

Old Dominion University

## ODU Digital Commons

---

Modeling, Simulation and Visualization Student Past Proceedings of the MSV Student Capstone  
Capstone Conference Conference

---

Apr 18th, 12:00 AM

### Proceedings, MSVSCC 2019

Old Dominion University, Department of Modeling, Simulation & Visualization Engineering

Old Dominion University, Virginia Modeling, Analysis and Simulation Center

Follow this and additional works at: <https://digitalcommons.odu.edu/msvcapstone>



Part of the [Engineering Commons](#)

---

#### Recommended Citation

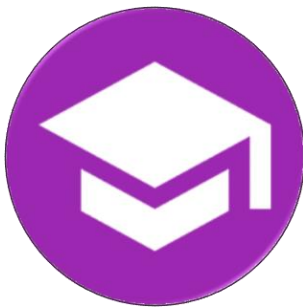
Old Dominion University. Department of Modeling Simulation & Visualization Engineering, & Virginia Modeling Analysis and Simulation Center. (2019, April 18). Proceedings, MSVSCC 2019. 13th Annual Modeling, Simulation & Visualization (MSV) Student Capstone Conference, Virginia Modeling, Analysis & Simulation Center, Suffolk, VA. 144 pp. <https://doi.org/10.25776/h6kt-wg87>

This Other is brought to you for free and open access by the Virginia Modeling, Analysis & Simulation Center at ODU Digital Commons. It has been accepted for inclusion in Modeling, Simulation and Visualization Student Capstone Conference by an authorized administrator of ODU Digital Commons. For more information, please contact [digitalcommons@odu.edu](mailto:digitalcommons@odu.edu).

# MSVSCC 2019

## THE 13<sup>TH</sup> ANNUAL MODELING, SIMULATION & VISUALIZATION STUDENT CAPSTONE CONFERENCE

April 18, 2019  
VMASC, Suffolk, VA



ABM, Education  
& Training



General Sciences  
& Engineering



Infrastructure Security  
& Military



Gaming  
& Virtual Reality



Transportation,  
Business & Industry



Medical  
Simulation

# PROCEEDINGS

# MSVSCC 2019

## PROCEEDINGS

Old Dominion University Department of Modeling, Simulation & Visualization Engineering (MSVE) and the Virginia Modeling, Analysis and Simulation Center (VMASC) held the 13<sup>th</sup> annual Modeling, Simulation & Visualization (MSV) Student Capstone Conference on April 18, 2019.

The Conference featured student research and student projects that are central to MSV. Also participating in the conference were faculty members who volunteered their time to impart direct support to their students' research, facilitated the various conference tracks, served as judges for each of the tracks, and provided overall assistance to the conference.

Appreciating the purpose of the conference and working in a cohesive, collaborative effort, resulted in a successful symposium for everyone involved.

These proceedings feature the works that were presented at the conference.

Sincerely,

Dr. Yuzhong Shen  
Capstone Conference Chair  
[yshen@odu.edu](mailto:yshen@odu.edu)

Daniel Perez  
Capstone Conference Student Chair  
[dpere013@odu.edu](mailto:dpere013@odu.edu)



# MSVSCC 2019

## KEYNOTE SPEAKER



### ANGEL SAN JOSE MARTIN

received his Master of Science in Operation Research by the Naval Postgraduate School in Monterey, CA, in 2001. He is the Head of the Science and Technology (S&T), Modelling and Simulation (M&S) Section of the NATO Headquarters Supreme Allied Command Transformation in Norfolk, VA. He is a NATO civil servant and former Spanish Navy commander. He is responsible for the M&S activities on the Allied Command Transformation (ACT), as well as the coordination with the Science and Technology Organization of NATO.

### KEYNOTE: Simulating Alliance Security. The Movie.

Modeling and Simulation opportunities in defense and security. Hot topics for the Alliance and allies that require attention from the M&S community. Beyond training, focus nowadays is expanding to M&S in support of analysis and wargaming to enhance decision making in NATO. Making M&S, AI, Data Sciences and optimization methods to work together is an essential element for innovation in NATO. Experts are required and opportunities will emerge.





# MSVSCC 2019

## BEST PAPERS PER TRACK

The tracks that the papers were divided up into included the following:

- Agent Based Modeling, Education & Training
- Gaming & Virtual Reality
- General Sciences & Engineering
- Infrastructure Security & Military Application
- Medical Simulation
- Transportation, Business & Industry

For each track there were awards given out: The Best Paper award, and the Best Presentation Award. Those recipients for best paper are listed below by track.

### **ABM, Education and Training**

Christine Odenwald, Germaine Badio, Jaron Stevenson, Travis Sullivan, Peter Marchione, Elise Feldt and Zhaohui Hu. "Virtual Autonomous System Training (VAST) Environment"

### **Gaming & VR**

Shan Liu, Yuzhong Shen and Jim Leathrum. "A Review of Modeling and Simulation Tools for Autonomous Vehicles"

### **General Sciences & Engineering**

Erik Jensen, Evan Coleman and Masha Sosonkina. "Randomized Asynchronous Linear Solvers"

Daniel Perez. "Capsule Networks: Review of the Structures and Visualization of Features"

### **Infrastructure Security & Military Application**

Yan Lu, Jonathan Graham and Jiang Li. "Deep Learning Based Malware Classification Using Deep Residual Network"

### **Medical Simulation**

Haben Yhdego, Nahom Kidane, Dr. Rick McKenzie and Michel Audette. "Deep Learning Method for ECG-based Virtual Pathology Stethoscope Tracking in Patient Heart Auscultation"

# MSVSCC 2019

## GENE NEWMAN AWARD

### Overall Best Paper – The Gene Newman Award

The overall best paper is awarded the Gene Newman award. This award was established by Mike McGinnis in 2007; the award is given for overall best presentation, best paper, and research contribution. The Gene Newman Award for Excellence in M&S Research is an award that honors Mr. Eugene Newman for his pioneering effort in supporting and advancing modeling and simulation. Mr. Newman played a significant role in the creation of VMASC by realizing the need for credentialed experts in the M&S workforce, both military and industry. His foresight has affected both the economic development and the high level of expertise in the M&S community of Hampton Roads. The Students receiving this award will have proven themselves to be outstanding researchers and practitioners of modeling and simulation.

For the 2019 Student Capstone Conference, The Gene Newman Award went to:

**Yan Lu**, for her paper entitled 'Deep Learning Based Malware Classification Using Deep Residual Network'

# MSVSCC 2019

## PROCEEDINGS

	<b>General Sciences &amp; Engineering</b>	<b>PAGE 1</b>
	<b>Gaming &amp; Virtual Reality</b>	<b>PAGE 50</b>
	<b>Medical Simulation</b>	<b>PAGE 65</b>
	<b>Agent Based Modeling, Education &amp; Training</b>	<b>PAGE 81</b>
	<b>Infrastructure Security/Military Application</b>	<b>PAGE 113</b>
	<b>Business, Industry &amp; Transportation</b>	<b>PAGE 132</b>



**OLD DOMINION**  
UNIVERSITY

# MSVSCC 2019

## PROCEEDINGS



## General Sciences & Engineering

Page 2	<b>Applying a Test and Evaluation Framework to an Unmanned Maritime System</b> Ntiana Sakioti, Andrea Robey, Thomas Laverghetta, Nathan Gonda and James Leathrum
Page 12	<b>A Game Theoretic Model of Pathogen Prevalence as an Outcome of Strategic Interactions Between a Bacterium and a Tick Vector</b> Sara Simmons and Holly Gaff
Page 14	<b>Band Selection in Multispectral Images for Seagrass Detection Using Deep Convolutional Neural Network</b> Kazi Aminul Islam, Daniel Perez, Victoria Hill, Blake Schaeffer, Richard Zimmerman and Jiang Li
Page 16	<b>Verbal Instruction to Perform Specific Running Styles Impacts Reaction Forces During Habitual and Non-Habitual Running</b> Collin Bowersock, Steven Morrison and Daniel Russell
Page 28	<b>Embossed Construction and Explorative Visualization of 3D Structures from 2D Images</b> Ahmet Saglam
Page 30	<b>Anisotropic Mesh Adaptation Pipeline for the 3D Laminar Flow over a Delta Wing</b> Christos Tsolakis and Nikos Chrisochoides
Page 38	<b>Capsule Networks: Review of the Structures and Visualization of Features</b> Daniel Perez

# APPLYING A TEST AND EVALUATION FRAMEWORK TO AN UNMANNED MARITIME SYSTEM

Ntiana Sakioti  
Andrea H. Robey  
Thomas Laverghetta  
Nathan Gonda  
James F. Leathrum, Jr.

Department of Modeling, Simulation, and  
Visualization Engineering  
Old Dominion University  
5115 Hampton Boulevard Norfolk, VA, USA  
[nsaki001@odu.edu](mailto:nsaki001@odu.edu), [arobe004@odu.edu](mailto:arobe004@odu.edu),  
[tlave002@odu.edu](mailto:tlave002@odu.edu), [ngond002@odu.edu](mailto:ngond002@odu.edu),  
[jleathru@odu.edu](mailto:jleathru@odu.edu)

## ABSTRACT

In order for an autonomous system to be confidently deployed in real world scenarios extensive behavioral testing must be conducted to ensure reliability. This is generally accomplished through the use of simulation followed by or in conjunction with testing in a physical environment. In this paper, the conceptual integration of an unmanned surface vehicle's software to a test and evaluation framework is presented. The utilization of the suggested framework in this application enables testing throughout the virtuality-reality continuum, enabling behavioral testing early in the lifecycle, seamlessly integrating autonomous software with either a test environment or the physical vehicle. This reduces safety concerns as well as cost. This paper considers the utilization of the framework by mapping autonomous software developed for the Maritime RobotX competition to the framework and discussing the potential benefits of this activity.

**Keywords:** autonomous systems, virtuality-reality continuum, test and evaluation framework, unmanned surface vehicle

## 1 INTRODUCTION

Due to their nature and expected behavior, autonomous systems introduce various challenges in testing and validation. Expected behavior must be studied for occurrence reliability, while the system must safely adapt to external factors and possible complications. Throughout the development cycle, testing is performed in both virtual and physical environments, introducing inputs and models to improve autonomous software response to both expected and unexpected inputs.

Utilization of a test and evaluation (T&E) framework is proposed so that system and subsystem testing can be conducted with augmented virtual and real data. This is accomplished by isolating autonomous software components from each other as well as from the operating environment, and ensuring that the autonomous software does not know the source of information. Isolating components and data enables testing to be conducted throughout the virtuality to reality spectrum, increasing the variety of tests and building confidence in the autonomous software (Leathrum et. al 2018a.).

The autonomous software that will utilize and be mapped onto the framework provides the autonomy to an unmanned surface vehicle (USV). The USV's software structure was designed around functionalities needed to complete the Maritime RobotX competition tasks (Maritime RobotX n.d.). Essential capabilities that the surface vehicle should demonstrate are listed below:

- Obstacle avoidance
- Navigation and control (circling an object, traversing through a gate, docking etc.)
- Object and symbol detection (buoys, totems, docks, a hole cutout etc.)
- Underwater beacon identification
- Station keeping (maintaining current position)

The remainder of the paper is organized into five sections. Section 2 details the framework that the authors propose for use in USV testing, while Section 3 presents the USV software architecture including the separate subsystems and methods of communication between them. Section 4 maps the surface vehicle's software architecture onto the proposed framework, with the additional testing capabilities of such application presented in Section 5, followed by concluding remarks in Section 6.

## **2 FRAMEWORK DESIGN**

This section presents the test & evaluation framework first described in (Leathrum et al. 2018a), (Leathrum et al. 2018b), and (Laverghetta et al. 2019). The framework facilitates the implementation of an effective test harness for different scenarios in the virtuality-reality spectrum. The main benefits to be reaped from the utilization of the framework are listed below:

- Isolating components of the autonomous software for testing purposes
- Isolating the autonomous software from its operating environment
- Isolating the autonomous software from knowledge of the source or use of information

### **2.1 Autonomous Software Model**

The approach used is to model the control system using a pipeline of the sense, plan, act functional modules (Gat 1998). At a high level, the system consists of the autonomous software, external environment, and hardware including sensors, actuators, and vehicle dynamics/state information. The vehicle dynamics and state information include physical attributes of the vehicle such as velocity, orientation, and fuel level. The software can then be further decomposed into the functional modules that generate and execute the plan for the system, as well as the world representation -an internal representation of the external environment and internal vehicle state. The world representation could include a panoramic view of distances to physical boundaries, sets of recognized objects and their computed attributes, or a map of the environment based on past experiences of the robot. The modules are described below:

- Sense – Computes a perception of the environment based on incoming raw data from the hardware sensors. This involves mapping the raw sensor data to the world representation
- Plan – Generates a plan composed of actions based on the system's current world representation, operational goals, and past experiences
- Act – Executes the plan by converting actions to control signals to send to the actuators

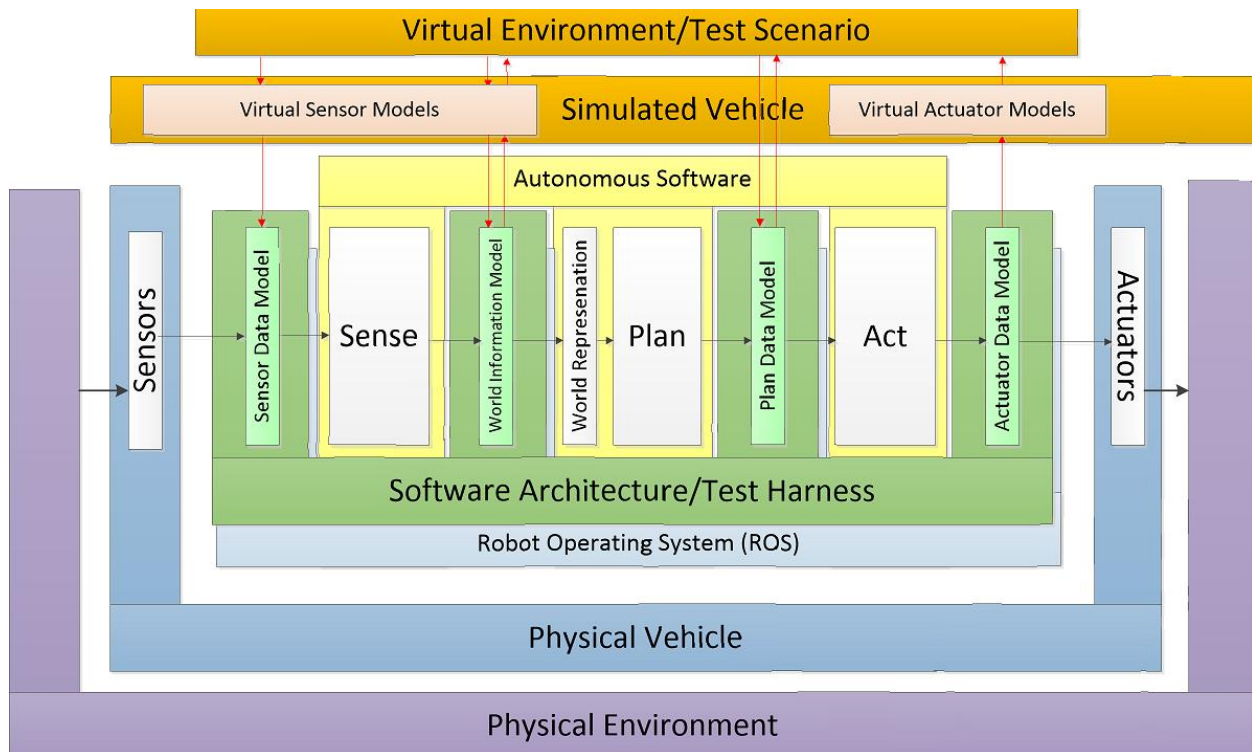
This model described above is employed to illustrate the ensuing work; however, the work is not solely relegated to this model and may include other stage decompositions such as the inclusion of a perception stage.



## 2.2 Framework Structure

The focus of the framework is to provide the capability to test each module in both isolation as well as during its integration to external system(s) without the need for reconfiguration or alteration the autonomous model, which can be intensive and time-consuming when applying it to different scenarios. Figure 1 illustrates a high-level view of this architecture. Four major sections are highlighted below.

- Physical Systems
  - The *Physical Vehicle*
    - *Sensors* – provide information for the autonomous software
    - *Actuators* – receive information as control signals to alter the vehicle’s operation
    - *Physical Plant* – provide power and motion
  - The *Physical Environment* – represent all the external factors and stimuli that can influence or be influenced by the physical vehicle.
- Virtual Systems - represent generated or simulated versions of the physical counterparts
  - *Simulated Vehicle*
    - *Virtual Sensor Models* – represent behavior of physical sensors
    - *Virtual Actuator Models* – represent the behavior of physical actuators
  - *Virtual environment* – represents a generated version of the environment external to the vehicle
- Autonomous Software – software implementing the system autonomy decomposed into *Sense*, *Plan*, and *Act*
- Test Harness – software framework isolating the autonomous software from its operating environment, and modules of the autonomous software from each other, to allow a control of information for testing purposes



**Figure 1. Test & Evaluation Architecture**

### 3 USV SOFTWARE DESIGN

This section details the autonomous surface vehicle's software architecture, which was centered around the Robot Operating System (ROS), a robotics middleware that facilitates communication via publish-subscribe services (ROS n.d.). Through the middleware, subsystem nodes communicate by publishing or subscribing to a topic characterized by a distinct name. Topics have anonymous publish-subscribe semantics, with nodes not generally aware of who they are communicating with (ROS reference).

The autonomous software was composed of seven major subsystems: AI, LiDAR, Guidance, Vision, Hydrophone, GPS, and Chassis. Each subsystem was in turn comprised of one or more nodes, each of which communicates necessary information through a variety of distinct topics. Simulation was extensively utilized to test USV response and behaviors. Subsystems that interfaced with hardware components contained separate nodes that mimicked sensor information or hardware response, with communication occurring through simulation specific topics. Figure 2 illustrates the autonomous system's software architecture, the components of which are described further in this section.

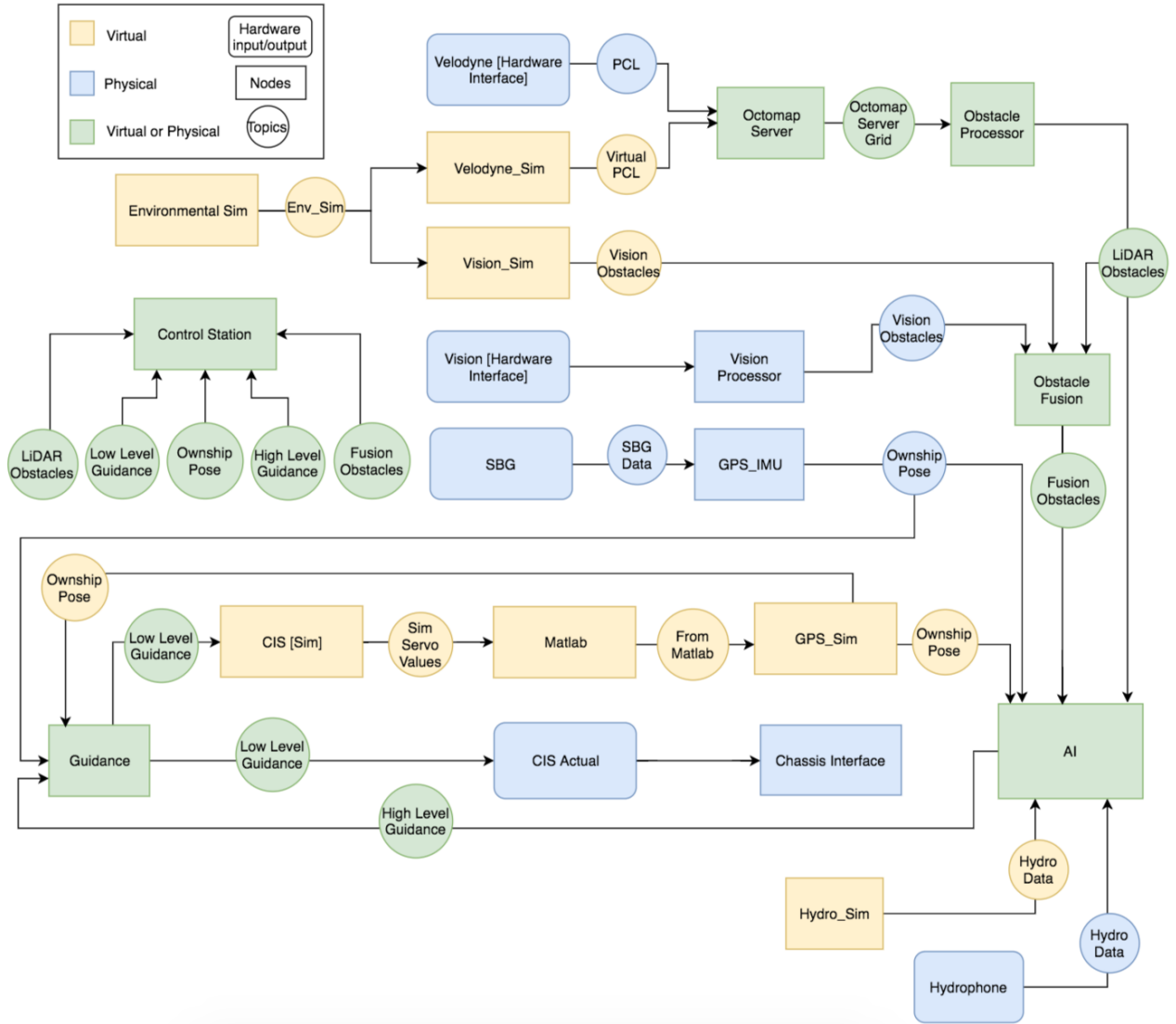


Figure 2. Autonomous Software Architecture

The LiDAR subsystem's function was to detect and classify environment objects, and consisted of four nodes. The Velodyne node directly interfaces with the LiDAR sensor, publishing point cloud data to the PCL topic, to which the Octomap Server node was subscribed to. The Octomap Server populated and published occupancy grid data to the Occupancy Grid topic, to which the Obstacle Processing node was subscribed to. Within the aforementioned node an obstacle list was realized and published to the LiDAR Obstacles topic that was listened on by other subsystems, such as Fusion and AI. To simulate sensor data for virtual testing, the Velodyne Sim. node was utilized. This node published point cloud data to the Virtual PCL topic; during testing in simulation the Octomap Server therefore listens to that topic instead of PCL.

The Vision subsystem's function was to aid in object recognition while performing color detection, and was comprised of two nodes, Vision and Vision Sim. The nodes published an obstacle list to Vision Obstacles, which was subscribed to by the Fusion and AI subsystem.

The Fusion subsystem in turn is implemented as one node and combines the object data generated by the LiDAR and Vision subsystems to provide a more comprehensive object list. Subscribing to Vision Obstacles and LiDAR Obstacles, it combines the two obstacle lists into a single list. That obstacle list is then published to the Fusion Obstacles topic, which is subscribed to by AI.

The GPS subsystem provides USV location data and is comprised of three nodes. The SBG node directly interfaces with the GPS sensor, publishing data to the SBG Data topic, which is then subscribed to by the GPS IMU node. The GPS IMU node converts this data into a message that includes information such as latitude longitude, yaw and pitch, and publishes that message to the Ownship Pose topic. The Ownship Pose topic is subscribed to by AI and Guidance. On the virtual side, the GPS device is represented on a simulated system built in MATLAB. The system publishes data similar to that of the GPS sensor which the GPS Sim node subscribes to, once again converting the data into the expected message which is published on the Ownship Pose topic.

The Hydrophone subsystem detects and calculates the strength of underwater ping data. It is comprised of two nodes, Hydro and Hydro Sim. Hydro directly interfaces with the hardware, calculating the strength of a detected ping and publishing it to the topic Hydro Data. Hydro Sim similarly acts as the simulated hardware interface, calculating and publishing the strength on the same topic.

The Guidance subsystem translates high-level USV control such as line-following or heading-speed control to low-level values in the form of percent effort of the motors and steering angle. The subsystem is implemented as one node that subscribes to the High Level Guidance topic, from which data is translated to low-level values. The translated data is published to the Low Level Guidance topic, which is subscribed to by the Chassis nodes and GPS Sim node.

The Chassis subsystem serves as an interface to the motor servos, and is comprised of two nodes, CIS and CIS SIM. Both nodes subscribe to Low Level Guidance to translate low-level motor values to servo values. The CIS sends those values to the servos while CIS SIM sends the values to the simulated system in MATLAB.

The AI subsystem is considered the "brain" of the USV. It receives sensor information from all the peripherals by subscribing to topics such as LiDAR Obstacles, Vision Obstacles, Fusion Obstacles, Ownship Pose, and Hydro Data. Utilizing the data gleaned of the topics and depending on the desired behavior decides USV movement or lack thereof. The movement data is published to High Level Guidance so that the required motion can be executed.

To monitor system behavior during physical testing a Control Station node was used that subscribed to all subsystem main topics and displayed important data to the screen for real-time data tracking. Implemented using Qt A visual of the physical environment is also displayed consisting of the USV as well as the environment components that were detected. The virtual environment is used to emulate the physical environment that the boat would be in in order to test off-shore. It subscribes to the Env Node to replicate wind and water currents as well as obstacles, as well as Ownship Pose to update the surface vehicle's

position. The virtual environment is visualized using RViz in a manner similar to the Control Station visualization. It is important to note that in terms of environment obstacles, the virtual visual displays obstacles that were intentionally placed, whereas the physical real-time visual displays the precepted obstacles.

#### **4 AUTONOMOUS MARITIME SYSTEM TO THE T&E FRAMEWORK**

This section presents the conceptual mapping of the USV's autonomous software onto the T&E framework. To accomplish this, the autonomous subsystem components were classified into different architecture components -virtual or physical sensors and data models, actuators, virtual environment and members of the Sense, Plan, and Act modules. To facilitate augmented testing, data combiner nodes - nodes handling routing physical and virtual information, were realized. These nodes receive both virtual and physical data and depending on the case perform a simple transfer or an augmentation. This eliminates the need for separate topics during simulation and ensures that nodes are unaware of the data source. Figure 3 illustrates the new architecture that is detailed in the following paragraphs.

The nodes classified as Sensors were SBG, Velodyne, Vision, and Hydro, with the Virtual Sensor Models being MATLAB SBG, Velodyne Sim, Vision Sim and Hydro Sim. Three Sensor Data Models – SBG Combiner, Vision Combiner, PCL Combiner, and Hydro Combiner receive the physical and virtual sensor data and make a decision on whether to send the Sense and Plan Modules physical, virtual, or augmented data through topics specific to each sensor. This is where virtual information -virtual sensor information, can be extracted from the framework to test the software's ability to realize the environment in which the system is placed.

The Octomap Server, Obstacle Processing, and GPS IMU nodes compose the Sense Module, from which data is fed into the World Information Model which consists of the Fusion node. Fusion acts as a combiner, choosing to send the Plan Module -through a single topic, obstacle data generated with the Obstacle Processing node, obstacle data generated through the Vision nodes, or a combination. Obstacle data along with GPS location data are sent through respective topics to the Plan Module.

The Plan Module is singularly comprised of the AI node, which as described in the previous section utilizes world data such as USV location and object location and type data to decide the desired course of action. Through a singular topic the expected vehicle behavior is sent to the Act Module.

The Act Module is similarly composed of one node, Guidance. The low level motor values are sent to the Actuator Data Model and Actuator CIS and CIS Sim respectively through a single topic that both nodes are subscribed to. The CIS converts low level guidance to servo values, sending them to the servos to drive the USV. CIS Sim similarly sends servo values to the simulated chassis in MATLAB. This is where information can be injected to the framework to test autonomous software response and update the virtual environment as well as any visualization utilized.

The Virtual Environment subscribes to the topics published by the Data Models and Sense module, keeping track of world information data, sensed objects, and simulated USV movement. The Visualization/Control Station then displays the environment and obstacles as well as the surface vehicle's movement through them. The Control Station can represent a virtual, physical, or augmented view depending on the testing scenario.

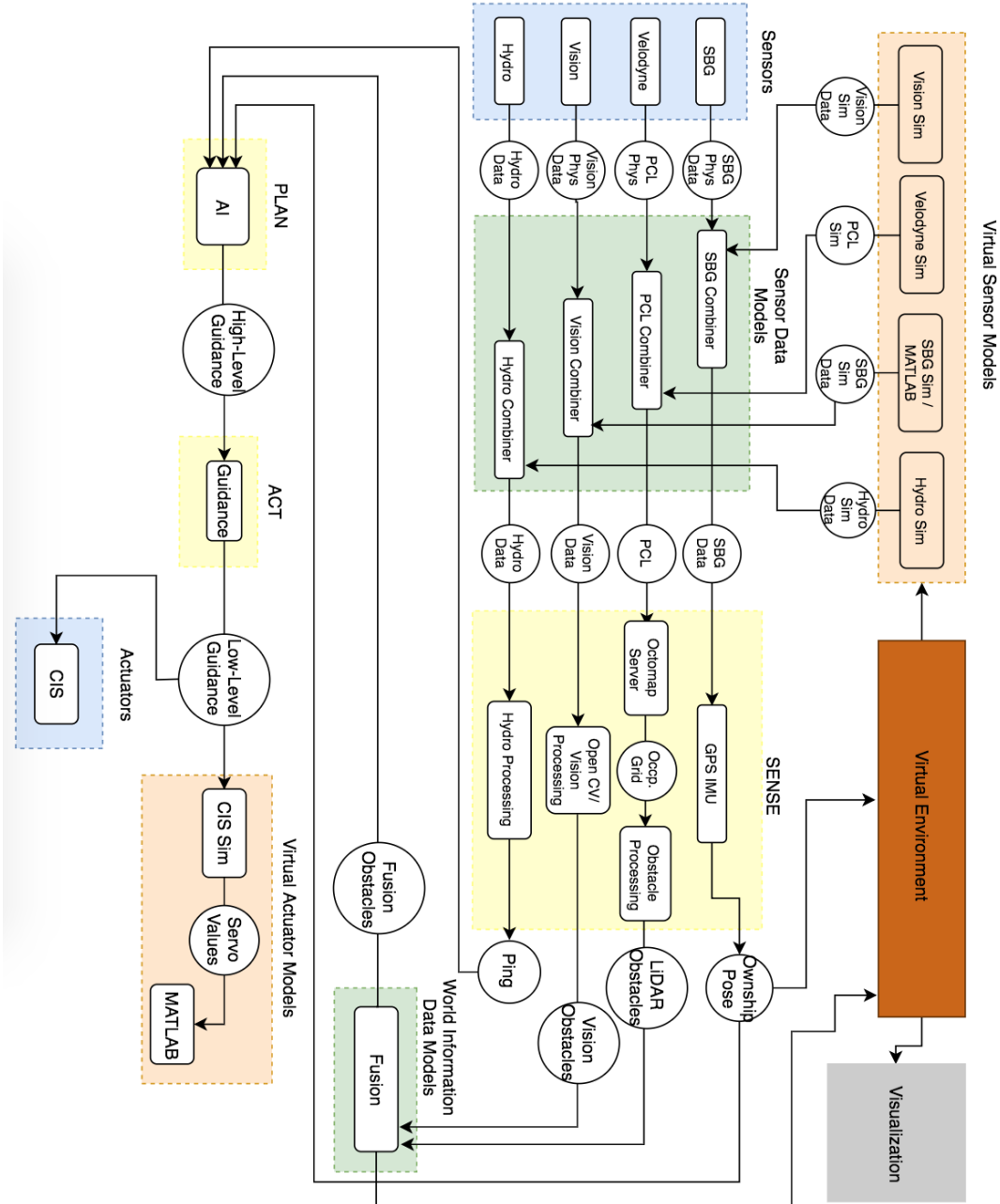


Figure 3. Autonomous Software onto Framework Architecture

## 5 ADDITIONAL FUNCTIONALITY

This section details the additional testing capabilities provided by utilizing the framework. The translation of the unmanned surface vehicle's autonomous software onto the suggested framework enables testing throughout the whole virtuality-reality continuum as well as the system's lifecycle without any modification on the autonomous software. Information (sensor or actuator) can be injected to the framework or extracted from the framework to perform a variety of testing scenarios. On the completely real side of the spectrum testing can be conducted on-shore, while a fully simulated system can also operate in a completely virtual reality. Furthermore, augmented virtuality introduces physical sensor input into the virtual environment to test the system's ability to analyze raw data as well as expected behavior. Finally, augmented reality allows

virtual information to be imposed on the real Sense information in cases where physical components are unavailable or if their introduction could lead to potential safety issues.

### **5.1 Augmented Virtuality**

Utilizing and injecting sensor data collected from the competition or on-shore testing in conjunction to virtual data not only permits the replication of competition behavior to hone automated functionality, but could also reveal crucial contrast between the virtual and physical system. The insertion of real GPS input would ensure that the expected values will not result in errors within the Sense Module, while similarly injecting real LiDAR data would test whether obstacles can be correctly identified with all the noise associated with a real environment. The question of whether the subsystem can identify and classify objects as totems, buoys, a dock, another boat, or even a judge on a canoe can thereby be answered; if the answer is negative modifications to the autonomous software can be continuously applied and tested until the system yields satisfactory results. This approach can also be applied to the use of real Vision data in order to determine if the subsystem can identify colors and aid in object classification.

### **5.2 Augmented Reality**

Superimposing virtual sensor data to the sensed physical environment can yield a number of benefits during sensor and behavioral testing including decreased cost, increased safety, and increased system reliability. To test the autonomous system's capability to perform obstacle avoidance, a small number of physical buoys can be placed on water, while a larger number of virtual buoys can be "placed" on the physical environment through the Virtual Sensor Models. This would acceptably test whether the surface vehicle behaves as expected. Similarly a dock can be superimposed to test the vessel's ability to park, while judges on canoes and other USVs can also be inserted to ensure that the vessel would not collide with either of the two in any case. This would solve the issue of not being able to perform physical testing for some tasks because certain physical components cannot be acquired, also saving the time it would take to place objects on the water as well as the cost of buying the objects. Safety would also not be compromised as a collision between a judge on a canoe or another USV would be impossible even if the surface vehicle does not respond as expected or required.

Utilizing the physical world visualization data displayed by the Control Station, virtual objects can be superimposed onto the physical environment. With accompanying technology such as VR Goggles a person could then look into the physical field and see both the physical USV and obstacles as well as the virtual objects. This would facilitate intelligent observation as all components of the testing environment would be visible.

## **6 CONCLUSION**

This paper presents the conceptual mapping of USV autonomous software onto a test and evaluation framework that is meant to be used throughout the development cycle to perform testing on the full reality-virtuality continuum. The autonomous software architecture along with the framework architecture are described, while the approach of mapping the USV software onto the framework is detailed. This approach introduces some of the benefits of utilizing this framework, the primary being the ability to develop and test autonomous software components throughout the development cycle without needing separate testing harnesses. Another advantage of utilizing said framework is the facilitation of augmented testing use to improve and validate autonomous system behavior by injecting or extracting data to and from the framework without autonomous software modification. Along with the benefit of increased and varied testing, framework utilization would result in a decrease of testing costs and safety concerns throughout the development and testing of autonomous systems.



## ACKNOWLEDGMENTS

The authors of this paper would like to thank and acknowledge the ODU RobotX team, including their advisors for their work on the USV's autonomous software, without which this paper would not have been possible.

## REFERENCES

- Erann, Gat. (1998). "On Three-Layer Architectures". *Artificial Intelligence and Mobile Robots*. AAAI Press.
- Leathrum, J., Shen, Y., Mielke, R., & Gonda, N. (2018). "Integrating Virtual and Augmented Reality Based Testing into the Development of Autonomous Vehicles". *MODSIM World 2018*, Norfolk, VA, April 24-26.
- Leathrum, J., Mielke, R., Shen, Y., and Johnson, H. (2018). "Academic/Industry Educational Lab for Simulation-Based Test & Evaluation of Autonomous Vehicles". *WinterSim 2018*, Gothenburg, Sweden, December 9-12.
- Maritime RobotX. "2020 Maritime RobotX Challenge". <https://robotx.org/>. Accessed March 10, 2019.
- Laverghetta, Thomas J., Leathrum, James F., Gonda, Nathan. (2019) "Integrating Virtual and Augmented Reality Based Testing into the Development of Autonomous Vehicles." *MODSIM World 2019*, Norfolk, VA, April 22-24.
- ROS. "Robot Operating System". <http://www.ros.org/>. Accessed March 10, 2019.

## AUTHOR BIOGRAPHIES

**NTIANA SAKIOTI** is a Graduate Assistant completing a Master of Science in Modeling, Simulation, and Visualization Engineering at Old Dominion University. She has earned a Bachelor Degree in Electrical and Computer Engineering, and has experience in computer programming and autonomous systems. Her research interests include autonomous systems in particular autonomous vehicles, simulation-based testing, and behavioral modeling. Her email address is [nsaki001@odu.edu](mailto:nsaki001@odu.edu).

**ANDREA H. ROBEY** is a Modeling and Simulation Analyst at SimIS Inc., completing a Bachelor of Science in Modeling, Simulation, and Visualization Engineering at Old Dominion University. Her research interests include autonomous systems in particular USVs and UAVs, modeling networks, and modeling networks and autonomous behaviors. His email address is [arobe004@odu.edu](mailto:arobe004@odu.edu).

**THOMAS LAVERGHETTA** is an undergraduate researcher working towards a Bachelor of Science degree in Modeling, Simulation, and Visualization Engineering at Old Dominion University; prospectively graduating in the May of 2021. His research interest includes simulation-based test and evaluation, big data analytics, systems and industrial engineering, and human behavioral modeling. His email address is [tlave002@odu.edu](mailto:tlave002@odu.edu)

**NATHAN GONDA** is a graduate student working towards a Master Degree in Modeling Simulation Engineering. He has earned a Bachelor Degree in Modeling and Simulation Engineering from Old Dominion University and has experience in computer programming and simulation software design for about 5 years. His research interests are in computer game design, virtual reality, and cybersecurity. His email address is [ngond002@odu.edu](mailto:ngond002@odu.edu).

**JAMES F. LEATHRUM** is an Associate Professor in the Department of Modeling, Simulation and Visualization Engineering at Old Dominion University. He earned a Ph.D. in Electrical Engineering from

Duke University. His research interests include simulation software design, distributed simulation, and simulation education. His email address is [jleathru@odu.edu](mailto:jleathru@odu.edu).

# A GAME THEORETICAL MODEL OF PATHOGEN PREVALENCE AS AN OUTCOME OF STRATEGIC INTERACTIONS BETWEEN A BACTERIUM AND A TICK VECTOR

Sara A Simmons

Department of Biological Sciences  
Old Dominion University  
Norfolk, VA, US  
smcdo010@odu.edu

Holly D Gaff

Department of Biological Sciences  
Old Dominion University  
Norfolk, VA, US  
hgaff@odu.edu

## ABSTRACT

Gulf Coast ticks collected along the leading edge of a geographic range expansion reveal a pattern of increased prevalence of a human pathogen *Rickettsia parkeri* compared to ticks within the historic range. Ticks are an intermediate host that act as vectors to transmit the bacterium to a human or other vertebrate animal. We used a game theory model to explore equilibrium outcomes based on strategies used by the pathogen to infect ticks, and the best response of the tick as an intermediate host in this system. We demonstrate how mixed strategies can produce different rates of infection in some tick populations, with strategies presented here as a proxy for phenotypic diversity.

**Keywords:** game theory, range expansion, invasion biology, pathogen dynamics

## 1 INTRODUCTION

*Rickettsia parkeri* (RP) is an obligate intracellular bacterium that has been identified as the causative agent of Tidewater spotted fever (TSF) in humans in the United States (Paddock et al. 2004). The highest incidence of confirmed human cases of TSF is clustered in or near the mid-Atlantic states through the Gulf Coast region in the southeastern U.S (Paddock et al. 2008). Environmental prevalence of RP can be determined through active surveillance of tick vectors. *Amblyomma maculatum*, commonly known as the Gulf Coast tick (GCT) is considered the primary natural vector of RP in the United States (Paddock et al. 2004). GCTs historically occurred within 200km of the Gulf Coast. Populations in Virginia occupy a frontier into which GCT migration has occurred relatively recently, where RP prevalence is higher than the species' historic range (Fornadel et al. 2011). Migration into new areas might favor cooperative interactions between RP and the GCT host, thus amplifying the presence of RP in invading GCT populations.

A game theoretical model was used to determine the equilibrium of a game in which each species can have either a beneficial or detrimental effect on the other. In game theory, the evolution of cooperation by direct reciprocity can be modeled in a straightforward game with two players, each of which has two strategies or phenotypes (Nowak 2006). Game theory provides an alternative to traditional epidemiological models, which typically use ordinary differential equations, by allowing us to explicitly model each species' tradeoffs in a situation where strategic tension exists between the cost of cooperation and the immediate benefit of defection. It is important to consider cooperation in this model because of the biology of RP infection in ticks. RP is transmitted from GCT mothers to their offspring and this ongoing relationship provides a basis for RP to benefit from promoting the fitness of female ticks, but only if the cost is minimal. In this model, the tick's phenotype is its physiological response to an infection, whereas the bacterial phenotype is associated with pathogenesis inside the tick host. If some tick phenotypes have a fitness advantage when infected, then the chance that this phenotype successfully invades new sites may increase. A differential fitness advantage such as this might explain disparities in pathogen prevalence between the core and the frontier of the GCT's geographic range.

This paper will use game theory to explore the relative difference in fitness advantage needed to maintain high levels of infections in GCTs under this scenario. This game theory model will be integrated with an agent-based model in NetLogo (Wilensky, 1999) to explore GCT population dynamics following introduction to novel environments.

## 2 PROPOSED METHODOLOGY

The proposed model is a normal form game with two types of players, a tick host (GCT) and a bacterium (RP). The bacterium makes the first move to establish an initial infection. RPs play two different strategies upon encountering a GCT: exploit or facilitate,  $S_{RP} = \{E, F\}$ . The GCT will either tolerate or attack the RP,  $S_{GCT} = \{T, A\}$ . To attack the RP, the GCT will mount an immune response, which incurs a cost to the GCT and a reduced payoff for the RP. These strategies will be integrated into an agent-based model to simulate the dynamics of the larger pathogen-tick-host network within a spatially structured environment.

## 3 PRELIMINARY RESULTS

Two Nash Equilibrium conditions exist in this model, one in which the bacterium has a symbiotic relationship with the GCT (facilitate, tolerate), and one in which the bacterium acts as a pathogen to the tick and stimulates an immune response (exploit, attack). These can coexist as mixed-strategies in the respective populations, with one strategy becoming more prominent as the payoff for that strategy increases.

## 4 DISCUSSION

This model presents a simple framework from which we can formulate hypotheses about biological mechanisms that may contribute to the distinct spatial pattern of pathogen prevalence observed in natural populations. Preliminary field data show that one site in Virginia with a single predominant GCT genotype has higher prevalence of RP (60%) compared to sites with at least two or more common GCT genotypes (range: 35-41%). Further field confirmation of a spatial pattern of pathogen prevalence, and a clearer understanding of the relationship between the tick and the bacterium will determine whether this hypothesis realistically captures the underlying biology of this system. Future work will be done to explore whether there is a beneficial relationship between RP and GCT as have been observed with other *Rickettsia* species.

## REFERENCES

- Fornadel, C. M., X. Zhang, J. D. Smith, C. D. Paddock, J. R. Arias, and D. E. Norris. 2011. High rates of *Rickettsia parkeri* infection in Gulf Coast ticks (*Amblyomma maculatum*) and identification of “*Candidatus Rickettsia andeanae*” from Fairfax County, Virginia. *Vector-Borne and Zoonotic Diseases* **11**:1535-1539.
- Nowak, M. A. 2006. Five rules for the cooperation of evolution. *Science* **314**: 1560-1563.
- Paddock, C. D., J. W. Sumner, J. A. Comer, S. R. Zaki, C. S. Goldsmith, J. Goddard, S. L. McLellan, C. L. Tamminga, and C. A. Ohl. 2004. *Rickettsia parkeri*: a newly recognized cause of spotted fever rickettsiosis in the United States. *Clinical Infectious Diseases* **38**:805-811.
- Paddock, C. D., P.-E. Fournier, J. W. Sumner, J. Goddard, Y. Elshenawy, M. G. Metcalfe, A. D. Loftis, and A. Varela-Stokes. 2010. Isolation of *Rickettsia parkeri* and identification of a novel spotted fever group *Rickettsia* sp. from Gulf Coast ticks (*Amblyomma maculatum*) in the United States. *Applied and environmental microbiology* **76**:2689-2696.
- Perlman, S.J, M.S. Hunter, and E. Zchori-Fein. 2006. The emerging diversity of *Rickettsia*. *Proceedings of the Royal Society B* **273**: 2097-2106.
- Wilensky, U. 1999. NetLogo. <http://ccl.northwestern.edu/netlogo/>

# BAND SELECTION IN MULTISPECTRAL IMAGES FOR SEAGRASS DETECTION USING DEEP CONVOLUTIONAL NEURAL NETWORK

Kazi Aminul Islam<sup>1</sup>, Daniel Pérez<sup>2</sup>, Victoria Hill<sup>3</sup>, Blake Schaeffer<sup>4</sup>, Richard Zimmerman<sup>3</sup> and Jiang Li<sup>1</sup>

<sup>1</sup>Department of Electrical & Computer Engineering, Old Dominion University, Norfolk, VA 23508

<sup>2</sup>Department of Modeling, Simulation & Visualization Engineering, Old Dominion University, Norfolk, VA 23508

<sup>3</sup>Department of Ocean Earth & Atmospheric Sciences, Old Dominion University, Norfolk, VA 23508

<sup>4</sup>Office of Research and Development, U.S. Environmental Protection Agency, Durham, North Carolina USA

## ABSTRACT

Seagrass is an essential factor for balancing oceanic ecological system. To maintain the seagrass growth properly, we propose an automated deep learning-based approach to detect seagrass using multispectral satellite image. Precisely, we utilized a deep convolutional neural network as a classifier to detect the seagrass in coastal area. We also used a sequential forward feature selection algorithm to find the correlated bands for seagrass detection in multispectral satellite image. Our proposed approach performed superbly by obtaining an overall training accuracy of 99.85% using yellow, green, red edge bands of worldview-2 satellite for seagrass detection.

**Keywords:** Convolutional neural network, Seagrass detection, Band selection

## 1 INTRODUCTION

Seagrass is an important component in marine biological ecosystem. It provides food, habitats and ecological balance to the fish and other marine animals. But, the seagrass growth declined over the years due to human activities and natural disasters. We propose an automated procedure to monitor the seagrass growth remotely from the conventional manual approach. To study the seagrass growth, we propose to develop a deep convolutional neural network classifier. This classifier may also gain a significant improvement in terms of computation, memory requirements and performance by removing un-correlated features. There are different methods for feature selection, e.g.: optimum, heuristic, randomized, filter and wrapper methods. We use sequential forward feature selection heuristic method for seagrass detection.

## 2 METHODS AND DATASET

Datasets:

We utilized 8 bands worldview-2 satellite image taken from Florida costal bay area. The bands are coastal blue, blue, green, yellow, red, red edge, NIR-1 and NIR-2. We used selected regions by an experienced operator. The selected regions contain five classes: seagrass, sea, sand, land and intertidal which are represented as green, blue, cyan, yellow and white in Figure 2.

Convolutional Neural Network (CNN):

CNN is the state of the art method for image classification in different applications. Our proposed deep CNN for seagrass detection shown in Figure 1. We used two convolutional layers, one flatten layer and one soft-max layer. The first convolutional layer has 32 filters with 2 by 2 kernel size and second convolutional layer has 16 filters with 4 by 4 kernel size. We used “Adam” optimizers with 10 epochs for training the CNN model.

