handles translating a heightmap into a three-dimensional mesh that can be used in the Unity game engine to create and render scenes.

2. Height Look-up

The height look-up component handles requests for height data from a heightmap. A heightmap stores three-dimensional data in a two-dimensional format. Height look-up converts these two-dimensional data to a three- dimensional data format that can be used in Unity and other three-dimensional software.

3. Object Recognition API

Object recognition is achieved by using the Vuforia SDK. Images to be recognized are saved in Vuforia's Target Manager. Any images in the Target Manager can then be used to create markers that are recognizable to Vuforia. Once an image has been saved in the Target Manager, an Image Target can be created within the Unity Editor. Image Targets can be connected to game objects whose locations are based on the locations of the corresponding Image Targets and that can otherwise be used like any other game object.

4. Vertex/Fragment Shaders

The vertex and fragment shaders are responsible for applying colored elevation data onto a terrain mesh. Shaders use height data from the heightmap to determine the color of the mesh at each point. Depending on the current mode of operation, standard elevation heights or differences in height may be rendered. When a saved terrain is loaded, the shaders depict the difference between the sand in the AR Sandbox and the height data from the saved terrain. In normal operation the shaders display just the difference between various points in the AR Sandbox.

5. Traffic Simulation Mode

The traffic simulation mode loads and runs a created road network. The objects that describe a scene, such as roads, vehicles, and buildings, are built as 3-D objects and placed on a terrain sized appropriately for the network. As the simulation runs, various functions are available for users to call by using Vuforia markers. For instance, users are able to add and remove work zones on roads, interact with traffic lights, and change simulation parameters such as displaying mesoscopic and microscopic simulations.

6. Traffic Simulator

Traffic simulation is achieved by using the Simulation of Urban Mobility software suite (SUMO) running as a background process. Road networks are fed into SUMO after they have been generated. When a simulation mode has been selected, the simulation sends output display data to an XML file. The Unity application then reads that XML data and updates car positions accordingly.

7. Traffic Simulation API

To interact with, and obtain data from, the traffic simulator (SUMO), a traffic simulator controller class as an API is created. This controller is able to generate road networks, load road networks, edit existing road networks, specify simulation modes, and read simulation output. The API allows us to separate our concerns when talking to the SUMO and reading simulated data.

8. Road Network Importer

SUMO's NETCONVERT takes in premade networks from other applications and converts them to SUMO-style networks. In addition, SUMO may use a heightmap to convert a two-dimensional network into a three-dimensional network.

9. Marker-Based Road Network Editor

To edit an existing road network, the Marker-Based Road Network Editor uses object recognition (OR) to determine predefined markers and their locations. When the location is determined, the system matches that object to the node in the road network on which it is placed and performs a predefined action related to that object. For example, if the object is defined as a construction zone of a given length, then the system will alter that section of road to be a construction node and reload the network to SUMO to relaunch the simulation.

10. Mass-Haul Mode

The mass-haul mode creates a Mass-Haul Diagram and a difference map on the basis of the topography of the sandbox and the desired topography. This diagram shows red where sand needs to be removed and blue where sand needs to be added. This difference can be used with the generated Mass-Haul Diagram to demonstrate how to follow a Mass-Haul Diagram.

11. User Interface

The user interface (UI) for the AR Sandbox consists predominantly of markers used to interact with the AR scene. The design details for the UI are described in the following section.

12. Renderer

Scenes are rendered by the Unity engine. Objects making up the scene are viewed by a virtual camera within Unity, and this view is projected into the sandbox. Any changes made to the scene are routed through the scene in Unity and rendered accordingly.

5.3 Demonstration of Traffic Visualization

The traffic simulation and visualization component that was developed for the AR sandbox enables users to select an area of interest for analysis by using a front-end interface provided by Open Street Map (OSM) and to view microscopic and macroscopic visualizations of traffic flow in the selected network. Users are also able to interact with the traffic network by using handheld markers. This section describes a case study demonstration in which the developed methodology was applied to the town of Corvallis, Oregon. The following three subsections describe (1) the area of study and traffic flow, (2) micro- and macroscopic visualization of traffic flow, and (3) interacting with traffic flow by using augmented reality.

5.3.1 Selecting the Area of Study and Traffic Flow

OpenStreetMap's (OSM) interface was used to enable users to select the region of interest for analysis of traffic flow, as shown in figure 5-2, which displays a map of the region. Figure 5-3 shows the user input options for the region of interest and the traffic options that were used in the simulation. The user could specify the number of cars, trucks, buses, motorcycles, bicycles, and pedestrians for the simulation. While the original scope of work intended to match these numbers with the actual traffic flows for the city of Corvallis, that could not be done because of the COVID-19 lockdown that was imposed during the final portion of the project. That work will therefore be undertaken in the future.