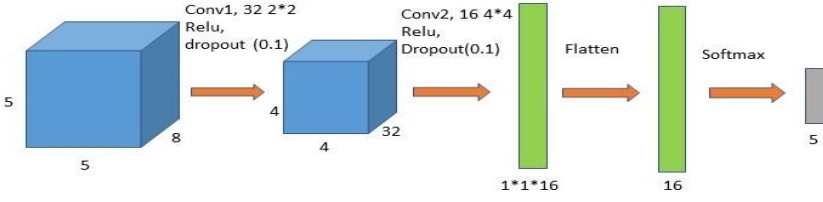


Figure 1: Architecture of deep convolutional neural network.

Sequential forward feature selection:

We used a sequential forward feature selection method with deep CNN classifier to find the effective bands for seagrass detection. We tried to find the optimum three bands for seagrass detection among available eight bands. We used the following procedure:

1. First, the best band is selected based on classification accuracy comparing all other bands,
2. In second step, we pair the band selected from the step 1 with the next best band. It is selected by comparing the improvement of classification accuracy for the remaining bands,
3. Next step, we form the triplets of bands by selecting one from the remaining bands using same procedure in previous steps,
4. This procedure is continued until we select our desired number of band or some stop criteria is met.

### 3 RESULTS

We implemented our proposed method using coastal blue, blue, green, yellow, red and red edge bands. We remove the NIR-1 and NIR-2 bands because these bands do not have any important information for seagrass detection. The feature selection results are shown in Figure 3 and Figure 4 for Saint Joseph Bay (SJB).

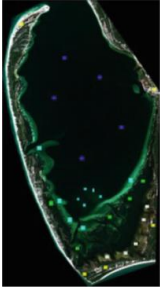


Figure 2: Saint Joseph Bay (SJB)

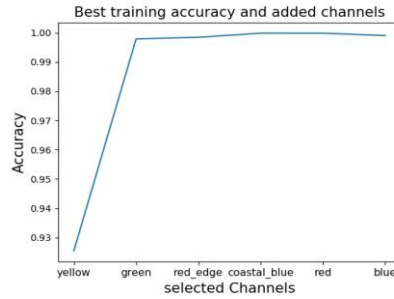


Figure 3: Feature selection results of SJB

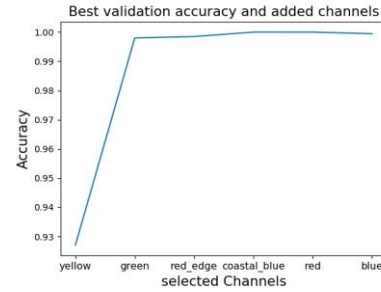


Figure 4: validation results of SJB

We achieved an overall training accuracy of 99.85% using yellow, green and red edge band compared to 99.90% using all six bands.

### 4 DISCUSSION AND CONCLUSION

Our proposed approach selected yellow, green and red edge as the most important three bands contributing to the seagrass detection in Figure 3. It achieved an overall training accuracy of 99.85% compared to 99.90% using all six bands. It achieved a best accuracy of 99.98% using four (yellow, green, red edge and coastal blue) or five bands (yellow, green, red edge, coastal blue and red). Using all the available bands, the overall accuracy decreased. It proves the significance of the feature selection algorithm. In Figure 4, we observe that the feature selection results aligned with the validation results. In conclusion, the proposed approach can detect seagrass proficiently using fewer band. Here, we implemented the proposed approach only in one location. In future, we will apply this method to other locations and compare with some baseline band selection approach to validate our proposed approach.



# VERBAL INSTRUCTION TO PERFORM SPECIFIC RUNNING STYLES IMPACTS REACTION FORCES DURING HABITUAL AND NON-HABITUAL RUNNING

Collin D. Bowersock

Steven Morrison

Daniel Russell

Old Dominion University  
School of Physical Therapy and Athletic Training  
4211 Monarch Way  
Suite 275, Norfolk, VA, USA  
cbowe005@odu.edu  
smorriso@odu.edu  
dmrussel@odu.edu

## ABSTRACT

The purpose of this study was to investigate the biomechanical differences between two popular strike patterns used during running, rearfoot and forefoot strike patterns. This study also examines how verbal instruction to perform these strike patterns changes running parameters when compared to uninstructed, preferred running. Overall, A forefoot running style results in decreased ground reaction forces while also slowing the progression of the center of pressure underfoot and decreasing the area underfoot to attend to these forces. Verbal instruction to perform a specific strike pattern lead to decreased ground reaction forces and an exaggerated center of pressure pattern, especially in the anterior-posterior direction. Thus, a biomechanical analysis performed after an acute switch in running technique should be interpreted cautiously. This work suggests that verbal instruction alone can affect biomechanical measures.

**Keywords:** running, center of pressure, ground reaction force, biomechanics.

## 1 INTRODUCTION

The predominant style used among recreational runners is one where the individual makes initial contact with the ground using the heel and then progresses through to push off with the toe, referred to as a rearfoot strike pattern (Daoud et al., 2012; Hasegawa, Yamauchi, & Kraemer, 2007; Larson et al., 2011; Shih, Lin, & Shiang, 2013; Williams III, McClay, & Manal, 2000). Despite the majority of running athletes naturally using a rearfoot running style, there has been a considerable amount of research and coaching promoting the utilization of a forefoot running style, where the forefoot makes initial contact with the ground (Diebal, Gregory, Alitz, & Gerber, 2011, 2012; Kulmala, Avela, Pasanen, & Parkkari, 2013; Yong, Silder, Montgomery, Fredericson, & Delp, 2018). The increase in popularity of a forefoot running style may be partially due to the claims of reducing the risk of certain musculoskeletal injuries (Oakley & Pratt, 1988). Studies have shown that the use of a forefoot strike pattern results in no vertical impact peak (Bobbert, Schamhardt, & Nigg, 1991; Kulmala et al., 2013; Lieberman et al., 2010b) and lower loading rates, ground reaction forces, joint stresses and joint moments (Kulmala et al., 2013; Laughton, Davis, & Hamill, 2003; Lieberman et al., 2010a), all of which have been suggested to contribute to running-related injuries. Also contributing to the popularity of a forefoot strike pattern is the use of this running style among elite athletes

(Hasegawa et al., 2007; Kasmer, Liu, Roberts, & Valadao, 2013). While these different strike patterns lead to different kinematic and kinetic features within the lower extremity, there is still a requirement for the individual to make subtle adjustments to compensate for gait-related forces generated during running irrespective of the selected running style.

Most studies of running patterns are typically based around the collection and assessment of kinetic and/or kinematic measures. Less common has been the utilization of center of pressure (COP) measures, arguably because of the difficulty in collecting COP measures during running. COP is a point location of the vertical ground reaction force, representing a weighted average of all the pressures over the surface of the area in contact with the ground (Winter, 1995). During running, the COP is always underneath a single foot and is, therefore, a point location of the weighted average of all the pressure underneath a single foot. Although researchers have discussed the importance of ground reaction force profiles and it's associated with running-related injuries and running styles, few studies have quantified where within the foot these forces are being applied while running (Cavanagh & LaFortune, 1980; De Cock, Vanrenterghem, Willems, Witvrouw, & De Clercq, 2008) and how the acute change of running patterns may affect the COP dynamics. Understanding where the forces and pressures are being regularly applied and sustained underfoot and how the human body as systems adapts to an altered running style could help researchers and clinicians identify an individual's running proficiency. As individuals choose to adapt their natural running pattern, further knowledge of how COP patterns are different between running styles and affected by the acute change in running styles could better help clinicians and researches advice and treat individuals with a variety of running strategies.

In order to measure the differences between a forefoot and rearfoot strike pattern, many researchers use a sample population of runners who naturally use only one running style. The authors will then collect running parameters while having the participants run using their preferred strike pattern and then the non-preferred strike pattern. This technique has largely been deemed acceptable as research has shown that individuals can replicate most kinematic and kinetic measures of a habitual strike pattern using a non-habitual strike pattern (Williams III et al., 2000). This implies that a rearfoot runner can run using a forefoot strike pattern and look comparable to an individual's who naturally uses a forefoot strike pattern, and vice versa. Only a few studies have investigated the subtle differences that occur when instructing individuals to perform a strike pattern that is not the natural pattern the runner would use (Knorz et al., 2017, Stearne et al., 2014). Further investigation of how an acute change in running style differs from a natural running style can help identify the risks and benefits of changing running techniques.

The principal aim of the current study was to assess the differences in COP patterns, spatiotemporal measures, and kinetic parameters during rearfoot and forefoot running styles for healthy individuals. It was predicted that the forefoot pattern would be reflected by decreases in VGRF, shorter steps, and decreased step time compared to a rearfoot strike pattern. We also hypothesized a forefoot strike pattern to exhibit decreased COP excursion compared to a rearfoot running style. A secondary aim was to investigate the effect of an imposed habitual and non-habitual strike pattern. It was expected that an imposed habitual strike would produce COP and kinematic patterns like those produced by an un-imposed habitual strike pattern.

## **2 METHODS**

### **2.1 Participants**

Nineteen healthy adults (7 male, 12 female; age 23.7±3.3 yrs.; 170±8 cm; 74.7 ±11.9 kg) participated in this study. Inclusionary criteria were being able to comfortably run at 5.5 mph (self-reported) and being of low risk for cardiac events during exercise according to American College of Sports Medicine guidelines (i.e., participants had no known cardiopulmonary disease or diabetes, no symptoms of cardiopulmonary disease, and no more than one major risk factor for cardiopulmonary disease).

## **2.2 Experimental Design**

After screening and consent procedures were completed, subjects were measured for height and mass. Prior to data collection, individuals were permitted a 5-minute warm-up period on the instrumented treadmill at their own pace using their own footwear. Following the warm-up, data collection for the specific running conditions was performed. Subjects performed three 60 sec trials for each condition, with 5-10 min of rest between each condition. The conditions performed were as follows: 1) running with the person's preferred strike pattern, 2) running with a forefoot strike pattern, and 3), running with a rearfoot strike pattern. All running conditions were performed at a constant speed of 5.5 mph or 2.45 m/s. For each participant, gait measures were recorded on an automated treadmill (H/P/Cosmos Mercury Med 4.0) with an installed pressure plate (FDM-T Zebris Medical GmbH, Germany). Data were collected at a sample rate of 120 Hz.

## **2.3 Data Analysis**

Data related to the spatiotemporal features, vertical ground reaction forces (VGRF) and center of pressure (COP) location in the mediolateral (ML) and anterior-posterior (AP) directions during each step was collected for each running condition. Participants were designated as either habitual forefoot strikers (n=5) or habitual rearfoot strikers (n=14) based upon visual analysis of COP patterns during the preferred running condition. All VGRF and COP data were analyzed using MATLAB software (Mathworks R2018a) and structured as follows:

### **2.3.1 Spatio-Temporal Gait Measures**

The specific measures related to the spatiotemporal features of the different gait patterns were stride length, stride time, step length, step time, and step width. These measures were obtained from the Zebris treadmill software.

### **2.3.2 Vertical Ground Reaction Forces**

The VGRF forces, derived from pressure by the Zebris treadmill software were obtained for each step and normalized to each individual's body weight. The specific measures were the mean VGRF, peak VGRF and cumulative VGRF per step.

## **2.4 Statistical Analysis**

All analyses were performed using repeated measures, mixed generalized linear models (GLM). Analyses were conducted on the individuals as a function of the three running conditions (i.e. preferred, forefoot, rearfoot) group (habitual rearfoot, forefoot) and limb (i.e., left, right). All statistical analyses were performed using SAS statistical software (SAS Institute Inc., NC), with the risk of Type I error set at  $p < 0.05$ . Given that no significant differences were found between the left and right limb, the results were collapsed over limb.

### 3 RESULTS

Figure 1 illustrates the COP location of one representative participant from each group during each running condition.

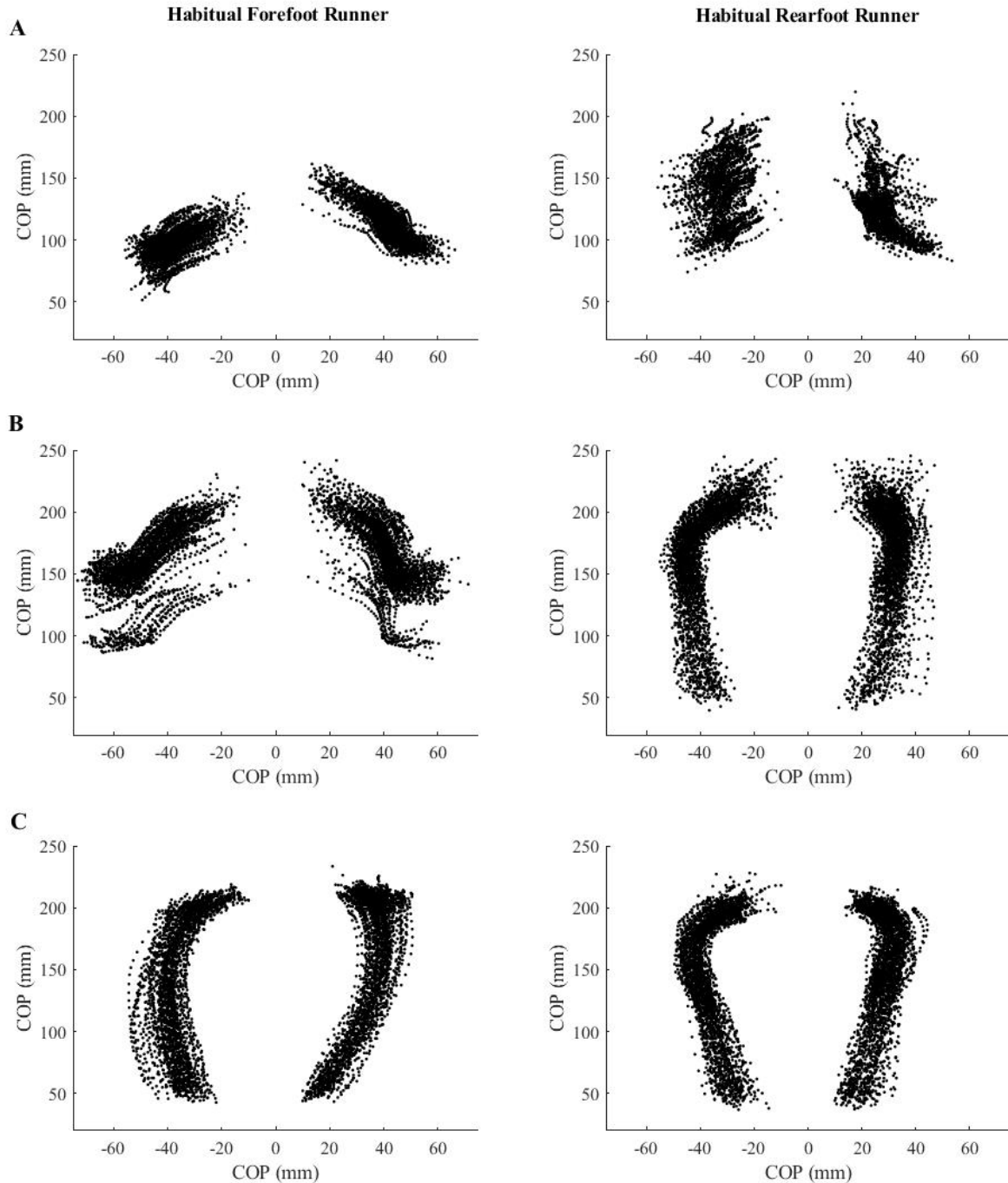


Figure 1: Representative COP location plots from each strike pattern group during each running condition. A.) Forefoot running condition. B.) Preferred running condition. C.) Rearfoot running condition

### **3.1.1 Spatio-Temporal Gait Measures**

Significant main effects for condition were observed for the stride length ( $F_{2,34}=16.25$ ;  $p<0.001$ ), and stride width ( $F_{2,34}=19.65$ ;  $p<0.001$ ). Planned contrasts revealed stride length and stride width to be significantly greater during the forefoot running condition compared to the other two running styles. No significant differences between conditions were found for stride time.

### **3.1.2 Vertical Ground Reaction Force Analysis**

Significant main condition effects were observed for mean ( $F_{2,34}=16.81$ ;  $p<0.0001$ ) peak ( $F_{2,34}=38.47$ ;  $p<0.0001$ ) cumulative ( $F_{2,34}=21.79$ ;  $p<0.0001$ ) VGRF per step. For the mean and cumulative VGRF, planned contrasts revealed differences between all three running styles with highest values being seen for the preferred running style and lowest values seen for the imposed forefoot running style (all  $p$ 's $<0.05$ ). For the peak VGRF, planned contrasts illustrated that the peak VGRF being lowest in the imposed forefoot running condition compared to the other two running styles ( $p$ 's $<0.05$ ). Figure 2 illustrates the differences in VGRF across the three running conditions. Significant condition-by-group interaction effects were also observed for the mean ( $F_{2,34}=8.19$ ;  $p=0.0013$ ) and peak VGRF ( $F_{2,34}=4.65$ ;  $p=0.0165$ ). Lastly, a significant group effect was found for cumulative VGRF. Habitual forefoot runners have decreased cumulative VGRF per step regardless of the running condition ( $F_{1,17}=4.83$ ;  $p=0.04121$ ).

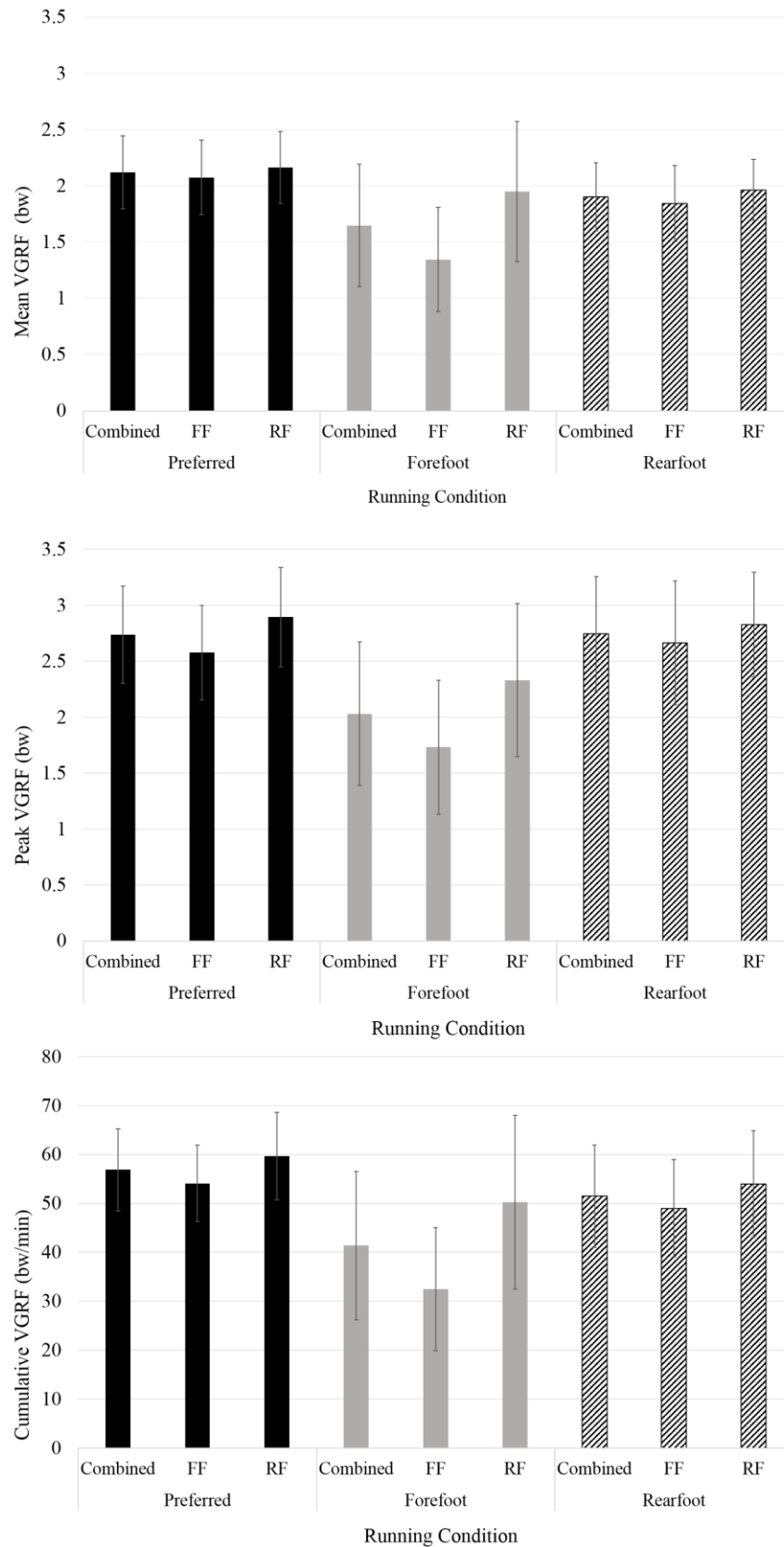


Figure 2. Mean, peak, and cumulative vertical ground reaction forces (VGRF) for each ground and running condition. Combined represents average of both groups. FF; represents forefoot group. RF; represents rearfoot group. Error bars represent the standard deviation.



### 3.1.3 Center of Pressure Analysis

Significant main effects for condition were found for COP velocity (Mean  $F_{2,34}=252.33$ ;  $p<0.001$ ), path length (Mean  $F_{2,34}=255.67$ ; SD  $F_{2,34}=7.90$ ;  $p's<0.001$ ) AP COP excursion (Mean  $F_{2,34}=22.19$ ; SD  $F_{2,34}=32.40$ ;  $p's<0.0001$ ) and ML COP excursion (SD  $F_{2,34}=5.27$ ;  $p's<0.01$ ). For the mean COP velocity, path length, and AP excursion planned contrasts revealed the differences were between all three running styles (all  $p's<0.05$ ). The imposed forefoot running style resulted in the lowest COP velocity, smallest mean and standard deviation of the path length, and smallest mean and standard deviation of the AP excursion while the imposed rearfoot running style resulted in the largest values for these measures. For the ML COP excursion, the smallest excursion occurred during the imposed rearfoot running condition while the other two conditions were not significantly different from one another.

A significant group effect was found for ML COP excursion ( $F_{1,17}=14.24$ ;  $p=0.0015$ ). Habitual forefoot runners had increased ML COP excursion regardless of the running condition. A significant group by condition interaction effect was also observed for AP COP excursion ( $F_{2,34}=55.0465$ ;  $p<0.0001$ ). Figure 3 illustrates the general pattern of change for selected COP measures (i.e., path length, COP velocity, COP excursion in the AP and ML directions) as a function of the different running styles.

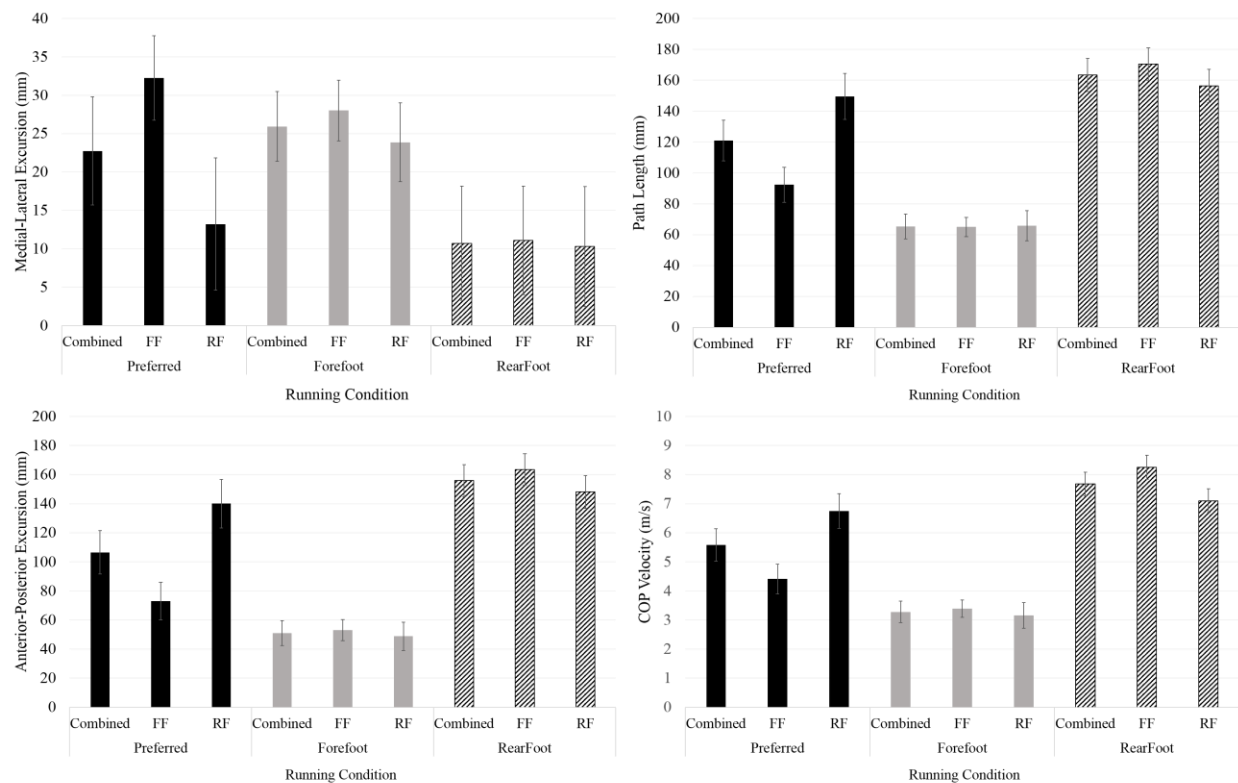


Figure 3. COP excursion (AP & ML), velocity and path length for each ground and running condition. Combined represents average of both groups. FF; represents forefoot group. RF; represents rearfoot group. Error bars represent the standard deviation.

## 4 DISCUSSION

This study was designed to examine a) the differences in spatiotemporal measures, VGRFs, and COP patterns between running styles and, b) the effect of imposing habitual and non-habitual running patterns.

The differences between running patterns will be discussed first, followed by a discussion about the effect of imposing habitual and non-habitual running styles

#### **4.1 The Difference between Running Techniques**

Overall, the results demonstrated that a forefoot running style leads to decreased VGRF's as well as COP velocity, path length, and AP excursion.

##### **4.1.1 Spatio-Temporal Gait Measures**

Specific increases in step length were observed for the forefoot running condition compared to the two other running styles. However, the increase in step length may be linked to how this variable is defined within the Zebris instrumented gait analysis system software. It may be that step length is calculated from the final contact of one limb to initial contact of the other. Therefore, during the rearfoot and preferred strike pattern, step length is toe off to heel strike. However, during the forefoot condition, step length becomes toe off to toe strike. Thus, the increase in step length may simply be a function of the length from one's heel to their toe, while their foot may still occupy the same space. This is supported by the lack of change in step time for any of the running conditions while the treadmill velocity was held constant. Indeed, previous studies have tended to report no change in step length with different foot strike pattern, supporting this previous assumption (Ahn, Brayton, Bhatia, & Martin, 2014; Almonroeder, Willson, & Kernozek, 2013; Ardigo, Lafortuna, Minetti, Mognoni, & Saibene, 1995; Kulmala et al., 2013; Vannatta & Kernozek, 2015). Research has also highlighted spatiotemporal differences between the Zebris instrumented gait analysis system and other gait measurement technologies (Reed, Urry, & Wearing, 2013; Wearing, Reed, & Urry, 2013) which may also support our interpretation of the step length result.

In contradiction to our finding of increased step width during a forefoot running condition, previous research has reported that utilizing a rearfoot strike pattern can lead to a wider step width accompanied by decreased iliotibial band strain and strain rate which the authors suggested may be beneficial to those with a history of iliotibial band syndrome or runners' knee (Boyer & Derrick, 2015). However, results in the same study (Boyer & Derrick, 2015) found habitual forefoot strikers to have a wider step length in both rearfoot and forefoot running conditions when compared to habitual rearfoot strikers. Further, habitual forefoot strikers were found to have the widest step during the rearfoot condition. While considering these varied findings and the previous mentioned spatiotemporal differences found when using the Zebris system, runners who suffer from runner's knee may take care before switching to an alternate foot strike pattern in an attempt to alleviate their symptoms.

##### **4.1.2 Vertical Ground Reaction Force**

Vertical ground reaction forces tended to be smaller during the forefoot running style compared to the other 2 running styles (figure 2). While this finding is consistent with previous literature which has also reported decreased GRFs when running with a forefoot strike pattern (Almeida, Davis, & Lopes, 2015) it is by no means a universal result. For example, other studies have reported increased GRFs (Kulmala et al., 2013; Laughton et al., 2003), and no differences in GRFs (Bowersock, Willy, DeVita, & Willson, 2016; Cavanagh & Lafortune, 1980) between the two running styles. Part of the reason for this inconsistency may be related to a range of intrinsic and extrinsic factors which determine ground reaction forces such as speed, habitual running pattern, runner experience, footwear, running environment, intervention length etc. Despite the aforementioned differences across studies, the finding of decreased VGRFs during forefoot running requires further elaboration. One explanation for the decreased VGRF during a forefoot strike pattern may be linked to greater ankle compliance when contacting the ground (Hamill, Gruber, & Derrick, 2014; Laughton et al., 2003). In this regard, increased ankle compliance can decrease the rate of tibial deceleration and increase the time to peak vertical ground reaction force in each step (Gruber, Boyer, Derrick, & Hamill, 2014), possibly contributing to the decrease in peak VGRF. Also, because step time remained unchanged across

running condition, the found decrease in peak VGRF per step relates directly to the decreases in mean VGRF per step during the forefoot running conditions.

#### **4.1.3 Center of Pressure**

The results of the COP profile between running condition as expected and consistent with previous findings (Cavanagh & LaFortune, 1980). A forefoot running style leads to decreased COP velocity, AP excursion, path length and a larger amount of ML excursion when compared to the other running conditions. Further, the SD of the AP excursion and path length were smaller in the forefoot running condition. Thus, although the VGRF's are less during a forefoot strike pattern, a forefoot strike pattern resulted in a COP profile that moves slower through a smaller area of that foot, with less step to step variance while step time is held constant. To the extent that force concentration and overuse phenomena attribute to injury risk, an acute change to a forefoot running style may result in some deleterious musculoskeletal effect.

### **4.2 The Effect of Switching Technique**

The second aim of this study was to study the effect an imposed strike patterns on habitual rearfoot and forefoot runners. Overall, we found imposing a strike pattern by way of verbal instruction led to an exaggerated COP pattern in both running techniques and a decrease in VGRF's. We also found imposing a habitual strike pattern to effect kinetic and COP measures when compared to an un-imposed preferred running condition.

#### **4.2.1 Vertical Ground Reaction Force**

VGRFs were largest when individuals ran using their preferred running style with no instruction and decreased when verbal instruction was given to perform a different running style. Specifically, following verbal instruction cumulative VGRF per step decreased regardless of a runner's habitual running style or the style that they were instructed to perform. This is in direct relation with Williams and colleagues who found converted forefoot strikers to produced lower kinetic forces compared to habitual forefoot strikers (Williams III et al., 2000). For the current study, the verbal instruction given to each individual to perform even their habitual strike pattern led to a decrease in the VGRF measures. This finding indicates that runners do not optimize their running kinematics to decrease ground reaction forces. The verbal instruction may enact an external focus, prompting more attention to their footfalls and possibly eliciting a softer landing. In relation to this discussion, a previous study included investigating how forefoot strike instruction in the form of biofeedback effects vertical loading rate and joint work (Baggaley, Willy, & Meardon, 2017). Subjects were visually cued to produce a foot-strike angle representative of a forefoot strike pattern. The results showed the cueing to be successful at decreasing loading rates while increasing joint work at the ankle and decreasing work at the knee. A similar phenomenon of increased mechanical work at the ankle and decreased work at the knee was seen when the biofeedback cued the runners to reduce vertical loading rate instead of producing a forefoot strike angle. These results combined with the results from the current study suggest that both direct and indirect instruction to switch running styles cause individuals to land softer, possibly requiring more mechanical work at the ankle and less at the knee. These effects may be more pronounced in habitual rearfoot strikers who switch to a forefoot strike. Stearne et al. found that when habitual rearfoot runners switch to a forefoot pattern there are increases in mechanical cost as well as increases in joint moments about the ankle and knee (2014). In contrast, habitual forefoot runners require less mechanical work and the knee and ankle to maintain the same running speed when they switch to a rearfoot running style (2014). Thus, clinicians and researchers should be cautious when promoting and reporting on the decrease in forces prompted by an acute switch in running style. Here we see evidence that the decrease in forces is possibly due to the instruction or external cueing to perform a certain running style and not the change in running style. Furthermore, rearfoot strikes may take additional caution if switching to a forefoot strike pattern due to the increase in joint kinetics even while VGRF are instantaneously reduced.

#### **4.2.2 Center of Pressure**

Visually, the COP patterns created by the two groups and three running styles have distinguishing features. Figure 1 shows a representative subject from the habitual forefoot and rearfoot group. When examining a rearfoot strike pattern, differences are seen between groups and within the rearfoot group. During the preferred condition of the habitual rearfoot runner, the COP pattern creates a shape similar to a question mark. The COP begins posteriorly at the heel, and progresses in a mostly linear fashion, anteriorly and slightly laterally towards the forefoot. Before toe-off, the COP shifts abruptly lateral while still advancing slightly anteriorly, signifying a push off from a region around the first metatarsal. When observing the habitual forefoot runners during an imposed rearfoot strike pattern, the group lacked the hook shape at the end of the COP path. Instead of a question mark pattern, it is more similar to a “C” pattern. The habitual forefoot strikers do not have the abruptly lateral shift towards the first metatarsal head before push off but have a more curved path throughout the COP path. This results further suggests that although runners can replicate most kinematic and kinetic measures of a habitual strike pattern using a non-habitual strike pattern (Stearne et al., 2014; Williams III et al., 2000), COP patterns, as well as kinetic measures, are not reproduced. This may contribute to the sudden injuries that are seen when runners switch to a new foot strike pattern.

COP excursion measures show that externally imposing a strike pattern tends to cause an exaggerated strike pattern. When imposing a rearfoot strike upon the habitual rearfoot runner, the path remained similar but was exaggerated. AP COP excursion increased from, from  $14.0 \pm 1.7$  cm to  $15.6 \pm 1.1$  cm. Furthermore, the habitual forefoot strikers had the largest AP excursion ( $16.4 \pm 1.1$  cm) during the imposed rear foot strike condition. The occurrence of an exaggerated non-habitual strike pattern has been previously noted. Knorz et al., (2017) found both habitual rearfoot strikers and habitual forefoot strikers to exaggerate their respective non-habitual strike pattern. Our results suggest it is not only the non-habitual strike pattern that is exaggerated but imposing the habitual strike pattern by verbal instruction also causes an exaggerated pattern. In support of this, we also found habitual forefoot strikers to decrease their AP COP excursion from  $7.3 \pm 1.3$  cm to  $5.3 \pm 0.7$  cm during an imposed forefoot strike pattern. This phenomenon should be considered when conducting biomechanical analysis between various strike patterns. Verbally instructing individuals to perform a certain strike pattern may affect the COP pattern that would be naturally performed.

#### **4.3 Conclusions**

This study highlights the spatiotemporal and kinetic differences between habitual rearfoot and forefoot running patterns as well as the effect of externally imposing habitually and non-habitually strike patterns. When compared to rearfoot running, a forefoot running style results in decreased VGRF's while also slowing the speed of COP progression velocity and utilizing a smaller area underfoot to attend to these forces. An imposed strike pattern leads to decreased VGRF's and an exaggerated COP pattern, especially in the AP direction. The findings are irrespective of an individual's habitual strike pattern and show that even imposing a strike pattern that is preferred by the runner by way of verbal instruction will lead to kinematic and kinetic differences.

## REFERENCES

- Ahn, A. N., Brayton, C., Bhatia, T., & Martin, P. (2014). Muscle activity and kinematics of forefoot and rearfoot strike runners. *Journal of Sport and Health Science*, 3(2), 102-112. doi: <https://doi.org/10.1016/j.jshs.2014.03.007>
- Almeida, M. O., Davis, I. S., & Lopes, A. D. (2015). Biomechanical differences of foot-strike patterns during running: a systematic review with meta-analysis. *Journal of Orthopaedic & Sports Physical Therapy*, 45(10), 738-755.
- Almonroeder, T., Willson, J. D., & Kernozek, T. W. (2013). The effect of foot strike pattern on Achilles tendon load during running. *Annals of biomedical engineering*, 41(8), 1758-1766.
- Ardigo, L. P., LaFortuna, C., Minetti, A., E, Mognoni, P., & Saibene, F. (1995). Metabolic and mechanical aspects of foot landing type, forefoot and rearfoot strike, in human running. *Acta Physiologica Scandinavica*, 155(1), 17-22.
- Baggaley, M., Willy, R., & Meardon, S. (2017). Primary and secondary effects of real-time feedback to reduce vertical loading rate during running. *Scandinavian journal of medicine & science in sports*, 27(5), 501-507.
- Bobbert, M. F., Schamhardt, H. C., & Nigg, B. M. (1991). Calculation of vertical ground reaction force estimates during running from positional data. *Journal of biomechanics*, 24(12), 1095-1105.
- Bowersock, C. D., Willy, R. W., DeVita, P., & Willson, J. D. (2016). Independent effects of step length and foot strike pattern on tibiofemoral joint forces during running. *Journal of sports sciences*, 1-9.
- Boyer, E. R., & Derrick, T. R. (2015). Select injury-related variables are affected by stride length and foot strike style during running. *The American Journal of Sports Medicine*, 43(9), 2310-2317.
- Cavanagh, P. R., & LaFortune, M. A. (1980). Ground reaction forces in distance running. *Journal of biomechanics*, 13(5), 397-406.
- Daoud, A. I., Geissler, G. J., Wang, F., Saretsky, J., Daoud, Y. A., & Lieberman, D. E. (2012). Foot strike and injury rates in endurance runners: a retrospective study. *Med Sci Sports Exerc*, 44(7), 1325-1334.
- De Cock, A., Vanrenterghem, J., Willems, T., Witvrouw, E., & De Clercq, D. (2008). The trajectory of the centre of pressure during barefoot running as a potential measure for foot function. *Gait & posture*, 27(4), 669-675.
- Diebal, A. R., Gregory, R., Alitz, C., & Gerber, J. P. (2011). Effects of forefoot running on chronic exertional compartment syndrome: a case series. *Int J Sports Phys Ther*, 6(4), 312-321.
- Diebal, A. R., Gregory, R., Alitz, C., & Gerber, J. P. (2012). Forefoot running improves pain and disability associated with chronic exertional compartment syndrome. *The American Journal of Sports Medicine*, 40(5), 1060-1067.
- Gruber, A. H., Boyer, K. A., Derrick, T. R., & Hamill, J. (2014). Impact shock frequency components and attenuation in rearfoot and forefoot running. *Journal of Sport and Health Science*, 3(2), 113-121.
- Hamill, J., Gruber, A. H., & Derrick, T. R. (2014). Lower extremity joint stiffness characteristics during running with different footfall patterns. *European Journal of Sport Science*, 14(2), 130-136.
- Hasegawa, H., Yamauchi, T., & Kraemer, W. J. (2007). Foot strike patterns of runners at the 15-km point during an elite-level half marathon. *Journal of Strength and Conditioning Research*, 21(3), 888.
- Kasmer, M. E., Liu, X.-c., Roberts, K. G., & Valadao, J. M. (2013). Foot-strike pattern and performance in a marathon. *International journal of sports physiology and performance*, 8(3), 286-292.
- Knorz, S., Kluge, F., Gelse, K., Schulz-Drost, S., Hotfiel, T., Lochmann, M., . . . Krinner, S. (2017). Three-dimensional biomechanical analysis of rearfoot and forefoot running. *Orthopaedic journal of sports medicine*, 5(7), 2325967117719065.
- Kulmala, J.-P., Avela, J., Pasanen, K., & Parkkari, J. (2013). Forefoot strikers exhibit lower running-induced knee loading than rearfoot strikers. *Medicine & Science in Sports & Exercise*, 45(12), 2306-2313.

- Larson, P., Higgins, E., Kaminski, J., Decker, T., Preble, J., Lyons, D., . . . Normile, A. (2011). Foot strike patterns of recreational and sub-elite runners in a long-distance road race. *Journal of sports sciences*, 29(15), 1665-1673.
- Laughton, C. A., Davis, I. M., & Hamill, J. (2003). Effect of strike pattern and orthotic intervention on tibial shock during running. *Journal of applied biomechanics*, 19(2), 153-168.
- Lieberman, D. E., Venkadesan, M., Werbel, W. A., Daoud, A. I., D'Andrea, S., Davis, I. S., . . . Pitsiladis, Y. (2010a). Foot strike patterns and collision forces in habitually barefoot versus shod runners. *Nature*, 463(7280), 531-535.
- Lieberman, D. E., Venkadesan, M., Werbel, W. A., Daoud, A. I., D'andrea, S., Davis, I. S., . . . Pitsiladis, Y. (2010b). Foot strike patterns and collision forces in habitually barefoot versus shod runners. *Nature*, 463(7280), 531.
- Oakley, T., & Pratt, D. (1988). Skeletal transients during heel and toe strike running and the effectiveness of some materials in their attenuation. *Clinical biomechanics*, 3(3), 159-165.
- Reed, L. F., Urry, S. R., & Wearing, S. C. (2013). Reliability of spatiotemporal and kinetic gait parameters determined by a new instrumented treadmill system. *BMC musculoskeletal disorders*, 14(1), 249.
- Richman, J. S., & Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6), H2039-H2049.
- Shih, Y., Lin, K.-L., & Shiang, T.-Y. (2013). Is the foot striking pattern more important than barefoot or shod conditions in running? *Gait & posture*, 38(3), 490-494.
- Stearne, S. M., Alderson, J. A., Green, B. A., Donnelly, C. J., & Rubenson, J. (2014). Joint kinetics in rearfoot versus forefoot running: implications of switching technique. *Medicine & Science in Sports & Exercise*, 46(8), 1578-1587.
- Vannatta, C. N., & Kernozek, T. W. (2015). Patellofemoral joint stress during running with alterations in foot strike pattern. *Medicine and science in sports and exercise*, 47(5), 1001-1008.
- Wearing, S. C., Reed, L. F., & Urry, S. R. (2013). Agreement between temporal and spatial gait parameters from an instrumented walkway and treadmill system at matched walking speed. *Gait & posture*, 38(3), 380-384.
- Williams III, D. S., McClay, I. S., & Manal, K. T. (2000). Lower extremity mechanics in runners with a converted forefoot strike pattern. *Journal of applied biomechanics*, 16(2), 210-218.
- Winter, D. A. (1995). Human balance and posture control during standing and walking. *Gait & posture*, 3(4), 193-214.
- Yong, J. R., Silder, A., Montgomery, K. L., Fredericson, M., & Delp, S. L. (2018). Acute changes in foot strike pattern and cadence affect running parameters associated with tibial stress fractures. *Journal of biomechanics*.

# EMBOSSSED CONSTRUCTION AND EXPLORATIVE VISUALIZATION OF 3D STRUCTURES FROM 2D IMAGES

Ahmet Saglam

Department of Modeling, Simulation, and Visualization Engineering

Old Dominion University

5115 Hampton Blvd, Norfolk, VA, USA

asagl001@odu.edu

## ABSTRACT

We present a practically applicable method for embossed construction and explorative visualization (ECEV) of 3D structures from a single 2D image. Using one picture or photograph, computer graphics techniques cannot be employed to obtain a 3D scene. By adopting pixel-based illumination to synthesize embossing effects, ECEV provides an effective and efficient representation of the 3D structure conveyed by a single image in support of interactive adjustment of the degree of accentuation and explorative visualization. Unlike existing methods, the proposed work is not intended to construct accurate 3D models in the scene. Instead we generate an embossed version of the input image which offers a perception of 3D look and feel, mapping it onto a height field to immerse the user into the 3D structure for interactive exploration. ECEV specifically aims at mobile platforms where users can take a photo and instantly apply our method to obtain pleasant and fancy results.

**Keywords:** data visualization, image embossing, interactive exploration

## 1 INTRODUCTION

Data visualization exploits techniques from computer graphics, image processing, high-performance computing, feature extraction, human perception, and other domain-specific knowledge to provide visual interpretation and exploration of what would be otherwise unseen from vast amount of data acquired in various scientific, engineering, and social disciplines. While visualization may answer some *questions* such as “given a 2D terrain map, what is the highest point?” or “given a computed tomography (CT) slice image, are there any abnormalities?”, it also helps *discover the unknown* (Telea, 2007) from a given dataset (e.g., an unexpected flow behavior otherwise might not be identified).

There are many techniques for representing 3D models or enhancing the delineation and perception of 2D images. While Image-based rendering (IBR) and image-based modeling (IBM) require a sampling procedure and a collection of images, direct volume rendering (DVR) methods such as Ray Casting (Levoy, 1988) assumes an array of 3D points with at least one associated physical property (e.g., pressure, temperature, intensity, et al) and intensive computation. Another depiction of three-dimensional structures can be done through terrain visualization in which height values are mapped to a 2.5D manifold. However, common approaches in terrain visualization also take a series of accurate images as the input in addition to the ambiguity (i.e., determining the direction of the normal vector) caused during *triangulation* phase.

In light of the aforementioned issues, we present a method for embossed construction and explorative visualization (ECEV) of 3D structures from 2D images. ECEV takes a single image as an input and does not require depth values or any other transitional stages. Given data in the form 2D image, an embossed image is synthesized by the grey-scale intensity of each pixel resulted from in combination of the normal



vector obtained through the local partial derivatives and the light direction. Then, a 2D embossed image of brightness is generated and blended with the original image that enables a sense of 3D look and feel. Furthermore, by mapping the resulting embossed image onto a height surface, which is in fact derived from the intensity values, the user is able to immerse into the structures and interactively explore the initial source image.

## 2 ECEV

As a pixel-based illumination method to synthesize embossing effects, ECEV uses Phong Reflection Model (Newman & Phong, 1975) to approximate the lighting effects in the scene. To emboss a source image, we begin with extracting Red-Green-Blue colors, creating an intensity field which is considered as a 3D surface (and “height” displacements to construct 3D scene). Using the input parameters (i.e., the light position), the light direction is calculated and then incorporated with the normal vectors to obtain the grey-scale intensity of each pixel. Once the shading per pixel is ready, it is synthesized with the original image to get an embossed version of the input.

Next, we apply data map and color map to the embossed image and render in 2D settings employing either pixel-based or geometric rendering techniques. Then, we construct the 3D scene using the modified version of the original illumination values, which are regarded as “height” displacements, to suppress otherwise sheer differences for surface rendering. This height field is delineated in the form of a curved surface in 3D scene by texture-mapping the aforementioned embossed image. Lastly, the visual depiction of the height field in 3D settings is supported by real-time immersive exploration equipped with yaw/pitch self-rotation, yaw revolution, and horizontal/vertical translation (Figure 1).

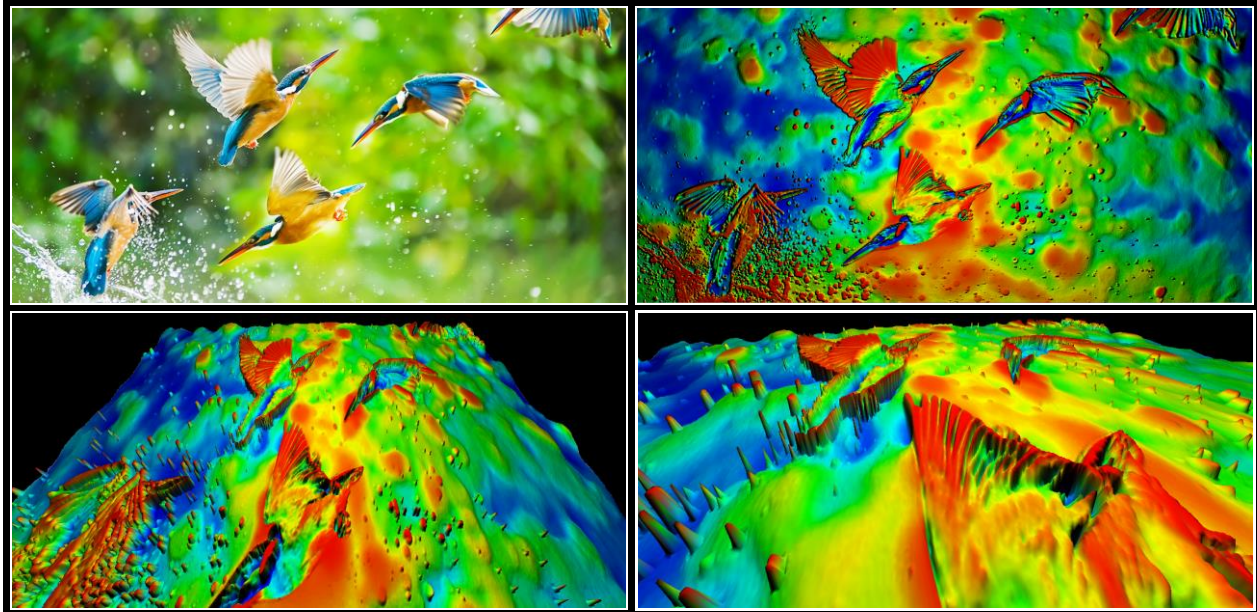


Figure 1. Some outputs from ECEV. The source image (**upper left**; the image courtesy of <https://imgur.com/>) and visualization of the *height field* in 2D (**upper right**) and 3D (**lower left**) settings through data map, color map, image embossing, and texture map, in support of real-time immersive exploration (**lower right**).

## REFERENCES

- Levoy, M. (1988). Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, (3), 29–30.
- Newman, W., & Phong, B. T. (1975). *Graphics and Illumination for Computer Generated Pictures*. <https://doi.org/10.1145/360825.360839>
- Telea, A. C. (2007). *Data visualization: principles and practice*. AK Peters/CRC Press



# ANISOTROPIC MESH ADAPTATION PIPELINE FOR THE 3D LAMINAR FLOW OVER A DELTA WING

Christos Tsolakis and Nikos P. Chrisochoides

CRTC Lab Computer Science Department  
School of Sciences  
Old Dominion University  
{tsolakis,nikos}@cs.odu.edu

## ABSTRACT

In this work, a mesh adaptation tool is loosely coupled with a metric based error estimator and a CFD solver into a computational pipeline. The pipeline is validated using a laminar flow over a delta wing. Drag and lift coefficients are computed and compared to similar simulations from the literature. The proposed anisotropic adaptation method delivers the same accuracy with an order of magnitude fewer resources than the more commonly used isotropic mesh adaptation.

**Keywords:** Anisotropic Mesh Generation, Metric Adaptation, Mesh Adaptation

## 1 INTRODUCTION

Computational Fluid Dynamics (CFD) is concerned with evaluating characteristics and quantities related to the flow of a fluid around a body. The computational aspect of the method is composed mainly of three parts. First, is the Geometry Definition and Geometry Discretization, Solution of the underline equations and finally visualization. While not new, the requirements of this pipeline have undergone significant changes following the increasing demands of industry and academia for better solution resolution, see (Alauzet and Loseille 2016). As a consequence, any new tool needs to be thoroughly tested and verified against previous methods and datasets. In this work, a pipeline consisting of a mesh adaptation software a suite of error estimator and interpolation tools, as well as a CFD solver, are brought together, and the configuration is tested against a well-documented case of a 3D laminar flow over a delta wing.

In this study, mesh adaptation is built upon previous work (Tsolakis, Drakopoulos, and Chrisochoides 2018). More specifically, metric-based adaptation is employed. The crux of this approach is to create an anisotropic mesh by mapping distance and quality evaluation to the metric space. This mapping  $\mathcal{M}$  is given by a  $3 \times 3$  positive definite matrix called metric tensor and it can be shown that it induces an inner product  $\langle u, v \rangle_{\mathcal{M}} := u^T \mathcal{M} v$ . Based on the inner product the familiar formulas for distance, angle and volume evaluation can be rewritten so that they take into account the information encapsulated in  $\mathcal{M}$ . Equipped with these size evaluation tools, the mesh adaptation algorithm will attempt to create elements that optimize the spacing and the quality in the metric space. A high-level description of the general adaptation procedure is given in Figure 1.

Section 4 demonstrates the effectiveness of this approach where quantitative characteristics like the lift and drag coefficients of the flow are evaluated with lower error using an order of magnitude fewer vertices and

consequently an order of magnitude less computational resources than the more commonly used isotropic mesh adaptation.

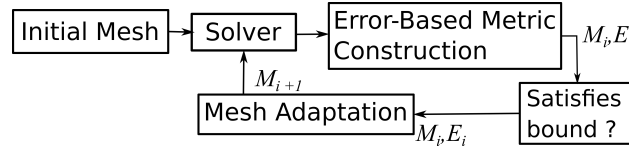


Figure 1: Mesh Adaptation pipeline : At each iteration, the solver provides feedback to the mesh generation module, about the areas of the mesh that need refinement and or coarsening.

## 2 METHODS

In this work, the mesh is adapted using CDT3D (Drakopoulos 2017) a multi-threaded mesh generation software based on advancing front point creation, direct point insertion and a speculative scheme for local reconnection (Drakopoulos, Tsolakis, and Chrisochoides 2017). SU2 (Economon, Palacios, Copeland, Lukaczyk, and Alonso 2016) is used as a CFD solver, it was chosen due to its ease-of-use and open availability. Finally, the `refine` suite of mesh mechanics (Park 2018) is used for evaluating the multiscale metric and interpolating the solution at each iteration to the new, adapted mesh.

### 2.1 Mesh Adaptation

The mesh adaptation software used in this work is CDT3D (Drakopoulos 2017, Drakopoulos, Tsolakis, and Chrisochoides 2017). CDT3D is designed to be the speculative component of the telescopic approach to mesh generation presented in (Chrisochoides 2016). As part of meeting the expectations of the CFD 2030 study (Slotnick, Khodadoust, Alonso, Darmofal, Gropp, Lurie, and Mavriplis 2014) which frames the future needs of the simulation and meshing community, metric adaptation was added (Tsolakis, Drakopoulos, and Chrisochoides 2018) and evaluated (Tsolakis, Chrisochoides, Park, Loseille, and Michal 2019) next to well-know and tested mesh adaptation methods.

The pipeline of CDT3D adaptation for this work is shown in Figure 2 can be divided into four main steps: (a) boundary adaptation which is performed using the mesh library MMGS (Dobrzynski and Frey 2009), (b) initial mesh construction which is starting from a surface mesh builds a tetrahedral mesh that conforms to the boundary, (c) mesh refinement introduces the new points improving the quality of the mesh along the way, and finally mesh quality optimization.

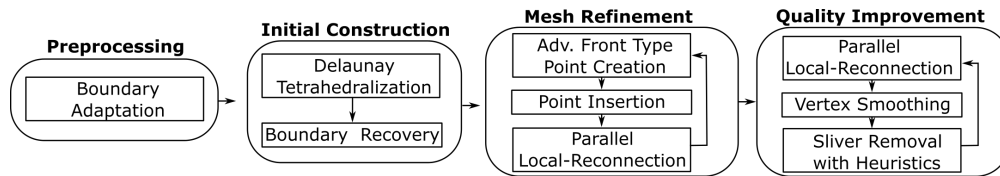


Figure 2: CDT3D mesh generation pipeline

### 2.2 CFD Solver

SU2 (Economon, Palacios, Copeland, Lukaczyk, and Alonso 2016) is a general framework for solving sets of governing equations for multi-physics problems. It is a vertex-based solver and uses dual control volumes for determining the required quantities. It can use either a finite volume method or finite element method

with an edge-based data structure. Both centered and upwind spatial integration schemes are provided. Moreover, it includes features like agglomeration multigrid and preconditioners for low-speed applications which are both utilized in this study. SU2 utilizes MPI to exploit parallelism in distributed memory environments.

### 2.3 Multiscale Metric

The multiscale metric (Loseille, Dervieux, Frey, and Alauzet 2007) is used in order to control the  $L^p$ -norm of the interpolation error of a given scalar field. The multiscale metric is evaluated based on the reconstruction the Hessian  $\mathcal{H}$  of the given scalar field.

It has been shown experimentally (Loseille and Alauzet 2011b) and theoretically (Loseille and Alauzet 2011a) that a mesh conforming to  $\mathcal{M}_{L^p}$  provides optimal control of the scalar field interpolation error in the  $p$ -norm.

In this study, the implementation of `refine`'s multi-scale metric is used (Park 2018). In particular, the  $L^2$ -projection gradient reconstruction scheme is used for Hessian reconstruction and the order of the metric norm is set to  $p = 2$ .

## 3 EXPERIMENTAL SET-UP

In order to meet the ever-evolving and growing demands of the CFD community, a simulation pipeline should be able to integrate a plethora of different tools. The T-infinity project (O'Connell, Druyor, Thompson, Jacobson, Anderson, Nielsen, Carlson, Park, Jones, Biedron, Lee-Rausch, and Kleb 2018), demonstrates a series of different use-cases where a high-level Python interface can be used to build complicated pipelines. In this work, due to the small scale (in terms of computational effort) and complexity of the application, a more straightforward pipeline was built communicating data solely through files. However, the API that the T-infinity project is proposing will be the goal for future applications.

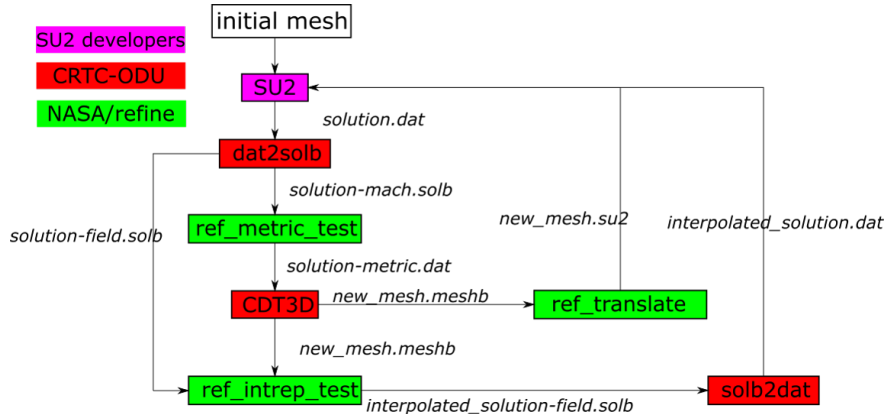


Figure 3: Execution pipeline for the simulation in this study

The absence of common file formats for meshes and solutions required the creation some converters. The final pipeline can be seen Figure 3. `dat2solb` and `solb2dat` are used to convert the solution format between SU2 and the `refine` tools. `ref_metric_test` creates the metric tensor out of the current solution. `ref_intrep_test` interpolates the old solution to the new mesh. This step allows the solver to restart the calculation from a state closer to the final solution than starting from the freestream values which is the default. The values of the previous solution are interpolated using linear interpolation. Finally,

`ref_translate` converts the mesh from the `.meshb` format to the `.su2` so that the solver can process it. Although, at first this pipeline may seem highly tailored to the specific format, the flow of data follows the abstract pipeline of mesh adaptation of Figure 1.

The simulation chosen for this study is a laminar flow over a delta wing. The flow conditions have been set so that they match the case used in the first three High-Order Workshops (Wang, Fidkowski, Abgrall, Bassi, Caraeni, Cary, Deconinck, Hartmann, Hillewaert, Huynh, Kroll, May, Persson, Leer, and Visbal 2013). In particular, the freestream conditions are 0.3 Mach, 4000 Reynolds number based on a root chord length of 1 and  $12.5^\circ$  angle of attack. The wing surface is modeled as an isothermal no-slip boundary with the freestream temperature. The Prandtl number is 0.72. The viscosity is assumed constant.

As initial mesh, a model from the Unstructured Grid Adaptation Working Group (UGAWG)<sup>1</sup> was used. This mesh can be seen in Figure 4. The computations were performed on a 24 core workstation (2 x Intel<sup>®</sup> Xeon<sup>®</sup> E5-2697 v2 @ 2.70GHz) with 757GB RAM.

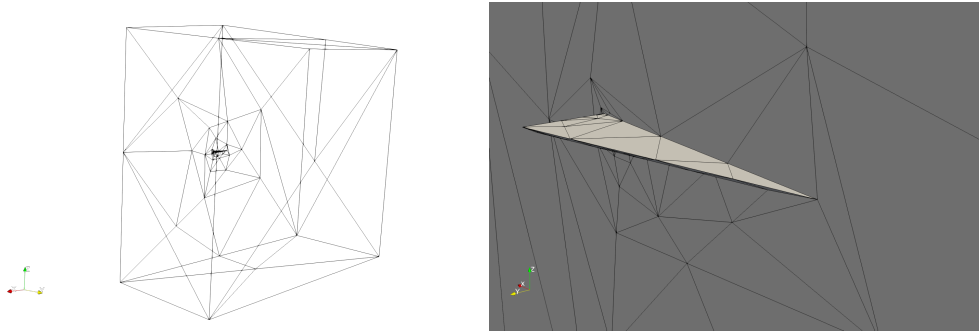


Figure 4: Initial mesh, # vertices 143 # tetrahedra 434, available at <https://github.com/UGAWG/solution-adapt-cases>

## 4 RESULTS

In this section, both quantitative and qualitative results are presented.

Figure 9 in appendix A depicts the evolution of the solution after each adaptation iteration. The generated Mach contour lines capture the features of the vortex and match the ones presented in (Leicht and Hartmann 2010).

Another important issue when simulating flow where vortices are present is that the vortex created by the flow should be resolved in the wake region of the wing. Since this is the expected behavior for these simulation parameters both in physical and computational tests. Figure 5 depicts the evolution of the solution at a distance of 4 cord lengths from the trailing edge.

The structure of the vortex core can also be seen in Figure 6 where streamlines are used to visualize the flow through the vortex.

One of the most significant advantages using metric-based anisotropic mesh adaptation is that the same level of accuracy can be achieved with a smaller mesh reducing thus the running time of all the components in the computational pipeline. To illustrate this result for this study, two more configurations were tested both using the same mesh adaptation tool. First, as a baseline, a uniform mesh refinement was used. In the uniform refinement case the size of all the elements was fixed and for every iteration it was approximately reduced to half. Second, an isotropic size was derived from the metric tensor by disregarding the directional

<sup>1</sup><https://ugawg.github.io/>

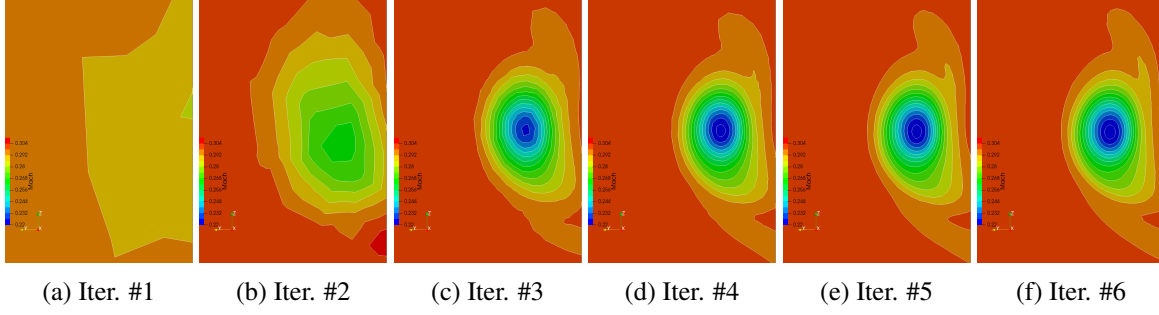


Figure 5: Slice of the vortex core in the wake region 4 cord lengths from the trailing edge at each iteration.

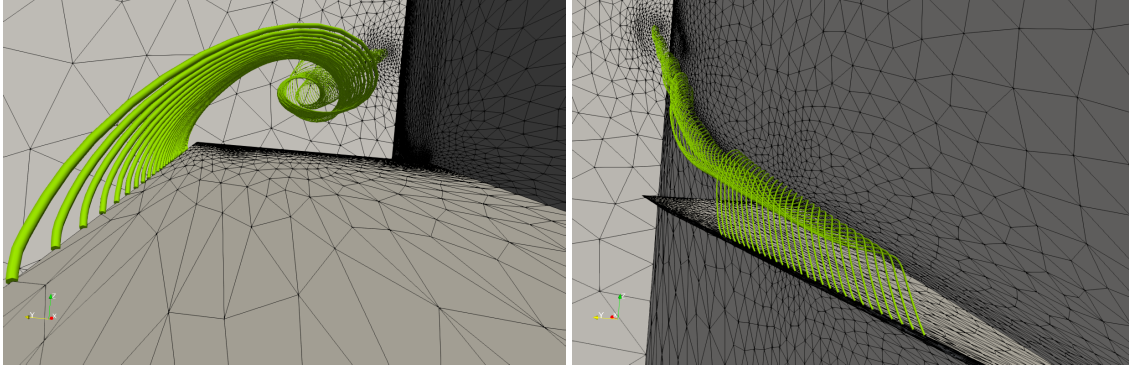


Figure 6: Streamlines of the final solution, # vertices 122,384 # tetrahedra 714,018

information (i.e. its eigenvectors) and using the smallest directional size ( i.e. the largest eigenvalue) to control the local sizing.

The three mesh refinement approaches were compared against each other using the lift and drag coefficients as calculated by SU2 ( see Figure 7). As a reference, the values  $C_l^{ref} = 0.347$  and  $C_d^{ref} = 0.1658$  have been used (Leicht and Hartmann 2010).

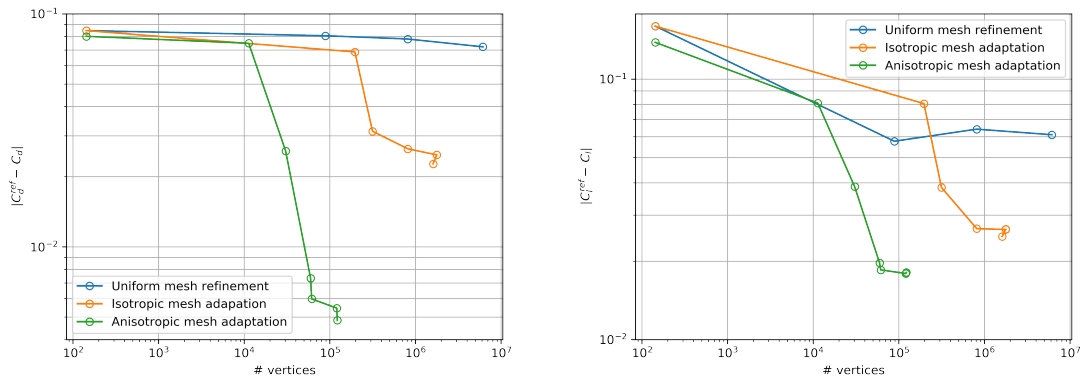


Figure 7: Absolute error of lift and drag coefficients, for three different types of mesh refinement.

	Mesh Refinement Method		
	Uniform (3 iterations)	Adaptive Isotropic (6 Iterations)	Adaptive Anisotropic (6 Iterations)
Solver Time	817.28 min	151.95 min	93.05 min
Mesh Adaptation Time	23.75 min	14.31 min	12.05 min

Table 1: Simulation time comparison for the three mesh refinement approaches

Figure 8 depicts the upper surface of the wing of the final iteration for each mesh refinement approach. The anisotropic approach packs more efficiently the points along the pressure contour lines. The uniform mesh is also included for reference.

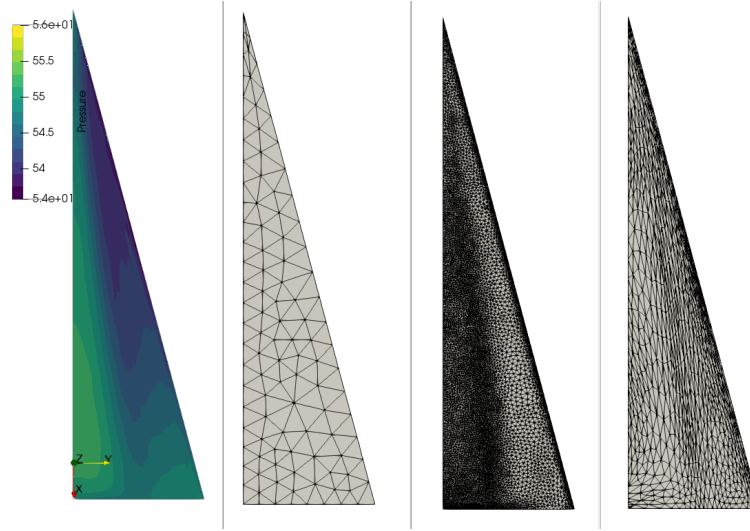


Figure 8: From the left : Pressure contours on the surface of the wing. Final mesh for the uniform, isotropic and anisotropic case. Respective number of vertices : 6.140.030, 1.609.346 , 122.384

The anisotropic mesh adaptation outperforms the other two approaches, in particular it can achieve lower error with an order of magnitude fewer vertices. Table 1 indicates that these gains are also reflected in the total time of the simulation.

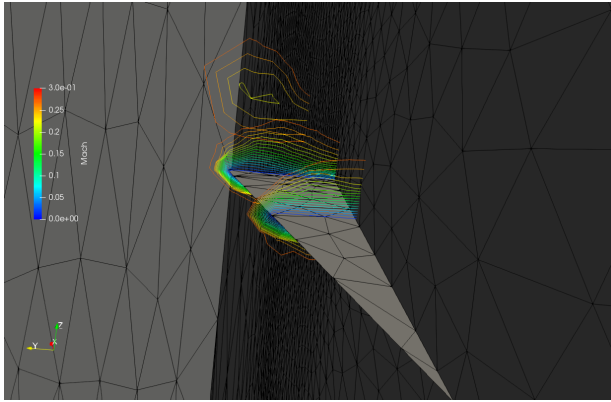
## 5 CONCLUSION

In this work, three different software components were brought together and a simulation pipeline was built. The simulation results are promising and agree with already published work. The use of anisotropic mesh adaptation enables lower error with an order of magnitude fewer elements, that can also be extrapolated to an order of magnitude lower use of resources, power and time optimizing thus user-productivity. In the future, the same case will be executed at higher complexity to explore both the solution convergence as well as to study the scalability of all three components.

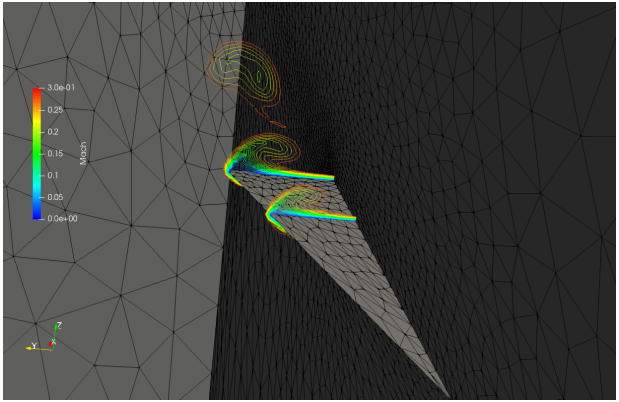
## ACKNOWLEDGMENTS

This research was sponsored in part by the NASA Transformational Tools and Technologies Project (NNX15AU39A) of the Transformative Aeronautics Concepts Program under the Aeronautics Research Mission Directorate, NSF grant no. CCF-1439079, the Richard T. Cheng Endowment and the Modeling and Simulation fellowship of Old Dominion University.

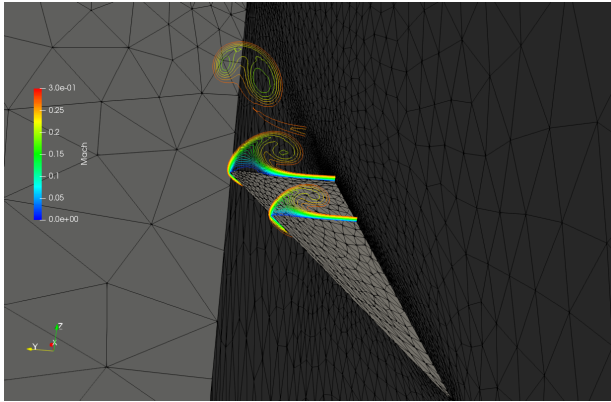
## A APPENDIX



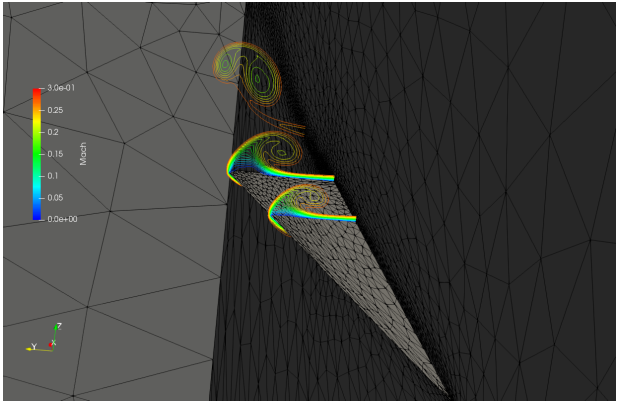
(a) # vertices 11,313 # tetrahedra 58,333



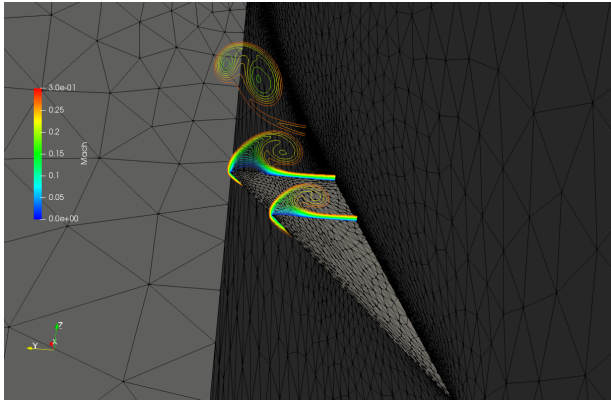
(b) # vertices 30,509 # tetrahedra 171,425



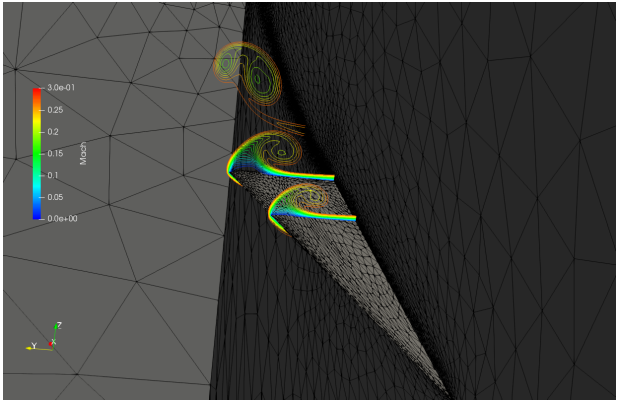
(c) # vertices 59,726 # tetrahedra 344,379



(d) # vertices 61,455 # tetrahedra 355,650



(e) # vertices 120,612 # tetrahedra 702,791



(f) # vertices 122,384 # tetrahedra 714,018

Figure 9: Adapted surface mesh and contour lines of the Mach number for each iteration.



## REFERENCES

- Alauzet, F., and A. Loseille. 2016, March. “A decade of progress on anisotropic mesh adaptation for computational fluid dynamics”. *Computer-Aided Design* vol. 72, pp. 13–39.
- Chrisochoides, N. P. 2016. “Telescopic Approach for Extreme-Scale Parallel Mesh Generation for CFD Applications”. In *46th AIAA Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics.
- Dobrzynski, C., and P. Frey. 2009. “Anisotropic Delaunay Mesh Adaptation for Unsteady Simulations”. In *Proceedings of the 17th International Meshing Roundtable*, edited by R. V. Garimella, pp. 177–194. Springer Berlin Heidelberg.
- Drakopoulos, F. 2017. *Finite Element Modeling Driven by Health Care and Aerospace Applications*. Ph. D. thesis, Computer Science, Old Dominion University.
- Drakopoulos, F., C. Tsolakis, and N. P. Chrisochoides. 2017. “Fine-Grained Speculative Topological Transformation Scheme for Local Reconnection Methods”. In *AIAA Aviation Forum*, pp. accepted.
- Economon, T. D., F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. 2016. “SU2: An Open-Source Suite for Multiphysics Simulation and Design”. *AIAA Journal* vol. 54 (3), pp. 828–846.
- Leicht, T., and R. Hartmann. 2010, September. “Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations”. *Journal of Computational Physics* vol. 229 (19), pp. 7344–7360.
- Loseille, A., and F. Alauzet. 2011a. “Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error”. *SIAM Journal on Numerical Analysis* vol. 49 (1), pp. 38–60.
- Loseille, A., and F. Alauzet. 2011b, January. “Continuous Mesh Framework Part II: Validations and Applications”. *SIAM Journal on Numerical Analysis* vol. 49 (1), pp. 61–86.
- Loseille, A., A. Dervieux, P. Frey, and F. Alauzet. 2007, June. “Achievement of Global Second Order Mesh Convergence for Discontinuous Flows with Adapted Unstructured Meshes”. In *18th AIAA Computational Fluid Dynamics Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics.
- O’Connell, M., C. Druyor, K. B. Thompson, K. Jacobson, W. K. Anderson, E. J. Nielsen, J.-R. Carlson, M. A. Park, W. T. Jones, R. Biedron, E. M. Lee-Rausch, and B. Kleb. 2018, June. “Application of the Dependency Inversion Principle to Multidisciplinary Software Development”. In *2018 Fluid Dynamics Conference*, AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics.
- Park, Michael A. 2018. “refine GitHub site”. <https://github.com/nasa/refine>.
- Slotnick, J., A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis. 2014, March. “CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences”. Technical report.
- Tsolakis, C., N. Chrisochoides, M. A. Park, A. Loseille, and T. Michal. 2019. “Parallel Anisotropic Unstructured Grid Adaptation”. In *AIAA SciTech Forum 2019*.
- Tsolakis, C., F. Drakopoulos, and N. Chrisochoides. 2018, April. “Sequential Metric-Based Adaptive Mesh Generation”. In *Modeling, Simulation, and Visualization Student Capstone Conference*. Suffolk, VA.
- Wang, Z. J., K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P.-O. Persson, B. v. Leer, and M. Visbal. 2013. “High-order CFD methods: current status and perspective”. *International Journal for Numerical Methods in Fluids* vol. 72 (8), pp. 811–845.



# CAPSULE NETWORKS: REVIEW OF THE STRUCTURES AND VISUALIZATION OF FEATURES

Daniel Perez

Department of Modeling, Simulation and Visualization Engineering  
Old Dominion University  
Norfolk, VA, USA  
dpere013@odu.edu

## ABSTRACT

Capsule networks are a recently proposed machine learning model that is producing significantly promising results in different datasets. There are currently two main approaches to develop capsule networks: the first one defines a capsule as a vector of instantiation parameters of an entity, and its length corresponds to the probability of finding that entity in the input space; the second one establishes a capsule as a pose matrix and an activation probability. In this paper, both structures are analyzed and compared, as well as the performance of the capsules in complex data. Additionally, a novel visualization of the capsule features is proposed to assess the performance of the model in the MNIST and CIFAR-10 datasets.

**Keywords:** Capsule Networks, Expectation Maximization, Routing, Feature visualization.

## 1 INTRODUCTION

Deep learning consists on learning representations with multiple levels of abstraction to achieve high-accuracy predictions on a set of data (LeCun et al. 2015). During the last years, applications of deep learning have been successful in numerous fields such as image classification (Krizhevsky et al. 2012, He et al. 2016), object detection and tracking (Redmon and Farhadi 2016, Ren et al. 2015), speech recognition (Hinton et al. 2012), autonomous driving (Chen et al. 2015, Huval et al. 2015) and cybersecurity (Sommer and Paxson 2010, Najafabadi et al. 2015). Among different deep learning models, deep Convolutional Neural Networks (CNNs) are significantly popular for various applications. A CNN consists of a set of convolutional layers followed by several fully-connected layers. The convolutional layers learn effective representations for the raw data and the fully connected layers perform classification or regression based on these learned representations. Many CNN based image classification systems can perform an end-to-end learning in which feature extraction (representation learning) is jointly optimized with classification, and it is believed that this automatic feature learning process plays a critical role to achieve the superb performances of CNNs.

However, CNNs have major drawbacks, and a new deep learning model has been recently proposed to overcome these problems: Capsule Networks (Sabour et al. 2017). This paper describes and analyzes the purpose of capsule networks and compares the two main structures to design these models. Specifically, the original capsule network (Sabour et al. 2017) is discussed as well as its performance when applied to complex data (Xi et al. 2017). Then, a different approach in defining capsule networks as matrices (Hinton

et al. 2018) is analyzed and compared to the original structure. Finally, a novel visualization of the last capsule layer is proposed to assert its performance in two different datasets.

The remainder of this paper is organized as follows. Section 2 discusses the current limitations of CNNs that capsule networks aim to overcome. The structure of the original capsule networks and its performance in different datasets is analyzed in Section 3, while Section 4 reviews the matrix capsules and their differences with the previous model. The visualization of the capsule network is proposed in Section 5, and the conclusions of the paper are drawn in Section 6.

## 2 DRAWBACKS OF CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks are one of the most popular deep learning models in the present. They have achieved state-of-the-art accuracies in multiple datasets, and they have proven to be specially efficient in image recognition and object detection tasks. However, there exist some fundamental problems with these models which capsule networks aim to solve.

The main problem of CNNs is that they are unable to detect the pose (position, rotation and scale) of the predicted entities. For instance, a CNN that is trained to predict whether an input image is a face might succeed on its prediction, but will be unable to determine the pose of the face in the image. This problem can lead to classification errors when the entity is found in an uncommon scenario within the image (e.g. the face is rotated 180 degrees). This problem occurs due to the loss of information as the image features are passed through the CNN layers. Lower layers of a CNN extract the representation features of the input image, and a routing operation is used to reduce the size of the layers and obtain higher representations of the data. This routing is generally achieved with an operation that is called max pooling. A max pooling layer takes an image and divides it into non-overlapping regions. For each region, it outputs the maximum number, generating an output image of reduced size. The pooling operation with a kernel of 2x2 (size of the regions) is shown in Figure 1. The problem with this routing mechanism is that there is a significant amount of loss of information from layer to layer. Consequently, the information about the entity's pose is lost through the layers of the CNN. Since the location of each feature is lost through the routing process, a new problem arises. A picture containing all the features of an entity in wrong positions (e.g. a picture with a nose, eyes, mouth and ears in random positions without forming a face) might be misclassified by a CNN. This effect occurs because CNNs treat features as invariant entities (i.e. independent from each other), whereas the capsule network's parameters are equivariant: they detect objects that can transform to each other.

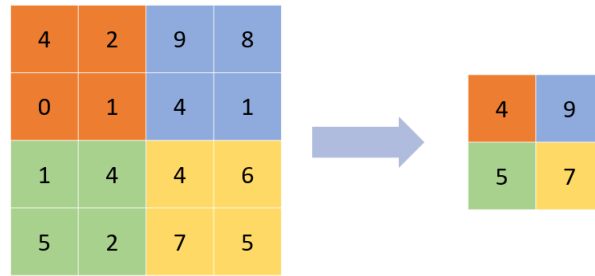


Figure 1: Max Pooling operation with a 2x2 kernel.

Additionally, CNNs are vulnerable to noise input attacks. These attacks consist on slightly modifying the input image based on the parameters of the CNN so that it is misclassified by the deep learning model. These modifications can be computed using different methods, being Fast Gradient Step Method (FGSM) (Goodfellow et al. 2014) one of the most popular ones. FGSM consists of computing the gradient of the

CNN's loss depending on the intensity of each pixel from the input image, and then changing the pixel intensity in a direction that increases the loss (Goodfellow et al. 2014). This can result in a targeted attack in an input image by updating its pixels to increase the classification probability of a wrong class. Another example of adversarial attacks against CNNs is the Basic Iterative Method, which is a slightly more complex version of FGSM in which the attack takes multiple small steps in the computation of the adversarial image (Kurakin et al. 2016). The ability of capsule networks to withstand these attacks is evaluated in the paper written by (Hinton et al. 2018), which will be discussed on Section 4.

### 3 ORIGINAL CAPSULES

Capsule networks had been previously discussed by G. Hinton in several occasions during the last decade, but they were officially proposed in 2017 by (Sabour et al. 2017). In their paper, the authors define a capsule as a group of neurons that represents the instantiation parameters of an entity in the input image, while the length of the capsule represents the posterior probability that the entity exists in the input image. The instantiation parameters include different properties of the image such as the pose, deformation, albedo, texture etc. Grouping these instantiation parameters into capsules ensures that the entity's information is not lost through the capsule layers. The probability of the entity existing in the image is computed by calculating the length of the capsule vector. This length can be greater than 1, but probability is a floating number between 0 and 1. To solve this issue, a non-linear operation called "squashing" is applied to each capsule vector so that short vectors' length is almost zero and large vectors' length is close to 1. The "squashing" operation is shown in Eq. 1, where  $s_j$  and  $v_j$  are the input and output vectors in capsule  $j$  respectively.

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (1)$$

The output of capsule  $j$  is calculated as a weighted sum of all the capsule vectors in the previous layer, as shown in Eq. 2. For each capsule vector  $u_i$ , there exists a weight matrix  $W_{ij}$  whose parameters are learned during the training process. The vector  $c_{ij}$  is calculated through a dynamic routing process called "routing by agreement."

$$s_j = \sum_i c_{ij} W_{ij} u_i \quad (2)$$

The routing function is a softmax operation in which the logits  $b_{ij}$  represent the prior probabilities that capsule  $i$  should be assigned to capsule  $j$ , as shown in Eq. 3.

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (3)$$

"Routing by agreement" offers an alternative to max pooling guaranteeing that information is not lost from capsule to capsule, and ensuring that capsules with similar instantiation parameters are grouped together. The routing process is then computed as follows. The initial logits  $b_{ij}$  for each input capsule  $i$  and output capsule  $j$  (in the next layer) are initialized to 0. Then, the following operations are performed for an arbitrary number of iterations:

1. For all capsule  $i$  in the input layer, compute the vectors  $c_{ij}$  using the softmax function from Eq. 3.
2. For all capsule  $j$  in the output layer, calculate the capsule's output using Eq. 2.

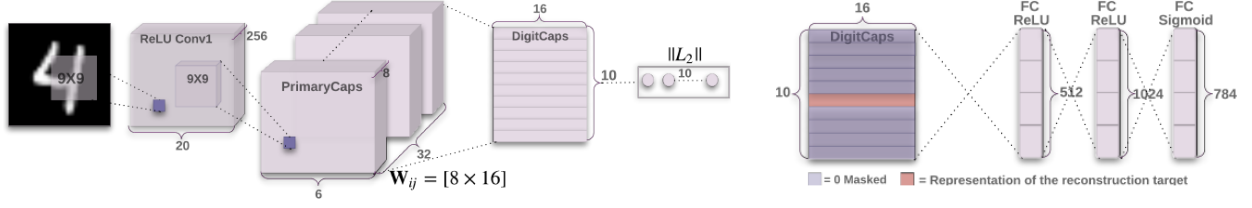


Figure 2: Structure of the Capsule network as proposed by Sabour *et al.*

3. For all capsule  $j$  in the output layer, perform the "squashing" operation (Eq. 1).
4. For all capsules  $i$  and  $j$ , increase the logit unit  $b_{ij}$  by the activation defined as the scalar product  $a_{ij} = v_j \hat{u}_{ji}$ , where  $\hat{u}_{ji} = W_{ij} u_i$ .

After trying different configurations, the authors set the number of iterations to 3 and establish their network structure as the one shown on Figure 2. Their capsule network has one simple convolutional layer for feature extraction, followed by a capsule layer called *PrimaryCaps* in which the neurons are grouped into capsules after a 9x9 convolutional kernel is applied to the previous layer. The last capsule layer, *DigitCaps* consists of 10 capsules, each of them corresponding to one the 10 classes that will be predicted by the network. The routing process is only computed between this layer and *DigitCaps*. The probability of an entity is computed as the length of its corresponding capsule in *DigitCaps*.

Additionally, the authors include a decoder that is able to reconstruct the input images from the *DigitCaps* layer. During training, the capsule vector corresponding to the training label of the image is used to reconstruct the input image as a regularization for the optimization. The error between the reconstructed image and the input image is then used to optimize both the reconstruction weights and the weights in the capsule network through back propagation. Thus, the weights  $W_{ij}$  are updated during the training process according to the decoder loss and the loss of the capsule network, for which the authors use the separate margin loss function shown in Eq. 4. The loss  $L_k$  is computed for each capsule  $k$ .  $T_k = 1$  if and only if an entity of class  $k$  is present.

$$L_k = T_k \max(0, 0.9 - \|v_k\|)^2 + 0.5(1 - T_k) \max(0, \|v_k\| - 0.1)^2 \quad (4)$$

The authors tested their capsule network in the MNIST dataset, which consists of 28x28 black and white images of digits 0-9 (LeCun et al. 2010). They obtained an accuracy of 99.75% which, at the time of writing this paper, represents state-of-the-art on this dataset. Additionally, the authors showed promising results on the classification of overlapping images (MultiMNIST). They also showed the efficacy of the decoder on reconstructing a digit from the output of *DigitCaps*. They demonstrated that applying the reconstruction as a regularization technique increased the accuracy of the capsule network, and they proposed an experiment to observe the implication of each capsule's features in the reconstructed image. To this end, they reconstructed a digit from the output of *DigitCaps*, and then they applied small perturbations to just one of the features of the capsule, while maintaining the other features frozen. Their experiments showed how each feature corresponded to an instantiation parameter such as scale, thickness, translation or skew.

### 3.1 Application of the Capsule Network on Complex Data

Sabour *et al.* mentioned that applying their capsule network to more complex data did not produce improvements from the current state-of-the-art and set their future work to test the network in more complex data, but they did not give specific results of their approaches. This problem was further investigated by (Xi

et al. 2017). In their paper, they attempted to find the best configuration of a capsule network that yields the minimal test error on the CIFAR-10 dataset. This dataset contains 32x32 colored images divided in the following 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck (Krizhevsky et al. 2014). To do this, they started with the original structure proposed by (Sabour et al. 2017) and investigated the effect of applying the following modifications:

- Adding more layers to account for the complexity of the CIFAR-10 dataset.
- Increasing the number of capsules in *PrimaryCaps* to allow the capsule network to learn more features about the data.
- Use ensemble averaging, which consists of training a group of neurons together and averaging their predictions during the testing phase. Sabour *et al.* obtained a 10.6% testing error using this method in the CIFAR-10 dataset.
- Change the scaling of each pixel for reconstruction loss to account for the higher complexity of the CIFAR-10 dataset.
- Increase the amount of simple convolutional layers prior to the capsule layers to extract more complex features from the data.
- Replacing the squash function by  $f(x) = (1 - \frac{1}{|e|^x}) \frac{x}{|x|}$ . The authors hypothesized that their function would be more sensitive to changes in the data ( $x$ ) which would lead to more separation between classes.
- Utilizing a class for the data that does not belong to any class ("none of the above" category).

The best accuracy they obtained was 71.55%. This result was achieved using a 4 ensemble and 2 convolutional layers in a model trained on 50 epochs. This is far from the state-of-the-art (96.53%) and significantly lower than the result reported by Sabour *et al.* (89.4%). In addition, it was demonstrated that the reconstruction network did not perform well when applied to a high-dimensional image. While most of their proposed method did not perform well on improving the capsule network's accuracy, the authors offered a comprehensive study on the application of capsule networks to complex data. They showed that the model proposed by Sabour *et al.* was still not optimally configured to be applied in complex data and set the future work in studying the performance of the matrix capsules, which is the structure that will be analyzed in Section 4.

## 4 MATRIX CAPSULES

A different version of the capsule network was released by the same team 3 months after proposing the original model. In this version, a capsule is defined as a 4x4 pose matrix and an activation probability. This implementation substitutes the instantiation parameter vector by a matrix and, instead of obtaining the probability by computing the length of the vector, they have a separate variable to store it. Similarly to the original model, the vote of assigning capsule  $i$  in the layer  $\Omega_L$  to capsule  $j$  in the layer  $\Omega_{L+1}$  is computed by multiplying the pose matrix  $M_i$  by a 4x4 trainable transformation matrix  $W_{ij}$ . This vote  $V_{ij}$  is then used as an input in the routing procedure to calculate the pose matrices and activations of the capsules in layer  $\Omega_{L+1}$ .

The routing procedure in this case is based on the Expectation Maximization (EM) algorithm, which is a method to fit a set of Gaussian models in an unlabeled dataset. With this method, the authors are able to adjust the mean, variances and activation probabilities of the capsules in the next layer and the assignment probabilities between all capsules. EM routing consists of performing the *M-step* and *E-step* for a number of iterations. The *M-step* consists of computing the mean  $\mu$  and the standard deviation  $\sigma$  of the input capsules based on the assignment probabilities, votes and activations of the children capsules. Besides, the activation

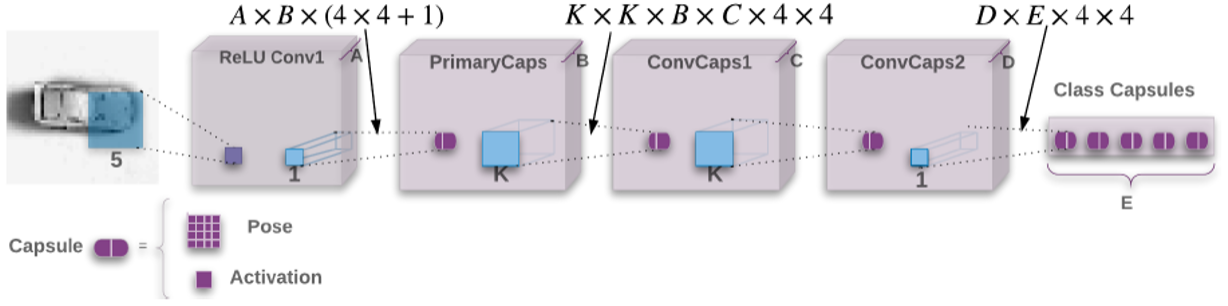


Figure 3: Structure of the Matrix-Capsule network as proposed by Hinton *et al.*

of the parent capsules  $a_j$  is recalculated based on  $\mu$  and  $\sigma$ . In the *E-step*, the assignment probabilities  $r_{ij}$  between capsules  $i$  and  $j$  are recalculated using the new  $\mu$ ,  $\sigma$  and  $a_j$ .

The authors set the number of iterations in the routing procedure to 3, and they proposed a capsule network structure as the one shown in Figure 3. The network consists of a convolutional layer with a 5x5 kernel followed by 4 capsule layers in which the neurons are grouped into 4x4 pose matrices and 1 activation probability. The first capsule layer *PrimaryCaps* acts as a filter so that the neurons can be grouped into capsules. Then, the 2 following convolutional capsule layers (*ConvCaps1* and *ConvCaps2* respectively) act as a regular convolution but applying the EM routing procedure to calculate the new output. The last layer (Class Capsules) outputs the poses and activations of each predicted class. The loss to train the capsule network's transformation matrix is changed to "spread loss" with the goal of maximizing the gap between activations. Eq. 5 shows how the loss between the activation of a wrong class  $a_i$  and the target class  $a_t$  is computed. In the equation,  $m$  is a margin variable that starts at 0.2 and it is linearly increased to 0.9 during the training process.

$$L_i = (\max(0, m - (a_t - a_i)))^2, L = \sum_{i \neq t} L_i \quad (5)$$

The authors performed different experiments to assess the performance of their model. They tested their matrix capsules in the smallNORB dataset, which contains 96x96 gray-level stereo images of 5 toys (airplanes, cars, trucks, humans and animals) in 18 different azimuths, 9 elevations and 6 lighting conditions (Huang, Fu Jie and LeCun, Yann 2009). Their main experiment consisted on downsampling the dataset images to 48x48 and train the capsule network on randomly cropped 32x32 patches, adding random brightness and contrast in each patch. Then they tested the capsule network cropping a 32x32 patch from the center of each testing image. They reported a best test error rate of 1.4%, which beats the previous state-of-the-art set at 2.56%. They also tested the original capsule network (Section 3) and obtained an error of 2.6%, proving that the new model performed better.