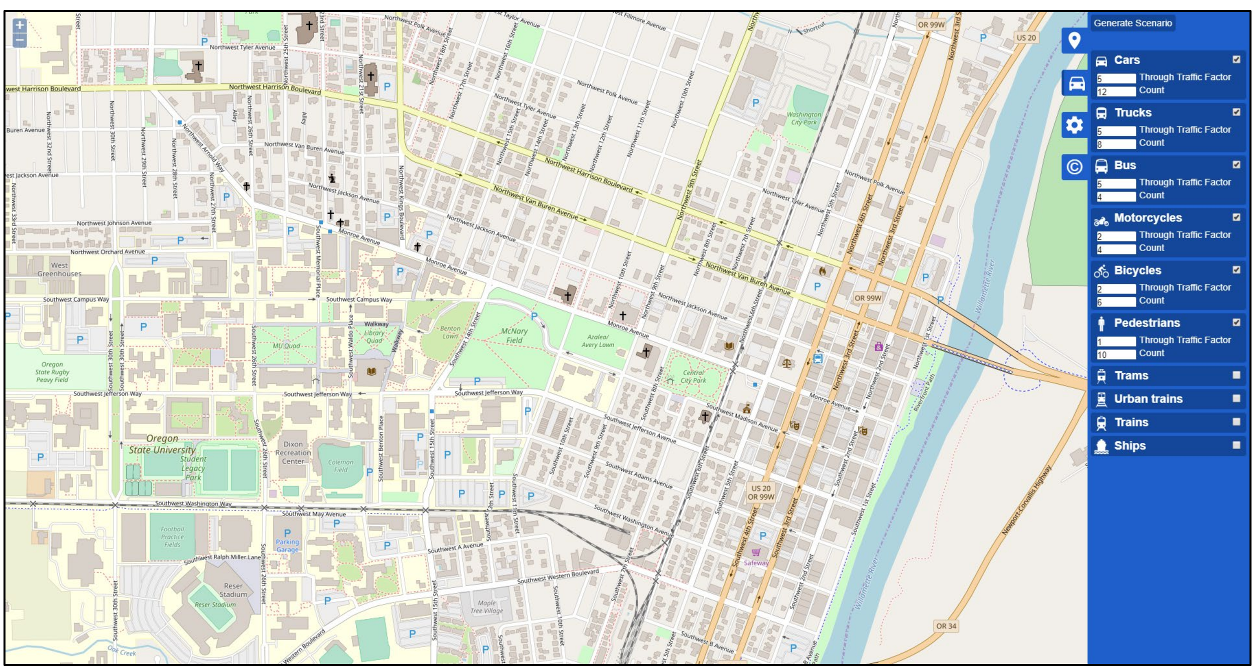

Figure 5-2. OpenStreetMap interface to allow users to select the region and traffic simulation parameters