Additionally, they performed two experiments to demonstrate the advantages of capsule networks against CNNs. The first one consisted on testing the capsule network and a baseline CNN on images with azimuths and elevations that were not included on the training set. They demonstrated that the error of applying the capsule network to novel viewpoints was reduced by about 30% with respect to the CNN. The second experiment consisted on testing the robustness of the capsule network against adversarial attacks such as FGSM and BIM. They applied these two attacks at different intensities to the baseline CNN and the capsule network, and they found that their model was significantly less vulnerable to this kind of attacks. The baseline CNN quickly dropped its accuracy to below 10%, whereas the capsule network did not perform below 50% in any of the attacks.

With matrix capsules, the authors built on their previous work and offered a model that improved the accuracy of the capsules in more complex data. Their new model provides a new structure for the capsules, a new routing algorithm and a new loss functions that overcomes some of the limitations of their previous model and produces better results. They reduced the number of errors in the smallNORB dataset by 45% with respect to the previous state-of-the-art and proved the advantages of capsule networks with respect to CNNs. There is a capability of the original capsule network that is missing in this implementation: the reconstruction of the input image from the capsule’s output. The authors do not make any reference to this functionality but, taking into account the study by (Xi et al. 2017) in which they prove how the reconstruction is not optimal on complex data, it could be assumed that they were not able to obtain good results in this dataset and decided to omit this feature. The future work of the authors is set to investigate the applications of capsule networks in significantly more complex datasets such as ImageNet.

## 5 VISUALIZATION OF CAPSULE PARAMETERS

Capsule networks are a very recent deep learning model and, at the moment of writing this paper, the research on this method is significantly limited. One of the areas that have not been explored yet is the visualization of the features learned by the model. Visualizing the features learned by CNNs is a well extended practice to understand what is going on inside of the neural network. The general method to visualize CNNs consists of generating images that correspond to the features learned by the deep learning model through deconvolution operations (Springenberg et al. 2014, Simonyan et al. 2013). However, a simpler visualization of the capsule features can be obtained using the original model developed by (Sabour et al. 2017). This visualization consists on a 2D plot of the capsules in the *DigitCaps* layer. The visualization is obtained through the following steps:

1. For each class in a dataset, collect all the images belonging to class  $X$  and pass them through the capsule network.
2. For each image, extract the feature matrix computed by the layer *DigitCaps*, which contains  $c$  capsules (where  $c$  is the number of classes), each of them with a size of  $n$  features.
3. Reshape each feature matrix into a 1-dimensional vector in which the first  $n$  numbers are the features corresponding to the first class, the next  $n$  are the ones corresponding to the second class and so on. This feature vector has a total size of  $c * n$ .
4. Average all the feature vectors belonging to class  $X$  and plot them in a 2D graph. Since the probability of an entity belonging to class  $X$  is measured by the length of its instantiation parameters (or features), the absolute value of the features belonging to class  $X$  should be significantly bigger than the rest of the features.

Figure 4 shows the visualization process from the *DigitCaps* layer to the 2D plot. This visualization was applied to the original structure shown in Figure 2 on the MNIST and CIFAR-10 datasets. In both cases, the number of classes is  $c = 10$  and the size of the features is  $n = 16$ , which results on a feature vector with a size of 160 features.

The capsule network was trained in Python using the open source implementation provided by (Guo 2017), which follows the original implementation proposed by Sabour *et al.*. The capsule network was trained for 50 epochs in both datasets. The validation accuracy was 99.66% in the case of the MNIST dataset and 72.83% in CIFAR-10, which is consistent with the literature.

Figures 5 and 6 show the visualizations produced when the capsule network was trained on MNIST and CIFAR-10 respectively. In the figures, a plot is generated for each class of the dataset. It can be seen that in the case of the MNIST visualization, the features from the class that is fed to the network are significantly

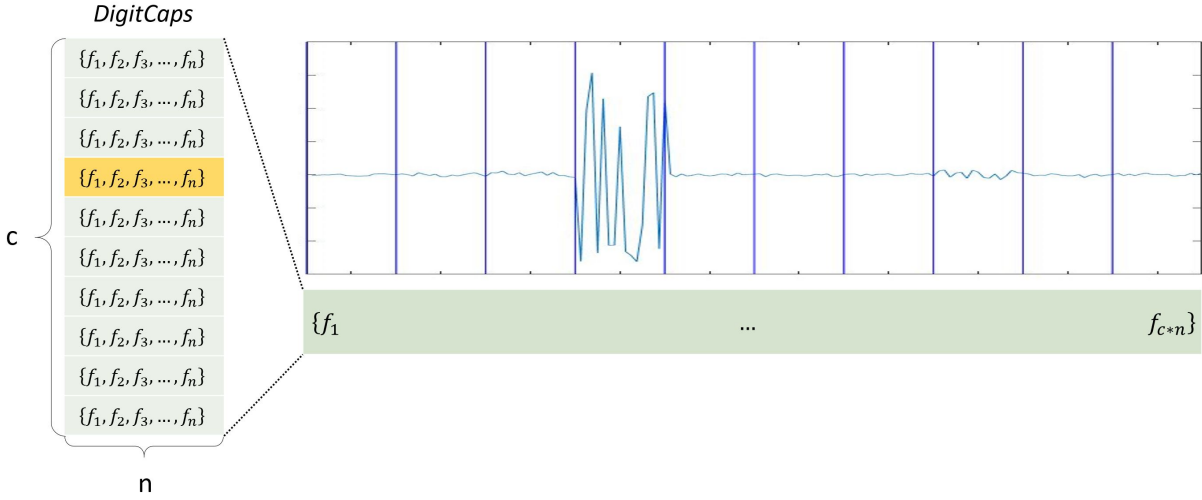


Figure 4: Creating the visualization plot of the capsules from *DigitCaps*

big, while the rest of the features are very close to zero. In the case of the CIFAR-10 visualization, the features belonging to the fed class are generally bigger, but it can be observed that some features are considerably large. This can cause the capsule network to misclassify some images, and it is the reason that the model performs worse in the CIFAR-10 dataset. It is interesting to observe that similar classes have similar features' size. For instance, in the case of the classes for cat and dog, there are certain features that are big in both graphs. It can be determined that these features are activated when a small animal is feed to the network. There are also similarities between the plots of deer-horse and automobile-truck. It is especially interesting the plot of the bird class, in which capsules belonging to all the other animals (cat, deer, dog, frog and horse) are activated, as well as the capsule feature's from the airplane class. This graph tells us that the capsule network finds specially difficult the prediction of the bird class. Knowing this, future steps can be taken to improve the network's performance.

It is important to note that this visualization is only applicable to the original structure of the capsule network. The visualization of matrix capsules following this method would not be meaningful, since their probability is not calculated using the length of the capsule vector. For this reason, the visualization of matrix capsules is not studied in this paper.

## 6 CONCLUSIONS

This paper offers a review and analysis of capsule networks and their main structures. Capsule networks are a very recent deep learning method that is obtaining very promising results, and future research on this topic is fundamental in the implementation of new machine learning solutions. To help in this regard, an analysis of the two main structures and a comparison between them was offered, and the application of capsule models to complex data was assessed.

Additionally, a novel visualization of capsule features was proposed. The visualization consisted on a 2D plot of the capsule features from the last network's layer. It was demonstrated that meaningful information can be obtained from the visualization that can lead to the capsule network's improvement. The future work of this project will be focused on offering solutions from the results of the visualization, as well as providing a visualization of the matrix capsules.



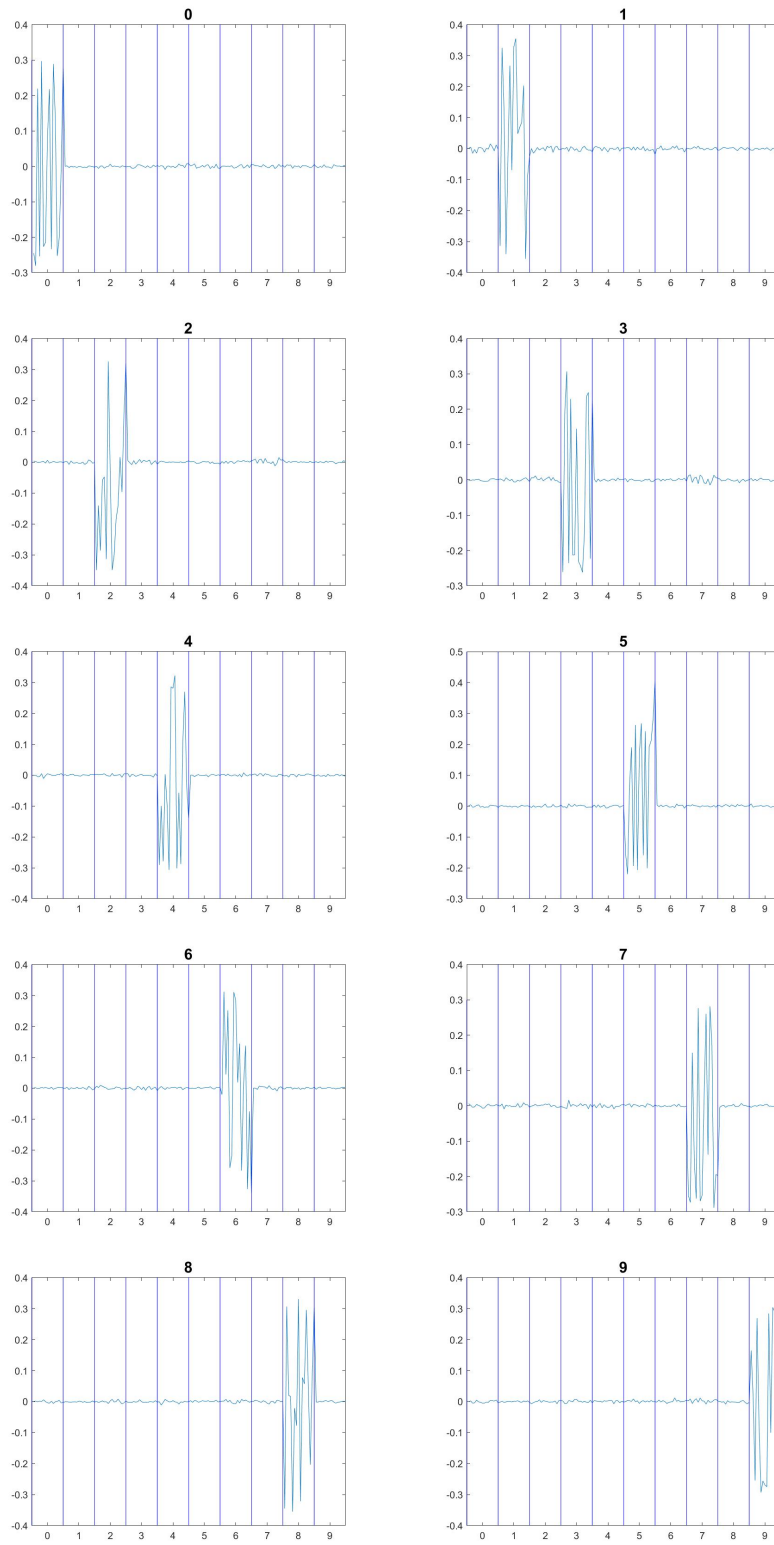


Figure 5: Visualization of Capsule features in the MNIST dataset.

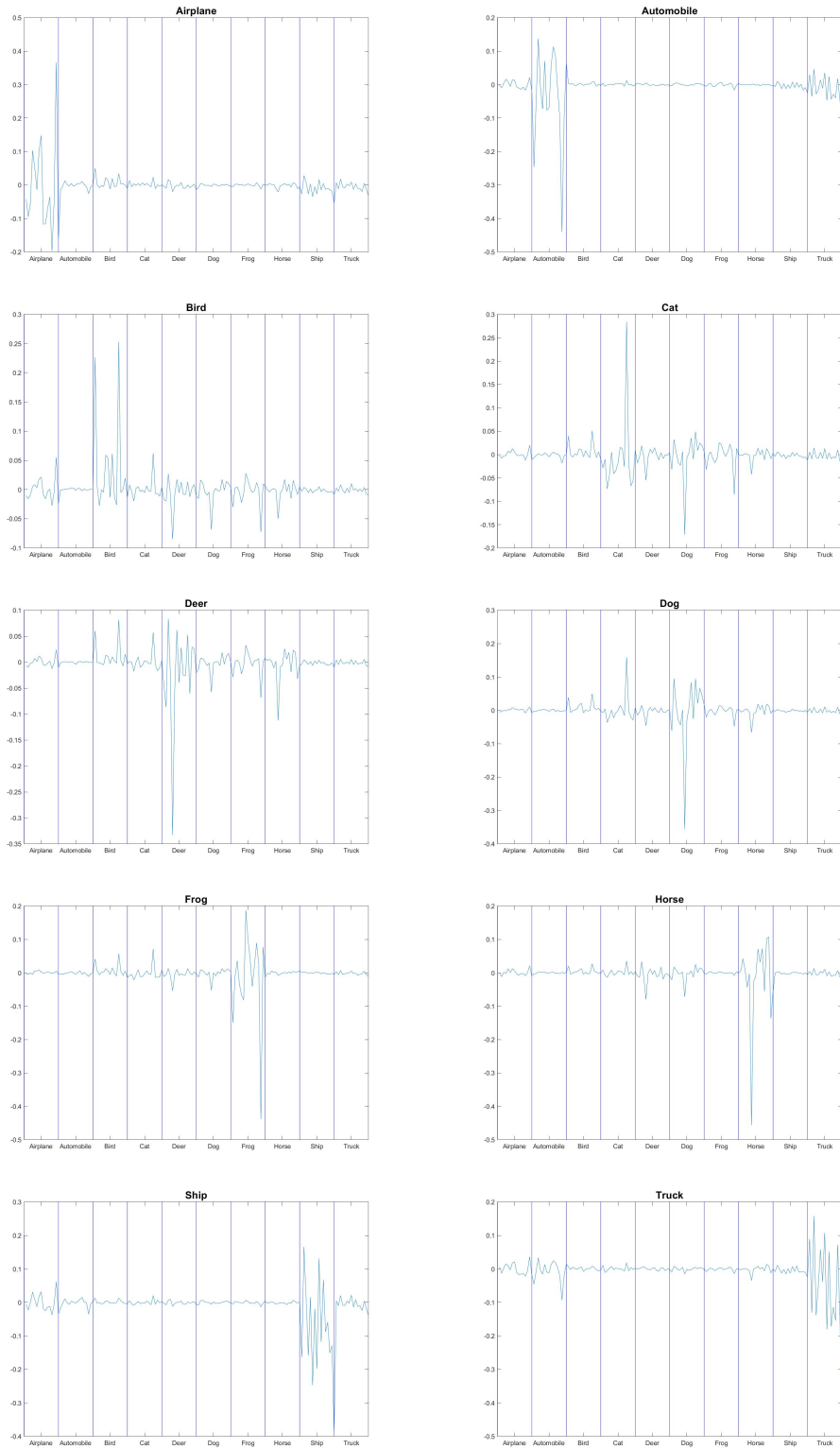


Figure 6: Visualization of Capsule features in the CIFAR-10 dataset.

## ACKNOWLEDGMENTS

The author would like to thank Dr. Jiang Li for his help and assistance on the introduction of capsule networks, as well as Dr. Eric Weisel, as this paper is a result from his class MSIM 830 - 'Simulation Formalisms.'

## REFERENCES

- Chen, C., A. Seff, A. Kornhauser, and J. Xiao. 2015. "Deepdriving: Learning affordance for direct perception in autonomous driving". In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pp. 2722–2730. IEEE.
- Goodfellow, I. J., J. Shlens, and C. Szegedy. 2014. "Explaining and harnessing adversarial examples". *arXiv preprint arXiv:1412.6572*.
- Guo, Xifeng 2017. "CapsNet-Keras: A Keras Implementation of CapsNet". <https://github.com/XifengGuo/CapsNet-Keras>.
- He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep residual learning for image recognition". In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
- Hinton, G., L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath et al. 2012. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". *IEEE Signal Processing Magazine* vol. 29 (6), pp. 82–97.
- Hinton, G., S. Sabour, and N. Frosst. 2018. "Matrix capsules with EM routing".
- Huang, Fu Jie and LeCun, Yann 2009. "The Small NORB Dataset, V1. 0, 2005".
- Huval, B., T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, and A. Y. Ng. 2015. "An Empirical Evaluation of Deep Learning on Highway Driving". *CoRR* vol. abs/1504.01716.
- Krizhevsky, A., V. Nair, and G. Hinton. 2014. "The CIFAR-10 dataset". *online: http://www.cs.toronto.edu/kriz/cifar.html*.
- Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. "Imagenet classification with deep convolutional neural networks". In *Advances in neural information processing systems*, pp. 1097–1105.
- Kurakin, A., I. Goodfellow, and S. Bengio. 2016. "Adversarial examples in the physical world". *arXiv preprint arXiv:1607.02533*.
- LeCun, Y., Y. Bengio, and G. Hinton. 2015. "Deep learning". *nature* vol. 521 (7553), pp. 436.
- LeCun, Y., C. Cortes, and C. Burges. 2010. "MNIST handwritten digit database". *AT&T Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist> vol. 2.
- Najafabadi, M. M., F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic. 2015. "Deep learning applications and challenges in big data analytics". *Journal of Big Data* vol. 2 (1), pp. 1.
- Redmon, J., and A. Farhadi. 2016. "YOLO9000: Better, Faster, Stronger.". *arXiv: 1612.08242*.
- Ren, S., K. He, R. Girshick, and J. Sun. 2015. "Faster r-cnn: Towards real-time object detection with region proposal networks". In *Advances in neural information processing systems*, pp. 91–99.
- Sabour, S., N. Frosst, and G. E. Hinton. 2017. "Dynamic routing between capsules". In *Advances in Neural Information Processing Systems*, pp. 3857–3867.
- Simonyan, K., A. Vedaldi, and A. Zisserman. 2013. "Deep inside convolutional networks: Visualising image classification models and saliency maps". *arXiv preprint arXiv:1312.6034*.
- Sommer, R., and V. Paxson. 2010. "Outside the closed world: On using machine learning for network intrusion detection". In *Security and Privacy (SP), 2010 IEEE Symposium on*, pp. 305–316. IEEE.

- Springenberg, J. T., A. Dosovitskiy, T. Brox, and M. Riedmiller. 2014. “Striving for simplicity: The all convolutional net”. *arXiv preprint arXiv:1412.6806*.
- Xi, E., S. Bing, and Y. Jin. 2017. “Capsule Network Performance on Complex Data”. *arXiv preprint arXiv:1712.03480*.

# MSVSCC 2019

## PROCEEDINGS



## Gaming & Virtual Reality

Page 51	<b>AR-PED: Augmented Reality Enabled Pedestrian-in-the-Loop Simulation</b> Daniel Perez, Mahmud Hasan, Yuzhong Shen and Hong Yang
Page 53	<b>Overview of Automated Label Placement for Three-dimensional Objects</b> Katherine Smith and Yuzhong Shen
Page 55	<b>A Review of Modeling and Simulation Tools for Autonomous Vehicles</b> Shan Liu, Yuzhong Shen and Jim Leathrum
Page 63	<b>Augmented Reality System for Marine Corps Weapon and Equipment Maintenance and Operation</b> Jayson Kreger, Edwin Merino, Noah Legrand, Xiaoyu Zhang and Yuzhong Shen

# AR-PED: AUGMENTED REALITY ENABLED PEDESTRIAN-IN-THE-LOOP SIMULATION

Daniel Perez, Mahmud Hasan, Yuzhong Shen, Hong Yang  
Department of Modeling, Simulation & Visualization Engineering  
Old Dominion University  
Norfolk, VA

## ABSTRACT

Pedestrian simulators are in high demand for many research and industry applications, such as evaluating pedestrian safety and testing of autonomous vehicles. This paper aims to develop an augmented reality (AR) enabled pedestrian-in-the-loop simulation (AR-PED) framework that enables the simulation of pedestrians in a large virtual scenario (e.g., a city) with an actual scale. The proposed AR-PED simulation framework is empowered by the latest advances in cloud computing and AR to allow multiple users to access the simulation at the same time. A client-server architecture allows users to act as pedestrians, drive a car or simply visualize the scene simultaneously. Additionally, other vehicles can be artificially generated by a microscopic traffic simulation (MTS) package that acts as the server. A prototype of the proposed framework is implemented and demonstrated by simulating a model of a city.

**Keywords:** Augmented Reality, Pedestrian Simulator, Traffic Simulation

## 1 INTRODUCTION

Pedestrian simulators are in great demand in both research and industry fields for various reasons and applications. With the recent advances in virtual reality (VR), head mounted display (HMD) devices are a very helpful option when developing pedestrian simulators. However, existing simulators are only capable of simulating a very small part of a city such as a midblock crosswalk or an intersection. This is due to the current limitation of HMD devices regarding the simulation area. Another challenge for pedestrian simulators is the realistic implementation of the vehicles' behavior. Programming vehicles at a single intersection or street crossing is a relatively simple task, but the implementation of traffic circulation in a large area is significantly more challenging. To overcome this limitations, we propose a user-controlled pedestrian simulation framework, offering a virtual environment in which user-controlled pedestrians can coexist with user-controlled vehicles for realistically simulating pedestrian-driver interactions. To summarize, the major contributions of this study are: (1) the development of a pedestrian simulator, AR-PED, with a wireless HMD device (Microsoft HoloLens) that allows the users total freedom of movement without any physical boundary restrictions; (2) the first implementation of adding user-controlled pedestrian to an MTS model that simulates traffic circulation in a large network; and (3) a framework where user-controlled vehicles and pedestrians can coexist with artificial vehicles generated by an MTS model.

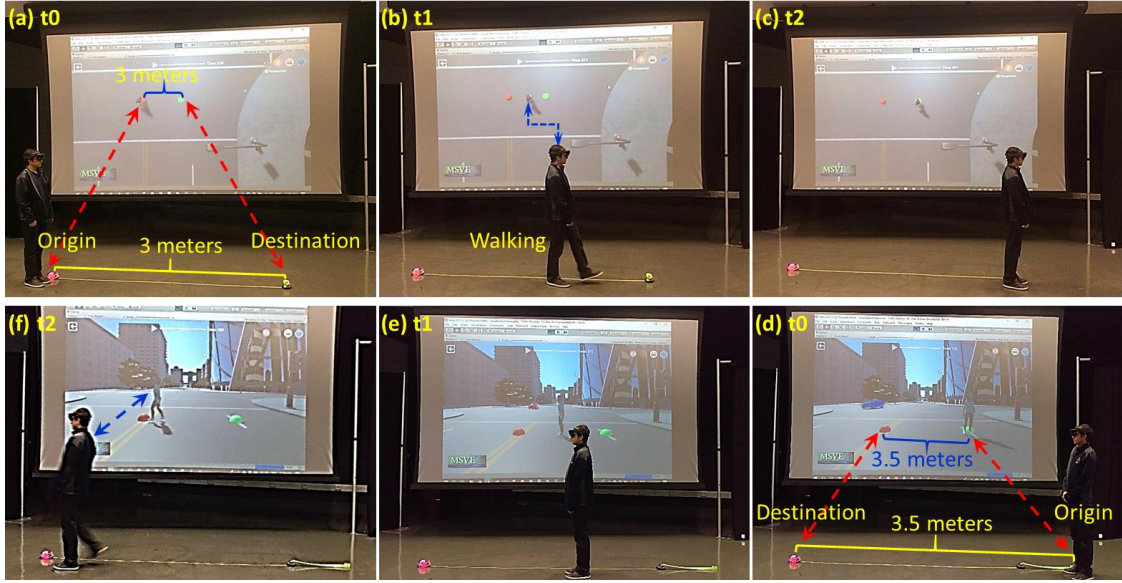
## 2 METHODOLOGY

The proposed AR-PED framework 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 server hosts the back-end MTS engine, which can be any commercial, open-source or custom-made MTS platform. The MTS engine generates the artificial vehicle data, traffic signal information, etc. Besides, the server has the function of synchronizing all the user-controlled data from pedestrian and vehicles outside the MTS engine. 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. Multiple clients can connect to the server simultaneously and perform different actions. 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. Once the clients receive the pedestrian data, they generate realistic 3D models of the pedestrians in the form of human avatars.

### 3 RESULTS

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 simultaneously displayed on a projector screen in real time. Additionally, we used 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 Figure 1. Through these test scenarios, we can determine that the measurements of our virtual scene correspond with the measurements in the real world. Figure 1 (a)-(c) show a person using the pedestrian client framework and the corresponding visualization from top view, and Figure 1 (d)-(f) show the profile view during the walk process.



**Figure 1. A User Wearing the HoloLens Headset Walks Straight and the Corresponding Virtual Pedestrian is Visualized in a Projector Screen. [Video demonstration of the framework: [click here](#)].**

We performed several experiments to assess the performance of our framework, and recorded an average latency of 0.15 seconds, which is significantly small for the purposes of our framework. Additionally, we measure the time difference between a user crossing a 14-meter crosswalk in the virtual and the physical world, and determined that, on average, pedestrians took 1.15 more seconds to cross the road in the virtual world. This can be due to the fact that they are wearing a headset that delays their walking time. Finally, we evaluate the accuracy of the virtual locations with respect to the actual location of the user wearing the HoloLens and, in average, we record an absolute error of 0.12 meters, which is expected due to the mapping limitations of the HoloLens headset and the different behavior of each user.

### 4 CONCLUSIONS

In this study, we presented the AR-PED pedestrian simulator that enables user-controlled pedestrians to coexist with user-controlled vehicles and artificial vehicles generated by an MTS server. The framework has been demonstrated in a sample city, but it can be easily deployed to any 3D model of a traffic environment if its structure is matched with the road network in the MTS server. The framework and its performance was successfully demonstrated. We observe significantly low latencies and deviations between the users' physical position and their virtual position. In the future, we plan to keep working on the framework to incorporate the latest advances in AR (HoloLens 2) and address its current limitations.

# OVERVIEW OF AUTOMATED LABEL PLACEMENT FOR THREE-DIMENSIONAL OBJECTS

Katherine Smith and Yuzhong Shen  
Department of Modeling, Simulation and Visualization Engineering  
Old Dominion University  
5115 Hampton Boulevard, Norfolk, VA, USA  
k3smith@odu.edu

## ABSTRACT

This abstract provides an overview of automated label placement for three-dimensional objects. A summary of the problem is presented along with several examples of previous work. Special consideration is given to latency which is a major consideration in augmented and virtual reality applications.

**Keywords:** label placement, visualization, augmented reality, virtual reality.

## 1 INTRODUCTION

When three-dimensional objects are viewed using augmented reality (AR) or virtual reality (VR), it is important to provide labels which give information and context for the object. These labels, and their respective leader lines, should not overlap with important parts of the object or each other as this would obstruct the user's view. The placement of these labels needs to update as the view changes to insure that these requirements continue to be met. In systems where the view is updated manually, including Google Maps (Google 2019) and 3D design software, the labels can be updated after the user is finished manipulating the view. In AR and VR, the view is updated continuously as the user moves and looks around. Therefore, the labels need to update continuously and smoothly. In VR and AR environments, latency is recommended to be kept below 15 milliseconds which is a challenge (Elbamby et al. 2018).

While there are many different types of labeling, this abstract will focus on external labeling with leader lines connecting each label to the relevant part of the view. The remainder of this abstract will be organized as follows. Section 2 will provide an overview of the external labeling problem presented as a constrained optimization problem. Section 3 will discuss previous algorithms that have been implemented to solve this problem along with a performance comparison to the latency requirement in VR and AR, where possible. Section 4 will conclude the abstract.

## 2 THE EXTERNAL LABELING PROBLEM

The labeling problem is a constrained optimization problem that has been shown to be NP-complete (Marks and Shieber 1991). The constraints on this problem vary slightly in the literature, but the following are an example set of constraints from (Cmolik and Bittner 2010).

Leader lines are the lines connecting the label to an anchor point on the object. Leader lines should not cross and should not be too close together. A set of principal directions should be defined and leader lines should always align with one of these principal directions. Leader lines should be as short as possible to keep labels close to the part to which they are assigned. Anchor points are the points where the leader line



attaches to the object associated with its label. Important points in the region to which the label corresponds should be used as anchor points. Anchor points should not be too close together and should not move too much from one frame to the next. Finally, there are constraints on the labels themselves to insure readability. Labels should not be too close together and must not overlap. The movement of labels from frame to frame should be small.

In order to solve this problem, it must be posed mathematically and then solved algorithmically. The next section will provide example solutions from the literature.

### 3 REVIEW OF PREVIOUS ALGORITHMS

A variety of approaches have been implemented to automate the placement of labels around three-dimensional objects. One common approach is to use a greedy algorithm for label placement (Cmolik and Bittner 2010, Stein and Decoret 2008). These algorithms provide good results with respect to the constraints from Section 2 when the view is held constant. When the view moves quickly, labels can take some time to settle after the conclusion of movement. Additionally, the latency reported by the authors of these studies is approximately 50 milliseconds and 30 milliseconds, respectively, which exceed the requirement for AR and VR applications. These latencies were achieved on standalone PCs with much of the computation performed on the GPU. Therefore, the results may not be repeatable on mobile devices. The final drawback of greedy algorithms is that they find local minima for optimization problems rather than seeking the global minimum.

Other approaches include algorithms based on ideas from physics and agent-based modeling. There has been work using dynamic potential fields for label placement however this work does not consider that labels should move smoothly from frame to frame or that leader lines should not cross (Hartmann, Ali, and Strothotte 2004). The same group used an agent-based approach for automated labeling (Götzelmann, Hartmann, and Strothotte 2006). They reported promising performance with a latency value of 20 milliseconds. However, it is difficult to assess this approach as the paper does not include any images of the resulting labeled objects. It is interesting to see that this newer technique has a promising result for latency. Hopefully, the application of other newer techniques will lead equally promising results.

### 4 CONCLUSIONS AND FUTURE WORK

While significant work has been done to solve the problem of automated labeling for three-dimensional objects, there are still significant challenges, especially with respect to implementation in AR and VR systems.

### REFERENCES

- Cmolik, Ladislav, and Jiri Bittner. 2010. "Layout-aware optimization for interactive labeling of 3D models." *Computers & Graphics* 34 (4):378-387.
- Elbamby, Mohammed S., Cristina Perfecto, Mehdi Bennis, and Klaus Doppler. 2018. "Toward Low-Latency and Ultra-Reliable Virtual Reality." *IEEE Network* 32 (2):78-84.
- Google. 2019. "Google Maps." Accessed 03-18-2019. <https://maps.google.com/>.
- Götzelmann, Timo, Knut Hartmann, and Thomas Strothotte. 2006. "Agent-Based Annotation of Interactive 3D Visualizations." *International Symposium on Smart Graphics*:24-35.
- Hartmann, Knut, Kamran Ali, and Thomas Strothotte. 2004. "Floating Labels: Applying Dynamic Potential Fields for Label Layout." *International Symposium on Smart Graphics*:101-113.
- Marks, Joe, and Stuart Shieber. 1991. The Computational Complexity of Cartographic Label Placement. In *Harvard Computer Science Group Technical Report*: Harvard College.
- Stein, Thierry, and Xavier Decoret. 2008. "Dynamic Label Placement for Improved Interactive Exploration." *International Symposium on Non-Photorealistic Animation and Rendering*:15-21.

# A REVIEW OF MODELING AND SIMULATION TOOLS FOR AUTONOMOUS VEHICLES

Shan Liu

Yuzhong Shen

Jim Leathrum

Department of Modeling, Simulation, and Visualization Engineering

Old Dominion University

4700 Elkhorn Avenue, Norfolk, VA 23529, USA

{sliu004, yshen, jleathru}@odu.edu

## ABSTRACT

This paper presents a brief review of modeling and simulation tools for autonomous vehicles. The paper first introduces the necessity, history and status quo of the autonomous driving systems, followed by the importance and research status of simulation research on the autonomous driving system. Four publicly available modeling and simulation platforms for the autonomous driving system are then presented. The paper concludes with discussions of the presented modeling and simulation tools for autonomous vehicles and some future work considered by the authors for advancing modeling and simulation capabilities for autonomous vehicles.

**Keywords:** autonomous vehicles, modeling, simulation

## 1 INTRODUCTION

An autonomous vehicle can sense its environment, control its speed and direction, and move with or without human input (Gehrig and Stein 1999). Five autonomous driving levels (L1-L5) have been defined for ground vehicles (UCS 2018). Each level describes the extent to which a vehicle takes over tasks and responsibilities from its driver, and how the vehicle and driver interact.

Due to the potential benefits of the autonomous driving, such as reduced costs, increased mobility and safety, reduced crime, and increased utilization efficiency (Technical and Aspects 2016), more and more companies and research institutions are actively conducting research and development in this field. The history of autonomous driving can be traced back to 1977, with the development of the first truly automated vehicle by Tsukuba Mechanical Engineering Laboratory in Japan (Weber 2018). Since then, several projects such as those funded by Defense Advanced Research Projects Agency (DARPA) (Schmidhuber 2009) and EUREKA (Schmidhuber 2009), have driven the rapid development of autonomous driving. On the other hand, with the development by commercial car manufacturers such as Tesla, Audi, BMW, and Mercedes, the second level of autonomous driving, partial automation, has been reached, wherein this level, in certain situations, the car can take over steering, braking, and acceleration functions. From 2015, some states in the US, such as Virginia and Florida, started to allow the testing of automated vehicles on public roads (Ramsey 2015). In December 2018, Waymo, a self-driving technology development company, commercialized a fully autonomous taxi service in the US (Laris 2018).

Although partially automated technology has been around for the last few years, the full fruition of potential benefits from the autonomous driving may be limited by foreseeable challenges, such as concerns about the safety of driverless in practice. There were several high-profile car fatalities caused by Tesla and Uber cars in the last few years (Green 2018, Vlasic and Boudette 2016, Greenemeier 2018). There are still many

challenges, and issues that need to be addressed before the higher level of autonomous driving (e.g., Level 4 and Level 5) can be achieved.

## **2 BACKGROUND**

With the rapid development of automobile industry and the rising of artificial intelligence, various autonomous driving technologies have been developed by automakers, researchers, and technology companies (UCS 2018), which can be divided into 3 categories: recognition to identify components of the surrounding environment, prediction to predict the future states of the environment, and planning to plan the future driving actions (Sallab et al. 2017). These driving technologies usually are integrated into autonomous vehicles, or the intelligent driver assistance systems. The software component of an autonomous driving system is integrated with the hardware component of the vehicle, such as sensors and actuators, and through them to the surrounding environment (Basarke, Berger, and Rumpe 2007). Regardless of the specific technologies utilized, autonomous vehicles must be fully and thoroughly tested for safe and intelligent operations. However, testing autonomous systems in the real physical world introduces unpredictable consequences and potential dangers, such as testing the autonomous driving system when a child runs onto the road ahead of the vehicle. Therefore, considering the risk of autonomous driving, it is necessary to run simulations before a new autonomous driving system is applied to a vehicle that runs on a road.

Simulation has long been used for training autonomous driving systems (Dean A 1988), but only recently has simulation been useful for building, training, and testing autonomous systems (Shah et al. 2017b, Dosovitskiy et al. 2017). Traditionally, simulation has been used to play back sensor data that cars collected in the real world and verify that the vehicle's self-driving software handles tricky situations appropriately. More advanced simulators, like Waymo's Carcraft and Microsoft's AirSim, can build entire road networks virtually, in which an autonomous vehicle is surrounded by many other vehicles, bicyclists, and pedestrians. Engineers can use these virtual worlds to test and re-test variations of specific scenarios and to train and test autonomous driving systems. After autonomous vehicles are thoroughly tested in virtual environments, they can be tested in real environments augmented with virtual objects (e.g., a pedestrian), and finally in completely real environments.

## **3 MODELING AND SIMULATION TOOLS**

This section discusses four publicly available, open-source tools for autonomous driving systems, including Microsoft AirSim, OpenPilot, Nvidia Drive, and Baidu Apollo.

### **3.1 AirSim**

AirSim is an open-source, cross-platform simulator designed for training autonomous systems built on Unreal Engine / Unity, from Microsoft AI & Research (Shah et al. 2017a). It supports both unmanned aerial vehicles (e.g., drones) and ground vehicles. Microsoft developed AirSim as a platform to employ various deep learning, computer vision, and reinforcement learning algorithms for autonomous vehicles. Figure 1 shows a snapshot from AirSim. AirSim provides a photo-realistic environment, vehicle dynamics, and multiple sensors for researchers to train autonomous vehicles that utilize artificial intelligence theories and methods to promote safe operation of these autonomous vehicles in the real world.

AirSim is composed of the core components and external parts. The core components include environment model, vehicle model, physics engine, sensor models, rendering interface, public API layer and an interface layer for vehicle firmware as depicted in Figure 2 (Shah et al. 2017b). It supports hardware-in-loop with popular flight controllers such as PX4 for physically and visually realistic simulations. AirSim also exposes APIs to retrieve data and control vehicles in a platform-independent way.

During simulation, AirSim provides the sensor data from the simulated world to the vehicle controller. The vehicle controller yields the actuator signals as the input of vehicle model in the simulator. The vehicle model is used to compute the forces and torques generated by simulated actuators. The forces generated from drag, friction, and gravity are inputs of the physics engine to compute the next kinematic state of the vehicle. The kinematic states with the environment model provide the ground truth for the simulated sensor models. Users can set the desire state inputs for the vehicle controller by remote control or companion computer in the autonomous system. The companion computer processes some higher-level computation, such as path planning, mission planning, etc. and interacts with the simulator via a set of APIs.

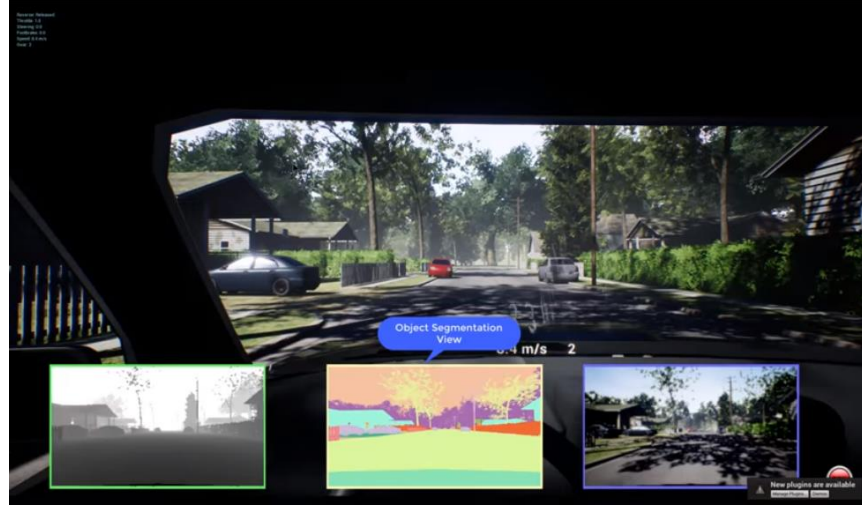


Figure 1: A snapshot from AirSim shows a vehicle driving on an urban road. This figure shows depth, object segmentation and front camera streams generated in real time (Shah et al. 2017a).

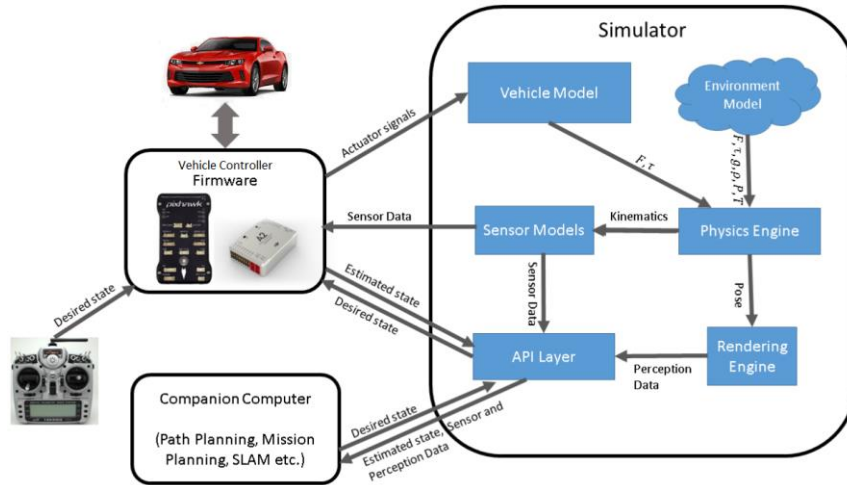


Figure 2: The architecture of the system that depicts the core components and their interactions (Shah et al. 2017b).

### 3.2 OpenPilot

OpenPilot is an open source driving agent that is used as a driver assist system. Currently, it performs the functions of Adaptive Cruise Control (ACC) and Lane Keeping Assist System (LKAS) for selected Honda, Toyota, Acura, Lexus, Chevrolet, Hyundai, Kia vehicles (Hotz 2016).

To utilize OpenPilot to assist a vehicle like any late model Toyota vehicle, three gadgets are necessary: 1) a modified OnePlus smartphone, called the EON, 2) a dongle already for sale called the Panda, that allows external electronics to interface with a vehicle's computer system, and 3) the Giraffe that connects to a vehicle's controlled area network (CAN) bus (Hawkins 2018). Figure 3 shows a picture of a driving vehicle without human control with OpnPilot system and its three gadgets, EON, Panda, and Giraffe, where the EON is connected to the vehicle's backup camera to provide a view from behind the vehicle.



Figure 3: A picture shows a vehicle drives on an urban road without human control that is loaded OpenPilot and its three gadgets, EON, Panda, and Giraffe (Hotz 2016).

OpenPilot system works with the driver to control his/her vehicle. A camera embedded in the EON monitors the driver's head movements as shown in Figure 4. If the driver looks down or away from the road for 2 seconds, a visual warning will appear on the EON's screen. After 4 seconds, the driver will hear an audible alert. And after 6 seconds, the system will disengage and begin to slow down (Hawkins 2018). OpenPilot can control the gas, brake, and steering on certain cars, reaching up to 6 minutes without user's actions (besides paying attention).

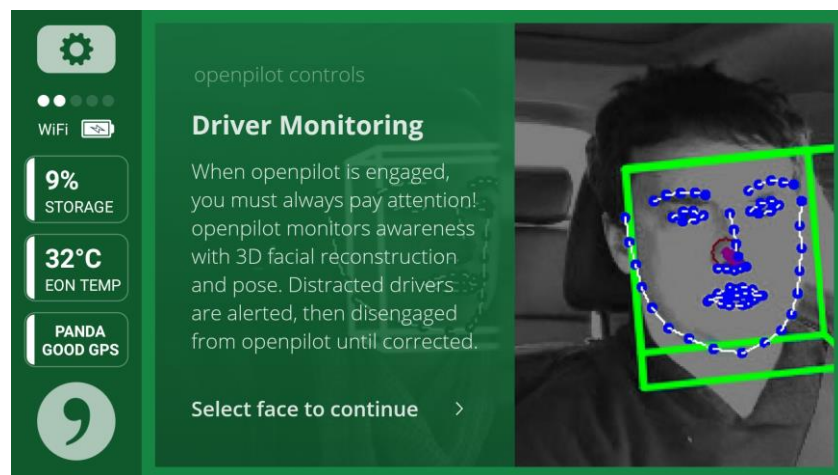


Figure 4: A snapshot from EON shows the interface of OpenPilot (Hawkins 2018).

OpenPilot provides a set of tools (openpilots-tools) that allows the user or developer to replay driving data, test, and develop OpenPilot from personal computers. Openpilots-tools runs on Linux (Ubuntu 16.04) and MacOS (10.14.2) and uses Python (2.7) as the development language. Openpilots-tools provides a very



primitive capability to run basic simulations and generate reports of OpenPilot's behavior along a driving direction (longitudinal control). The simulation capability does not offer graphical user interface or virtual environment, but only high-level abstractions of overly simplified scenarios.

### **3.3 NVIDIA Drive**

NVIDIA provides a series of driver assist products including both hardware and software systems. NVIDIA DRIVE AutoPilot is a level 2+ automated driving solution that is world's first commercially available driving system, set for production in 2020 and uniquely provides both world-class autonomous driving perception and a cockpit rich in artificial intelligence capabilities. DRIVE AutoPilot is part of the open, flexible NVIDIA DRIVE platform, which is being used by hundreds of companies worldwide to build autonomous vehicle solutions that increase road safety while reducing driver fatigue and stress on long drives or in stop-and-go traffic (Shapiro 2019), shown in Figure 5.

NVIDIA DRIVE Software is an open software suite that's powered by the compute capabilities of the DRIVE AGX platform (Sundaram 2019). It consists of the DRIVE OS operating system, DriveWorks software framework, a rich SDK to develop software applications for autonomous driving, as well as DRIVE AV and DRIVE IX software applications for autonomous driving. The photorealistic data streams are generated to create a wide range of testing environments and scenarios as depicted in Figure 6, which provides a safe, scalable solution for testing and validating autonomous driving systems. And, advanced driver assistance systems (ADAS) technology is used to provide a powerful foundation for building applications that computationally leverage intensive algorithm for object detection, map localization, and path planning.



Figure 5: A picture shows the interface of NVIDIA DRIVE AutoPilot (Shapiro 2019).

### **3.4 Apollo**

Apollo is an open-source, high performance, flexible architecture developed by Baidu, which accelerates the development, testing, and deployment of autonomous vehicles (Li and Miao 2017). Apollo is composed of accurate perception, simulation, HD map and localization, end-to-end autonomous solution, planning, intelligent control, open data platform, hardware development platform, map engine, internet of vehicles (IOV) operating system (Dosovitskiy et al.) and security product.

In the perception component, various sensors, such as LiDAR, cameras, and radar, collect environment data surrounding the vehicle. By using sensor fusion technology, perception algorithms can determine in real time the type, location, velocity, and orientation of objects on the road. Simulation component provides the ability to virtually drive millions of kilometers daily using an array of real-world traffic and autonomous

driving data. Through the simulation environment, users can gain access to a large number of autonomous driving scenes to quickly test, validate and optimize models with comprehensive coverage in a way that is safe and efficient. Planning component includes prediction, behavior, and motion logic, which can adapt to real time traffic conditions. Currently, the planning component operated on a fixed route in both night/day conditions. Figure 8 is a snapshot from Apollo shows a scene where pedestrians pass ahead of vehicles (Baidu 2018).



Figure 6: A picture shows one scenario in NVIDIA DRIVE Software (Sundaram 2019).

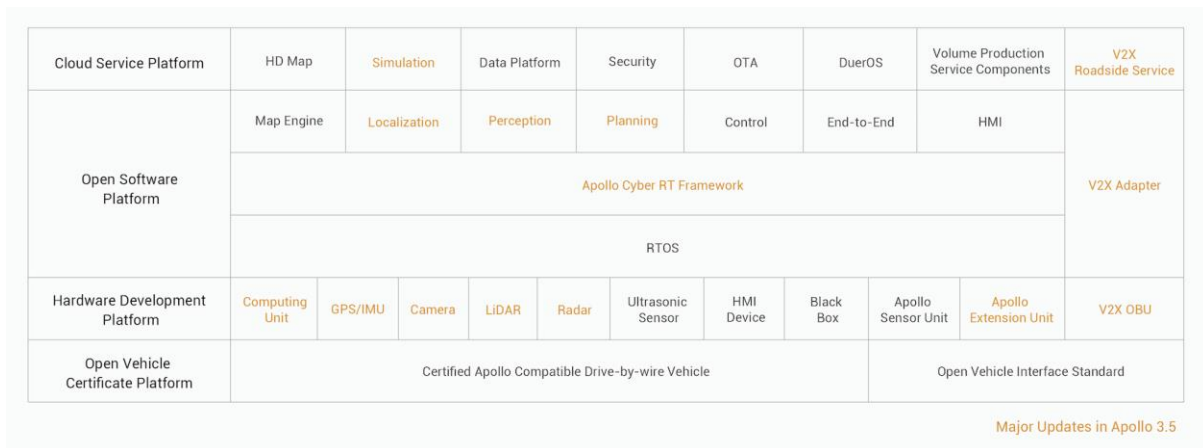


Figure 7: The architecture of Apollo 3.5 (Li and Miao 2017).



Figure 8: A snapshot from Apollo shows a scene where pedestrians pass ahead of vehicles (Baidu 2018).

## 4 DISCUSSION AND CONCLUSION

The four publicly available modeling and simulation tools for autonomous vehicles discussed in this paper are all open-source. AirSim is based on the Unreal Engine and produces a highly realistic simulation environment (e.g., photorealistic rendering). It currently supports Unreal Engine and Unity, but Unity is still in experimental stage. Besides, its visual sensor simulation is ideal presentation of the surrounding environment, using cameras to collect various images.

Apollo has a well-defined architecture that is easy to understand, with loosely coupled subsystems. However, Apollo lacks specific implementation of the components in its system and still needs substantial development to populate various algorithms and techniques to train and test vehicles in virtual environments. And, its simulation system is not based on full virtual environments, but simplified ones. Also, it should be built on Linux system, which is not suitable for Windows users.

Nvidia DRIVE has been applied by more than 300 companies to develop autonomous driving. However, it is not open to all developers although it is claimed as open source.

OpenPilot works on a real vehicle with a clearly defined software architecture. OpenPilot does not need expensive hardware, so it is a good option to combine it with other autonomous driving platforms. However, it is an open problem to integrate OpenPilot with other autonomous driving simulators. OpenPilot has been utilized by many car manufacturers. The authors believe open source autonomous driving systems should be promoted as they benefit from openness and the vast volumes of vehicles that utilize these systems. However, the current modeling and simulation capabilities provided by OpenPilot is very primitive, limiting its adoption by more manufacturers and customers. Development of an open source modeling and simulation tool with high fidelity for OpenPilot is one task considered by the authors, such as integration of OpenPilot and AirSim.

## ACKNOWLEDGMENTS

The authors would like to thank anonymous reviewers for their invaluable suggestions and helpful comments on the paper.

## REFERENCES

- Baidu. 2018. "Open Platform." <http://apollo.auto/developer.html>.
- Basarke, Christian, Christian Berger, and Bernhard Rumpe. 2007. "Software & Systems Engineering Process and Tools for the Development of Autonomous Driving Intelligence." *Journal of Aerospace Computing, Information, and Communication* 4 (12):1158-1174. doi: 10.2514/1.33453.
- Dean A, Pomerleau. 1988. "ALVINN: An autonomous land vehicle in a neural network." *Neural Information Processing Systems*:305-313.
- Dosovitskiy, Alexey, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. "CARLA: An Open Urban Driving Simulator." 1st Conference on Robot Learning, Mountain View, United States.
- Gehrig, S. K., and F. J. Stein. 1999. "Dead reckoning and cartography using stereo vision for an autonomous car." Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients, Oct. 17-21, 1999.
- Green, Jason. 2018. "Tesla: Autopilot was on during deadly Mountain View crash." Last Modified Mar. 30, 2018. <https://www.mercurynews.com/2018/03/30/tesla-autopilot-was-on-during-deadly-mountain-view-crash/>.
- Greenemeier, Larry. 2018. "Uber Self-Driving Car Fatality Reveals the Technology's Blind Spots." Last Modified Mar. 21, 2018. <https://www.scientificamerican.com/article/uber-self-driving-car-fatality-reveals-the-technologys-blind-spots1/>.



- Hawkins, Andrew J. 2018. "GEORGE HOTZ IS ON A HACKER CRUSADE AGAINST THE 'SCAM' OF SELF-DRIVING CARS." Last Modified Jul. 13, 2018. <https://www.theverge.com/2018/7/13/17561484/george-hotz-comma-ai-self-driving-car-scam-diy-kit>.
- Hotz, George. 2016. "open source driving agent." Last Modified Feb. 25, 2019. <https://github.com/commaai/openpilot>.
- Laris, Michael. 2018. "Waymo launches nation's first commercial self-driving taxi service in Arizona." accessed Dec. 5. [https://www.washingtonpost.com/local/trafficandcommuting/waymo-launches-nations-first-commercial-self-driving-taxi-service-in-arizona/2018/12/04/8a8cd58a-f7ba-11e8-8c9a-860ce2a8148f\\_story.html?noredirect=on&utm\\_term=.02a0a50460c3](https://www.washingtonpost.com/local/trafficandcommuting/waymo-launches-nations-first-commercial-self-driving-taxi-service-in-arizona/2018/12/04/8a8cd58a-f7ba-11e8-8c9a-860ce2a8148f_story.html?noredirect=on&utm_term=.02a0a50460c3).
- Li, Dong, and Calvin Miao. 2017. "An open autonomous driving platform." Last Modified Jan. 7, 2019. <https://github.com/ApolloAuto/apollo>.
- Ramsey, John. 2015. "Self-driving cars to be tested on Virginia highways." accessed Jun. 1, 2015. [https://www.richmond.com/news/self-driving-cars-to-be-tested-on-virginia-highways/article\\_b1168b67-3b2b-5274-8914-8a3304f2e417.html](https://www.richmond.com/news/self-driving-cars-to-be-tested-on-virginia-highways/article_b1168b67-3b2b-5274-8914-8a3304f2e417.html).
- Sallab, AhmadEL, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. 2017. "Deep Reinforcement Learning framework for Autonomous Driving." *Electronic Imaging* 2017 (19):70-76. doi: 10.2352/issn.2470-1173.2017.19.Avm-023.
- Schmidhuber, Jürgen. 2009. "Prof. Schmidhuber's highlights of robot car history." Last Modified Jul. 15, 2011. <http://people.idsia.ch/~juergen/robotcars.html>.
- Shah, Shital, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. 2017a. "Airsim open source platform at github." Last Modified Jan. 11, 2019. <https://github.com/Microsoft/AirSim>.
- Shah, Shital, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. 2017b. "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles." the 11th International Conference on Field and Service Robotics, Jul. 18, 2017.
- Shapiro, Danny. 2019. "Introducing NVIDIA DRIVE AutoPilot: AI-Powered System Delivers Safer Vehicles Today, Autonomous Driving Tomorrow." accessed Jan. 7. <https://blogs.nvidia.com/blog/2019/01/07/nvidia-drive-autopilot/>.
- Sundaram, Shri. 2019. "DRIVE Software 8.0 Enables Surround Perception, AR for Safe Automated Driving." accessed Jan. 7. <https://blogs.nvidia.com/blog/2019/01/07/drive-software-8-0-automated-driving/>.
- Technical, Legal, and Social Aspects. 2016. "Autonomous Driving." In, ed Mark Maurer, J. Christian Gerdes, Barbara Lenz, and Hermann Winner: Springer Nature.
- UCS, Vehicles. 2018. "Self-Driving Cars Explained." accessed Feb. 21, 2018. <https://www.ucsusa.org/clean-vehicles/how-self-driving-cars-work>.
- Vlasic, Bill, and Neal E Boudette. 2016. "Self-Driving Tesla Was Involved in Fatal Crash." Last Modified Jun. 30, 2016. <https://www.nytimes.com/2016/07/01/business/self-driving-tesla-fatal-crash-investigation.html>.
- Weber, Marc. 2018. "Where to? A History of Autonomous Vehicles." Last Modified Jul. 26, 2018. <https://www.computerhistory.org/atcm/where-to-a-history-of-autonomous-vehicles/>.

# AUGMENTED REALITY SYSTEM FOR MARINE CORPS WEAPON AND EQUIPMENT MAINTENANCE AND OPERATION

Jayson Kreger<sup>1</sup>, Edwin Merino<sup>1</sup>, Noah Legrand<sup>1</sup>, Xiaoyu Zhang<sup>1,\*</sup>, Yuzhong Shen<sup>2,\*</sup>

<sup>1</sup> Department of Mechanical Engineering, Old Dominion University, Norfolk, VA, USA

<sup>2</sup> Department of Modeling, Simulation, & Visualization Engineering, Old Dominion University, Norfolk, VA, USA

\* Corresponding Authors: Xiaoyu Zhang ([x1zhang@odu.edu](mailto:x1zhang@odu.edu)); Yuzhong Shen ([yshen@odu.edu](mailto:yshen@odu.edu))

## ABSTRACT

The present project focused on developing an augmented reality (AR) system for the U.S. Marine Corps weapon (i.e. MK 18<sup>1</sup>) and equipment maintenance and operation. Additive manufacturing technology (aka. 3D printing) was implemented to fabricate physical objects (i.e. MK 18) to interact with the created coded AR system. This project utilized low-cost, market-leading AR software, including Unity and Vuforia, to develop an interactive AR application for maintenance and operation of MK 18 rifle. The AR application contains interactive presentation and visualization of MK 18 rifle exterior and interior parts and 3D animations for maintenance and troubleshooting procedures. We also conducted preliminary research on software framework, interface, and file structures for easy integration of different objects under the same AR application.

**Keywords:** Augmented Reality, 3D Printing, MK 18 Rifle

## 1 INTRODUCTION

Augmented Reality (AR)<sup>2</sup> augments the real world by overlaying information and/or virtual objects that are spatially aligned to the objects in the real world. The information and/or virtual objects augmented to the real world help the user understand and interact with the real world in a more intuitive and efficient way. With recent advances in computing hardware and software technology, AR is making gradual but solid progress in a wide range of fields, such as manufacturing, construction, maintenance, training, and healthcare. For example, how to assemble, disassemble, or repair complex equipment, is a common but difficult task in many occupations. Technicians often dedicate a large amount of time to studying manuals, since it is often impossible to memorize all detailed procedures. Augmented reality, on the other hand, can



Figure 1: Augmented Reality System



Figure 2: 3D Simulation System

present instructions directly superimposed in the field of view of the technician and reduce the amount of time needed for the maintenance or repair, leading to improved work efficiency.

This project focused on developing an AR system for the U.S. Marine Corps weapon (i.e. MK 18 <sup>1</sup>) and equipment maintenance and operation. A 1:1 scale MK-18 was fabricated using Stereolithography (SLA) 3D printer. This project utilized low-cost, market-leading AR software, Unity and Vuforia, to develop an interactive AR application for maintenance and operation of MK 18 rifle, the standard service weapon for the U.S. Marine Corps. The AR application contains interactive presentation and visualization of MK 18 rifle exterior and interior parts and 3D animations for maintenance and troubleshooting procedures. The project is elaborated below.

## **2 DEVELOPMENT OF AR SYSTEM FOR MK 18**

The system includes an AR (Figure 1) and a 3D Simulation (Figure 2) aspect with an interactive menu to switch between. Both AR and 3D Simulation includes maintenance/operation procedures and a exploded view function for the MK-18. These procedures include a step by step instructions with animations of specific parts that is indicated by flashing red for the user to easily follow. The 3D Simulation system also includes interactive features such as pan, zoom, and rotate. The AR application uses Model Targets and Vumarks for the AR system to recognize when to display the Mk-18 model in the real world. Model Targets use the outline of the MK-18 model to recognize and overlay the virtual model onto the gun in the real world. Vumarks uses a man-made image embedded with a QR code to recognize when to display the model over the gun.

## **3 3D PRINTING OF MK 18**

In order to begin the 3-D printing process a CAD software, Fusion 360, was used in order to disassemble the MK-18 into individual and partially individual components. A middle man software known as Preform was used to import the Fusion 360 file to the Formlabs 2 3-D stereolithography (SLA) printer. The print would take place and would have to be cleaned in an Isopropyl alcohol wash of at least 90% and removed of its build supports. Figure 3 shows the print-out of MK-18 for AR.



a

## **4 ACKNOWLEDGEMENT**

This project is supported by Office of Naval Research (N00014-18-S-F003).

## **REFERENCES**

1. Close Quarters Battle Receiver, available at [https://en.wikipedia.org/wiki/Close\\_Quarters\\_Battle\\_Receiver](https://en.wikipedia.org/wiki/Close_Quarters_Battle_Receiver)
2. R. Palmarinia, J.A. Erkoyuncua, R. Roya, H. Torabmostaedi, Robotics and Computer-Integrated Manufacturing, 2018, 49:215-228
3. F.P.W. Melchels, J. Feijena, D.W. Grijpm, Biomaterials, 2012, 31: 6121-6130

# MSVSCC 2019

## PROCEEDINGS



## Medical Simulation

Page 66	<b>Deep Learning Method for ECG-based Virtual Pathology Stethoscope Tracking in Patient Heart Auscultation</b> Haben Yhdego, Nahom Kidane, Rick McKenzie and Michel Audette
Page 72	<b>Multi-resolution Simplex Mesh: Towards a Deformable Digital Cranium Atlas</b> Shrabani Ghosh and Michel Audette
Page 74	<b>A Visualization Tool for Protein Anomaly Detection in Chimera</b> Mitchell Sheep, Eduardo Santos, Brandon Baker and Lin Chen

# Deep Learning Method for ECG-based Virtual Pathology Stethoscope Tracking in Patient Heart Auscultation

Haben Yhdego<sup>1</sup> Nahom Kidane<sup>1</sup> Frederic D McKenzie<sup>2</sup> and Michel Audette<sup>2</sup>

**Abstract**—This paper propose a method to improve the Cardiac Auscultation skills of medical students by virtual pathology simulation that utilizes modified stethoscopes in patients. It is ECG (electrocardiogram) based virtual pathology stethoscope (VPS) tracking method for placing virtual symptoms in correct auscultation landmarks of patients. It presents ECG tracking method using transfer learning, making knowledge learned from sufficiently transferable existing trained model. As tremendous progress has been made in object recognition with deep convolutional neural networks (DCNNs), we can pre-train on the currently available large-scale annotated datasets. Those pre-trained convolutional layers are reused to transfer knowledge to other classification tasks. First, the ECG datasets are changed to images using time-frequency analysis resulting in scalograms by inputting these datasets into a continuous wavelet transform filter bank. Subsequently, data augmentation is performed on our datasets to increase the accuracy by complementing the limited labeled ECG sensor data and perform transfer learning from the existing pre-trained model. Our experimental results demonstrate that transfer learning leads to a better performance than the existing methods in the case of scarce labeled training data and the additional data augmentation helps to boost the capability of classification results.

**Keywords:** ECG Analysis, Auscultation, Virtual Pathology, Deep Learning, Medical Training.

## I. INTRODUCTION

Cardiovascular Disease (CVD) is the leading cause of death in United States [20][1] (accounting 25% of the total death each year). According to a projection done by the American Heart Association [20], by 2030: 40% of the U.S. populations expected to suffer from CVD. The treatment expenditure increases- 272.5 in 2008 to 818.1 billion dollars by 2030. In addition, there is also a short supply of primary care and specialty physicians—an estimated shortage of 16,000 cardiologists by 2050. Thus, there is an urgent need to prepare more health-care workers—including primary care doctors, specialists, and nurses—for the expected demand.

Cardiac Auscultation (CA) as shown in figure 1 is the primary method for detecting structural cardiovascular pathologies. It involves listening to the heart's sound with a stethoscope. Cardiac diseases such as congestive heart failure, systemic arterial hypertension, and coronary artery disease can be diagnosed with proper CA [14]. Several surveys [14][2][15][5][3], have highlighted a rapid decline in

CA skill among medical trainees and residents. A study [14] conducted in 2006, showed that CA skills among trainees did not improve after the third year of medical studies. Wayne et.al (2009) surveyed hundred medical students [3], and found that only one third of the trainees achieved a minimum passing standard (MSP) in CA on their first try. There are many reason for the decline in CA skills: limited patient access due to health-care management restrictions, a patient population with high short hospital stays, shortage of both training time experienced auscultation educators, and reliance on comparably expensive competing technologies.



Fig. 1. Auscultation

This describes a method that improves the CA skills of medical students by virtual pathology simulation that utilizes modified stethoscopes in tandem with standardized patients. A cost-efficient and novel orientation-invariant tracking method for placing virtual symptoms in correct auscultation areas based on electrocardiogram (ECG) signals. It would allow trainees to perform realistic CA and hear abnormal heart or lung sounds in otherwise healthy patient actors. Points of Cardiac auscultation are the safest, most convenient, and least expensive method for detecting cardiovascular pathologies. Accurate placement of the stethoscope chest piece reveal heart murmurs associated with valvular abnormalities. CA areas generally correlated with the cardiac valves: Aortic area (right intercostal space), Pulmonic area (Left intercostal space), Tricuspid area (Left sternal boarder), Mitral area (Apex of the heart) as shown in figure 2.

Medical simulators have been used to teach CA [17], which provides a safe and controlled environment for repetitive practice and immediate training opportunities (no need to wait for suitable real patient). But this method is complex, noisy and costly. Our research describes ECG based distinctive stethoscope head placement using transfer learning and continuous wavelet analysis of ECG signal data to classify the four auscultation areas by using the pre-trained CNNs Alexnet architecture. Wavelet-based time-frequency representations of sensor data are used to create vector-valued scalograms, which map to RGB-formatted images

<sup>1</sup>H. Yhdego and N. Kidane are PHD student in the Department of Modeling, Simulation and Visualization Engineering, Old Dominion University, Norfolk, USA. {hyhde001, nkida001}@odu.edu

<sup>2</sup>Dr. R. McKenzie and Dr. M. Audette are faculty in the Department of Modeling, Simulation and Visualization Engineering, Old Dominion University, Norfolk, USA. {rdmcken, maudette}@odu.edu



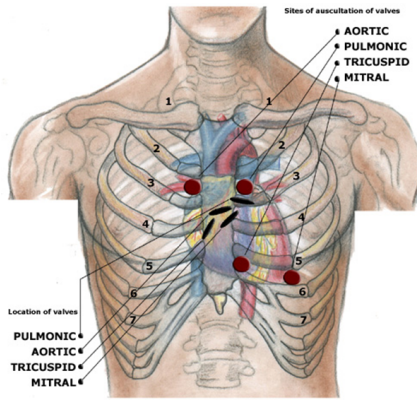


Fig. 2. Points of auscultation [14]

suitable for processing by AlexNet. Data augmentation on those images are performed to increase the size of the datasets that are used to fine-tune the Alexnet-based deep CNNs.

## II. BACKGROUND

For two decades, convolutional neural networks have been applied to problems in image and video processing. Since LeCun’s seminal paper in 1998 [23], CNN architectures achieved excellent performance in applications like handwriting detection and face recognition. LeNet [23], a system for handwritten digit recognition, is among the first to achieve near human accuracy. LeNet notably includes an interactive visualization system that displays featurized versions of input images as well as predictions. The visualizations allow direct and compelling demonstration of important properties of the system like invariance to translations and deformations of the input. While LeNet’s visualizations provide evidence for the CNN system’s merits and add to its merits, since an inspectable model is better than a black box, they do not serve as design aids to practitioners.

Increases in processing units have allowed researchers to train deeper and more complex convolutional neural network. The imagenet classification using DCNN by Krizhevsky [13] and his colleagues achieved record results in the large scale visual recognition challenge with an eight-layer network trained on GPUs. In most image recognition and classification tasks imagenet [9] is used to train large models. Decaf, a system written in Python and developed to make convolutional network processing more approachable [4]. Many feature visualisation tools today like Deepvis [10] uses decaf extensively for interactive image featurization and for offline statistical computations.

Recent work by Jarret and his colleagues [15] explores the growing design space of convolutional networks to evaluate architectural variations of different hand-designed networks on several data sets. Other developed by Yamins group [21] use Bayesian methods to automatically search the parameter space of convolutional networks. The interactive visualization tool Deepvis [5] is designed to help users develop some

heuristics to guide search and evaluation in this increasingly complex space.

With DCNN, visualization has been used in computer vision as a tool in feature evaluation. Visualization of image and video features helps to understand the problems of a model [20]. Many algorithms of visualization use feature inversion algorithms, whereby an image is featurized and then recovered to a transformed but more intelligent model, to give us access to abstract feature representations.

Feature inversion has been applied to convolutional neural networks to obtain several interesting results. One method of Khavjou et al. [18] performs inverse optimization on a network trained by unsupervised learning to construct the optimal inputs for specific neurons. In particular, they find single deep neurons trained to respond to faces and bodies. Later, Espiritu-Santo-Rincon et al. [6] use a type of feature inversion called deconvolution to render re-weighted versions of input that highlight the areas, patterns, and textures of an image deemed most important by a particular part of the network. These re-weighted images are both accessible and informative, and the authors used insights from these images to refine their network design to achieve state-of-the-art performance.

Deep learning, or the use of deep (i.e., many-layered) convolutional neural networks (DCNN) for machine recognition and classification, is advancing the limits of performance in domains as varied as computer vision, speech, and text [25][11]. Improvements in both hardware and software performance and the availability of large datasets have enabled the development of larger networks that have achieved record results[13]. The promise of deep learning is to automate feature engineering, a task that otherwise requires application of both domain expertise and machine learning expertise.

Deep convolutional neural networks have recently shown novel research results on image classification problems [12][13][8]. However, their inner workings remain a mystery to machine learning experts, particularly when compared to better studied and less complex algorithms such as SVM and Logistic Regression. As a result, constructing and debugging effective convolutional neural networks is time-consuming and error-prone, as it often involves a substantial amount of trial and error.

ECG consists of the analysis of cardiac wavefronts, that are formed by the activation taking place inside the heart, propagate and cause cell-to-cell activation. Cells generate electrical current flow within the body and results in a potential difference on the surface of the skin. ECG amplifies and records these signals as they travel through the body. The first clinically important ECG measurement system was developed in early 1900s by Willem Einthoven. Normal ECG is comprised of the following wave components: P -the atrial depolarization, QRS -ventricular depolarization, and T wave -ventricular repolarization. ECG signals recorded at four auscultation areas with the stethoscope diaphragm positioned a horizontal orientation were used for virtual tracking [18]. However, different individuals have distinctive hand movements; left-handed while others Place stethoscope

head at various angle when performing CA. This research describes a method that analyzing ECG signals at four auscultation areas— aortic, pulmonic, mitral and tricuspid with a wide range of stethoscope orientations for tracking stethoscope.

### III. TRANSFER LEARNING

In practice, we don't usually train an entire DCNN from scratch with random initialization. This is because it is relatively rare to have a dataset of sufficient size for the depth of network required. Instead, it is recommended to pre-train a DCNN on a very large dataset and then use the trained DCNN weights either as an initialization or a fixed feature extractor for the task of interest which we call transfer learning.

The low-level and high-level features learned by a CNN on a source domain can often be transferred to augment learning in a different but related target domain. For target problems with a large dataset, we can transfer the low-level features (such as edges and corners), and learn new high-level features specific to the target problem. For target problems with limited amounts of data, learning new high-level features is difficult. However, if the source and target domain are sufficiently similar, the feature representation learned by the CNN on the source task can also be used for the target problem. Deep features extracted from CNNs trained on large annotated datasets of images have been used as generic features very effectively for a wide range of computer vision tasks [4][19].

However, in transfer learning as we use a pre-trained network, we are slightly constrained in terms of the model network. For instance, we can't arbitrarily take out convolutional layers from the pre-trained network. Due to parameter sharing, we can easily run a pre-trained network on images of different spatial size. This is clearly evident in the case of convolutional and pool layers because their forward activation function is independent of the input volume spatial size. In the case of Fully Connected (FC) layers, this assertion still holds true because FC layers can be converted to a Convolutional Layer [22].

#### A. Fine-tuning the Transferred Model DCNNs

Fine-tuning the transferred model allows us to bring the power of DCNN models to new domains where insufficient data and time, as well as cost constraints might otherwise prevent their use. Transfer learning strategies depend on various factors, but the two most important ones are the size of the new dataset, and its similarity to the original dataset. DCNN features are more generic in early layers for different dataset and more dataset-specific in later layers [22].

The transformed ECG RGB images dataset is smaller in size and very different in content compared to the original imagenet dataset. As the data is small and very different, it is good to train a classifier from activations somewhere earlier in the network [22]. We modify the weights of the pre-trained DCNN by continuing the backpropagation [7]. It is possible to adjust all the layers of the DCNN, or its possible to keep

some of the earlier layers fixed (due to overfitting concerns) and only customize some higher-level portion of the network. This is motivated by the observation that the earlier features of a DCNN contain more generic features (e.g. edge detectors or color blob detectors) that should be useful to many tasks, but later layers of the DCNN becomes progressively more specific to the details of the classes contained in the ECG Scalogram RGB images dataset [22].

The inputs for the Alexnet architecture should be three-channel images, so we change the ECG signal to Alexnet correspondent RGB image input. To change those ECG signals to RGB image inputs, we first create time-frequency representations of the ECG signals. These representations are called scalograms that are saved as RGB images to pretrain our convolutional neural network model [16]. A scalogram is the absolute value of the continuous wavelet transform coefficients of a signal. Subsequently, data augmentation is performed on the ECG dataset to make the signal invariant to the position of the sensors and also to increase the size of our dataset complementing the limited labeled ECG sensor data and perform transfer learning from the existing pre-trained model. The data augmentation method is performed; it consists of rotation by twenty degree in the X and Y axis. We thereby obtain RGB scalogram images as shown in figure 3.

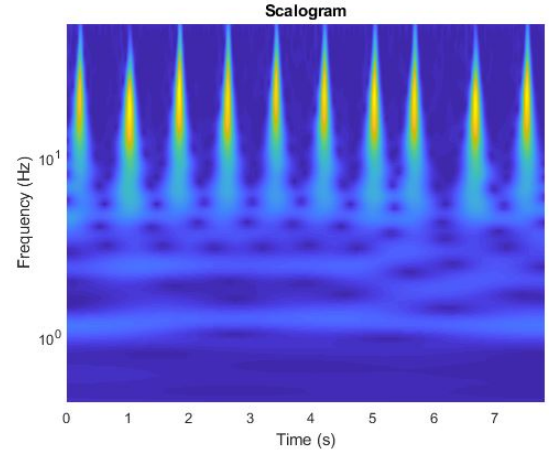


Fig. 3. Sample Scalogram

Hence, if we want to use the imagenet for our ECG datasets like in our case. We first train the Alexnet [9] architecture shown in figure 4 on the imagenet dataset and then remodel the Alexnet to be used for this ECG RGB image dataset. The ECG RGB image dataset is insufficient to train an architecture as complex as Alexnet and so we use weights from the imagenet trained on this network. We harmonize all layers. Moreover, We use a smaller learning rate for DCNN weights that are being redesigned under the assumption that the pre-trained DCNN weights are relatively good. We don't wish to distort them too quickly or too much. The last three layers of the Alexnet network are changed and train our new dataset on those layers as the last layers are helps us to get specific features whereas the earlier layers

give us more generic features. And also We replace the classification output layer of 1000 classes with a new output layer according to the number of classes supported by our ECG image dataset, which is four [22]. By training the ECG RGB images on the deeper layers of the updated Alexnet we can track the training progress of our model over the training iteration number as shown in figure 5.

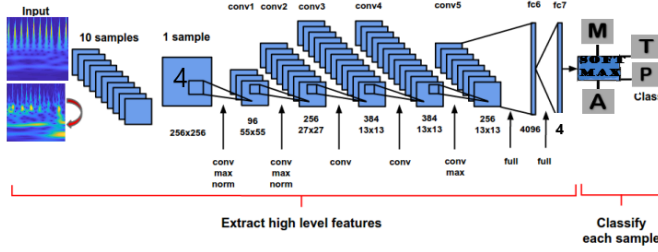


Fig. 4. Alexnet [9]

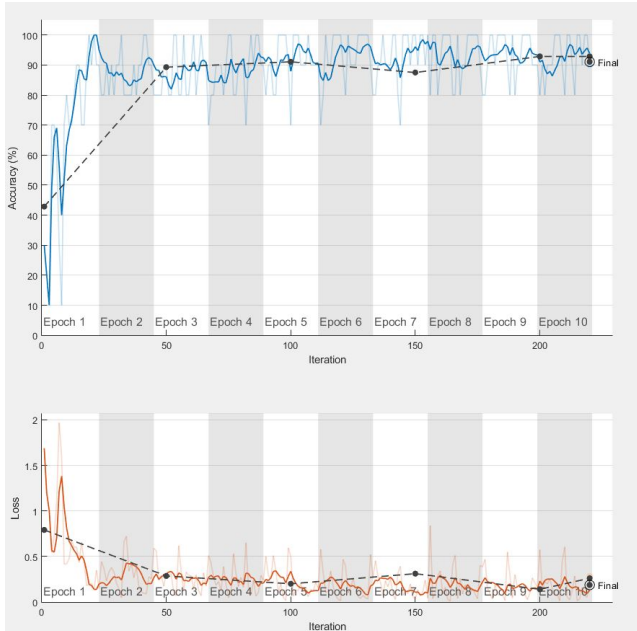


Fig. 5. Training Progress

The previous methods of Bayes Network and Decision Tree [18] are compared with the TL (transfer learning) method for tracking the four auscultation areas with the stethoscope diaphragm. We achieved the results shown in table I, using the formulas where TP is True positive, TN is True Negative, FN is false negative and FP False positive. The result of our transfer learning has better accuracy than those of old machine learning algorithms (Bayes Network, SVM and Decision trees). As a result, this findings would significantly aid in extending the capabilities of standard patients and allow medical student trainees to perform realistic auscultation and hear abnormal heart or lung sounds.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP}$$

TABLE I  
COMPARING THE RESULTS OF DIFFERENT METHODS.

Metrics	BayesNet	DecisionTree	TL(Our Method)
Accuracy	89.6	91.3	94.6

#### IV. CONCLUSIONS

In this paper, we use deep learning methods to boost the performance of an orientation invariant ECG-based virtual pathology stethoscope tracking method for placing virtual symptoms in correct auscultation landmarks of patience. The deep learning method is used for classifying four auscultation regions, aortic, mitral, pulmonic, and tricuspid. DCNN is challenging to apply as the training data is very scarce in many sectors. But we can apply transfer learning to such applications for which it is difficult to obtain enough datasets for training like our ECG dataset and fine-tune the transfer learning-based neural network based on relevant training data. For image data, the first layers of the network typically learn low-level features such as edges and corners, and further layers learn high-level features such as textures and objects [24]. As a result, we can exploit those features learned at the shallow layer for different dataset as they are generic features for any dataset, while training the deeper layers with our limited ECG dataset to get specific features related to our dataset. The experimental results demonstrate that transfer learning leads to a better performance than the existing methods in the case of scarce labeled training data and the additional data augmentation helps to boost the capability of classification results. Furthermore, data augmentation by applying arbitrary rotations to the existing ECG data helps as a way of simulating different sensor placements.

In the future work, classification results can be further improved by performing an online classification that makes predictions based on a sequence of QRS waves from the incoming ECG signal. As real-time system would have to be robust enough to identify the auscultation areas with shorter ECG signals ( $< 3sec$ ), thus the previous classifiers must be tested and optimized for real-time application. Sequentially classifying the signals must also be studied; such system will perform real-time classification and make predication's on the incoming beat or section of an ECG signal. For example, if the current signal is from a particular region, the next signal has higher likelihood of being from the same region.

#### REFERENCES

- [1] J. Xu A. M. Minino S. L. Murphy and K. D. Kochanek. "Deaths: final data for 2008," in: *Natl Vital Stat Rep* (2011).
- [2] M. A. Chizner. "Cardiac Auscultation: Rediscovering the Lost Art". In: *Current Problems in Cardiology* (vol. 33, pp. 326- 408, 2008).
- [3] E. R. Cohen D. B. Wayne J. Butter and W. C. McGaghie. "Setting Defensible Standards for Cardiac Auscultation Skills in Medical Students," in: *Academic Medicine*, (vol. 84, pp. S94-S96, Oct 2009.).



- [4] Jia Y. Vinyals O. Hoffman J. Zhang N. Tzeng E. Darrell T. Donahue J. "A deep convolutional activation feature for generic visual recognition." In: *DeCAF: arXiv:1310.1531* (2013).
- [5] D. LeFrancois E. S. Spatz and R. J. Ostfeld. "Developing cardiac auscultation skills among physician trainees," in: *International Journal of Cardiology* (vol. 152, pp. 391-392, 2011).
- [6] A. Espiritu-Santo-Rincon and C. Carbajal-Fernandez. "ECG feature extraction via waveform segmentation". In: *in Electrical Engineering Computing Science and Automatic Control (CCE), 2010 7th International Conference* (on, 2010, pp. 250-255.).
- [7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. "Deep Learning." In: *MIT Press. p. 196. ISBN 9780262035613* (2016).
- [8] A.G. Howard. "Some improvements on deep convolutional neural network based image classification". In: *arXiv 1312.5402* (2013).
- [9] R. Socher L.-J. Li K. Li J. Deng W. Dong and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database." In: *IEEE Computer Vision and Pattern Recognition (CVPR)* (2009).
- [10] Anh Nguyen Thomas Fuchs Jason Yosinski Jeff Clune and Hod Lipson. "Understanding Neural Networks Through Deep Visualization". In: *International Conference on Machine Learning (ICML)* (2015).
- [11] Rajat Monga Kai Chen Matthieu Devin Quoc V. Le Mark Z. Mao Marc'Aurelio Ranzato Andrew Senior Paul Tucker Ke Yang Andrew Y. Ng Jeffrey Dean Greg S. Corrado. "Large Scale Distributed Deep Networks". In: *Google Inc., Mountain View, CA* (2012).
- [12] Shaoqing Ren Jian Sun Kaiming He Xiangyu Zhang. "Deep Residual Learning for Image Recognition." In: *CVPR* (2015).
- [13] Sutskever I. Hinton G. Krizhevsky A. "Imagenet classification with deep convolutional neural networks." In: *NIPS* (2012).
- [14] j. M. Vukanovic-Criley. "Competency in cardiac examination skills in medical students, trainees, physicians, and faculty: A multicenter study (vol 166, pg 610, 2006)," in: *Archives of Internal Medicine*, (vol. 166, pp. 1294-1294, Jun 26 2006).
- [15] S. Mangione. "Cardiac auscultatory skills of physicians-in-training: A comparison of three English-speaking countries," in: *American Journal of Medicine* (vol. 110, pp. 210-216, Feb 15 2001).
- [16] Matlab. *Classify Time Series Using Deep Learning*. URL: <https://www.mathworks.com/help/wavelet/examples/signal-classification-with-wavelet-analysis-and-convolutional-neural-networks.html#dl20e7570>.
- [17] FD McKenzie et al. "Augmented standardized patients now virtually a reality." In: *In Third IEEE and ACM*

- International Symposium on Mixed and Augmented Reality* (2004).
- [18] Nahom K. Salim C. Li J. McKenzie D. Hubbard T. "Orientation invariant ECG-based stethoscope tracking for heart auscultation training on augmented standardized patients". In: *The Society for Modeling and Simulation International* (2013 ).
  - [19] Bottou L. Laptev I. Sivic J. Oquab M. "Learning and transferring mid-level image representations using convolutional neural networks." In: *CVPR* (2014).
  - [20] O. A. Khavjou J. Butler K. Dracup M. D. Ezekowitz E. A. Finkelstein Y. Hong S. C. Johnston A. Khera D. M. Lloyd-Jones S. A. Nelson G. Nichol D. Orenstein P. W. F. Wilson P. A. Heidenreich J. G. Trogon and Y. J. Woo. "Forecasting the Future of Cardiovascular Disease in the United States". In: *Circulation* (2011).
  - [21] J. Pan and W. J. Tompkins. "A Real-Time QRS Detection Algorithm". In: *Biomedical Engineering, IEEE Transactions on*, (vol. BME-32, pp. 230-236, 1985.).
  - [22] Anusua Trivedi. *Transfer Learning and Fine-tuning Deep Convolutional Neural Networks*. URL: <http://blog.revolutionanalytics.com/2016/08/deep-learning-part-2.html>.
  - [23] Y. Bengio Y. LeCun L. Bottou and P. Haffner. "Gradient Based Learning Applied to Document Recognition." In: *Proceedings of the IEEE*, 86(11):2278-2324 (1998).
  - [24] M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks." In: *CoRR abs/1311.2901* (2013).
  - [25] Matthew D. Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". In: *ECCV* (2014).

# Multi-resolution simplex mesh: Towards a Deformable Digital Cranium Atlas

Shrabani Ghosh

Modeling, Simulation & Visualization Engineering

Old Dominion University

1300 Engineering & Computational Sciences Building

Norfolk, VA 23529

Email: {sghos003, maudette}@odu.edu

## Abstract:

Digital atlases are important tools for medical and surgical operation. These atlases provide guidance to identify anatomical structures. Accurate, patient specific structural representation is especially important because a small inaccuracy can lead to a dangerous complication. Multi-material surface meshes have shared boundaries. It is important to change the mesh resolution globally in order to adapt the complexity of the multi surface model. In image registration, coarse-to-fine approach is widely used to increase robustness. This research effort extends multi resolution scheme on non-shared faces that is based on the tessellation of the dual triangle mesh into multi resolution scheme for shared surfaces decimation or refinement to a simplex surface. A synthetic data and bone-foramen model is presented from coarse-to-fine approach to show the effectiveness of mesh optimization scheme.

**Keywords:** Decimation, Cranial foramen, Super-resolution, Contour model, Marching cubes

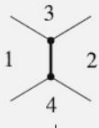
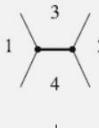
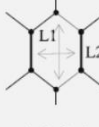

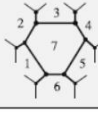
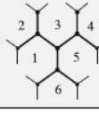
## 1 INTRODUCTION

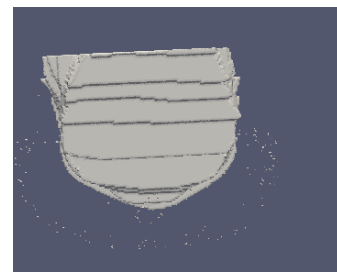
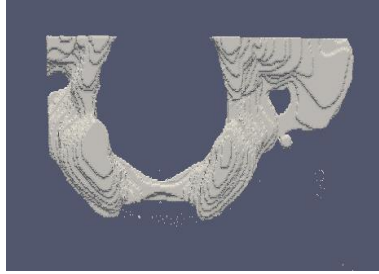
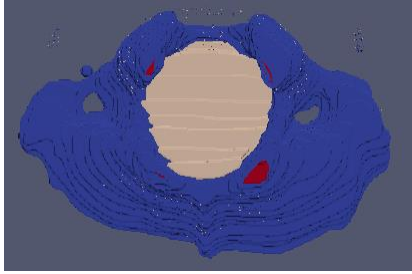
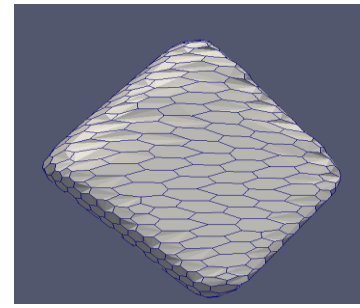
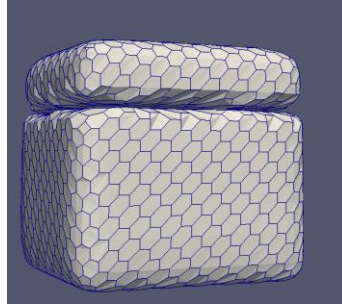
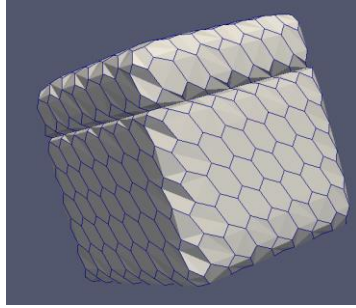
Printed and digital atlases are important tools for medical interventions. An accurate depiction is important for surgical intervention because even a small inaccuracy leads to a dangerous complication. Human skull has 34 cranial foramina that transmit cranial nerves, blood vessels and other structures- these are collectively referred to as the cranial foramina. These foramina and bones share common surfaces. In image registration, mesh resolution control is important, coarse-to-fine approach gives more accurate representation of interest. To prevent complications in skull base neurosurgery, the cranial foramen of the patient must be presented in the planning and patient specific simulation of the procedure.

## 2 METHODS

Decimation is important to control resolution to solve problem of producing tractable nonrigid registration. Without a coarse-to-fine approach, the fine mesh can become easily entangled in false minima. The geometric quality (uniformity of vertices repartition) and topological quality (uniformity edge number among faces) of simplex meshes are important to ensure a good matching of the model with the object we want to represent. We want to keep the object topology (same genus and number of holes) while optimizing the mesh topology. The decimation below is part of Delingette-type which has two operators (TO1 and TO2). It is possible to combine them to form micro-operators such as the exchange operation that changes the repartition of edges across faces without changing the total number of vertices, edges and faces. The sparse representation of cranium from coarse-to-fine level gives better understanding of bones and foramina surfaces curvatures.

Summary of mesh topology adaptation operators.  $n_i$ ,  $S_i$  and  $L_i/L_2$  are respectively the number of edges, surface and principal lengths of a face  $i$ .  $\bar{S}$  is the target face surface.

Operators/criteria		
	Exchange	
	$\rightarrow n_1 + n_2 - n_3 - n_4 > 2$ $\leftarrow n_3 + n_4 - n_1 - n_2 > 2$	
	$TO_1/TO_2$	
	$\rightarrow L_1 > L_2 \sqrt{2}$ $\rightarrow S > \bar{S} \sqrt{2}$ $\leftarrow \sum_2 S_i < \bar{S} \sqrt{2}$	
	$TO_4/TO_1$	
	$\rightarrow \sum_6 S_i > \sqrt{42} \bar{S}$ $\leftarrow \sum_7 S_i < \sqrt{42} \bar{S}$	



**Figure:** **Upper left** Summary of mesh topology, **center row** synthetic model multi-resolution scheme on shared boundary and non-shared surfaces, **last row** Occipital Condyle Bone and Foramen Magnum multi-resolution scheme on shared boundary and non-shared surfaces.

#### 4 CONCLUSION AND DISCUSSION

Within a multi-material deformable model framework, we have performed scalable methods (mesh topology adaptation, multi-resolution scheme) both in shared boundary and non-shared surfaces. Our work represents multi-resolution scheme applied on sparse representation of occipital condyle and foramen magnum multi-surface model. The deformable surface model of cranium will be able to differentiate cranial nerves, holes and blood vessels. The most exciting part is way to do the registration of our whole cranium model on clinical human dataset. We believe our model will have a good potential for clinical use although we need validation and extensions to handle.

#### 5 References

- Delingette, Hervé. *Simplex meshes: a general representation for 3D shape reconstruction*. Sophia Antipolis, France: INRIA, 1994.
- Gilles, Benjamin, and Nadia Magnenat-Thalmann. "Musculoskeletal MRI segmentation using multi-resolution simplex meshes with medial representations." *Medical image analysis* 14.3 (2010): 291-302.

# A VISUALIZATION TOOL FOR PROTEIN ANOMALY DETECTION IN CHIMERA

Mitchell Sheep

Department of Mathematics & Computer Science  
Elizabeth City State University  
Elizabeth City, NC, USA  
mlsheep226@students.ecsu.edu

Eduardo Santos

Department of Natural Science  
Elizabeth City State University  
Elizabeth City, NC, USA  
eduardos526@gmail.com

Brandon Baker

Department of Natural Science  
Elizabeth City State University  
Elizabeth City, NC, USA  
bmbaker214@students.ecsu.edu

Lin Chen

Department of Mathematics & Computer Science  
Elizabeth City State University  
Elizabeth City, NC, USA  
lchen@ecsu.edu

## ABSTRACT

Cryo-electron microscopy (cryo-EM) has become a significant technique for protein structure determination. However, in our previous study, we noted that there are plenty of suspicious defects in proteins solved from cryo-EM in the Protein Data Bank (PDB). Those defects may be due to that the current validation pipe-line in PDB lacking the features to identify those errors. We proposed an unsupervised model to help recognize those anomalous previously. The output is a score for each residue in a protein according to a histogram based analysis from a high-resolution reference X-ray dataset ( $<1.5\text{\AA}$ ). To help structural biologists identify reasons causing conformation defects in proteins, we designed a visualization tool embedded in the Chimera platform. The users can visualize the location of each suspicious residue in both three-dimensional (3D) space and one-dimensional (1D) space. By clicking residues in the sequence section of the graphic user interface (GUI), users can highlight the target residues in 3D visualization window. Scientists can easily confirm the reasons for generating those defects during the modeling process and fix it.

**Keywords:** visualization; Chimera; protein structure; anomaly; validation;

## 1. INTRODUCTION

Validation of protein structures has become a major issue in the structural biology community due to the serious errors occurring in many published structures, especially in protein structures derived from cryo-EM [1]. Since the early 1990s, plenty of validation tools [2-6] have been developed by validating experimental data, protein models and the fit between experimental data and protein models. In order to create comprehensive assessment criteria, the Worldwide Protein Data Bank (wwPDB [7]) convened a Validation Task Force (VTF) consisting of experts in the field to define policy issues, validation standards, and format specifications. The wwPDB VTF includes X-ray VTF, NMR VTF, and EM VTF for X-ray, NMR, and EM data respectively. In 2012, EM VTF released their recommendation reports based on the X-ray VTF report in 2011 [8]. wwPDB and its partners accept and curate depositions by wwPDB Deposition & Annotation (D&A) system. wwPDB generates validation reports for new depositions which contain the

results from rigorous tests of structure model quality. The D&A system updated the validation reports for X-ray structures in March 2017 with 2016 statistics. The validation reports for EM structures in PDB are available since May 2016.

wwPDB D&A system is an integral validation pipeline for X-Ray, NMR and EM models. The recommendation checks of X-Ray VTF include bonding geometry, conformation, quality of data set, and fit to experimental data [9]. Geometric criteria include covalent bond lengths, bond angles, chirality, and planarity. Target values are compared with the expected value in wwPDB compilation 2012 and calculated for corresponding Z-scores. All residues with a Z-score greater than 5 are over-fitted residues and labeled as outliers. Conformational criteria include Ramachandran [10] and Rotamer [11] analysis. The residues with unfavorable combinations are labeled as outliers. Both geometry and conformation analyses are performed by MolProbity [5-6], and the results are listed in the model quality section of validation reports. EM VTF recommended validation of both EM density map and models due to the high uncertainty of cryo-EM technique, the flaws in the cryo-EM density map can mislead model generation. However, in the current wwPDB EM validation reports, no model validation methods based on cryo-EM characteristics are used.

In our previous study [12], we observed the difference between the cryo-EM structures and more accurately determined X-ray structures in PDB. Low-risk conformations in validation reports had been noted to have high risks of having flaws [12]. It serves as a reminder that the current validation tool initially designed for X-ray proteins may not be suitable for cryo-EM proteins. An efficient validation tool for cryo-EM proteins is in urgent need. Chen proposed an unsupervised model with five geometrical features [13]. Their model uses high-resolution X-ray proteins as the reference and returns a histogram based outlier score (HBOS). However, using the raw values generated from the model to validate residues in 3D space is time-consuming. A visualization tool is needed to narrow down the validation space with the built unsupervised model.

UCSF Chimera is a highly extensible visualization platform for interactive visualization and analysis of proteins and corresponding cryo-EM density maps [14]. With the supports of Chimera APIs, researchers can develop tools for their research and embed them as plugins into Chimera [15]. Most of the current Chimera built-in tools are written in Tkinter, a graphic design Python module. The unsupervised model proposed by Chen’s team was written in Biopython, which is a widely used computational biology open source Python module [16]. In the current study, we designed a visualization tool running the unsupervised model in Chimera platform.