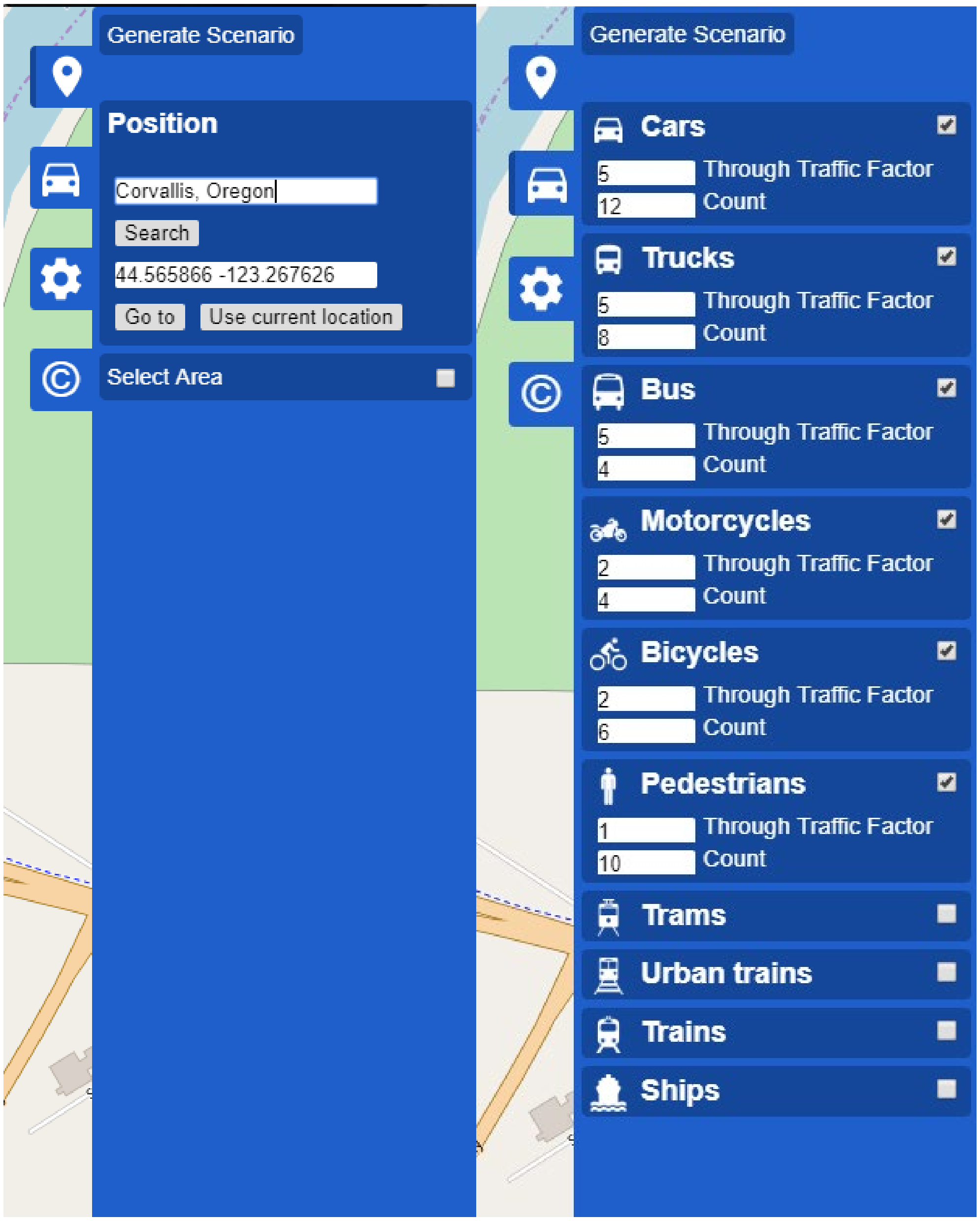

Figure 5-3. Close-up view of user input options for region and traffic parameter selection

5.3.2 Visualization of Traffic Flow

Once the traffic simulation parameters and the region had been selected, the OSM map was converted into a roadway network for use in the SUMO traffic simulation tool inside the Unity game engine. Figure 5-4 shows the roadway network used inside SUMO that was generated from the previously selected region from OSM.

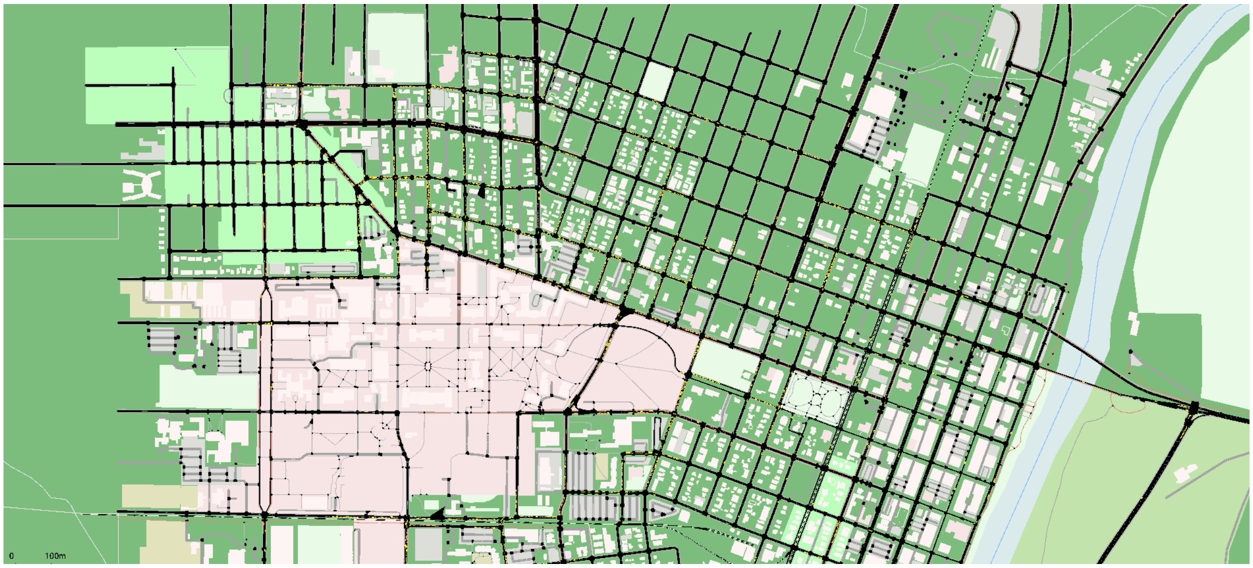

Figure 5-4. Traffic network for simulation in SUMO

Figure 5-5 shows a closer view of the same network with the flow of vehicles based on user input. These can be seen in yellow in figure 5-5.