## 2. METHODS AND SOFTWARE DESIGN

### 2.1 Methods

The histogram based plots were generated and saved as a dataset for the visualization tool. In Chen’s study [13], five features, block length  $d_{Block}$ , side-chain length  $d_{SC}$ , backbone torsion angle Phi  $\phi$  and Psi  $\varphi$ , and first side-chain torsion angle  $\chi_1$ , were used to describe the 3D conformations of a residue.  $d_{Block}$  is the distance between the CA atom on backbone and mass center of distal block of a specific residue [12].  $d_{SC}$  is the distance between CA atom on residue backbone and mass center of its side-chain. 18 of 20 residues were used since glycine (GLY) and alanine (ALA) have no  $\chi_1$  due to their small size of side-chains. For each of the five features, a probability density function (pdf) was generated for each of the 18 residues using X-ray-1.5 reference dataset which contains 9131 X-ray PDB protein structures having the resolution better than or equal to 1.5 Å. The bin size of  $5^\circ$  for  $\phi$ ,  $\varphi$ , and  $\chi_1$  and 0.05 Å for  $d_{Block}$  and  $d_{SC}$  were used for calculating the pdf. Each pdf then was normalized with its highest peak value. A total of 90 normalized pdfs (npdf) were generated from 18 types of residues and five features for each type. The generated pdfs were saved into five csv files for five features.

$$HBOS = \sum_{i=1}^5 HBOS(i) = \sum_{i=1}^5 \log \left( \frac{1}{npdf_i(v_i)} \right) \quad (1)$$

The Histogram-Based Outlier Score (HBOS) of a residue is the summation of log values of its inverted npdf value of five features according to equation (1), in which ‘ $i$ ’ is the index of a specific feature. The npdf <sub>$i$</sub>  represents the probability of having a particular value for feature ‘ $i$ ’. If the value of npdf <sub>$i$</sub>  is less than 0.001, the corresponding feature value is rarely observed from reference X-ray dataset. The corresponding HBOS( $i$ ) value in that bin is assigned a value of 5 to avoid infinite value. Based on the analysis in Chen’s study [13], the probability of an X-ray protein residue having large HBOS value (>9) nears to zero. Compared with X-ray residues, there are plenty of EM protein residues with HBOS value greater than 9. We use X-ray residues in the dataset as the benchmark. An EM residue with HBOS greater than 9 may have errors in its conformation. To have high confidence, in this study, we chose a more restricted threshold 10 as the default cutoff value for anomaly labeling. A residue having HBOS greater than or equal to 10 were labeled as suspicious residue. In the visualization tool, thresholds are parameters entered by users with a default value 10.

## 2.2 Software Architecture

The code of the visualization tool was organized with the Model-View-Controller (MVC) model. As shown in Figure 1, the visualization tool contains a GUI, control code, model code, and histogram data generated from X-ray reference data. The protein loaded in Chimera platform can be accessed by GUI and control modules. The GUI has input and output widgets which are coded in the Python Tkinter module. The input widgets receive chain ID and threshold values from the user, these values are then passed to the control module. The control module uses the received values as parameters for the model to calculate HBOS scores for each residue in the selected protein chain. The model module loads the pre-generated histogram-based data and identifies suspicious residues with entered thresholds. The residues with HBOS higher than or equal to the threshold value are labeled as anomalies and returned to the control module. Those labeled residues are colored with yellow color and displayed in both the GUI and Chimera window by the control model.

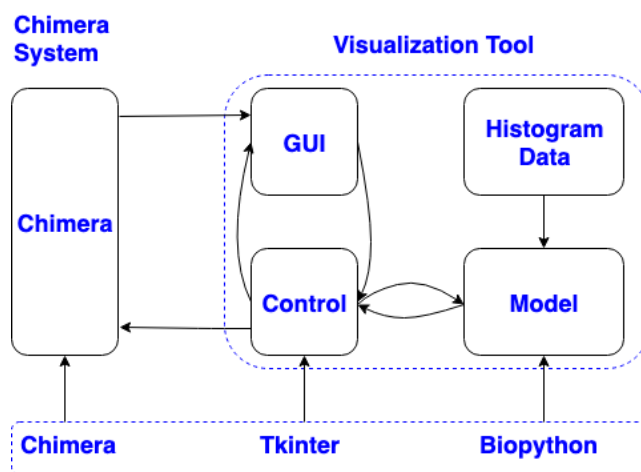


Figure 1. The architecture of the visualization tool.

The tool was coded mainly in three Python modules: Chimera, Tkinter, and Biopython, with Chimera platform software being an isolated visualization system containing the Chimera module. The Chimera module provides Application Program Interfaces (API) to access and manipulate protein data loaded in the Chimera software. The Python modules installed in Chimera platform are separated from the modules installed in the default Python distribution (i.e. not part of the python development kit). Tkinter is one of the embedded Python modules in Chimera platform to support GUIs written by Chimera engineers and

third-party research groups. Biopython is a Python module that has been used in bioinformatics and computational biology for more than fifteen years. However, Biopython uses a different data structure to save and process protein data and is not a default embedded module which called for the control module to be coded as a communication interface for Chimera and Biopython. With the control module, the residue information collected by Biopython can be passed to Chimera without converting Biopython protein data structure to Chimera protein data structure.

## 2.3 Delivery and Use

The visualization tool was designed for the Chimera platform as an open-source utility. Its functionalities need the support of Chimera as a foundation. The tool has been tested in Chimera 1.13.1, this can be downloaded from the Chimera official website (<https://www.cgl.ucsf.edu/chimera/download.html>). The model module in the tool needs the Biopython module that can be obtained through download from the Biopython website (<https://biopython.org/wiki/Download>). It should be installed manually to Chimera directory instead of using default installation path by pip or conda since they use the path of standard Python or Anaconda Python as the default install destination. For example on Mac, `python setup.py install --home=/Applications/Chimera.app/Contents/Frameworks/Python.framework/Versions/2.7/`. The source code of the visualization tool in Python has been deposited to Github at <https://github.com/lin-chen-VA/chimeraplugin>. A readme file including installation instructions has been saved with the code.

The visualization GUI is shown in Figure 2. To label anomalous residues in a protein: load the target protein from Chimera, then click “Tools -> Utilities -> ECSU Label” to start the tool. The GUI has three sections: inputs, sequence, and labels. Users can choose the chain and input the threshold value in the inputs section. In Figure 2, we labeled the suspicious residue in chain C of protein 3izx. The default threshold value 10 was used for validation purpose. After choosing input arguments, click the “Check” button to run the labeling code. The sequence section will show the residue sequence of the selected chain, in which each residue is represented with its single-letter name. The identified anomalous residues are highlighted with yellow color in the sequence. The corresponding residues are also colored with yellow with side-chain shown in the Chimera window (right side in Figure 2). In labels section: residue name, residue index, and HBOS scores for five features are displayed for each labeled residue. The significant high HBOS scores are highlighted with the red color.

The visualization tool can help users observe the conformations of labeled residues and their neighbors. To understand why a residue is labeled as an anomaly, users can observe the target residue as well as the conformation of its neighbors in sequence. By clicking the neighbor residue around the target residue in sequence in the sequence section, the Chimera window will highlight (red color) and show the side-chains of clicked residues. With the HBOS scores listed in the label section, users can have a brief conclusion of the causes of those conformation defects.

## 3. RESULTS AND DISCUSSION

The validation tool narrows down the validation space. To validate a residue manually, scientists check its conformation, neighbor residues, and cryo-EM density map around it carefully. It is an impossible task to validate every residue in a moderate protein manually. Chen’s team have exposed many residues labeled as perfect in their validation reports at PDB, but are noted as suspicious residues by biologists [12]. In those cases, the current validation pipeline at PDB does not perform well. Validation biologists cannot guarantee the generated models have no defects in the modeling process due to the significantly large validation spaces. Therefore, plenty of proteins containing suspicious residues have been deposited into PDB. The visualization tool is complementary to the PDB validation pipeline. It compares the conformation of target proteins with the data collected from reference dataset. The suspicious residues can be labeled by adjusting the threshold value. Those anomalies are a small percentage of residues. It is feasible to double validate them manually by structural biologists.



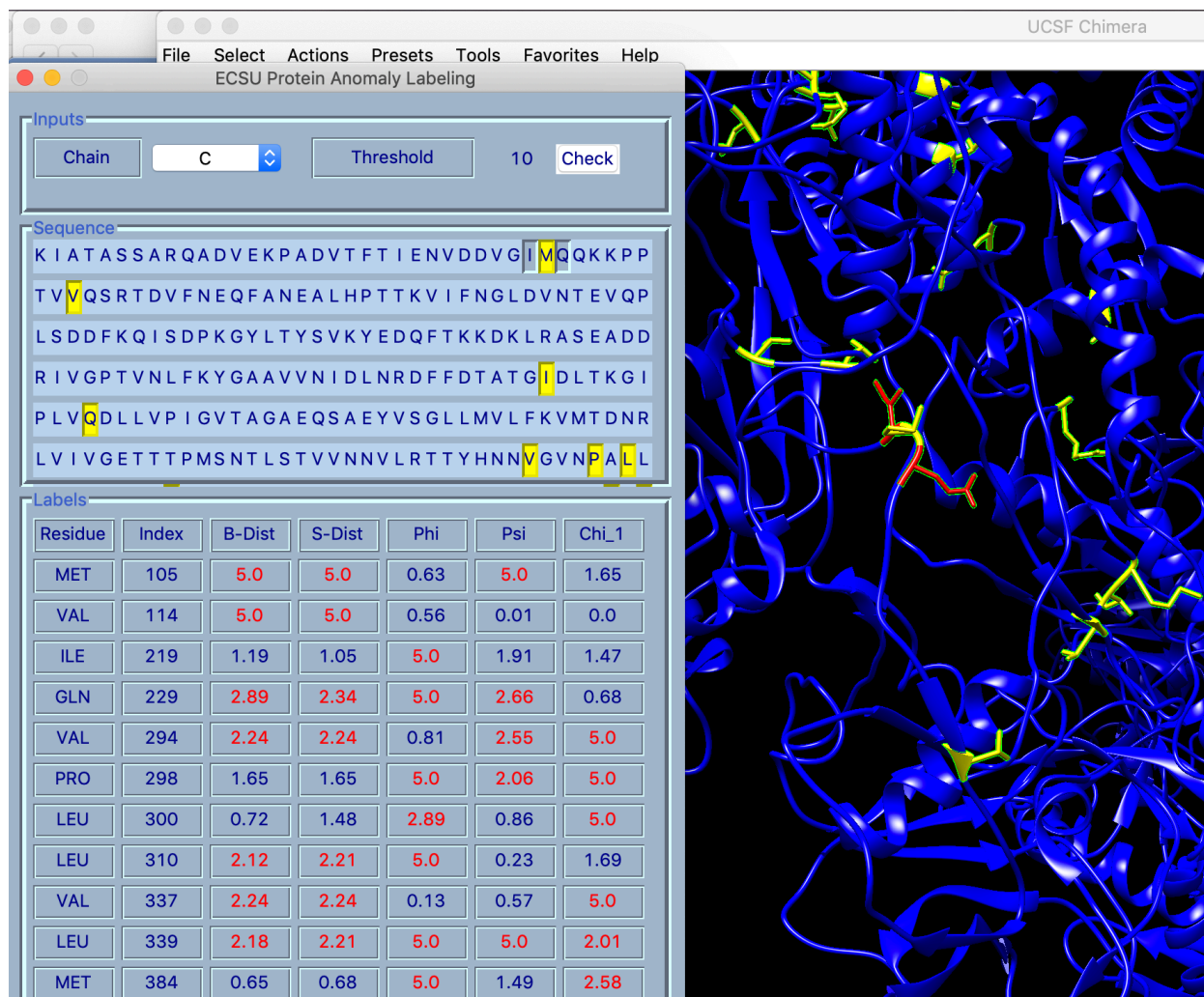


Figure 2. Anomalous residues (yellow) labeled by the visualization tool in chain C of protein 3ixx.

The visualization of a labeled residue in 3D space can help biologists make the decision quickly. The tool infuses validation and visualization. Users can read 1D sequence information, 3D space information, and HBOS scores within one window. In Figure 3, Methionine-105 (MET) in chain C of 3ixx was labeled as an anomaly. We can observe its neighbors by clicking ILE-104 and GLN-106 next to target MET-105. The conformations of two neighbor residues are shown in Chimera windows. The side-chain interactions among those residues can be observed directly. Load the corresponding cryo-EM density map, we note the side-chain of MET-105 are fitted into a small size of density map. Reading the HBOS scores listed in the label section, especially the score highlighted as red color, confirms that the side-chain of MET-105 are over-folded. Since MET-105 has high HBOS score in  $d_{Block}$ ,  $d_{SC}$ , and  $\phi$ , structural biologists need to adjust both side-chain atoms and backbone atoms to fix the conformation.

The tool uses Biopython module in Chimera platform. Biopython and Chimera have a different data structure to save protein structures. Therefore, they are not able to share data directly. The current tool uses an interface code to pass data between them. The data passing increases CUP time and memory cost. To make the visualization tool run more efficiently, we may need to rewrite validation code from Biopython version to Chimera version. Chimera is a primary tool in protein visualization. Currently, it only supports Python 2. Including the current visualization tool, Chimera built-in tools and third-party tools were written

in Python 2. However, the Python community has decided to stop the support for Python 2 in 2020. Since different research teams wrote the tools in the Chimera platform, it is not feasible to ask all those teams to rewrite and update their code to Python 3. It is predictable that Chimera community will stick with Python 2 for an extended period.

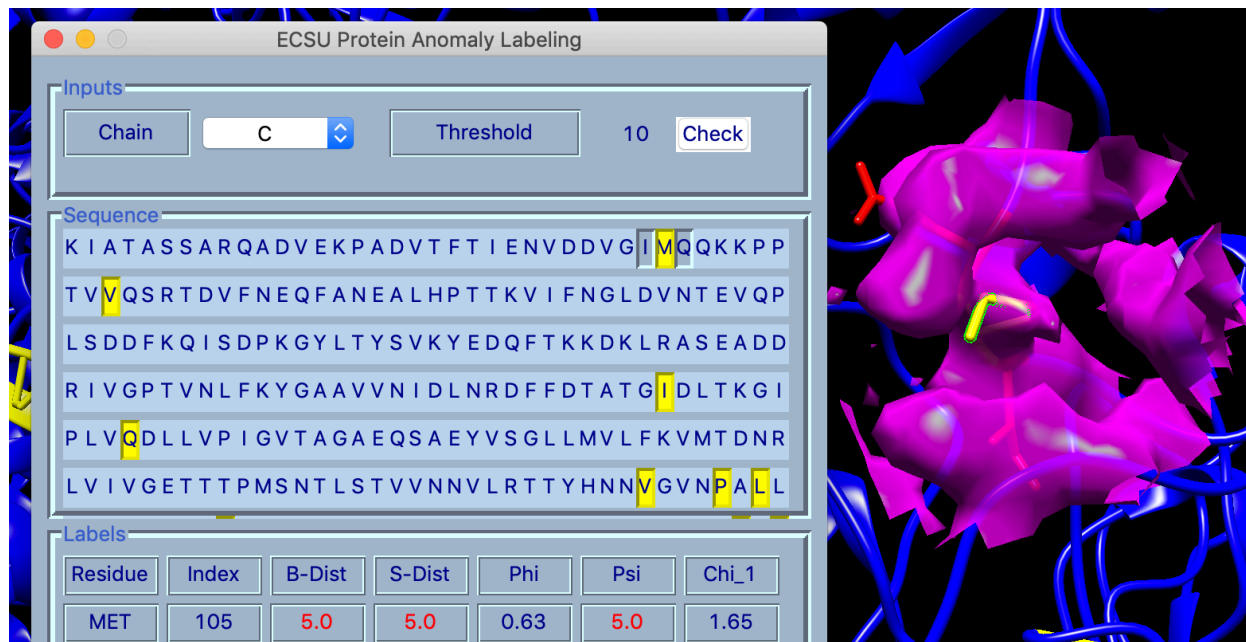


Figure 3. Methionine (MET) in chain C of 3izx and cryo-EM density map around it.

#### 4. FUTURE WORK

The current tool consumes plenty of CUP time on validation of large proteins. We may optimize the code of control module and GUI module to speed up the communication and display. The table in the label section was implemented with layout label widgets. It takes plenty of memory to hold those labels. We may use widgets built in tix module to optimize it. The display controls in the current tool were implemented with Chimera Midas commands which may have extra running overhead. Rewriting those code with Chimera Python module only may help increase running performance.

#### ACKNOWLEDGMENTS

NIH 2T34GM100831-06 supports the work in this paper.

#### REFERENCES

- [1] Kleywegt, G. J. and T. A. Jones (2002). "Homo Crystallographicus—Quo Vadis?" *Structure* 10(4): 465-472.
- [2] Jones, T. A., et al. (1991). "Improved methods for building protein models in electron density maps and the location of errors in these models." *Acta Crystallogr A* 47 ( Pt 2): 110-119.
- [3] Laskowski, R. A., et al. (1993). "PROCHECK: a program to check the stereochemical quality of protein structures." *Journal of Applied Crystallography* 26(2): 283-291.
- [4] Hoof, R. W., et al. (1996). "Errors in protein structures." *Nature* 381(6580): 272.

- [5] Lovell, S. C., et al. (2003). "Structure validation by Calpha geometry: phi, psi and Cbeta deviation." *Proteins* 50(3): 437-450.
- [6] Davis, I. W., et al. (2004). "MOLPROBITY: structure validation and all-atom contact analysis for nucleic acids and their complexes." *Nucleic Acids Res* 32(Web Server issue): W615-619.
- [7] Berman, H., et al. (2007). "The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data." *Nucleic Acids Res* 35(Database issue): D301-303.
- [8] Henderson, R., et al. (2012). "Outcome of the First Electron Microscopy Validation Task Force Meeting." *Structure*(London, England:1993) 20-330(2): 205-214.
- [9] Gore, S., et al. (2012). "Implementing an X-ray validation pipeline for the Protein Data Bank." *Acta Crystallogr D Biol Crystallogr* 68(Pt 4): 478-483.
- [10] Ramachandran, G. N., et al. (1963). "Stereochemistry of polypeptide chain configurations." *J Mol Biol* 7: 95-99.
- [11] Shapovalov, M. V. and R. L. Dunbrack, Jr. (2011). "A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions." *Structure* 19(6): 844-858.
- [12] Chen, L. and J. He. (2014). "A distance- and orientation-dependent energy function of amino acid key blocks." *Biopolymers* 101(6): 681-692.
- [13] Chen, L., He, J., Using Combined Features to Analyze Atomic Structures derived from Cryo-EM Density Maps. The 9th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, Washington D.C., August 29 – September 1, 2018.
- [14] Pettersen, E.F., Goddard, T.D., Huang, C.C., Couch, G.S., Greenblatt, D.M., Meng, E.C., Ferrin, T.E. UCSF Chimera--a visualization system for exploratory research and analysis. *J Comput Chem.*2004 Oct;25(13):1605-12.
- [15] Pintilie, G., Chiu, W. (2018) Assessment of structural features in Cryo-EM density maps using SSE and side chain Z-scores. *J Struct Biol* 204:564–571.
- [16] Cock PA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B and de Hoon MJL (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25, 1422-1423.

# MSVSCC 2019

## PROCEEDINGS



## Agent Based Modeling, Education & Training

Page 82

**Evacuation Model of Michie Stadium at West Point, New York**  
Delanie Weliver

Page 87

**Virtual Autonomous System Training (VAST) Environment**  
Christine Odenwal, Germaine Badio, Jaron Stevenson, Travis Sullivan, Peter Marchione, Elise Feldt and Zhaohui Hu

Page 99

**Utilizing Accelerometry to Determine Body Posture of Dental Hygienists During Instrumentation**  
Eric Jenkins

Page 101

**Is the Bang Worth the Buck? A Value-based and Statistical Analysis of the U.S. Army's Movement from the 5.56 to the 6.8 Round**  
Chris Banga and Gene Lesinski

# EVACUATION MODEL OF MICHIE STADIUM AT WEST POINT, NEW YORK

Delanie A. Weliver  
Department of Systems Engineering  
United States Military Academy  
West Point, New York  
delanie.weliver@westpoint.edu

## ABSTRACT

Over the past 10 years, there have been 51 mass casualty shootings in the United States, and large public gatherings seem to be the preferred targets. Large gatherings in one condensed area, such as sporting events, have high risk for mass casualty. Recent increased emphasis on force protection at the United States Military Academy aims to ensure top security for all personnel living on or visiting West Point. In this research, a discrete-event simulation model of Michie Stadium at West Point, New York was created to simulate how a full stadium would evacuate during a man-made disaster, such as a mass shooting or bombing. The average evacuation time was found to be 17 minutes. These findings have a direct impact on how emergency responders will react to man-made disasters. Measures to decrease evacuation time include adjusting key parameters such as walkway and tunnel width and increasing signage of gates.

**Keywords:** Mass Evacuation, Geographic Information System (GIS) Model, Discrete-Event Simulation

## 1 INTRODUCTION

In the United States, the most popular past time activity is watching a football game, whether it is a collegiate or professional game, you are able to find fans in stadiums come Saturday night. The U.S. Homeland Security (DHS) has identified that sport stadiums meet the criteria for being a potential terrorist threat. With the average attendance to an NFL football game being approximately 65,000, there is an immediate concern for the safety and well-being for all participants. In the past 10 years, the number of mass casualty shootings has hit a historic high with 51 occurrences. This makes up more than half (57%) of all recorded mass shootings in the United States. In order to ensure safety for others, emergency evacuation plans are being reevaluated and analyzed to ensure we, as a nation and leaders, are issuing the best solution for responding to an emergency by getting to a designated safe area and providing medical care.

### 1.1 Background

At the United States Military Academy, located at West Point, New York, the Corps of cadets and 38,000 Army West Point fans come out to Michie Stadium for seven football games, fourteen lacrosse games, and the highly spectated graduation every year. Michie Stadium is a two-level stadium that has a seating capacity of 38,000 within forty sections. The seating within the stadium is composed of metal bleachers that are screwed into the floor. Additionally, there are eight gates for spectators to enter and leave the stadium. Due to the number of high profiled military personnel and the majority of future junior officers

of the Army located in the same location increase the likelihood of a terrorist attack. A terrorist's goal is to kill the maximum amount of people, whether it is done so by mass shootings or a suicide bomb.

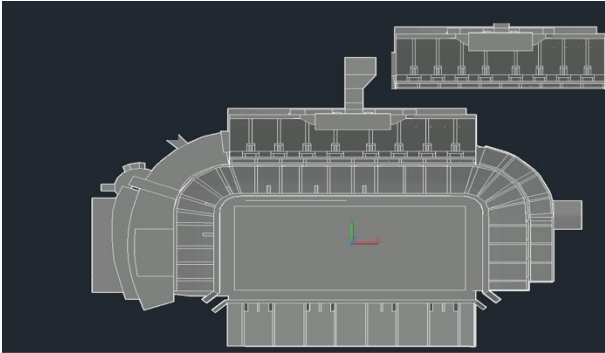


Figure 1 ProModel Map of Michie Stadium

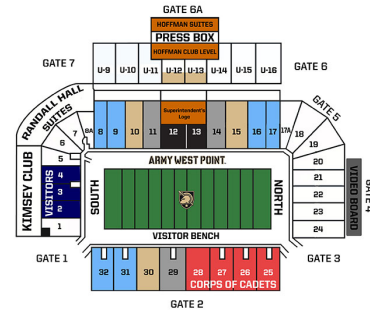


Figure 2 Seating Map of Michie Stadium

## 1.2 Literature Review

There are several resources focused on Modeling Stadiums and Security Resources. A case study, completed by the Cartographic Modeling Lab (CML) of Penn University, on an evacuation incident of Franklin football field. Within this model, CML used SimWalk to evaluate the time it would take to evacuate the graduation event which comprises of 15,000 to 18,000 personnel. Their model explicitly considers special requirements of 75 disabled personnel. The simulation results showed that it would take approximated 15 to 16 minutes to evacuate all personnel. Despite the stadium layouts being different, this case study provides an example for concept design and comparison.

Zale and Kar conducted this study by modeled the University of Southern Mississippi (USM) football stadium, M.M. Roberts Stadium. Zale and Kar using Geographic Information System (GIS) software model with the objectives of identifying the number of evacuees and their potential origins and destination, identifying evacuation routes and the shortest travel time and compute the total evacuation time of the stadium. On average, the case study found that it took 2.1 hours to evacuate 33,000 personnel from the stadium. The USM model includes the evacuation off campus when the Michie Stadium is strictly evaluating how long it takes to evacuate the personnel from the stadium.

Large venues such as stadiums are used for many different types of events such as sporting, graduations, and concerts. At Fayetteville State University, Wagner and Agrawal utilized an agent-based model to simulate crowd evacuation in the presence of a fire disaster. When modeling a stadium environment of 4000 people, it took on average 16.9 minutes for personnel to be evacuated. This model shows that through an agent-based system (ABS), it can help evaluate different scenarios to help improve safety and evacuation time. This model represented the effects of a fire disaster as adding fire as an agent in the system. Additionally, this study is useful to see the effects of a fire disasters drastically slows down personnel evacuation time. This is important to consider when modeling different emergency scenarios in Michie Stadium.

College football attracts spectators across the nation every fall. One of the top universities in college football, University of Alabama, created a model based upon a geographic information system (GIS). Within this model, Alabama found that the evacuation time estimate to be 23.54 minutes. Paul was able to calculate this through factoring distance and speed throughout the football stadiums. Analysis of similar

stadium evacuation models highlighted key factor consideration and methodology that better signage and fan education of exits, along with wider walkways would benefit the Michie Stadium model.

## 2 METHODOLOGY

Figure 1 illustrates the methodology of the methodology for this study. The model consists of entities arriving in grouped sections within the stadium. From each section, based off of the data collected through video, queues are created to illustrate the natural flow of personnel in Michie Stadium. User distributions and attributes are assigned for sections with multiple exits. The percentages used in these distributions were created through counting the flow rates of each gate. From each sections, queues are created to simulate the flow of entities out of the stadium. Capacities were placed on the size of the queues to directly reflect the amount of personnel that can actually stand within the area covering the queue.



Figure 1. Research Methodology

### 2.1 Stakeholder Analysis

The stakeholders for this project are West Point's Emergency Manager, the Superintendent (the University President), the Corps of Cadets, and the Director of Athletics. The initial client meeting began by understanding the current evacuation plan for Michie Stadium. Additionally, feedback and previous data that has been collected over the years in regards to Michie's attendance record. The goal for this project is to find friction points within the current evacuation plan which will allow a safer plan for the future.

### 2.2 Data Collection

In order to create this model, data was collected at three home football games. During this collection, multiple individuals were tasked to time and video record the movement of individuals out of Gates 1, 3, and 6a in order to identify choke points and service rates. This data can be broken down into three categories: structural, numerical, and operational data.

#### 2.2.1 Structural Data

Structural data in the context of our project is the layout of the stadium and the capacities of the sections. During the football games, I observed that each row could hold approximately 25 people. Some sections were slightly larger however because seats at Michie are a bench, rather individual seats, most patrons spaced out comfortably rather than trying to sit in their exact spot. Additionally, ticket sale were used to find out how many rows each section had.

#### 2.2.2 Numerical Data

The primary numerical data I gathered was the rate that entities departed the stadium. Specifically, I looked at the rate they passed through the bottlenecks. I reasoned that if I got the rate that people got through the bottlenecks, I could then get an average service time. At the conclusion of each football game I stood near chokepoints and recorded the amount of people who passed the point every 30 seconds.

### **2.2.3 Operational Data**

At the conclusion of football games, I stayed afterwards and filmed the behavior of entities leaving the stadium. I later analyzed this footage in order to come up with routing logic for each section to different exits. The last structural data gathered was the location of bottlenecks at the stadium. The bottlenecks at the stadium were not the gates themselves, but rather the aisles that lead to open areas.

### **2.2.4 ProModel Data Requirements**

For this simulation, service rates differed by the type of area entities were trying to get through. If the area was a walkway the service rate was .532 seconds with a standard deviation of .08 seconds. If the means to exit was a tunnel, the service rate was .499 seconds with a standard deviation of .07 seconds. Additionally, arrival rates were instantaneous as the disaster occurred. All 36,000 entities entered the system to be processed through pathways and tunnels to get out of one of the main exits of the stadium.

## **3 MODEL**

### **3.1 Verification and Validation of Model**

I was able to verify this model by viewing the amount of time it takes for the model to process all of the entities and comparing to the raw data that was taken at the home football games this season. The time it took for all entities to be processed out of the system was 2.91 minutes and on average it took around 17 minutes in our data collection. And since our entities are one-tenth of the actual amount of a maxed out stadium, it is expected to be 29 minutes. This time makes sense because we are not utilizing all of the gates in Michie in our model. Another way we were able to verify our model was by watching the animation of all the entities moving through the queues and gates. However, as I watched the animation, it significantly lagged the model. Throughout the creation of the project, I used the trace tool to ensure entities were going through the proper queues and gates. All three of these tools allowed me to verify that the model was reasonable and functioning properly.

The primary method of validating this model came from comparing the model to the real life system. When I timed the exit of Michie Stadium after a sold out football game, the time it took for the stadium to be totally evacuated took roughly 17 minutes. The results of the Promodel simulation demonstrated similar results. This means the Promodel is a good simulation of the real life scenario. I was able to get a utilization equal to 1 for the Bottleneck in the southeast of Michie Stadium. Finally, when I consulted our stakeholder, he informed me that a simulation model of the whole stadium has never been conducted. Due to this I am unable to compare my model to other models.

### **3.2 Assumptions**

The assumptions for this model include the following. Evacuees will prioritize their own safety, evacuees will only exit the stadium through designated exits (ex. No climbing fences), evacuees will move at a pace appropriate to the danger posed (ex. Running in life/death), there will be sufficient stadium personnel to facilitate an evacuation, all gates will be operational and accessible, no resources or personnel will re-enter the stadium once they have left, each row will hold 25 people, sections 6-8a, and section 17a-20 are corner sections and their capacity is 2/3 that of a section with the same amount of rows, and, people went to the exit that was most visible to them rather than the closest to their seat.



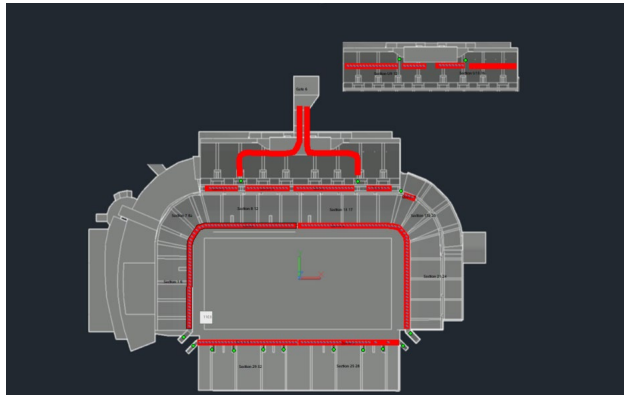


Figure 3. ProModel of Michie Stadium

#### 4 CONCLUSION

With the information provided from our data and Promodel there are a couple of recommendations I would make in order to improve the evacuation speed of Michie Stadium. With the current model it takes on average 17 minutes. The primary recommendation would be to increase the width of the concrete walkways that are directly connected to the seats for the fans. Additionally, proper signage of locations of all gates would help increase the throughput of entities exiting the stadium. It was found that many fans were unfamiliar with the layout of Michie Stadium. Even if after a regular game they showed a map of all of the gates, the time in system would drastically reduce. Additionally, the use of flashing lights/arrows will help fans move to the closest gate to exit the stadium.

#### REFERENCES

- Paul, D. 2015. "GAME DAY EMERGENCY: A FLOW AND VECTOR BASED GIS STADIUM EVACUATION MODEL"
- Savannah Simulations AG "Football Stadium. Evacuation Simulation Study Verified."
- Wagner, N. and V. Agrawal. 2013. "An agent-based simulation system for concert venue crowd evacuation modeling in the presence of a fire disaster."
- Zale, J. and B. Kar. Spring 2012. "A GIS-Based Football Stadium Evacuation Model" *Southeastern Geographer*. Vol 52, No. 1, pp 70-89.

#### AUTHOR BIOGRAPHY

**DELANIE A. WELIVER** is an Engineering Management Major with a focus in Biomechanics and Human factoring at the United States Military Academy. Her email address is [delanie.weliver@westpoint.edu](mailto:delanie.weliver@westpoint.edu).

## VIRTUAL AUTONOMOUS SYSTEM TRAINING VAST (ENVIRONMENT)

Germaine Badio  
Department of Modeling,  
Simulation, & Visualization  
Engineering  
Old Dominion University  
[gbadi001@odu.edu](mailto:gbadi001@odu.edu)

Elise Feldt  
Department of Modeling,  
Simulation, & Visualization  
Engineering  
Old Dominion University  
[efeld002@odu.edu](mailto:efeld002@odu.edu)

Zhaohui Hu  
Department of Modeling,  
Simulation, & Visualization  
Engineering  
Old Dominion University  
[zxxhu002@odu.edu](mailto:zxxhu002@odu.edu)

Peter Marchione  
Department of Modeling,  
Simulation, & Visualization  
Engineering  
Old Dominion University  
[pmarc002@odu.edu](mailto:pmarc002@odu.edu)

Christine Odenwald  
Department of Modeling,  
Simulation, & Visualization  
Engineering  
Old Dominion University  
[coden001@odu.edu](mailto:coden001@odu.edu)

Jaron Stevenson  
Department of Modeling,  
Simulation, & Visualization  
Engineering  
Old Dominion University  
[jstev00@odu.edu](mailto:jstev00@odu.edu)

Travis Sullivan  
Department of Modeling,  
Simulation, & Visualization  
Engineering  
Old Dominion University  
[tsull003@odu.edu](mailto:tsull003@odu.edu)

### ABSTRACT

Autonomous Vehicles (AVs) have the potential to be the future for public transportation, delivery services, military, and other applications. Lockheed Martin (LM) sees this potential and has developed autonomous systems for air, land, and water vehicles. These AVs require testing by introducing them to different scenarios that a vehicle could encounter. This paper contains the design review for the Virtual Autonomous System Testbed (VAST). This design review discusses a more detailed design of the solution than the previous proposal, the Proof-of-Concept currently under development, VAST's software testing methodology, and anticipated outputs of VAST.

### Keywords:

av testing environment, navigation simulation ,collision detection, scenario object

## 1 INTRODUCTION

The Corporate Engineering, Technology & Operations (CETO) Mission Development and Integration (MDI) Branch of LM wants to rapidly test AVs but needs to build scenarios in the most cost-effective manner. They want to have a test harness that enables their Testers to make decisions on the effectiveness and suitability of an AV in varying environments.

The VAST conception will build on existing research on navigation simulation, as well as integrate the design, development, and testing of software features necessary to the system implementation. It is within the scope of VAST to parse the sensor inputs and movement outputs of AVs, as well as the simulation initialization inputs and simulation run outputs.

As demonstrated in Figure 1 below, the overall system is comprised of three major modules: the navigation simulation(s), the AV/software, and VAST. VAST is comprised of the AV Testbed and the User Interface (UI) with its data and visualization components. User-defined parameters

and metrics will be entered into the User Interface, and the system will generate a visualization based on these parameters and metrics. Each auxiliary module (simulation, AV / software, and UI) will send and receive information to and from the AV Testbed.

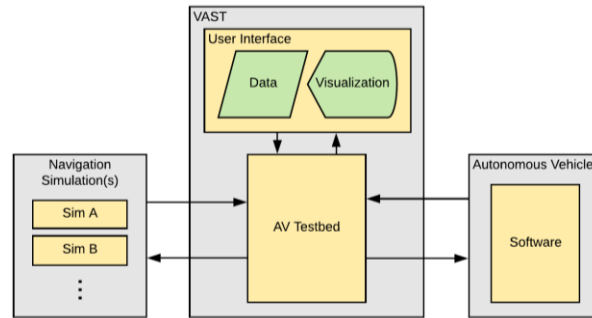


Figure 1: Testbed System Architecture

### Requirements Overview

This section lays out the requirements of the AV Testbed in Table 1. The requirements are broken up into three categories: the AV Testbed Process (what the internal testbed components are and how they interact), AV Testbed Inputs (from the user interface, the navigation simulation(s), and the AV / software), and AV Testbed Outputs (to the user interface, the navigation simulation(s), and the AV / software). Each requirement also has acceptance criteria associated with it, which describes the requirement in more detail.

Table 1: System Requirements

Requirement	Acceptance Criteria
<b>AV Testbed Process</b>	
AV Testbed shall accurately represent data input to clients.	With sensor parameters provided, the simulation must be able to understand the AV's decision. This decision must be representative of the AV's actual decision.
AV Testbed shall visualize the environment accurately.	The virtual environment visualized must represent AVs and other obstacles' position and geometry accurately.
<b>AV Testbed Inputs</b>	
AV Testbed shall be abstracted to handle multiple sensor and vehicle types that can be inputted by the user. These sensor and vehicle types shall have acceptable and realistic parameters.	The user must be able to add new sensor types with different parameters and functionality. The sensor parameters must provide the simulation with enough information to understand how the sensor will act including what types of data are associated with it.
AV Testbed shall receive and store scene simulation information.	The simulation information must be abstracted to handle multiple types of simulations.
<b>AV Testbed Outputs</b>	
AV Testbed shall accurately state the output made by the AV.	The geometric output may include information about the state of the environment surroundings, the AV, the sensor readings, and the resulting output of the vehicle.
AV Testbed shall evaluate whether a failure/operational mission failure has occurred and output its result.	A failure is attributed to any software issue that results in unpredicted behavior. An operational mission failure (OMF) is a failure that prevents the entire system from being used (e.g. crash).
AV Testbed shall report relevant performance measures	Performance measures include quantitative and qualitative information regarding the overall

performance of the AV. Relevant measures are described in Section 2.3.
--

## 2 SOLUTION DESIGN

### 2.1 Testbed Components

The VAST Testbed refers to the software executable at the center of the VAST system seen in Figure 1. When the Tester is ready to begin a scenario, she will configure VAST Testbed to control the start and run time of the clients (AV navigation logic and the navigation simulation). Modules in the VAST Testbed will receive known port names and variable configurations for communication with AV logic and the navigation simulation. The data will be intercepted for scenario analysis. Modules in VAST Testbed will perform data relay, data storage, and analysis calculations. Further modules will interface with the VAST Testbed user.

The data flow between components is illustrated in Figure 3. Each component block is a module of functionality planned for one or more source code files. An arrow describes the movement of data from source to destination.

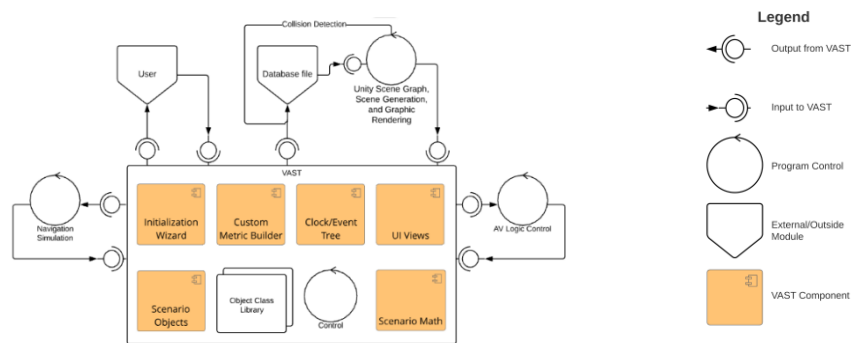


Figure 2: VAST Component Diagram

### Organization

The configuration wizard classes will perform complex routines involving the naming of scenario objects, the configuration of communication ports, and initialization variables needed for the clients' to be run via VAST. During initialization the user will also be prompted to select metrics and the location of the output database.

The scenario objects will extend from the Object Class Library to create objects with open port connections and data structures to complement the translation of incoming client data to outgoing client data. Events generated by port activity will disseminate to the associated scenario objects and the event tree.

As the development of the VAST program has evolved, the importance of the types of data flow has emerged as seen in Figure 3. Every arrow in the diagram below indicates part of the VAST program that was written to convey meaningful information from component to component. All of the data is part of the testing of AVs.

The initial black arrow going from the user to the Configuration Wizard symbolizes the programming of the QT graphical user interface and the initial efforts the user will undergo to set up their VAST system components. Subsequent orange arrows symbolize the programming inside of VAST that will use the configuration specifics to initialize components and TCP connections. This initialization is a form of automation for the VAST user, allowing them to access VAST features from a single control point once the components have been decided.

Internal to the VAST program, data will flow via updates (blue) and events (purple), with the EventTree and its Clock time as a hub that receives and organizes data flow by time stamp and

sends it to the database file. An update would be characterized by any information modified from external sources that changes attributes of the sending component. A sendEvent would be characterized by information the EventTree has organized and is ready to distribute to all the components in the system who may want to hear about changes to other components.

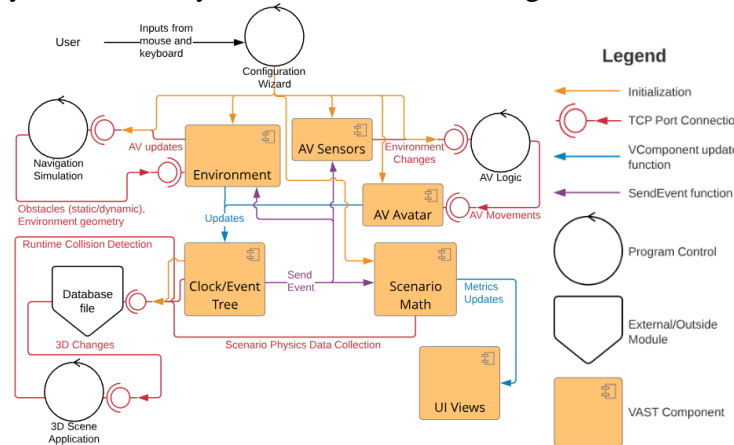


Figure 3: VAST Data Flow Diagram

### Extension

It is the aim of the VAST product to make extensible classes for the AV tester to create their own Environment, Obstacle, AV, and Sensor classes. Extensibility of these base classes will allow the user to create complex AV sensor types and more elaborate environments in which to test the AV.

There are currently no designs for a user interface for visual programming of extended Scenario Object classes.

### 2.2 Software Design

The VAST program will be written with Object Oriented Design, as referenced in the Testbed Components section above. The source code for VAST is stored in a GitHub repository. A user guide will be generated for VAST developer-user reference. The user guide should be developed as VAST Scenario Objects are implemented and functionality is expanded. VAST's Scenario Objects Library base class initial design UML has been included in the Appendix.

The abstract classes that are used in the current prototype include the Scenerio-Metric, AV, Sensor, Environment, and the VComponent class, their parent. All these classes allow the user to set up different extended specialization classes that can be used anywhere that the original class is used. This allows the user to create metrics that they would like to track as well as add more autonomous vehicles that will fall under the AV class. It is the aim of the VAST product to make extensible classes for the AV tester to create their own Environment, Obstacle, AV, and Sensor classes. Extensible base classes will allow the user to create tests with complex AV sensor types and more elaborate environments.

### Time, Events, and Synchronization

A simulation cannot exist without a concept of time, but an AV is assumed to operate in the real world, with continuous real time. A simulation can use continuous or discrete time progression. Both the AV and the navigation simulation must perform calculations with respect to state changes over time. Metrics and calculations are more readily accessible with a constant unit of time by which they measure those state changes. Because of this higher relevance for regular intervals of calculations, and the desire to update a visualization, the choice was made to follow a discretized method of time progression.

In order to bring together the state changes of these two client applications, VAST will adopt its own discretized event recording. The advancement of the clock time will be driven by VAST using multi-threading.

It is important to the virtual environment testing to run multiple replications and multiple scenarios as fast as possible, so the AV system logic may likely be required to take in and output a timestamp of its own, or implementation of the AV may be instructed to ask the EventTree for the current time.

The VAST Tester will ideally choose a minimum time slice with which the EventTree will prioritize incoming Events. Events that fall between time slices will be grouped and considered a single event, with priority given to the latest added event.

### ***Database***

Database set up based on SQLite3. Current database file is able to let user to connect between C++ and SQLite by using SQLite library. If the database does not exist, then it will be created and finally a database object will be returned. The Unity module will query data from the open database in order to generate scene graphics and perform collision detection during the run. The collision detection in the scene graph is during run time and that data will be published to the database file. The Scenario Math classes will perform live metric calculations and general physics calculations at each Scenario Object update. And the calculation will be sent to database file. The Event Tree/Clock class will perform the function of collecting and maintaining AV and Environment events and publishing them.

### ***2.3 Proof-of-Concept***

#### ***Proof-of-Concept Updates***

The Proof-of-Concept has shifted slightly after an evaluation of the capabilities of SUMO and new developments in the previously-researched simulation software AirSim. The primary method for controlling a vehicle in SUMO would be to use the MoveToXY command [1], which would require an update every time-step to continue moving the vehicle. Although using MoveToXY may be feasible, the control methods in other simulation software products are far more intuitive.

Considering these limitations and strengths, the updated Proof-of-Concept will leverage the 3D rendering and ease of control of AirSim, in coordination with the traffic simulating capabilities of SUMO. This will not only allow for additional features in the simulation but will also demonstrate the ability of VAST to interface with multiple simulations and create an ideal simulation for training AVs.

### ***Implementation***

The current plan for implementing both simulations for use with the AV is to use OpenStreetMap [2] along with various tools that have been developed to convert OpenStreetMaps to other formats (including Unity and Unreal environments, and SUMO networks) [3], [4], [5]. This would allow for the same real-world location to be represented in both simulation products, easing the process of having consistent scenarios.

This coordination of multiple simulation products will not be without challenges, as this introduces factors such as time-synchronization between the two simulations. However, this will be an ideal way of showing the full functionality of VAST, as well as helping to ensure VAST is developed with enough abstraction to enable such feats.

#### ***Collision Detection and Post-Simulation Visualization***

Another consideration is that not all navigation simulations have the capability to detect collisions, which is an integral part of training and testing an AV. To account for this, the Proof-of-Concept will also include Unity-driven collision detection.

The final component of the Proof-of-Concept involves the generation of a post-run visualization (as described later in section 3.3). This functionality allows users the opportunity to visualize a scenario after it has been run and interact with it: reversing, fast-forwarding, etc. This is highly useful in cases where the navigation simulation does not support 3D visualization inherently.

## 2.4 Verification for Design

Verification of VAST must be performed to ensure that it meets all project requirements as addressed in section 2.4. Each requirement will have a test description, criteria for accepting that the requirement is met, and description on how these requirements solve the problem statement. Criteria will include measures that may be thresholded or un-thresholded. The Minimum Viable Product (MVP) of VAST is considered complete when each measure of verification passes. A thresholded measure has a specific valued condition that must be met as a condition for the system to pass the test; in other words, it is a critical measure. However, an un-thresholded measure means that the value is calculated and tracked but is not required for the system to pass. Critical Operational Issues (COIs) will be addressed as well in the test plan; these include effectiveness (mission support) and suitability (reliability, maintainability, logistic supportability, and availability) determinations. This plan must be developed in accordance with the requirements and its acceptance criteria, which are described in Table 1 and will utilize design of experiments (DOEs).

Table 2 below lists the unit test procedures, associating each requirement with specific components within AV Testbed.

The collision detection section includes information about what requirement it is fulfilling, some information on what the collision detection module needs to report to the database, and descriptions of the test scenarios. The requirement that it is fulfilling is the testbed outputs requirement of evaluating whether a failure/operational mission failure has occurred and output its result". The module should report the run number, scenario number, time of the collision, and object AV collided with to the database. There are four test scenarios: post-simulation example to file, post-simulation example to the database, faster-than-real-time example to file, and faster-than-real-time example to database. The test scenarios with the files will be completed within the unity editor; however, the scenarios with the database will be a Unity executable application.