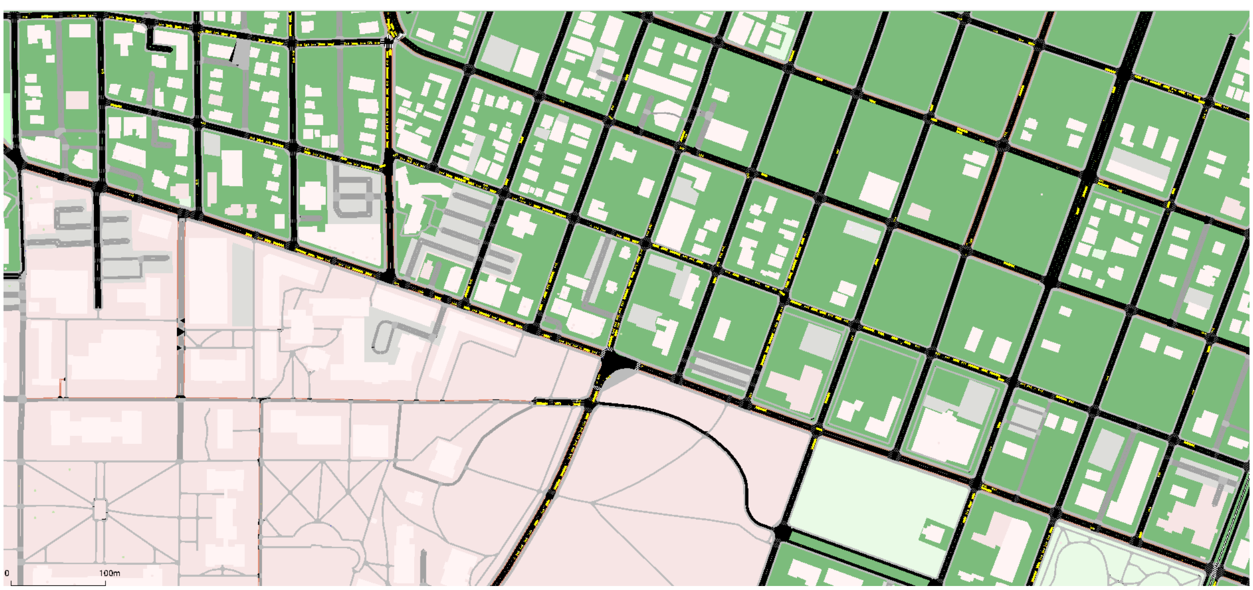

Figure 5-5. Microscopic simulation shown in the traffic network

While figures 5-4 and 5-5 show 2-D views of the traffic simulation, a 3-D view was obtained and is shown in figures 5-6 and 5-7.

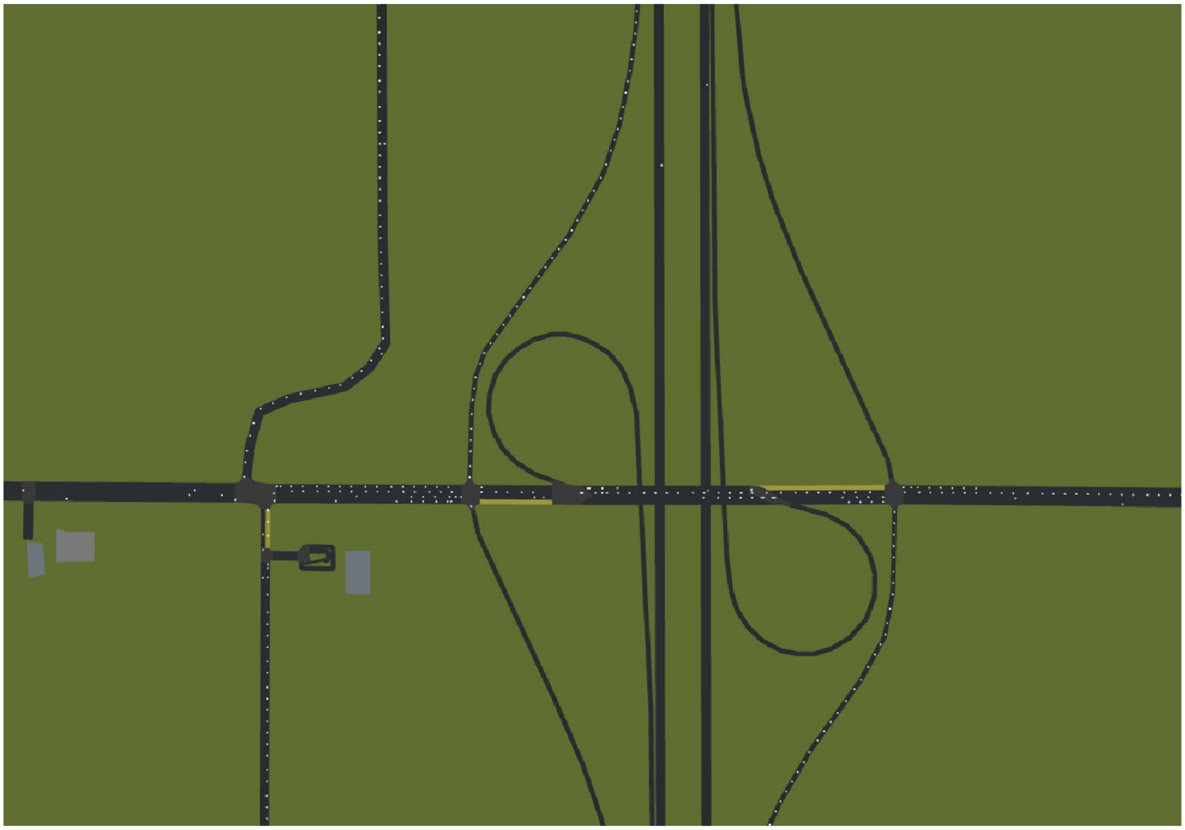

Figure 5-6. 3-D microscopic simulation overview

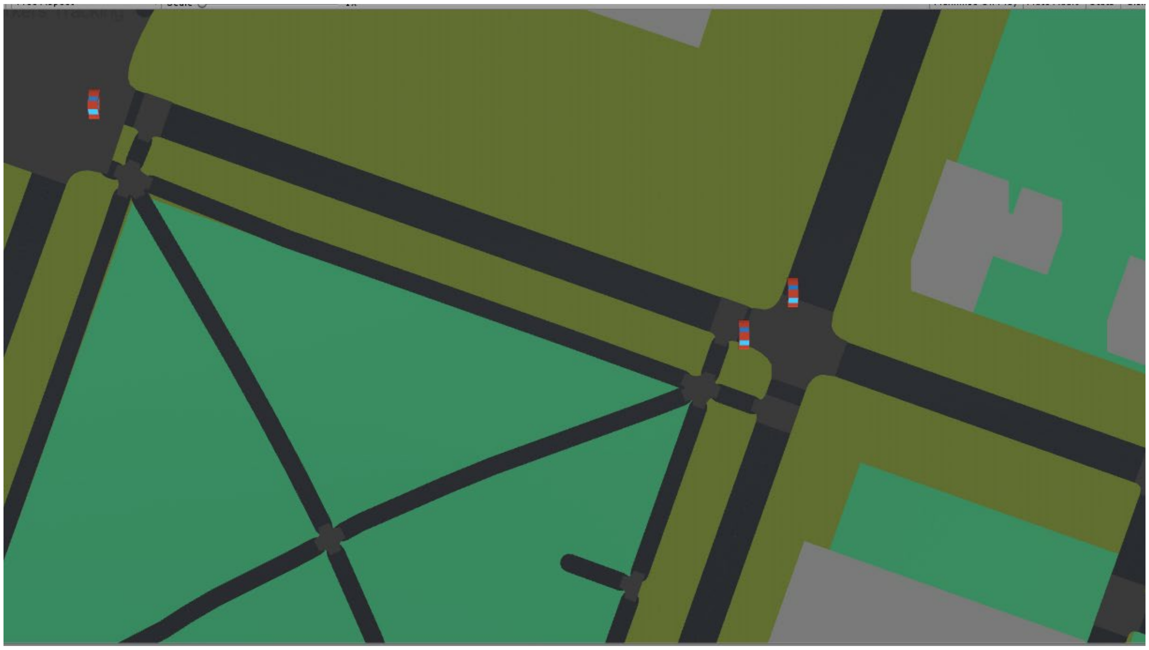

Figure 5-7. Close-up of 3-D microscopic simulation

Finally, macroscopic or mesoscopic views of traffic flows were visualized in which different road segments were colored green or red to indicate free flowing or congested traffic, as shown in figure 5-8.