The Proof-of-Concept demonstration will be used to validate that the entire system works together properly; thus, it serves as the integration test for VAST. While the focus of the test plan has been on the unit testing of the classes, there is also the integration testing to take into consideration. An integration tests serves as the validation of the component interactions; the unit tasks prove the inner functionality of each component in isolation. Thus, a combination between these two tests will validate the functionality of our project.

*Table 2: Unit Test Procedures and Associated Measures*

Requirement	Unit Test Procedures	Measures
<b>Process:</b> AV Testing Environment shall accurately represent data input to clients.	-Tester will run scenarios that require data transfer between AV testing environment and the simulation/AV. -Information regarding data accuracy and transfer rate is recorded and analyzed.	-Data format reliability -Data transfer success rate
<b>Process:</b> AV Testing Environment shall visualize the environment accurately.	-Generate a virtual environment based on the simulation and AV interaction. -Determine if the AV position in the AV testing environment scene graph is reliable.	-AV position reliability in visualization
<b>Inputs:</b> AV Testing	-Tester selects multiple sensor types.	-Sensor list



Environment shall be abstracted to handle multiple sensor and vehicle types that can be inputted by the user with acceptable and realistic parameters.	-AV testing environment is expected to instantiate these sensors and associate them with the AV. -Sensor list is printed out and checked against the expected sensor listing.	reliability
<b>Inputs:</b> AV Testing Environment shall receive and store scene simulation information.	-Tester runs a scenario with a chosen simulation. -Scene information is stored internally to AV testing environment. -Stored scene information is compared to the expected scene information.	-Scene object position reliability -Scene update rate
<b>Outputs:</b> AV Testing Environment shall accurately state the output made by the AV.	-Tester runs multiple scenarios. -AV testing environment sends information to the AV and receives input back from the AV. -Output is compared to the expected output..	-Output reliability
<b>Outputs:</b> AV Testing Environment shall report whether a failure/operational mission failure has occurred.	-Tester runs scenarios that force collisions between the AV and other vehicles. -AV testing environment outputs the collision information.	-Collision detection rate
<b>Outputs:</b> AV Testing Environment shall report relevant performance measures.	-Tester selects metrics. -AV testing environment runs a test scenario with already-known metrics. -AV testing environment generates a metrics report. -Tester compares the generated metrics with the known metrics.	-Proper number of measures are reported -Measures are correct for the scenario

Overall, there will be limitations to testing VAST. The major limitation is the fact that this test does not include hardware-in-the-loop, which means that this validation is purely on the software. Even if this software is fully validated, new problems are likely to arise once both hardware and software are tested in conjunction. Specifically, in defense acquisition, multiple test events are conducted. In an Operational Test (OT) report from Fiscal Year (FY) 17, 64 OTs were conducted with Director, Operational Test and Evaluation (DOT&E) oversight. 34.4% OTs conducted in FY17 discovered critical, new problems in addition to old problems. 15.7% discovered only new, critical problems.

Additionally, there is no way a system can account for every scenario without testing how the system of systems (SOS) integrates with the system under test (SUT). This includes how the hardware interacts with the software (impossible to test the hardware) along with other simulation types.

### 3 VAST TESTBED USER EXPERIENCE (UX) AND OUTPUT

#### 3.1 System Flow

One of the most important steps in the software process is the basic user experience (UX), which for this project focuses around initialization and configuration, which consists of its own “wizard,” laid out in the configuration wizard UX. This process also accounts for a post-run view that will display metrics selected in the initialization step. This view will include aggregated metrics such as average speed, maximum speed, minimum speed, etc. Multiple replications can be selected, all replications will be run before this screen is displayed, and metrics will be

displayed per replication. VAST will generate an output database (.db) file. Within the run summary screen, the user will have several options as to how to continue. The simulation can be re-run, the user can go back to the initialization step, an interactive visualization can be generated based on the data from the run or the program can be exited.

The configuration wizard allows a user to select configurations for the simulation and AV, determine important metrics, select number of replications, choose sensors to use, and determine the location for any output file. The UI and flow for the wizard is depicted in the configuration wizard. The most important part of the configuration wizard is the behind-the-scenes initialization of the simulation and AV that occurs based on the information provided by the user in configuration files. This allows the user to determine things like how many vehicles should be present in the simulation, what AV logic to use, starting position of the AV, etc.

### 3.2 Metrics

Performance metrics are necessary because the user can choose to perform simulations faster than real time. It would be infeasible for a user to only visually determine the information they need in such a scenario.

Table 3 includes performance metrics chosen for their suitability to be implemented into VAST by default. “By default” means that the metric is applicable to most AVs and scenarios. Many of the metrics below would make excellent measures of AV performance, and the Tester will have the ability to extend the ScenarioMetric base class in a desired implementation.

*Table 3: Viable Metrics*

<b>Minimum distance</b>	The distance between the autonomous vehicle and any object. The object can be dynamic – like another vehicle in the simulation – or static – such as a building that a drone must avoid.
<b>Maximum acceleration /deceleration</b>	These depend upon the spatial paradigm of the scenario: feet, meters, etc. They measure, respectively, the maximum rate of increase and decrease in the AV’s velocity.
<b>Miles per collision</b>	Viable if the simulation does not end after the AV makes a collision. If this is the case, VAST will keep track of how many collisions occur and how many miles occur between them.
<b>XYZ location, average speed/ acceleration</b>	All viable metrics since they are related to one another. Since we are keeping track of X, Y, and Z location by default, we can calculate the average speed and average acceleration from this information.
<b>Congestion</b>	A metric that represents how many dynamic obstacles are around the AV within a given space. This will be useful in determining if the AV is generating traffic due to the way it behaves.
<b>Worst time gap between dynamic obstacles (front/back)</b>	Represents the smallest distance “in time” between two ground vehicles. For example, if driving on the highway at 55 mph and the car in front of the AV passes a pole, and it takes 3 second for the AV to pass that pole, then the time gap between them is 3 seconds. The way this is different from following distance is that if both vehicles in the example are driving at 25 mph, then the distance between them can be smaller, but the time gap between them might still be 3 seconds.
<b>Stop frequency</b>	How often the vehicle must stop along its route. This may be useful to determine if the AV is making more stops than necessary.

Additionally, a standardized testing arena is something the Intelligent System Division at the National Institute of Standards and Technology is proposing in order to facilitate standard performance metrics. These Autonomous Road Driving Arenas (ARDA) would ideally promote

the creation of AVs by providing a standard testing method and “lifting the standards of development within the community by providing comprehensive data sets, publicly accessible arenas, and competitions.” It would make sense that a virtual replication of said arenas could be equally as beneficial for testing an AV’s decision making during early development [6].

One crucial aspect related to performance metrics is the feasibility of this testing arena, although it is not within the scope of this project to determine that feasibility. In the use case of a car AV, it would be impossible to measure the aggressiveness of the onboard AI if there are no other vehicles for the car to interact with, nor would it be possible to determine how well it handles the aggressiveness of other vehicles on the road. It is also important to have a repeatable scenario in order to see the results adjustments to the AV logic. The feasibility of the scenario lends to the feasibility of the AV performance metrics.

A feasible testing arena for any AV type would have features that can adequately allow the AV to perform a feature test. For a car AV’s camera feature, a feasible testing arena must include objects visible in varying degrees of the color and light spectrum that is picked up by that camera. Objects must move on and off screen and move at *and* faster than the camera frame rate. Objects must range from smaller than a pixel size to larger than the camera frame size. In this way, the metrics developed for the camera feature testing will confidently report the failure parameters of the AV’s camera sensor.

### 3.3 Visualization

The Proof-of-Concept will have an application based in Unity to be optionally used as a plug-in by the VAST Tester. The purpose of the Unity-based visualization is to show the user the combined scenario of the simulation and the AV. The overall logic of the visualization as seen in Figure 4 below begins by reading in an output file from SUMO that contains second-by-second position, rotation, and speed data for each vehicle in the simulation. The number of vehicles is determined based on this file, and the vehicles are instantiated into the scene after runtime. The user of the visualization selects playback speed, and Unity program updates all vehicle information in the viewing window (position and rotation) accordingly.

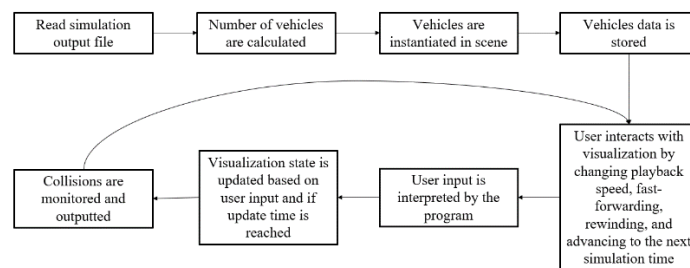


Figure 4: Visualization Logic

In the demo version of this visualization, there are multiple public variables (a reference to vehicle meshes/camera/text objects and time interval between simulation update) and private variables (a reference to the stream reader, current line of the data file, number of vehicles, vehicle list, current visualization time, space pressed, and enter pressed). The program initializes the update time, reads through the entire data file, interpret the first line of the data file to determine the categories and number of vehicles, and declare arrays and list.

Additionally, the update function is called once per frame, checks multiple conditions, and updates certain state conditions. The camera is updated based on the position and rotation of the first vehicle. The user’s input is received. The user can press the spacebar, the enter button, and the arrow keys.

The input received is interpreted by the program and is used to update all user interface information. These include five text object references: time text, car following text, position text, rotation text, and update time text (time between visualization update). Figure 5 below demonstrates an example scenario where the bus collides with a car. The simulation time, the name of the vehicle being followed, the world position, the rotation of the vehicle, and the time between visualization update in seconds are all included in the left panel. Additionally, the “paused” visual feedback, graphics/audio statistics, and user input instructions are included.



Figure 5: Visualization Example Scenario

Additionally, collision detection information is outputted using the “OnCollisionEnter” built-in Unity function [7]. To increase performance of the visualization and collision detection, a simple box collider is used to encompass all obstacles in the scene, while a mesh collider is used to encompass the AV. The major difference between these two colliders is the mesh collider is shape-specific [8]. Collision detection is controlled by the physics system in the Unity program [9]. The layer collision matrix defines what object types can collide with other object types. Additionally, Unity allows for two major broad-phase types: sweep/prune and multi-box pruning. Continuous collision detection is also an option, which ensures that objects with a high velocity do not simply pass through other objects without the physics system recognizing the collision [10]. While it may reduce the performance of the simulation, it is necessary for certain use cases (e.g. high-speed vehicles).

## 4 GLOSSARY

Term	Explanation
AV Performance Metric (“metrics”)	A calculation of AV or scenario data in a specified formula that lends meaningful analysis to the Testers of the AV about its performance in any scenario.
Packet	A unit of encapsulated data that can be sent and received over a network and unpacked to become meaningful information for the recipient.
Scenario	A combination of 3D environment and objects and other potentially non-3D elements that together create a distinctive set of parameters for an AV to navigate.
Testing Arena	The precise combination of scenario elements that must be present to test the performance of the AV.
Tester	The future user of VAST, a developer of AV logic, or even potentially a developer of navigation simulations.

## 5 ACRONYMS

2D: Two-dimensional

3D: Three-dimensional

AI: Artificial Intelligence

ARDA: Autonomous Road Driving Arenas

AV: Autonomous Vehicle

CETO: Corporate Engineering, Technology & Operations  
COIs: Critical Operational Issues  
CPU: Central processing unit  
DOEs: Design of experiments  
DOT&E: Director, Operational Test and Evaluation  
FY: Fiscal Year  
GPU: Graphics processing unit  
ID: Identifier  
IQ: Intelligence quotient  
LMC: Lockheed Martin Corporation  
MDI: Mission Development and Integration  
MVP: Minimum Viable Product  
ODU: Old Dominion University  
OMF: Operational mission failure  
OT: DOT&E operational test  
Sim: Simulation  
SOS: System of systems  
SUMO: Simulation of Urban Mobility  
SUT: System under test  
TCP: Transmission Control Protocol  
UI: User Interface  
UX: User Experience  
VAST: Virtual Autonomous System Testbed

## **6 REFERENCES**

- [1] "TraCI/Change Vehicle State," 28 November 2018. [Online]. Available: [http://sumo.dlr.de/userdoc/TraCI/Change\\_Vehicle\\_State.html](http://sumo.dlr.de/userdoc/TraCI/Change_Vehicle_State.html). [Accessed 6 February 2019].
- [2] "OpenStreetMap," [Online]. Available: <https://www.openstreetmap.org/>. [Accessed 6 February 2019].
- [3] M. Fricker, "Street Map Plugin for UE4," 18 June 2018. [Online]. Available: <https://github.com/ue4plugins/StreetMap>. [Accessed 6 February 2019].
- [4] Mapbox, "Maps SDK for Unity," [Online]. Available: <https://docs.mapbox.com/unity/maps/overview/>. [Accessed 6 February 2019].
- [5] "Networks/Import/OpenStreetMap," 26 January 2019. [Online]. Available: <http://sumo.dlr.de/wiki/Networks/Import/OpenStreetMap>. [Accessed 4 February 2019].
- [6] C. Scrapper, S. Balakirsky and B. Weiss, "Autonomous Road Driving Arenas for Performance Evaluation," 24 August 2004. [Online]. Available: [https://ws680.nist.gov/publication/get\\_pdf.cfm?pub\\_id=822519](https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=822519). [Accessed 6 February 2019].
- [7] Unity Technologies, "Collision," 17 January 2019. [Online]. Available: <https://docs.unity3d.com/ScriptReference/Collision.html>. [Accessed 6 February 2019].
- [8] Unity Technologies, "Colliders," 17 January 2019. [Online]. Available:

- <https://docs.unity3d.com/Manual/CollidersOverview.html>. [Accessed 6 February 2019].
- [9] Unity Technologies, "Physics," 17 January 2019. [Online]. Available: <https://docs.unity3d.com/Manual/class-PhysicsManager.html>. [Accessed 6 February 2019].
- [10] Unity Technologies, "Continuous Collision Detection (CCD)," 17 January 2019. [Online]. Available: <https://docs.unity3d.com/Manual/ContinuousCollisionDetection.html>. [Accessed 6 February 2019].
- [11] "Database (SQLite) Setup for Unity," 8 July 2014. [Online]. Available: <https://answers.unity.com/questions/743400/database-sqlite-setup-for-unity.html>. [Accessed 6 February 2019].
- [12] S. Cope, "TCP/IP Ports and Sockets Explained," 6 July 2018. [Online]. Available: <http://www.steves-internet-guide.com/tcpip-ports-sockets/>. [Accessed 6 February 2019].
- [13] C. R. Rickarby, "Autonomous Driving Platform Performance Analysis," 2017. [Online]. Available: <https://scholars.unh.edu/cgi/viewcontent.cgi?article=1344&context=honors>. [Accessed 9 February 2019].
- [14] W. G. Phillip J Durst, "Levels of Autonomy and Autonomous System Performance Assessment for Intelligent Unmanned Systems," April 2014. [Online]. Available: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a601656.pdf>. [Accessed 6 February 2019].
- [15] Nvidia, "Nvidia Pro Pipeline," [Online]. Available: <https://developer.nvidia.com/nvidia-pro-pipeline>. [Accessed 6 February 2019].
- [16] S. Tamilarasan, D. E. Jung and L. Guvenc, "Drive Scenario Generation Based on Metrics for Evaluating an Autonomous Vehicle Controller," in *WCX World Congress Experience*, 2018.
- [17] D. Stanek, E. Huang, R. T. Milam and Y. A. Wang, "Measuring Autonomous Vehicle Impacts on Congested Networks Using Simulation," in *Transportation Research Board Annual Meeting*, Washington, DC, 2017.
- [18] "Welcome to AirSim," [Online]. Available: <https://microsoft.github.io/AirSim/>. [Accessed 6 February 2019].
- [19] Unity, [Online]. Available: <https://unity3d.com/>. [Accessed 6 February 2019].
- [20] Institute of Transportation Systems, "SUMO - Simulation of Urban MObility," [Online]. Available: [https://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931\\_read-41000/](https://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/). [Accessed 7 February 2019].
- [21] A. Fox, "The Difference Between a CPU and a GPU," Make Tech Easier, 31 January 2017. [Online]. Available: <https://www.maketecheasier.com/difference-between-cpu-and-gpu/>. [Accessed 6 February 2019].

# UTILIZING ACCELEROMETRY TO DETERMINE BODY POSTURE OF DENTAL HYGIENISTS DURING INSTRUMENTATION

Eric Jenkins  
College of Physical Therapy and Athletic Training  
School of Health Sciences  
Kinesiology and Rehabilitation  
Old Dominion University  
ejenk002@odu.edu

## ABSTRACT

The physical stress of clinical practice is an occupational risk factor for developing musculoskeletal disorders (MSDs) in dental hygienists. The utilization of dental magnification loupes shows promise in allowing the practitioner to maintain a neutral posture, enhancing quality of care and a reduction of physical stress. Dental hygiene students, and professional practitioners were recruited to perform simulated scaling instrumentation in a repeated measures design while wearing loupes, and not wearing loupes. Accelerometers were placed on bony landmarks of the head, neck, and spine to determine body posture while performing an instrumentation. Preliminary data suggest a positive correlation between wearing loupes and a more neutral body posture. Loupes were also perceived by the participants to improve posture as well as instrumentation performance.

**Keywords:** Accelerometry, posture, training, professional equipment education

## INTRODUCTION

The physical stress of clinical practice is an occupational risk factor for developing musculoskeletal disorders (MSDs) in dental hygienists. The incidence of MSDs is a well-documented concern in the dental profession and attests to the work-related trauma exerted on the practitioner. Additionally, dental hygiene practice requires the clinician to sit in a fixed posture position for long periods of time adding to the risk for cumulative trauma and MSDs. Since MSDs threaten work productivity, income, career longevity and the overall health of the clinician, it is imperative to understand contributing factors to implement solutions mitigating the negative effects of MSDs. Dental loupes are specially manufactured eyewear consisting of magnification lenses which direct the line of sight downward to the horizon, denoted as the declination angle. The utilization of dental magnification loupes may decrease neck flexion, allowing the practitioner to maintain a neutral posture while working and thus reduce physical stress.

## METHODS

A convenience sample of 24 participants, consisting of 12 second-year dental hygiene students and 12 practicing dental hygienists were enrolled in the study. Six second-year dental hygiene students and 6 practicing dental hygienists wearing loupes with a predetermined declination angle comprised the experimental group; while 6 second-year dental hygiene students and 6 practicing dental hygienists wearing loupes without a predetermined declination angle comprised the control group. Participants were screened to ensure there was no history of injuries or disabilities of the head, neck, or trunk



regions. Additionally, participants were required to own either predetermined or non-predetermined declination angle loupes.

Generating a model of the human body can be done with a multitude of technologies and varied techniques based on the needs of the given project. To determine the posture of a hygienist while performing an instrumentation, accelerometers placed at key points of the body allow an appropriate three-dimensional representation of the body posture during the task. Through the utilization of multiple accelerometers placed at certain locations, it is possible to construct a human body posture model. Each accelerometer records the three-dimensional position of the sensor in units of gravity. Through comparing the recorded data to the calibrated position relative to the horizon, each sensor records the X, Y, and Z orientation relative to both the horizon and body. Orientation data is compared to that of the baseline calibration of a neutral posture to account for differences in individual posture. Because acceleration is being used to ascertain position, appropriately filtering the signal to remove excess noise is critical. Three accelerometers are placed on the head, neck, and mid-back which enable the model construction of a human torso to better understand the effect of dental loupes on posture. By placing each sensor at the dedicated location, the generation of a head and trunk model represents an accurate model of posture during the trials. Each sensor was averaged over the course of the trial to determine average body position during the task.

Chair mounted typodonts (simulated head models including teeth) with artificial calculus were used to represent a simulated oral environment. Participants' testing order was counter-balanced, and they were assigned to wear loupes during either the first or second segment of the experiment. The instrumentation procedure consisted of scaling in each quadrant of the mouth for two minutes, totaling eight minutes. At the end of the study, an end user opinion 5-point Likert scale survey on the perception of loupes and their influence on instrumentation and posture was completed. Data from the accelerometers was processed using a custom MATLAB script to average the orientation of each sensor throughout each two-minute trial in order to ascertain body posture.

## **RESULTS**

Preliminary results suggest a positive relationship between wearing loupes and a more neutral body posture while performing an instrumentation.

## **CONCLUSION**

Wearing magnification loupes may improve posture while performing dental instrumentation when compared to not wearing loupes. These results suggest a benefit of dental hygienists wearing loupes while performing routine instrumentation to mitigate the occurrence of occupational based MSDs. Future investigation can provide useful information on whether altering the severity of declination angle of the loupes further influences posture.

# IS THE BANG WORTH THE BUCK? A VALUE-BASED AND STATISTICAL ANALYSIS OF THE U.S. ARMY'S MOVEMENT FROM THE 5.56 TO THE 6.8 ROUND

Christopher Banga  
Department of Systems Engineering  
United States Military Academy  
West Point, NY, USA  
Chrsitopher.banga@westpoint.edu

Gene Lesinski  
Department of Systems Engineering  
United States Military Academy  
West Point, NY, USA  
Eugene.lesinski@westpoint.edu

## Abstract:

The United States Army recently decided to invest in next generation weapon systems utilizing a 6.8mm munition and replacing the current M855 5.56 NATO round. This research models, analyzes, and compares the cost versus value of the 5.56 NATO round to the proposed program implementing the 6.8mm round highlighting the potential benefits and limitations the change makes on the performance of a standard infantry unit. To analyze performance, several scenarios are created and implemented within Infantry Warfare Simulation (IWARS), the Army's authoritative model for small unit constructive simulation. Key metrics of interest include lethality, survivability, and cost. The analysis finds the 6.8mm round improves lethality in all aspects, especially as engagements increase in range. Differences in terms of survivability, were negligible. However, movement to a 6.8mm round provides a greater cost-value return than almost any small arms investment by the U.S. military to date.

**Keywords:** Cost-Value, IWARS, Constructive Simulation, Lethality, Survivability

## 1 INTRODUCTION

### 1.1 Background

Currently, the United States Army fields a 5.56mm M4A1 carbine possessing a barrel length shorter than 16 inches for close to midrange combat scenarios. The U.S. Army also fields a 5.56mm M249 (SAW) belt-fed light machine gun intended to be a large casualty producing and suppressive weapon which is cheaper than the 7.62mm alternative, namely, the M240 general purpose machine gun. Together, these three weapons have provided the general basis of combat power for U.S. military operations for the past two decades.

The adoption of the 5.56mm round dates back to the end of the first World War. In the late 1920s, John Garand developed a .276 caliber rifle as the ideal combat weapon. The U.S. Military, however, decided to halt the development of the .276 caliber in 1932 due the excessive amount of .30-06 caliber M1906 ammunition stockpiled from the end of WWI. Although the .276 caliber proved to be the superior ballistic round, the now famous M1 Garand adopted the .30-06 caliber for WWII. While the U.S. and all of NATO had the opportunity to rectify this problem following WWII when the British designed the .270 (6.8mm) caliber and .280 (7mm) caliber ammunitions, they did not. Dr. Gary Roberts, an expert on ballistics terminal effects, expressed this decision as a "colossal weapon system selection error" in a presentation during the

2008 annual National Defense Industrial Association Armament Systems Forum (Roberts). Instead of the British round, the U.S. designed the 7.62mm T65 cartridge which showed “nearly identical ballistic characteristics to the battle-proven .30-06” (South).

The late 1950s saw the NATO militaries push for a more lightweight ammunition and weapon system. This in turn, pushed the development of the 5.56 NATO round, a round based on the .223 Remington which was designed for “varmint hunting” (South). The adoption of the lighter round was based on an effect known as “hydrostatic shock” or rounds that cause more damage from the tumbling of rounds and fragments following impact. In 1972, the 5.56mm NATO round was determined to lack the effective range and terminal performance of the 7.62mm NATO round, but the 7.62mm NATO round was still deemed too heavy. To create a middle ground, the U.S. military focused efforts on a 6mm Squad Automatic Weapon (SAW), but development was ceased four years later in fear of irritating the rest of the NATO community. Out of that development came the XM249/M249 SAW used today in the 5.56mm NATO configuration.

The 5.56mm round became the mainstay of NATO forces from the 1950s through current day. Overtime, the round has seen a variety of modifications from the original 55gr M193 to the currently used SS109 and 62gr FMJ (full metal jacket) M855. The Vietnam War indicated that the M193 was “very lethal at the relatively short engagement distances” (Crist). The round was able to easily penetrate walls and structures made of bamboo in Vietnam, but difficulties became apparent in Somalia when the M855 was unable to penetrate walls and other obstacles that existed in urban environments. Feedback from combat experienced soldiers noted that the M249 provided “good firepower” but “needed greater range and penetration power” and that the M4 carbines “too often required multiple hits to neutralize an opponent” (Crist). Dr. Roberts found that the 5.56mm NATO rounds suffered from a variety of difficulties from their inability to penetrate obstacles to the inconsistency at which the round yaws, fragments, or deforms leading to “relatively insignificant wounds” (Roberts).

These significant problems (lack of range, penetration, and lethality) prompted the U.S. military to begin looking at alternatives. This led to the lightweight small arms technologies (LSAT) development program in the early 2000s. The aim was to develop a replacement to the M249 by decreasing the weight of the system and ammunition by half. This led to Textron’s development of telescoped caseless and polymer rounds. The Army became “interested in more performance than the 5.56mm NATO round could offer” (Williams). Eventually, by 2010, the project would become known as the Cased Telescoped Small Arms Systems (CTSAS), which included a 6.5mm round.

During that development, a 2006 Joint Service Wound Ballistics-Integrated Project team would show the 6.8mm round as the “clear and unequivocal best performing” cartridge (South). This study prompted the Army’s 5<sup>th</sup> Special Forces Group to develop a new round with the help of the United States Army Armament Research, Development and Engineering Center (ARDEC). Through testing, the group arrived at the Remington 6.8mm special purpose cartridge (SPC) “offering the best combination of ‘combat accuracy, reliability, and terminal performance for zero to 500 yard engagements’” (South). In the end, the 6.8mm SPC was not adopted due to “modest aerodynamic properties” with little improvement in trajectory or wind drift at longer distances outside 500 yards. In 2013, the Army would undergo the Caliber Configuration Study (CCS) supporting two programs known as the Combat Lightweight Automatic Weapon System (CLAWS) and Lightweight Dismount Automatic Machinegun (LDAM). The two programs would later become grouped under the Next Generation Squad Weapon (NGSW) program and the CCS became the Small Arms Ammunition Configuration (SAAC) study. The Army began to publish results of the SAAC in 2017 stating advanced bullet technology, energy at the target, and fire control as the most important factors. By 2018, the U.S. Army released two Prototype Project Opportunity Notices (PPON) for the NGSW weapon systems.

## 1.2 NGSW and the PPON

Militaries around the world have begun to move away from the 5.56mm NATO round. The French, like the U.S. are looking to replace the M249, while Germany has already begun replacing many of their 5.56mm SAW systems with a 7.62mm-MG5 modular GPMG. The British have also begun to implement more of the 7.62mm round and have also already replaced their M249s with a 7.62mm variant. Turkey, the second largest NATO army now uses the MPT-76 which is a 7.62mm-rifle, as well as the Indian Army (Williams). Around the globe, NATO forces are drifting away from the 5.56mm NATO round and the pressure to maintain the use of the round for all NATO forces is diminishing. Many of the NATO concerns focus on overmatching enemy forces in a modern combat which would require outranging the 7.62mm PKM used by the Russians and Chinese. Further, there is growing concern about the readily available nature of level IV ballistic plates which can survive multiple armor-piercing 7.62mm rounds at a cost of as little as \$115. A quick search on the internet exposes many cheap, Chinese-made, ballistic plates that provide protection against a variety of heavy armor piercing calibers readily available to even the civilian populace. These concerns have driven the creation of the NGSW program.

The PPON provides “an opportunity to build a prototype based on certain requirements” (Bahde). The PPON issued by the U.S. Army requires the development of the NGSW to develop weapon systems that weigh less than 12 pounds, ammunition that is 20% lighter than current models, a specific dispersion tolerance, a weapon length of less than 35 inches, fire control of three pounds or less, a classified lethality requirement, a specific rate of fire, the ability to attack and use a suppressor system, and quick and accurate weapon controllability. The PPON discusses a classified 6.8mm general purpose cartridge which must be incorporated into the system. The bullet must weigh at least 120 gr. but cannot be larger than 130 gr. The Chief of Staff of the U.S. Army stated the round will “penetrate any existing or known body armor” (Milley). To maintain the energy necessary for that task, the round would need a muzzle velocity in excess of 2,800 fps (Bahde). The NGSW Program encompasses the NGSAR program (a more recent add-on) which details requested designs for a carbine or rifle intended to replace the M4A1, while the NGSW itself details requested designs for a light machine gun intended to replace the M249 SAW.

Currently there exists a variety of options to achieve many of these difficult tasks. Publicly, 6.8mm terminal ballistics can best be viewed by looking at the already designed Remington 6.8mm SPC. A physical analysis of the 6.8mm SPC against the .223 Remington showed that both had nearly identical lengths and maximum average pressure, but the 6.8mm SPC utilized a 120 grain bullet weight versus the 62 grain of the .223/5.56 (Gash). Gun World fired the 6.8mm SPC and recorded their ballistics reports. Combined with 5.56mm M855 data accessible on [shootercalculct.com](http://shootercalculct.com) a comparison table was generated as Table 1.

Table 1. 6.8mm SPC versus 5.56mm M855 Ballistics

Round	Range (yds)	Drop (in)	Velocity (fps)	Energy (ft-lbs)
5.56 M855	0	-1.01	2989	1230
	250	-7.24	2257	701
	500	-57.30	1632	367
	750	-193.29	1118	172
	1000	-496.75	954	125
6.8 SPC	0	-2.60	2460	1612
	250	-10.10	1955	1018
	500	-75.00	1524	619
	750	-225.00	1196	381
	1000	-299.00	1008	271

The 6.8mm SPC in its current form, while having slightly less velocity, maintains the energy necessary for terminal ballistic effects to far greater ranges than the standard 5.56mm M855 NATO. In fact, at 500 yards, the 6.8mm SPC has almost twice the amount of energy remaining for impact on the target.

Despite the advantage in terminal effects, the 6.8mm SPC does have some shortcomings. The round remains too heavy and when used in current magazine, allows only for 25 rounds per magazine. For a soldier to carry the same combat load as current, a soldier would need about 8 magazines as opposed to the standard seven but would also incur the same weight load as if they were carrying 300 round of 5.56mm NATO (Crist). Understanding, the burden of weight is extremely important as a fatigue exercise conducted by the Army found accuracy of hits dropped from 8.5 out of 10 to about 6.8 out of 10 when a soldier is fatigued by a 35 lb load (Ito). An Australian Army study showed that soldiers routinely carry weights exceeding 50kg (Brady). These excessive weights reduce a soldier's agility, produce fatigue and increase the likelihood of injury within soldiers (Brady). Natick found that excessive soldier loads increase energy expenditure by .7-1.0% for every extra .1 kg of load (Knapik). The Australian study concluded that reducing soldier load may require making items smaller and lighter and to invest in new technologies that meet that need rather than removing equipment, in order to maintain the soldiers benefit of having all possible and necessary equipment.

Options for reducing round weight do exist, to include the Sig Sauer's belt-fed machinegun which uses a hybrid ammo offering a "20% reduction in overall cartridge weight as well as increased velocity over conventionally produced ammunition" (Soldier Systems). There is also Textron's telescoping ammunition which was mentioned earlier. An Army study on decreasing the weight of the 7.62mm case found a mix of polymer casings and polymer links could decrease ammunition weight by 35-45%. According to their survey, a soldier carrying the standard load of 600 rounds for an automatic gunner could save over 12 lbs or replace this reduced weight with additional ammunition and increase the potential of "stowed kills" (Cloutier).

Cost is always an important concern in light of decreasing military budgets. The NGSW program is expected to cost about \$10M in the first year and \$150M per year as production increases for 10 years. The production award is expected to include 250,000 total weapon systems to begin replacing the 1,000,000 M4s and M16s currently in service, starting with front-line combat units (Soldier Systems). This, however, does raise issues on how the weapons would be serviced. The Army intends to keep the M4 within use for soldiers not directly in combat. This means maintaining the production of the 5.56mm round for the foreseeable future. Further, the U.S. Army would need to begin creating a stockpile of 6.8mm ammunition which, unlike in the past with the 7.62mm and the 5.56mm, is not available. In order to achieve this goal, the U.S. Army would thus need to support two weapon systems and two ammunition productions simultaneously to keep soldiers supplied for combat.

## **2 PROBLEM DEFINITION**

The proposed change from the current standard military system consisting of the 5.56mm NATO round to a newly designed 6.8mm round will undoubtedly see an increase in ballistics and range at the potential sacrifice of other factors. In order for the U.S. Army to properly justify the expense involved in the development and implementation of a new round and supporting weapon system, the value gained from the investment must be estimated. Constructive battlefield scenarios allow the Army to analyze a variety of statistical information related to the performance of soldier equipment and analyze the effects of that equipment. With the comparison of the expected costs and the current performance of the standard loadout, the U.S. Army is able to compare the cost-value benefits to determine the best return on investment (ROI) of the rounds to support the U.S. Army's true area of focus: the battlefield.

## 2.1 Problem Statement

Model, analyze, and compare the real world effects of the newly proposed 6.8mm versus the current 5.56mm. The effects must be quantified within the U.S. Army's core desires to increase soldier survivability and lethality in a variety of battlefield situations in a cost effective manner.

## 3 METHODOLOGY

### 3.1 Experimental Process

The experimental process is generated into four separate phases as indicated in Figure 1 below.

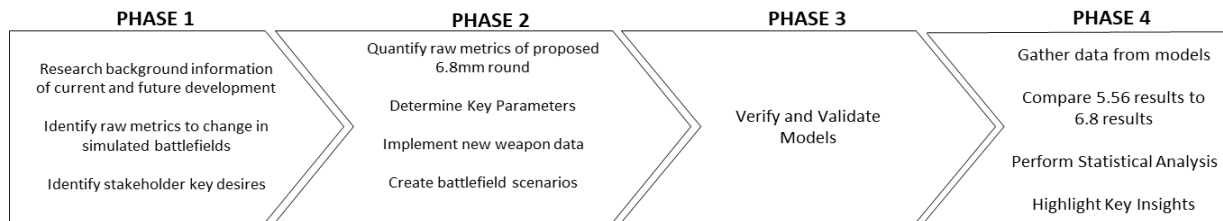


Figure 1. Methodology Process

#### 3.1.1 Phase 1

Phase 1 consists of the initial background research, identification of the raw metrics necessary to change in the IWARS database for effects in the simulated battlespace, and the identification of the stakeholder requirements. Background research revealed the historical impact of round changes, the justification behind the movement to a 6.8mm round, an analysis on how exterior effects can affect combat (basic load impacts), ballistic research on similar existing rounds, the requirements of the new weapon system and round, as well as currently developed technology that can help meet the new requirements. Stakeholder analysis was derived exclusively from the PPONs as they detail the U.S. Army's core desires and requirements from the 6.8mm round and its accompanying NGSWs and NGSARs. The PPON highlights that the system must have a weight of less than 12 pounds, an ammunition that is 20% lighter than current models, a specific dispersion tolerance, a weapon length of less than 35 inches, a classified lethality requirement, a specific rate of fire, and quick and accurate weapon controllability. Phase 1 also requires acquisition of ballistic data from current systems to include what current 6.8mm research has already been completed as well as the current metrics of the 5.56mm. Within IWARS, there are certain metrics that would need to be modified within the simulation database in order to model the effects of a new round. Within IWARS, further research identified adjustable factors within the constructive battlefield system and included: delivery accuracy, casualty assessment, weapon cyclic rates of fire, weapon weights, ammunition weights, ammunition capacity, ammunition carried, ammunition weight, weapon drag coefficients, and average weapon muzzle velocities. These factors affect the way a soldier shoots and kills enemies within IWARS.

#### 3.1.2 Phase 2

In Phase 2, the key parameters are developed from the stakeholder's desires and requirements, and the raw metrics from the proposed 6.8mm round are developed from a mixture of the current 6.8mm technology blended with the desired improvements outlined in the PPON. This data is then imported into the IWARS database and a variety of combat scenarios are developed within the IWARS system in order to analyze the potential varying effects on key parameters when soldiers are exposed to different combat

complications. Table 2 highlights the changes made within the IWARS database to replicate the new round. To fully analyze the NGSW program the base weapon systems currently in use, the M4A1 and the M240 SAW had to be analyzed as a baseline. Further, in order to analyze the NGSW program in its entirety, replacements for the M4A1 and M249 SAW with requirements detailed in the PPON were instituted within the IWARS database referred to as the NGSAR and NGSW, respectively.

Table 2. Data Changes for IWARS

System	Mean Muzzle Velocity (m/s)	Max Range (m)	Munition Capacity (per magazine)	Weight (kg)	Cyclic ROF (rounds/min)	Range50 (m)	Reload Time (s)
M4	905.267	3600	30	3.12	700	125	4
NGSAR	900.000	4000	25	2.95	650	250	4
M249	956.243	3600	200	7.44	600	200	9
NGSW	950.000	4000	200	5.50	600	300	9

The basic components for each weapon system were altered from the base M4 and M249s in order to create new weapons within the IWARS database. The NGSW velocity was dropped slightly based on the ballistic charts and information gained during the background research. The actual average velocity for the currently developed 6.8mm SPC was lower, but the value was adjusted to better reflect the new round which would require higher average muzzle velocities in order to maintain an effective penetration rate against enemy body armor, a stakeholder requirement. The max ranges and Range50 (range at which a hit has a 50% chance of incapacitating the enemy) were altered based on the ballistic charts as presented in Table 1. The weights were reduced to reflect new lightweight technology as well as to maintain the base requirement that the overall system weight must be less than 12 pounds. The NGSW reflected a drop in magazine capacity due to the Army's desire to use already fielded support equipment to include magazines. As such, only 25 rounds of 6.8mm fit within the standard M4 magazine instead of 30 5.56mm. Cyclic ROF was also reduced for the NGSW to reflect the larger caliber and increased heat buildup in the barrel for a system which is incapable of barrel changes. Lastly, the reload times remained constant as the same reloading systems are consistent for each weapon pairing. Further changes included dropping the ammunition weight by 20%, as well as adjusting the casualty assessment and delivery accuracy metrics using a mixture of restricted and unrestricted source material. Drag coefficients for the weapon systems were maintained constant between the weapon systems as no information on how they affect the new system could be found.

Following the U.S. Army's primary goals of introducing the new weapon system, the key performance parameters of interest are Lethality and Survivability. To measure these, an average over 50 replications, in a variety of combat scenarios, the killed red (enemy) soldiers and killed blue (friendly) soldiers are recorded as outputs. Simulated enemy elements are equipped with AK-47 and PKMs, which are the primary weapon systems of most U.S. near peer military threats. Lastly, three unique scenarios were developed in order to include the potential effects of engagement range, scenario lethality, and scenario complexity.

### 3.1.3 Phase 3

Phase 3 includes the verification and validation of the battlefield models. Phase 3 ensures the scenarios operate within IWARS as desired and that the imported data is creating the desired results. In order to validate each scenario, they were run with the base database. This method ensured that no-background data changes affected the base workings of the scenario. Blue force soldiers were given M4s to start each scenario while red force soldiers were given 9mm handguns. This method ensured that the



blue force soldiers were able to engage and kill the red force while also accomplishing the missions programmed into the scenario. Once success was achieved, the weapons were switched between the forces to ensure the red force was able to stop and kill the blue force from accomplishing their mission. The M4 and 9mm handgun were chosen due to the already known significant difference in their performance abilities in range and lethality. In order to validate the new database and weapon changes, a base scenario was designed with soldiers on flat ground. The blue force was armed with the new NGSW or NGSAR while the Red Force was unarmed. This ensured that the new weapons and new ammo were both functional and able to kill the enemy force.

The only method of verification during phase three was a submission and peer review of the mission programming, new database, and scenarios by resident IWARS subject matter experts located in the United States Military Academy Systems Engineering Department (USMA-SE). The Department helped to ensure the models were achieving the data analysis required of the project. Further verification is possible by submitting the project to the U.S. Army Natick Soldier System Center (NSSC).

#### **3.1.4 Phase 4**

Phase 4 concludes the process by applying the developed metrics to the constructed battlefields and generating numerical effects on the identified key parameters. The data generated from the battlefield scenarios is then statistically analyzed to determine statistical significance and compared with the expected cost to develop a cost-value analysis. With the quantified effects and a statistical analysis of those effects, the key insights are summarized on the value and benefit of the proposed change for the U.S. Army from the 5.56mm to the 6.8mm round.

### **3.2 Key Performance Metrics**

The U.S. Army's main focus on the change from the 5.56mm round to a redesigned and more lethal 6.8mm system is to maximize a soldier's ability to accomplish their mission in ground combat. In particular, a soldier's weapon system in regards to the accomplishment of their mission is to maximize the soldier's ability to survive the mission and complete the task while also maximizing the soldier's ability to eliminate threats. These two effects can be summed up as a soldier's lethality and survivability in combat. The 6.8mm round is already known to have an increased stopping power and increased range over the already existing 5.56mm round. In theory, these increases are intended to allow the soldier to neutralize the enemy more effectively and quickly. The proposed increase in lethality stem from the idea that the greater stopping power allows the soldier to neutralize more enemy soldiers with less ammunition, thus leaving more ammunition to engage other enemy forces in the area. Further, the greater stopping power should allow the Army soldier to neutralize the enemy before being impacted by the enemy soldier's weapon. The increased range is expected to allow the soldier to have a longer duration in which they are capable of applying effects on the enemy as well. The converse is also true in terms of survivability as the Army soldier should be able to out survive the enemy through a mixture of stopping power and range when compared to the 5.56mm round. The goal is to maximize a soldier's lethality and survivability and then determine if the value incurred justifies the allocation of resources. In order to measure the value created in terms of lethality, the number of enemy killed in action (KIA) will be analyzed through the battlefield scenarios to determine the increase or decrease, respectively, dependent on the round used in the scenario. Similarly, survivability will be tested by analyzing the number of friendly KIA in each scenario. Together these data points will be analyzed and paired with their estimated resource costs to determine the cost-value benefit allowing the 5.56mm and 6.8mm rounds to be compared. The key performance metrics are summarize in Table 3 below.

Table 3. Key Performance Metrics

Priority	Metric	Data Collected	Goal
1	Lethality	Enemy KIA	Maximize Lethality
2	Survivability	Friendly KIA	Maximize Survivability
3	Cost-Value	Expected Cost Lethality and Survivability	Pareto-Efficient Cost-Value

## 4 MODELS

### 4.1 Overview

In order to analyze the effects of the NGSW program on soldier lethality and survivability a variety of scenarios were developed varying the level of complexity, the size of enemy, and the distance of the engagement. The table below details a basic description of the models, and they are fully described in the sections below.

Table 5. Model Descriptions

Scenario	Friendly Size	Enemy Size	Complexity	Distance	Danger
Scenario 1a (Base)	1 Rifleman	1 Rifleman	Low	100m	Low
Scenario 1a (Base)	1 Rifleman	1 Rifleman	Low	300m	Low
Scenario 1a (Base)	1 Rifleman	1 Rifleman	Low	500m	Low
Scenario 2 (Ambush)	1 Squad	1 Squad	Moderate	<100m	High
Scenario 3 (React to Contact)	1 Platoon	4 Rifleman	High	>400m	Moderate

### 4.2 Scenario 1

Scenario 1 is a base combat scenario in its most simplistic form. Included are one blue force soldier and one red force soldier. Both forces are only capable of shooting at the opposing force from a standing position. The red force soldier was equipped with a standard loadout of one AK-47, body armor, and 210 rounds of 7.62x39 ammunition. The blue force soldier was equipped with an M4, body armor and 210 rounds of 5.56mm ammunition in the base scenario. The scenario was then repeated, but the blue force soldier was given an NGSW instead of an M4 and 175 rounds of 6.8mm ammunition instead of 210 rounds of 5.56mm (this ensured that soldiers were given a standard loadout of 7 magazines in each scenario). Scenario 1 was run three times as scenario 1a, 1b, and 1c. During 1a, the soldiers were placed 100m apart, 300m apart in 1b, and 500m apart in 1c. This allowed a raw gauge of soldier lethality and survivability without the potential effects of complex scenarios. Each different version of the scenario showed ranges to identify the change in the weapons lethality and survivability at first a range where they should both be effective and accurate (close range), a range where the M4 is at its max effective point range (mid-range), and a range where the M4 is past its most effective range and is at the NGSW's max effective point range (long range). These ranges were decided on using the ballistics data and bullet-drop off rate in table 1. Figure 1 shows a demonstration of scenario 1.

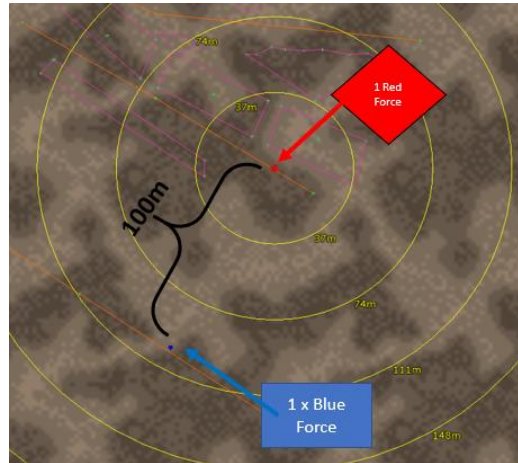


Figure 1. Scenario 1a

### 4.3 Scenario 2

Scenario 2 intended to add complexity, participant size, and scenario lethality to the simulation. In this scenario, a blue force squad with a standard loadout (M4s, M249s, and ammo) move North to South through an urban terrain. The squad splits into two teams as they move through the buildings and both teams are ambushed by an enemy squad minus. In the second run of the scenario the M4s were swapped with the NGSW and the M249s were swapped with the NGSAR. The scenario was intended to put stress on the effects of the lethality and survivability of the weapon systems using a realistic complex scenario that would normally be difficult and dangerous for real life soldiers in combat and analyze what effects the new weapon system can have in that type of scenario. The blue force soldiers mission is to push through the kill-zone as quickly and effectively as possible and are able to find and seek cover during the engagement. Figure 2 demonstrates scenario 2. Similarly, the enemy was equipped with a standard squad loadout comparable to that of the blue force and were equipped with Ak-47s instead of M4s and PKMs instead of M249s.

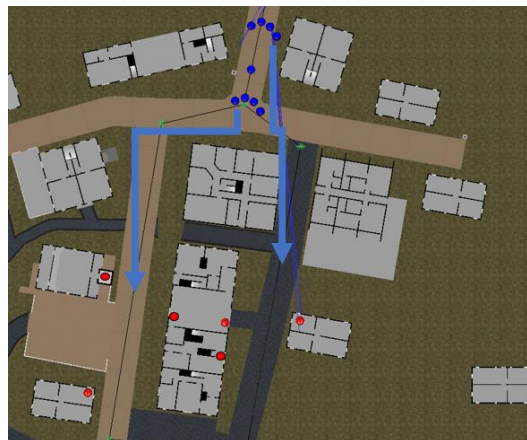


Figure 2. Scenario 2 Squad Ambush

### 4.4 Scenario 3

Scenario 3 added the most complexity, participant size, and range. As opposed to the close range of the squad ambush, scenario 3 included a platoon blue force reacting to contact against four red force soldiers in the mountains at ranges greater than 400m. This scenario was developed by NSCC for their own analysis and adapted for this project. As with scenario 1 and scenario 2, the first run was completed using the

standard infantry loadout, but included M240s as per the U.S. Army's standard loadout for a platoon. In the second run, the M4s again were swapped for the NGSW and the M249s were swapped for the NGSAR, but the M240s remained constants to best accurately represent a true infantry platoon. In this scenario, the red force is not fixed and is able to retreat, go prone, and fire on the blue force, while the blue force is able to move, seek cover, and moves up the hill to fix and destroy the red force. Figure 3 depicts scenario 3.