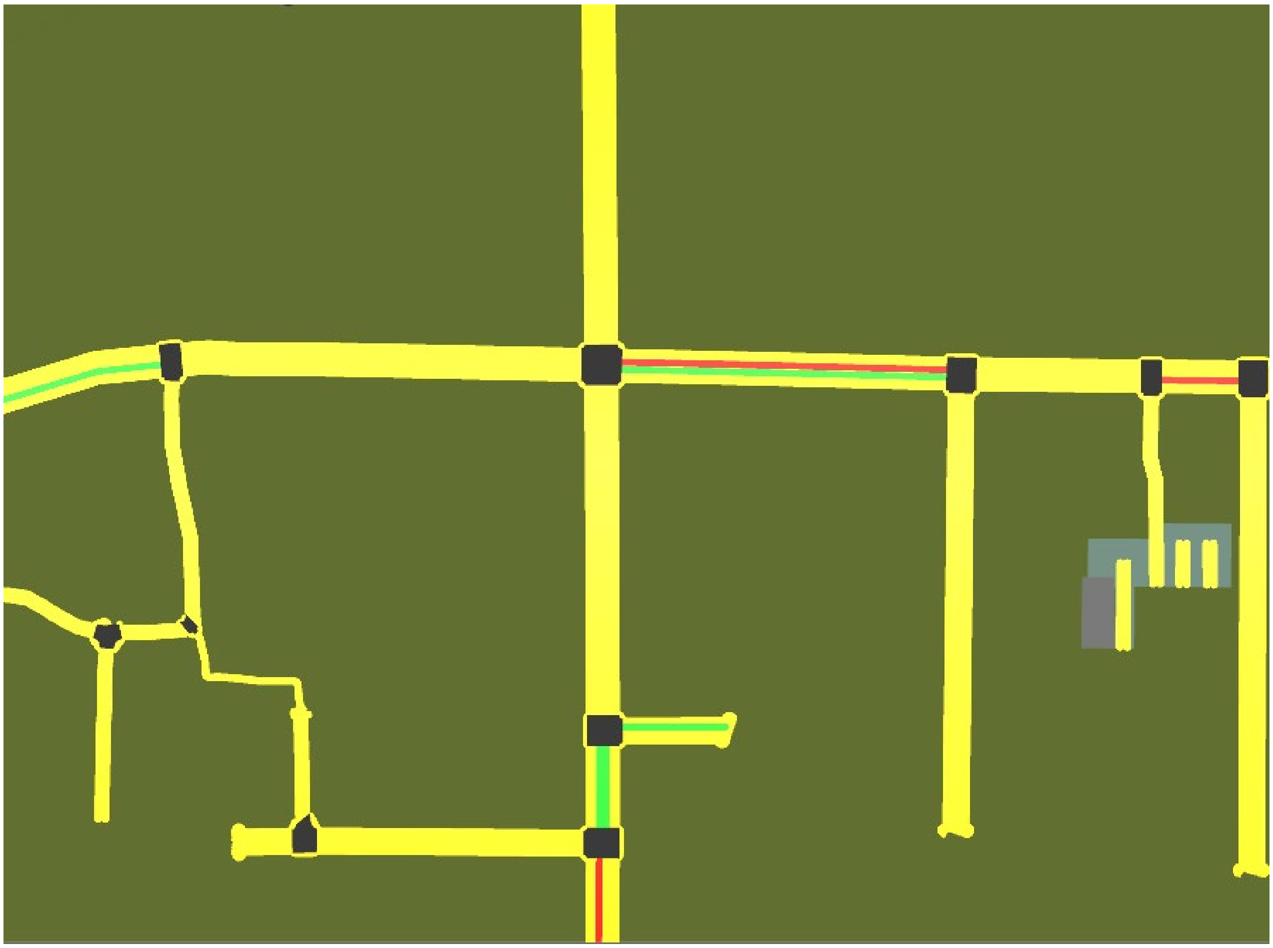

Figure 5-8. Mesoscopic view of traffic flow in the network

The above views showcase the various means by which user-defined areas from the real world can selected and used to create various traffic simulations based on user input of traffic volumes.

5.3.3 Interacting with Traffic Simulation Using Augmented Reality

After the traffic simulations and visualizations were created, they were projected onto the sandbox using the projector. The goal was to enable multiple users to huddle around the sandbox to visually observe and analyze the traffic flow patterns. Such a projection is shown in figure 5-9.

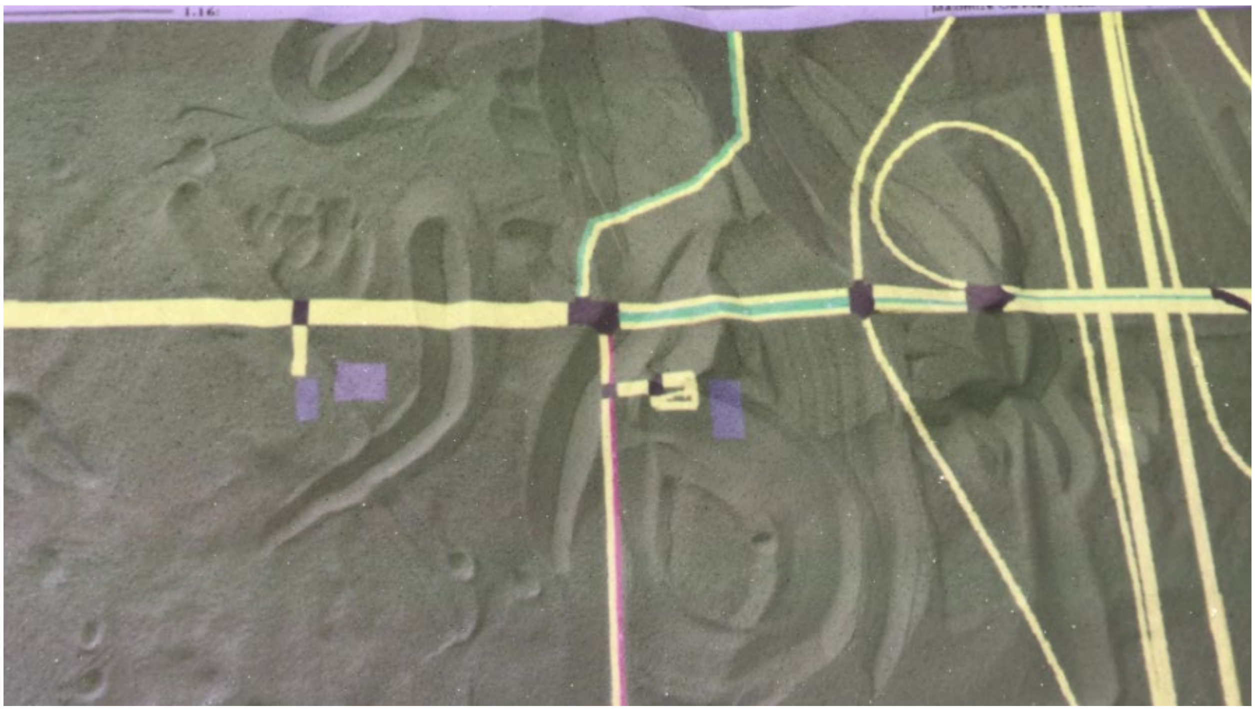

Figure 5-9. Projection of a traffic simulation on the sandbox

To enable a greater degree of interaction with the simulation by multiple users simultaneously, an AR-based interaction system was created. This system enables users to drop printed AR markers upon roadway segments and turn them into work zones with reduced traffic capacity. The SUMO simulation then routes traffic accordingly, thus showing the effects of the work zone on traffic flow in real time. Figure 5-10 shows a user dropping an AR marker on a road segment. The creation of the work zone on the segment is indicated by the half of that segment shown in yellow.

Figure 5-10. Using an AR marker to interact with the simulation

While simulation results were not analyzed in depth to validate the effects of work zones on road networks, this research did show that AR markers could provide a new means of interacting with simulations. This was proved by the reduction in capacity of the selected roadway segment. The success of AR markers in interacting with the simulation also motivated the development of a novel tangible user interaction mechanism for navigating through the AR sandbox applications with the development of AR menu systems, as shown in figure 5-11.

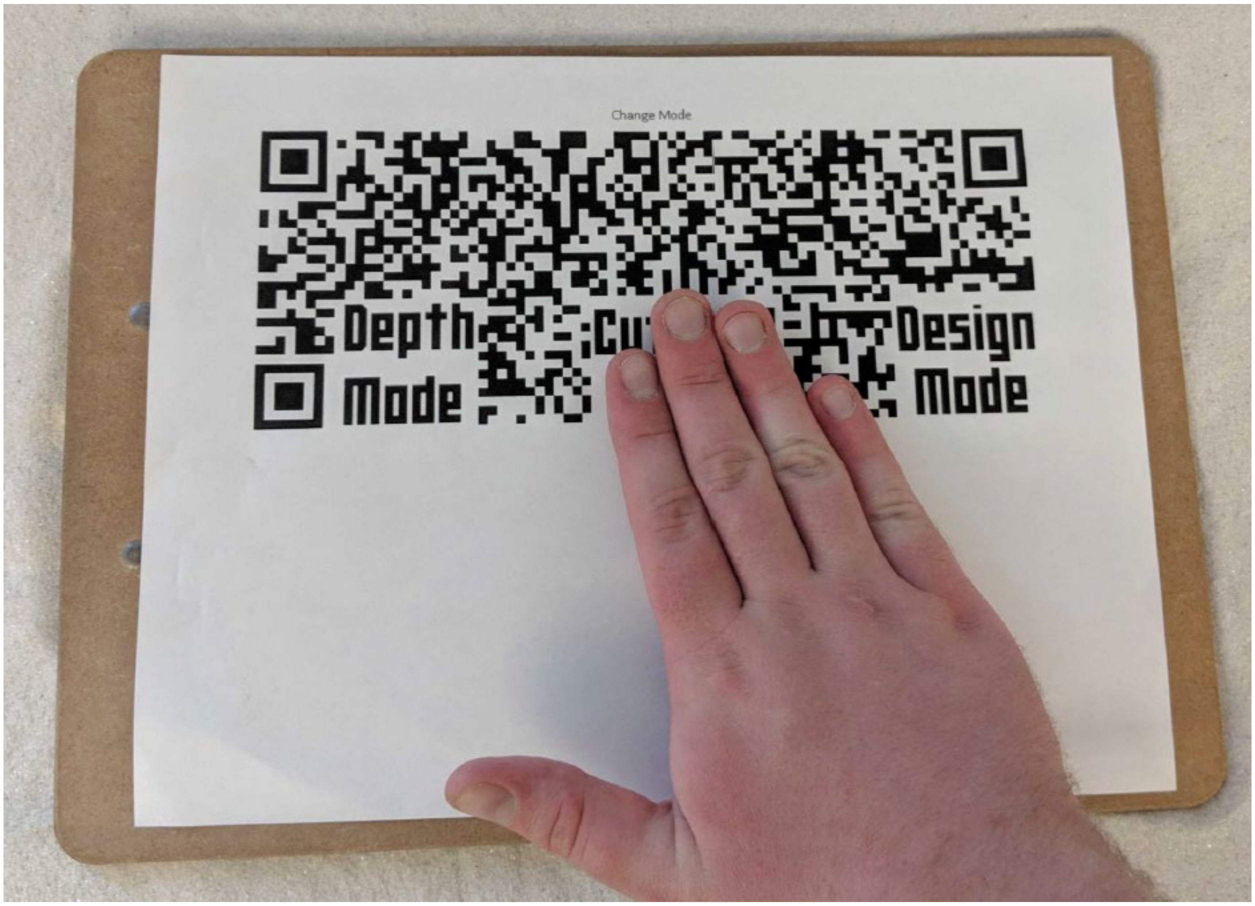

Figure 5-11. AR-based menu system for navigating through the program

Future work could expand upon the novel use of AR to provide greater functionality and more intuitive interactivity for users of the AR Sandbox.

Chapter 6: Conclusions

This report describes the development of AR sandbox applications for earthmoving and traffic simulations that are intended to be used for educational and visual analysis purposes. To this end, the hardware development of the sandbox is first described, followed by the architecture of the software required for the AR sandbox to work. Following the development of the sandbox itself, two applications geared toward transportation engineering were developed for the AR Sandbox.

The first application provides users with a novel means of visualizing earthmoving calculations that are required before the construction of roadways. Cut and fill sections can be visualized along the centerline of the roadway, and users can obtain real-time updates of cut and fill sections as the work of earthmoving progresses. Future work in this area will relate to testing the earthmoving visualization as an educational aid for teaching concepts related to the construction of highways and the estimations involved. Future work will also need to be performed to analyze the mass-haul table that is created.

The second application is a novel traffic simulation and visualization tool that was created for the sandbox. Users can select a region of interest and specify traffic parameters and view traffic simulations based upon their input onto the sandbox. Furthermore, they can interact with the traffic simulation by using a novel user interaction method enabled by marker-based augmented reality. Specifically, users can drop an AR marker upon a roadway segment to indicate to the simulation that the segment contains a work zone and then see the effects of the work zone on traffic flow in real time. Future work on this application will include the use of historical traffic data for the selected case study for the city of Corvallis and the validation of simulation results. Also, user satisfaction with the novel AR-based interaction method should be evaluated to assess the appropriateness of the developed tool for multi-user interactions with traffic simulations.

Despite the limitations of testing that could not be performed because of a lack of access to the sandbox during the last six months of the project, the research efforts did result in novel applications for transportation engineering education and visualization that use the AR Sandbox. It is expected that the work presented in this report will enable future development of novel visualization methods for transportation engineering.

References

Abdelmohsen, A. Z., & El-Rayes, K. (2016). Optimal trade-offs between construction cost and traffic delay for highway work zones. *Journal of Construction Engineering and Management*, *142*(7), 05016004.

Ables, A. (2017, October). Augmented and virtual reality: Discovering their uses in natural science classrooms and beyond. In *Proceedings of the 2017 ACM SIGUCCS Annual Conference* (pp. 61-65).

Bae, J., Choi, K., & Oh, J. H. (2017). Multicontextual Machine-Learning Approach to Modeling Traffic Impact of Urban Highway Work Zones. *Transportation Research Record*, *2645*(1), 184-194.

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

Peurifoy, R. L., & Ledbetter, W. B. (2003). Construction Planning, Equipment, and Methods (McGraw-Hill Series in Civil Engineering).

Sánchez, S. Á., Martín, L. D., Gimeno-González, M. Á., Martín-Garcia, T., Almaraz-Menéndez, F., & Ruiz, C. (2016, November). Augmented reality sandbox: a platform for educative experiences. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 599-602).

Savova, D. (2016, June). AR Sandbox in Educational Programs for Disaster. In *6th International Conference on Cartography and GIS* (p. 847).

UC Davis (2018) *Augmented Reality Sandbox* https://arsandbox.ucdavis.edu/, Accessed 4/25/2018

Woods, T. L., Reed, S., Hsi, S., Woods, J. A., & Woods, M. R. (2016). Pilot study using the augmented reality sandbox to teach topographic maps and surficial processes in introductory geology labs. *Journal of Geoscience Education*, *64*(3), 199-214.

Zhang, G., Gong, J., Li, Y., Sun, J., Xu, B., Zhang, D., ... & Yin, B. (2020). An efficient flood dynamic visualization approach based on 3D printing and augmented reality. *International Journal of Digital Earth*, 1-19.