Figure 3. Platoon React to Contact

## 5 RESULTS

Each scenario was run for fifty iterations under two cases. Each case represented one fielding the traditional 5.56mm ammunition with the necessary equipment and one fielding the 6.8mm ammunition with the necessary equipment. The scenarios were run through the IWARS batch analysis and the data was saved to excel with the Red KIA and Blue KIA for each iteration. Table 5 highlights the key results.

Table 5. Key Results

Statistic	Scenario 1a	Scenario 1b	Scenario 1c	Scenario 2	Scenario 3
5.56 Survivability	18%	40%	66%	49%	99%
6.8 Survivability	68%	76%	76%	52%	98%
5.56 Lethality	70%	74%	38%	65%	8%
6.8 Lethality	100%	98%	70%	84%	18%

To determine the significance, a paired t-test was used to compare the changes in data between the two cases. Scenario 1a showed a significant increase in lethality and survivability for the 6.8mm round. The analysis also showed a p-value of  $5.59 \times 10^{-7}$  for survivability and a  $3.18 \times 10^{-5}$  for lethality showing significance for both parameters. Scenario 1b similarly saw an increase in lethality and survivability with a survivability p-value of  $1.31 \times 10^{-5}$  and a lethality p-value of  $8.3 \times 10^{-4}$  again showing significance. At longer range like in scenario 1c, there was still a small increase in survivability but a p-value of .3018. For survivability, the null hypothesis that there is no difference between the 5.56mm and 6.8mm data sets for survivability cannot be rejected, and the statistic is not significant. Despite the non-significance of survivability, the effect did not translate to lethality as the p-value and increase for lethality remained significant with a  $4.897 \times 10^{-3}$ . Scenario 2, like in scenario 1c, showed an insignificant survivability with a p-value of .3153 and a significant p-value of  $3.55 \times 10^{-6}$  for lethality. Scenario 3 followed the same trend as the two earlier scenarios with a p-value of .3082 for survivability and .0023 for lethality. In all scenarios,

the lethality statistic increased when the 6.8mm was in use versus the 5.56mm. Survivability also increased for all scenarios, except scenario 3, but not always to a statistically significant amount.

## **6 CONCLUSIONS**

The 6.8mm round outperforms the 5.56mm round in terms of lethality regardless of distance or scenario complexity. However, it is not statistically significant enough to claim how much the change affects the survivability of soldiers. While it seems if a soldier was able to kill more enemies in a shorter amount of time, then a soldier's survivability should also be affected. This thought stems from the belief that a soldier would experience less time in an engagement. Unfortunately, that belief did not hold credence through the constructive battlefield scenarios. Indisputably, however, is the effect that the larger round has on lethality. The increased range and terminal effects showed a significant increase in the amount of enemy soldiers that were killed in each scenario. The effects of the 6.8mm round seem to be more pronounced in scenarios where a soldier fights another soldier one on one. This effect is believed to be the result of other confounding factors present in a realistic combat scenario. While the 6.8mm round showed incredible improvement in the one-on-one engagements at closer ranges for both survivability and lethality, those effects broke down as the complexity and range of the scenario increased. In those scenarios, other factors outside the scope of this project helped to effect the viability of the equipment in combat. Nonetheless, the significant return of lethality remained constant despite the changes made to the scenarios, however, to varying degrees.

There are a variety of factors that could still be considered and it is possible that not all the benefits of the change are fully captured in these scenarios. While the physical effect of body load had some effect on the scenarios outlined in this project, they could have even greater effects on simulations that experience longer engagements and further distances moved as the physical exertion of the soldier increases. There still remains a large amount of potential factors that could add to the true benefit of the 6.8mm round than what is captured in this project and these results.

Much of the information present in the scenarios is not perfect. Much of the information is either classified or unavailable for a variety of reasons. While some restricted information was made accessible, some assumptions still needed to be made. The project simulates a round that is not completely made, thus, the weight, lethality, and other weapon statistics are assumed. These assumptions were based off of the requirements of the PPON, existing similar technology, and viable proposed future technologies, but the actual NGSW, NGSAR, and 6.8mm round could have statistics that are either higher or lower than what was assumed in this project. Further, the baseline information for the M4, M249, and 5.56mm present in the IWARS system are not completely true data points. This information is also restricted and some of that data was unavailable for this project. As such, the input for the M4 and M249 were adjusted to make the data as accurate as possible, but again some simple assumptions needed to be made which could place some of those values as higher or lower. Nonetheless, the relative comparison of the two data sets developed for use in the IWARS database are expected to be similar to the actual and expected values, thus giving this project credence in terms of a proof of concept or theory. The actual specific data may cause an increase or decrease in the true lethality and survivability, but the difference from this project and the actual is expected to be minimal based on the research and information outlined through phase 1.

Lastly, the cost of the change is significant with costs of roughly \$150 million per year. When compared to the \$210M dollar contract that the U.S. Army made with FN and COLT to produce the M4A1 over the course of five years, a \$1.5B dollar contract over 10 years is noteworthy. The value gained from the contract with COLT and FN, however, produced marginal value returns and when compared to the value potentially gained from the NGSW, the returns of the FN and COLT contract become nearly insignificant. The value gained in the ability of the 6.8mm round to help the U.S. Army neutralize the enemy and achieve the mission cannot be overlooked. The weapon system has the ability to help soldiers kill the enemy and survive in combat while using less ammunition further creating the potential to save the U.S. Army a significant amount of money over time. As such, due to the statistical and potential return of the 6.8mm round, the round should be considered to replace the 5.56mm.

## REFERENCES

- Army Contracting Command. October 04, 2018. "Draft Prototype Opportunity Notice (PON) for Next Generation Squad Weapons (NGSW)." Department of the Army.
- Army Contracting Command. July 12, 2018. "Draft Prototype Opportunity Notice (PON) for Next Generation Squad Automatic Rifle (NGSAR)." Department of the Army.
- Bahde, Dave. November 9, 2018. "What Will the Next-Gen US Army 6.8mm Round Look Like?" Tactical-life.com
- Brady, Chris, et. al. December 2011. "A Review of the Soldier's Equipment Burden." Defense Science and Technology Organisation, Australian Government Department of Defense
- Cloutier, Todd. May 2005. "Development of Lightweight Ammunition Concept Using An Alternative Case Material and Enhanced Propellant." Veritay Technology. East Amherst. NY.
- Crist, Stanley C. 2006. "Small Arms Ammunition for the 21st Century: High-Performance Alternatives to the 5.56 NATO Round." Infantry, no. 3: 28.
- Ito, M. A. Et. al. October 2000. "Rifle Shooting Accuracy During Recovery From Fatigue Exercises." US Army Research Institute of Environmental Medicine.
- Knapik, Joseph. June 1989. "Loads Carried by Soldiers: A Review of Historical, Psychological, Biomechanical, and Medical Aspects." US Army Research Institute of Environmental Medicine.
- Milley, Mark. "AUSA 2018 Press Conference." U.S. Army Professional Forum.
- Roberts, Gary. May 21, 2008. "Time For a Change. U.S. Military Small Arms Ammunition Failures and Solutions." NDIA, Dallas, TX.
- Soldier Systems. "AUSA 18 – SIG SAUER Unveils Belt-Fed Machine Gun, Carbine and Hybrid Ammunition." Soldier Systems.com
- South, Todd. December 10, 2018. "More than a Rifle: How a New 6.8mm Round, Advanced Optics will make Soldiers, Marines a lot Deadlier." Military Times.
- Williams, Anthony. October 19, 2018. "The US Army's Next Generation Squad Weapon Programme." UK Land Power.
- Gash, Steve. April 3, 2018. "Starline 6.5 Grendel vs. 6.8 SPC." Starline Brass.
- "5.56mm Ballistics Data." Shoortercalculator.com.

# MSVSCC 2019

## PROCEEDINGS



### Infrastructure Security/Military Application

Page 114

#### **Simulation and Analysis of the T700 Repair Process at the Corpus Christi Army Depot**

Tucker Pearson, Andrew Lekowski, Timothy Le, Tesfaye Larmey, David Oh and Eugene Lesinski

Page 126

#### **Deep Learning Based Malware Classification Using Deep Residual Network**

Yan Lu, Jonathan Graham and Jiang Li



# SIMULATION AND ANALYSIS OF THE T700 REPAIR PROCESS AT THE CORPUS CHRISTI ARMY DEPOT

Tucker Pearson

Department of Systems Engineering  
United States Military Academy  
West Point, NY, USA  
Tucker.Pearson@westpoint.edu

Timothy Le

Department of Systems Engineering  
United States Military Academy  
West Point, NY, USA  
Timothy.Le@westpoint.edu

Andrew Lekowski

Department of Systems Engineering  
United States Military Academy  
West Point, NY, USA  
Andrew.Lekowski@westpoint.edu

Tesfaye Larmey

Department of Systems Engineering  
United States Military Academy  
West Point, NY, USA  
Tesfaye.Larmey@westpoint.edu

David Oh

Department of Systems Engineering  
United States Military Academy  
West Point, NY, USA  
David.Oh@westpoint.edu

**Abstract:** The Corpus Christi Army Depot (CCAD) has a critical mission of repairing all rotary winged helicopters for the US Military. This research project aimed to improve the CCAD T700 Engine repair process. Two discrete-event simulation models were created in both ProModel™ and ProcessSimulator™ in order to identify and analyze process inefficiencies within the system. These models incorporate an Excel-based user interface that allow the user to conduct a “what if” analysis by manipulating different production variables to examine implications and results. A Multiple-Criteria Decision Analysis model was developed to evaluate and compare the cost versus value of competing production alternatives. Analysis of model outputs identified the plating shop and power turbine shaft as the key bottlenecks, with the most constrained resource as workers. The baseline scenario has a modeled capacity of 430 engines per year. The developed models will assist in analyzing the efficiency of the T700 engine repair process.

**Keywords:** Multiple-Criteria Decision Analysis, Cost-Value Analysis, Discrete-event Simulation, Depot-level Maintenance

## 1. Introduction

### 1.1 Background

Maintenance and acquisition in the Department of Defense (DoD) involves millions of different pieces of equipment that vary in complexity and magnitude. To streamline the maintenance process, the DoD grouped maintenance repairs into three distinct levels: basic-level repairs, intermediate levels of support, and depot maintenance. For basic-level repairs, the US Army employs organizational maintenance, which focuses on small, unit-level repairs to their own equipment. Units performing organizational maintenance utilize their own equipment and resources to complete the repairs. For more complex repairs, the Army implements intermediate-level maintenance. The Army typically provides intermediate levels of support during training and in war theaters at centralized locations. Intermediate repairs are applied primarily to units on deployment, in which they send in equipment to a specialized

maintenance station. Depots perform major equipment overhaul, maintenance, and the complete repair of parts.

Founded in 1961, the Corpus Christi Army Depot (CCAD) provides “helicopter maintenance, repair, recapitalization, and overhaul capability to all the U.S. military services, as well as several foreign governments” (History of CCAD). CCAD currently employs over 3,200 civilian employees and occupies facilities valued at over \$746 million (USD) on approximately 154 acres at the Naval Air Station (History of CCAD). The mission statement of CCAD explains that its goals are to repair rotary wing aircraft components of the DoD and other government organizations to ensure that the repairs are of the highest level of quality while also maintaining low costs, in the shortest amount of time possible (History of CCAD).

## **1.2 Model Motivation**

The CCAD motivation for developing two models of their T700 engine repair process is driven by a greater demand for process efficiency and economy of resources. This stems from escalating expectations of the U.S. Army to meet operational tempo. Increasing the efficiency of the T700 engine repair process will assist CCAD in meeting their goals of repairing more engines each quarter, while simultaneously decreasing their costs for each fiscal year. Furthermore, by analyzing the metrics valued by CCAD (throughput and utilization), this research will assist CCAD in identifying process inefficiencies to improve the operational availability of the rotary wing aircraft fleet. Previous CCAD modeling and analysis studies have been conducted on the Aircraft Corrosion Control Facility (ACCF), but not the engine repair shop (Delimarschi et al. 2018).

## **1.3 Problem Statement**

The purpose of this project is to improve the CCAD T700 Engine repair process by identifying bottlenecks in the system. In order to do so, this project will utilize two discrete event simulation models of CCAD’s T700 Engine Repair Shop process, analyze key performance metrics, incorporate a multi-objective decision support tool, analyze production scenarios, prioritize resource allocation, and improve shop productivity and efficiency.

## **1.4 Related Research**

This section identifies key concepts related to the development of the models described in this tech report. The major concepts discussed include simulation modeling, process improvement methodologies, production metrics, and buy vs fix. These concepts were selected and analyzed because of its relationship to the problem statement of improving the efficiency of the T700 engine repair process.

### **1.4.1 Simulation Modeling**

Simulation modeling is essential towards improving the T700 engine repair process at the Corpus Christi Army Depot. It is necessary to fit the needs, assumptions, nature, and scope of the problem with the proper simulation type to meet this criterion the best. With all model types considered, the T700 engine repair process will be modeled as a discrete-event simulation because the process steps for the engine repair are performed in a discrete sequence of events. Overall, this simulation can be modeled realistically as a series of discrete steps, thus making it feasible for both ProModel™ and ProcessSimulator™.

### **1.4.2 Process Improvement Methodologies**

An important component of the United States Military’s lethality is its logistic capabilities. Military logistics support “deals with everything required to provide war fighters with the right stuff at the right time at the right place at the right cost” (Bessant, 1997). Process improvement methodologies are systems that can help improve the efficiency of military logistics. Process improvement methodologies stem from W.E. Deming’s early lectures on quality management, which inspired numerous methodologies such Lean, Theory of Constraints, and Six Sigma. All these methodologies can be used to improve efficiency of production, which will be helpful for building the CCAD model. Lean focuses on removing waste from the production system; the Theory of Constraints focuses on evaluating the constraints that create bottlenecks in the system; and Six Sigma uses statistical analysis to reduce the variation of the system’s production of products. These three process improvement methodologies are useful tools for project

managers who need to efficiently manage their production systems. These processes were not directed applied in the creation of the ProModel™ and ProcessSimulator™ models, yet their optimizational influence can be seen residually throughout the study.

### **1.4.3 Production Metrics**

Each organization has different metrics and methods of measuring data, which adds complexity to creating simulation models. For example, some use a balanced scorecard that considers customers, internal processes, innovation, and finance. (Kleijnen & Smits, 2003). Others use graphs for metrics, such as a Kiviat eight metric graph or a five point spider diagram. Some use distributions and calculate performance based on defects per million opportunities, or the 6-sigma process capability standard. (Heravi & Mohammad, 2017). Others like the Overall Equipment Effectiveness use a combination of availability, performance, and quality to determine their metric (Anand, 2013). Thus, a variety of methods are available to measure and present data.

These metrics use different distributions and calculations, prompting programmers to decide what fits their simulation best. Stakeholders often use previous data or research to measure their results. Proper research is often the best way to discover a method to measuring productivity metrics, and merely applying such metrics is the most efficient and simple way to achieve success. However, in most cases, strong abilities in statistical analysis is required to produce results that accurately represent the model (Hubbard, 2014). CCAD's primary focus is on quality and efficiency, not on growth and profit margins. CCAD primary metrics of interest are throughput per week, time in system, resource utilization, and downtime.

### **1.4.4 Buy vs. Fix**

Excluding outliers, it is generally more cost efficient for an organization to pursue repairing or refurbishing a successful system over producing or buying a new system. If an engine arrives at the CCAD facility, the Army has already made the decision that it is more cost efficient to repair the engine over buying a new one. It then becomes a matter of repairing and refurbishing these engines as cost efficiently as possible.

### **1.4.5 Related Research Summary**

The topics of simulation modeling, process improvement methodologies, production metrics, and buy vs fix are influential towards one's understanding of the professional application of project management and systems engineering. Major takeaways from the related research for the T700 engine repair process are the simulation modeling research and production metrics research. The T700 engine repair process will be modeled as a discrete-event simulation because the process steps for engine repair move in a consecutive series of events. Additionally, CCAD stakeholders approved the production metrics, which will be a framework for evaluating and determining the best alternative scenarios from the model. These research takeaways are influential towards the modeling and understanding of the inefficiencies of the CCAD T700 engine repair process.

## **2. Methodology**

The Corpus Christi Army Depot (CCAD) has the critical mission of repairing all rotary winged assets in the US Military. The helicopter repair process is complex and requires a break down of sequential steps. These steps for the methodology of the CCAD T700 engine repair process are in **Figure 1**. The objective of this research is to improve the CCAD T700 Engine repair process by identifying bottlenecks and making recommendations. Data will be collected from historical records and stakeholder interviews. Two discrete-event simulation models are developed in both ProModel™ and ProcessSimulator™ to identify and analyze process inefficiencies. A Multiple-Criteria Decision Analysis model is developed to evaluate and compare the cost versus value of competing production alternatives. In order to experiment with problem scenarios, these models will incorporate an Excel based user interface that allows the user to conduct a "what if" analysis, by manipulating different production variables and examining their implications.

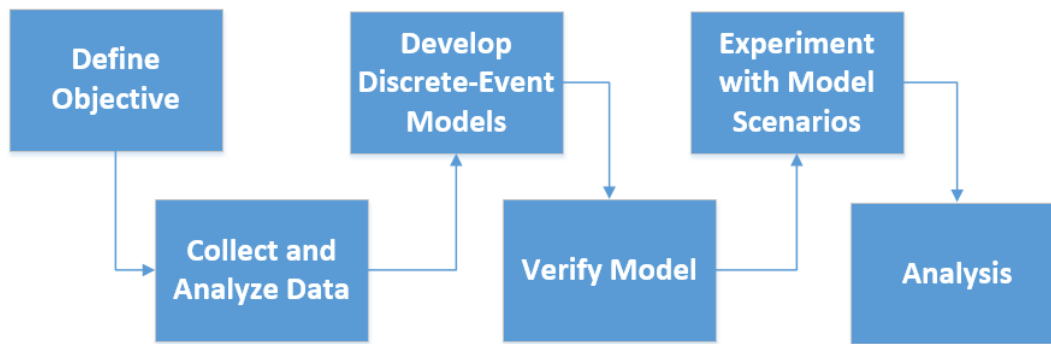


Figure 1. Research Methodology

## 2.1 Define Objective

An initial meeting with CCAD stakeholders was necessary to discuss the objectives of the models. The primary objective of the ProSim™ and ProModel™ is to improve the T700 Engine repair process by identifying bottlenecks and process inefficiencies within the system. The scope of this model is limited to processes within CCAD facilities, and all other facilities will be treated as a black box. Workers will be modelled as resources and assigned to work clusters (groups of workstations that share a common task). These work clusters are extended across the engine repair shop and cross between the four subassemblies of the T700 Engine.

This model will simulate the process flow of the four critical components of each major subassembly: PT Shaft (PT subassembly), Ball Bearings (Accessory subassembly), Combustion Liner (Hot subassembly), Midframe (Cold subassembly). These components have the longest lead times in the T700 engine repair process, so it is assumed that the engine repair process can be accurately modelled by following these four components.

## 2.2 Collect and Analyze Data

The data used in this research study was collected and analyzed from resources and objectives that were clarified by the stakeholders. CCAD's industrial engineers gathered this historical data for their own purposes and projects, so it was necessary to first filter and process the data. This filtering process consisted of both removing and renaming mislabeled data and removing data points with no values. The end result of this data scrubbing was viable distribution data, able to be used by the models to accurately reflect the process on the factory floor.

### 2.2.1 Process

The first step for data gathering was to verify and confirm the process flow diagram. In accordance with the project scope and objective, the process flow for the parts with the longest lead times were mapped through Microsoft Visio™. CCAD uses the Complex Assembly Manufacturing Solution (CAMS) to track the process of components. Using these CAMS steps, it was possible to create a process flow diagram. Finalizing the steps for the model included the removal of steps that were insignificantly used a few times during the lifetime of CAMS and also the removal of redundant activities in the process flow.

### 2.2.2 Data

CCAD also provided CAMS steps for the Level 0 descriptions of the CCAD T700 engine repair process. The Level 0 is the portion of the assembly line before and after it is broken into the four subassemblies. This included a datasheet consisting of engines by IDs, the processes the engines went through, the statuses of the engines, the time data of these processes, and other miscellaneous data. This data was filtered using several criteria:

- The engine must have been repaired within the past two years
- The engine must be completely repaired within CCAD facilities using CCAD assets
- Parts must be using the process steps that are listed by the stakeholders
- The process ID must match the process nomenclature, and
- The data must contain the time for the process

The values that the data consists of are as followed: NIIN number, location ID, location name, operation number, time in process, count of operations, and count of hours. Hours in this dataset are the time spent in the operation, which is the time the employee was occupied with the part before staging the it for the next operation. The data then was consolidated and prepped for statistical analysis.

### 2.2.3 Statistical Analysis

ProModel™ offers a Stat:Fit™ function that can be used to build a data distribution and also gather statistical data. Stat:Fit™ produces the necessary information to determine the data's: independence, variability, fitness for the provided distributions, regressions for these distributions, and the ProModel™ code to input these distributions into the model. Data populations without significant data values or populations with no matching distributions were replaced with the CAMS time data as a constant figure. These values were used so that the stakeholders would receive an accurate and realistic time in system, while allowing adjustments in the future with larger samples and corrected process data.

### 2.2.4 Resource Mapping

Resource variables such as resources, downtimes, and worker efficiency were added into the model. In the models, the only type of resources are workers. These workers are organized into seventeen different clusters, which are assigned to different workstation groups around the repair shop. Workers can be added or removed through the user interface, which gives the user the option to edit the quantity of workers per cluster.

Additionally, each workstation has a specific number of workers needed to perform that task. Although the majority of tasks only require one worker, there are a few that require two workers per task. These workers in the resource cluster act as constraints on the system, as parts cannot be processed without using workers. Lastly, it is of note that only processes inside the scope of the engine repair shop use workers; those that are treated as black boxes only have processing times.

Worker downtime, which was originally a variable of interest for CCAD, was changed to an input variable through further stakeholder analysis. This parameter can be set in the user interface and it indicates how often a worker is available as a percentage. This is because if worker downtime is modelled as an input, it is no longer measured as a metric of performance.

## 2.3 Develop Discrete-Event Model

The ProModel™ and ProSim™ were created as discrete-event models. The locations of these models were the workstations, which each have a designated processing time set by the collected data. The entities of these models are the T700 Engine components of each subassembly that is being sent through the system: T700 engine for the Level 0 initial disassembly, power turbine shaft for the power turbine subassembly, combustion liner for the hot subassembly, midframe diffuser for the cold subassembly, and the bearing ball cylinders for the accessory subassembly. These entities move through the discrete event model and follow the processing logics of each activity in the process flow. Resources are workers, who are used to process the entities through each workstation and they will be free to process other entities once the processing time for that activity is over. Lastly, arrivals are set as a user interface parameter, attributes and variables are set for collecting output data, and macros are assigned to design opening and closing of the PT shaft second line. All subassemblies are color schemed to show the location each activity takes place. Additionally, each subassembly is linked to the user interface for the processing times and resources required for each activities of each subassembly.

### 2.3.1 Model Choices

ProModel™ and ProSim™ were used to model the T700 engine repair process. The two models are managed by the ProModel™ Corporation. Both models use similar processes, but provide different perspectives towards the T700 engine repair process.

ProModel™ gives a visual simulation of the paths of activities and components off of the CCAD factory floor. ProSim™ provides an overview of the process flows of the activities and components of the T700 engine repair process.

ProSim™ is an additional modeling tool with the ProModel™ software. Similarly to ProModel™, it identifies the impact of interactions occurring in the system. ProSim™ is used to animate the process flow and also analyze input data. The main purpose of ProSim™ is that it enables the user to quickly achieve an optimized process for a realistic running model. Each process on the flow diagram becomes a series of activities which can be manipulated by different user inputs to optimize a model. The simulation is then run to identify any bottlenecks in the process.

ProSim™ and ProModel™ are used to identify any bottlenecks in the engine repair shop. Each activity will be linked to statistical distribution times and assigned worker resources. The processing times and resources are recorded in the user interface, which is then integrated to the simulations using arrays and attributes. The results of the simulation will influence necessary changes that will allow the model to meet CCAD's objectives and also provide outputs that will show the best option to optimize workshop efficiency.

### 2.3.2 User Interface

To provide easier access to adjusting parameters in the model, a Microsoft Excel sheet was built linking distributions and variables to the models. This enables stakeholders to alter the information in the simulation without directly adjusting the values in the model and potentially altering values they did not intend to. This increases the ease of use, as changing values in a spreadsheet is much more efficient than altering values in the model.

Both the ProSim™ and ProModel™ are manipulated by this user interface excel sheet. This user interface has a scenario tab, which allows the user to manipulate several different input parameters including opening and closing the second line and changing resources. There is also a Processing Times tab, which has all the processing times distributions for each tasks. The purpose of ProSim™ is to identify the bottlenecks in the subassemblies that affect the overall process flow.

### 2.3.3 Engine Assembly

The T700 Engine starts at the Level 0, which is the initial disassembly and inspection of the engine. The T700 engine is disassembled into the four subassemblies: power turbine Shaft (power turbine subassembly), combustion liner (hot subassembly), midframe and diffuser (cold subassembly), bearing ball cylinder (accessories subassembly). Repair activities for each subassembly were provided by the CCAD CAMs system and were all labeled as workstations in both models. All engines undergo the Level 0 process until they are broken up into the four individual subcomponents. At this point, they are routed through the subassemblies and undergo their individual processes until they are recombined into the Level 0 before exiting. Part integrity is maintained during this disassembly and assembly. For example, if Engine A is disassembled, all subcomponents will be reassembled back into Engine A once it is processed through the system. No subcomponents from Engine B are allowed to be used to reassemble. Engine Assembly contains nine steps, involves seven unique workstations, and also requires ten workers per step. This process takes an average of 60.87 hours to complete.

### 2.3.4 Power Turbine Shaft

The power turbine shaft is the component with the longest lead time and takes an average of 60.5 hours. This subassembly is the system bottleneck, as identified by CCAD in previous studies on the plating shop. The Power turbine goes through 27 processes, uses 6 separate workstations, and involves 27 workers for the baseline model. After cleaning, the turbine goes through the plating shop, which is modelled as a black box in the simulations. After cleaning, the turbine goes through several machining steps and priming shops intermittently. The component finally

goes through welding and priming before returning to the level 0 process. Activities have been added that have a second line, the use of the second line is implemented as a macro from the user interface. This second line was introduced to increase flow through the PT shaft, and it consists of seven independent workstations that can be opened or closed from the user interface.

### **2.3.5 Combustion Liner (Hot Subassembly)**

The combustion liner repair process takes an average time of 13.35 hours. The combustion liner repair process goes through 20 individual steps, uses 6 workstations, and uses 20 workers in the baseline model. The combustion liner goes through several sheet metal inspections and cleaning processes. Additionally, nondestructive testing and welding processes are conducted in subassembly.

### **2.3.6 Midframe Diffuser (Cold Subassembly)**

The midframe diffuser repair process takes an average of 9 hours. The midframe diffuser goes through 15 individual processes, utilizes 5 workstations, and requires 15 workers in the baseline. This part goes through cleaning before it is machined and metal sprayed. Then, it gets cleaned again for priming, returning to the Level 0 once fully processed.

### **2.3.7 Bearing Ball Cylinder (Accessory Subassembly)**

The bearing ball cylinder repair process takes an average time of 2.8 hours. The bearing ball cylinder undergoes 10 separate processes, utilizes 2 workstations, and uses 10 workers. After preshop assembly, the bearing ball cylinder finishes its remaining processes in the bearing shop. It is cleaned, machined, metal sprayed, machined again, cleaned, then primed for assembly and finally returned to the Level 0 process.

## **3. Verification and Validation**

### **3.1 Verification**

To verify the model, observations of the simulation and the data were used to determine feasibility and functionality. The data was spot checked to see if the values were reasonable and feasible for our models. This method provides insight into coding errors and corrections in the layout. These steps towards verification has made the model's process flow more efficient, which allows the model to produce more reliable results in accordance with the parameters. The “debug” and “display” statements were the methods of verifications that were used to create a reasonable output. Additionally, verifying the model's variables allows the user to see in real time if the output is realistic based on model parameters.

### **3.2 Validation**

The project was demonstrated to stakeholders in order to receive feedback on the accuracy of the model to the factory floor. Blueprints were provided to check the correct locations and interviews with floor workers provided validation to the model. Running the model and correlating the results with historical data also provided more insight into the ability of the model to reflect the real factory process. Lastly, the output metrics for throughput per week and time in system were compared to the expected values from CCAD historical data, which gave the best means for verification.

## **4. Results and Analysis**

### **4.1 Model Results**

Based on the results of the engine repair shop and the previous research of the plating shop, the plating shop

was the source of the largest bottleneck in CCAD operations. Scenarios doubling the capacity in the plating shop created the greatest increase in engines repaired and greatest decrease of the time engines take in the system. Creating more capacity or increasing efficiency in the plating shop would be the most beneficial solution. Thus the recommendation to the depot is to spend effort in improving the plating shop.

However, the plating shop is outside the scope of the engine repair shop operations; therefore, alternative scenarios need to be looked at to determine additional bottlenecks in the repair shop. The following 15 scenarios were modeled to give better alternatives for ways to improve engine repair shop bottlenecks:

- *Scenario 1: Base line:* This scenario is given as the ideal workshop as CCAD currently sees it.
- *Scenario 2: PT Second Line Open/Max Resources PT:* This scenario has the second line completely open on the PT shaft and maximum resources for all clusters associated with the PT shaft.
- *Scenario 3: PT Partially Open/More Resources PT:* This scenario has half the second line open: only the second line workstations before the plating shop. Additionally, a medium level of resources are added for the PT shaft cluster.
- *Scenario 4: Max Resources PT:* This scenario has maximum resources for all clusters associated with the PT shaft.
- *Scenario 5: PT Second Line Open/ALL Max Resources:* This scenario has the second line completely open on the PT shaft and maximum resources for all clusters.
- *Scenario 6: Maximum Engine Testing Capacity:* This scenario focuses on improving the engine testing capacity by adding more engine test cells and decreasing engine test cell downtime.
- *Scenario 7: Max Worker Productivity/Workshop Efficiency:* This scenario optimizes human labor and system efficiency. This includes measures such as adding more floor managers, replacing old machines with heavy downtime, modernizing the factory, increasing operations tempo, and setting more rewards or punishments for worker performance.
- *Scenario 8: Max Bearing Shop Efficiency:* This scenario minimizes the amount of time in the accessory subassembly by simulating additional stations in the ball bearing assemblies.
- *Scenario 9: Max Resources:* This scenario maximizes all workers within given constraints.
- *Scenario 10: Min Resources:* This scenario minimizes all workers within given constraints.
- *Scenario 11: Min Engine Testing Capacity:* This scenario focuses on minimizing the costs associated with the engine testing station by reducing engine test cells and increasing engine test cell downtime.
- *Scenario 12: Only PT Second Line Open:* This scenario only adds second line for the PT shaft.
- *Scenario 13: Mid resources, Higher Efficiency:* This scenario has average amount of resources and overall higher efficiency in the system.
- *Scenario 14: Lowest Cost:* This scenario sets all user parameters to the lowest cost option, acting as the left bound for all scenarios.
- *Scenario 15: Highest Cost:* This scenario sets all user parameters to the highest cost option, acting as the right bound for all scenarios.

## 4.2 Multi-Objective Decision Analysis Model

All 15 scenarios were run and results were collected using Output Viewer. In **Figure 2**, the results of the alternatives were recorded. For throughput per week, the greater the number, the better the value score. For time in system, the lower the number, the better the value. With these results, a multi-objective decision analysis model can be used to compare the scenarios by cost and value. Stakeholder analysis determined that the two criterias measured were throughput and time in system. The stakeholders valued throughput at 90 out of 100 and time in system at 70 out of 100. The best output scenarios for each individual variable were used as the standard to base off each scenarios performance. Below are the equations that determined the value of of each variable output for the simulation and the equation for the cost relative to one worker in the system.

$$\text{Scenario Value} = (\text{Throughput})/(\text{Max Throughput}) * (\text{Throughput Value})/(\text{Total Value}) +$$



$$(Time)/(Max Time) * (Time Value)/(Total Value)$$

Figure 2. Value Matrix Equation

	Raw Throughput	Throughput Data	Raw Time	Time Data	Total Value		Throughput Value	90
Scenario 1	8.28679613	0.430723674	162.4936	0.335975	0.7666983		Time Value	70
Scenario 2	8.157515282	0.42400403	168.2531	0.324474	0.7484779		Ideal Throughput	10.82207
Scenario 3	8.144569607	0.423331151	172.0857	0.317247	0.7405785		Ideal Time	124.7857
Scenario 4	8.251441416	0.428886038	157.7412	0.346097	0.7749828			
Scenario 5	8.157519717	0.42400426	168.2492	0.324481	0.7484857			
Scenario 6	8.286839451	0.430725926	162.419	0.336129	0.7668549			
Scenario 7	10.31020259	0.535894485	140.2129	0.389363	0.9252578			
Scenario 8	8.28679613	0.430723674	162.4936	0.335975	0.7666983			
Scenario 9	8.251451474	0.428886561	157.7248	0.346133	0.7750195			
Scenario 10	6.29577212	0.32723601	351.6922	0.155232	0.4824676			
Scenario 11	8.230050764	0.427774214	177.4854	0.307596	0.7353698			
Scenario 12	8.191775188	0.425784761	161.5298	0.337979	0.7637642			
Scenario 13	10.82207249	0.5625	135.4388	0.403088	0.965588			
Scenario 14	6.656855498	0.346004078	318.9732	0.171155	0.5171587			
Scenario 15	10.49243214	0.545366249	124.7857	0.4375	0.9828662			

Figure 3. The value matrix for the criteria.

After determining the overall values of each scenario, costs factors is then assigned to each scenario for comparisons. The cost of a scenario was determined based on the value it took to hire and employ one worker. Using this as the standard base, costs were determined for adding an extra workshop, increasing system efficiency, and for all of the user parameters. The scenarios then were plotted and compared in **Figure 3**. Scenarios that are dominated by others in cost and value are eliminated as viable options, which gives stakeholders a range of scenarios vary in different costs and value. The full cost equation is shown in **Figure 4**.

$$Cost = (10 * Lines Cost) + (1 * Cluster Cost 1) + (1 * Bearing) + (1 * Rate of Downtime) + (10 * Number of Bays) + (1 * Workshop Efficiency) + (1 * Worker Downtime)$$

Figure 4. Cost Equation

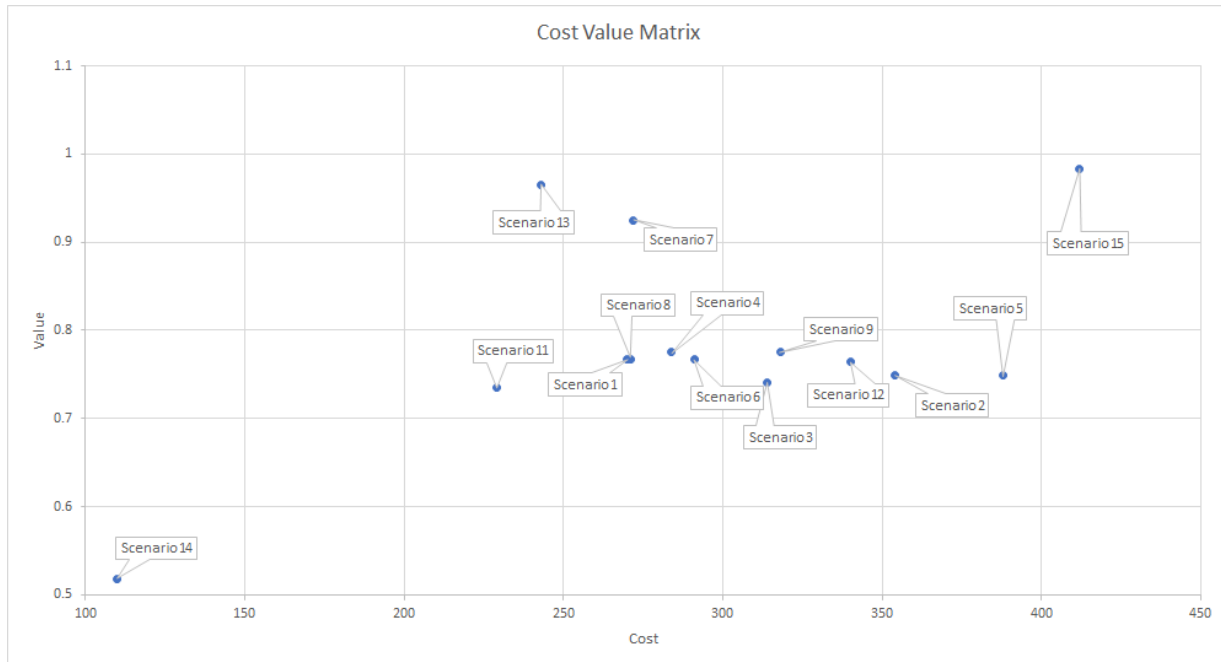


Figure 5. Cost Value Matrix

## 4.3 Results

**Figure 4** shows the dominated scenarios: costly scenarios that are surpassed in value by others. For example, Scenario 13 dominates Scenario 5 since it is more value and also less cost. Overall, the only non-dominated solutions are Scenario 14, Scenario 11, Scenario 13, and Scenario 15. Because Scenario 14 and Scenario 15 were designed to be the least and most costly scenarios, this makes sense. Because Scenario 14 and Scenario 11 are both lesser value than the baseline, Scenario 1, it is unlikely CCAD would accept these on the principle of saving cost. Thus, both Scenario 13 and Scenario 15 are left to choose from.

Figure 5. Value Chart for Throughput Per Week and Time In System

### 4.3.1 Recommendations

Despite Scenario 15 being higher value than Scenario 13, there is a large cost difference between the two for a very small return in value. This small difference in values can be seen better in **Figure 5**. Thus, Scenario 13 is the recommendation to CCAD, Mid resources, Higher Efficiency.

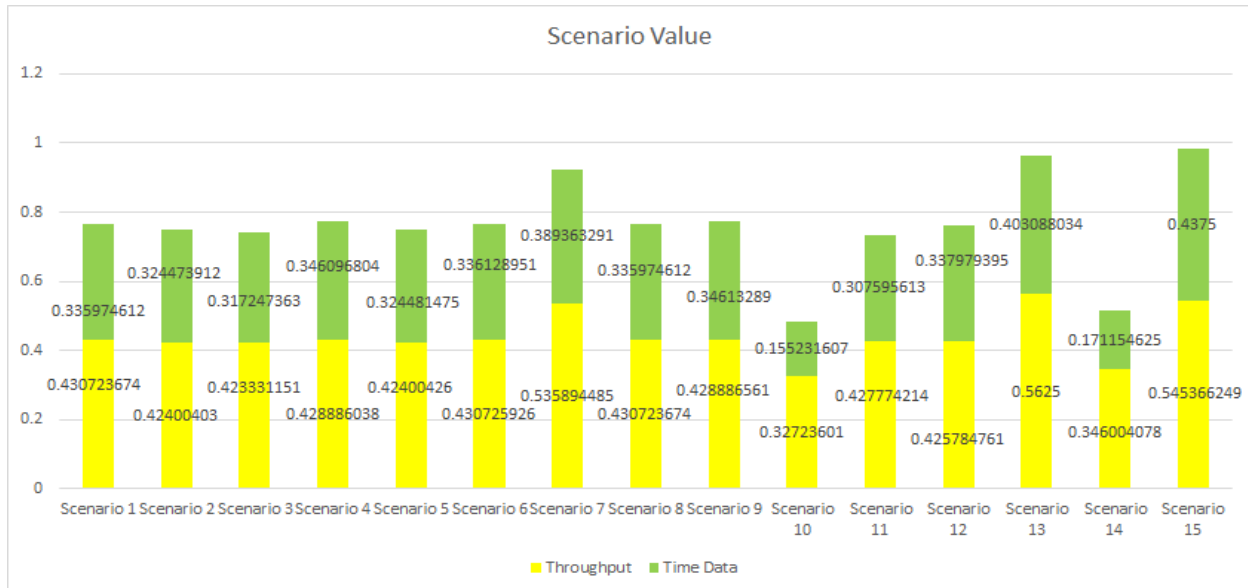
Once interesting trend in all the non-dominated solutions is that the only three scenarios broke a value score of over 80/100. In fact, all three of these scenarios were tied to an increase in workshop efficiency. Because increasing efficiency is not as tangible as adding workers or a workshop, Scenario 4: Maxing Resources in the PT Shaft, is recommended as a viable option if looking for tangible ways to improve value at little added cost.

To increase overall efficiency in the system and reduce worker downtime, the following measures ought to be considered. Firstly, CCAD should add more floor managers to increase the productivity of workers. The floor layout can be changed to reduce moving distance and old machines can be replaced to decrease processing time. Lastly, the operations tempo can be increased by creating a sense of urgency and/or setting more incentives. One of the interesting pieces of information that CCAD gave during the stakeholder analysis phase was that during wartime, throughput per week is dramatically increased. This signifies that if needed, the facility leadership and workers can

increase throughput, decrease time in system, and minimize worker downtime just through a sense of urgency.

### 4.3.2 Takeaways

Several takeaways can be discussed from this research. Firstly, it is of tremendous note that the plating shop represents the major bottleneck for the system. Even if resources are maxed in the PT shaft, only a slight improvement is observed. The majority of scenarios focus on trying to increase efficiency in the PT shaft, yet no matter how many



resources are added, the plating shop processing time prevents progress.

The only way to significantly improve value in the system is to increase workshop efficiency, as this method directly decreases all processing times by a factor. Because this is harder to represent in specific plans of action, it is recommended that CCAD invest more research in how to improve general workshop efficiency through changing the footprint of their shop. Because the engine repair shop will move to another facility within the next five years, this is the perfect time for CCAD to invest in how efficient footprints can cut down on time in system.

Lastly, in regard to opening a second line in the PT shaft, this was found to have no effect on increasing throughput per week or decreasing time in the system. Therefore, extra workstations are not necessary to add given the current constraints.

## 5. Conclusion

In conclusion, the research from the T700 Engine repair shop process has led to several valuable pieces of information for CCAD. They were provided with two adjustable models, one in ProModel™ and the other in ProcessSimulator™, which has a user interface that can simulate scenarios for the T700 engine repair process. By doing these simulations, the different scenario output on both throughput and time in the system can be observed. Through running 15 different scenarios, the most effective scenarios are those that increase general workshop efficiency. Also, efficiency can be increased by adding more workers to those clusters which directly work on the PT shaft subassembly. A second line to PT shaft should not be added, as this has no significant effect on increasing value. Using these simulations, CCAD can explore more options towards improving the T700 engine repair process for any future potential challenges.

## 6. References

Anand, R. (2013). Cloud Computing OEE (Overall Equipment Effectiveness) for Reducing Production Downtime. *SAE International Journal of Materials and Manufacturing* 6(3), 481-86. <http://www.jstor.org/stable/26268544>.

Bessant, J., Caffyn, S. (1997). High-involvement innovation through continuous improvement. *International Journal of Technology Management*. 14(1), 7-28.

Delimarski, P., Griffith, J., Howard, M., & McBryde, S. (2017). Discrete Event Simulation and Analysis of New ACCF. Technical Report prepared for the Corpus Christi Army Depot.

Heravi, G., and Mohammad, F. (2017). Production process improvement of buildings' prefabricated steel frames using value stream mapping. *International Journal Of Advanced Manufacturing Technology* 89, ( 9-12) 3307-3321. *Academic Search Premier*, EBSCOhost (accessed September 6, 2018).

History of CCAD. Accessed February, 2019. <http://www.ccad.army.mil/history.html>.

Hubbard, D, W. (2014). *How to Measure Anything Finding the Value of Intangibles in Business*. Hoboken: Wiley.

Kleijnen, J. P. C., Smits M. T. (2003). Performance Metrics in Supply Chain Management. *The Journal of the Operational Research Society* 54(5), 507-14. <http://www.jstor.org/stable/4101738>.

# DEEP LEARNING BASED MALWARE CLASSIFICATION USING DEEP RESIDUAL NETWORK

Yan Lu

Jonathan Graham

Jiang Li

Dept. of MSVE  
Old Dominion University  
5115 Hampton Blvd  
Norfolk, VA, USA  
Yxxlu003@odu.edu

Dept. of CS  
Norfolk State University  
700 Park Ave  
Norfolk, VA, USA  
Jmgram@nsu.edu

Dept. of ECE  
Old Dominion University  
5115 Hampton Blvd  
Norfolk, VA, USA  
JLi@odu.edu

## ABSTRACT

The traditional malware detection approaches rely heavily on feature extraction procedure, in this paper we proposed a deep learning-based malware classification model by using a 18-layers deep residual network. Our model uses the raw bytecodes data of malware samples, converting the bytecodes to 3-channel RGB images and then applying the deep learning techniques to classify the malwares. Our experiment results show that the deep residual network model achieved an average accuracy of 86.54% by 5-fold cross validation. Comparing to the traditional methods for malware classification, our deep residual network model greatly simplify the malware detection and classification procedures, it achieved a very good classification accuracy as well. The dataset we used in this paper for training and testing is Maling dataset, one of the biggest malware datasets released by vision research lab of UCSB.

**Keywords:** Malware Classification, Deep Learning, Deep Residual Network.

## 1 INTRODUCTION

Malicious software (malware) had been growing exponentially in the past years, based on the purpose of the malwares, the malwares could be divided into different categories, such as adware, spyware, virus, worm, trojan, and ransomware etc. The traditional ways of detecting malwares are mostly heuristic and signature-based methods. There are two types of traditional malware analysis methods: the static methods and dynamic methods. The static methods which extract the malware features from the static malware bytecodes, such as processor instructions, null terminated strings and library imports; the dynamic methods extract the features while the code is being executed, collecting information how the executed codes interact with the operating system such as system API calls or interactions with the other OSs and the network.

The feature extracting procedures in the traditional malware detection approaches could be time consuming and also rely heavily on domain expert knowledge, it also needs special tools and software environment and could be computation resource consuming as well. Besides, the traditional malware detection methods do not adapt well when the malwares are modified polymorphically or metamorphically by purpose to hide the real code when evolve or propagate. The most common ways that the polymorphic or metamorphic approaches use to hide the code are encrypting and decrypting the raw code for multiple times to mutate the original code, changing the registers to use, or inserting non-operation instructions etc.

Machine learning especially deep learning models achieved significant success in image classification and computer vision fields in the past years, the deep models' performance had exceeded human's performance on some dataset such as MNIST dataset. To take advantage of the high performance of the deep learning

model in object detection and classification, in this paper we proposed a deep learning-based malware classification scheme which converts the malware into 3-channel RGB images and then uses a 18-layer deep residual network to classify and detect the malwares. Our experimental results show that the deep residual network model does not only greatly simplify the malware classification procedure comparing to the traditional methods, but also achieves a very good classification accuracy on the Maling dataset.

The remainder of the paper will be organized as follows. At first, in the methods section it will provide an overview of our method how to convert malware samples to images. Next, the deep residual network model structure we used will be introduced. In the next following section, our experiment result by applying the deep residual network model on the Maling dataset will be discussed. In the last section, we conclude our work in this paper for future work.

## 2 METHODS

### 2.1 Malware as Images

The malwares in the same malware family appear very similar in layout and texture, the new malwares with modifications in the same malware family have the visual similarity with the existing ones as well. As Fig 1. Shown, the variants from the FYI malware family and Dialplatform malware family, the visual dissimilarity in one malware family are very minimal.

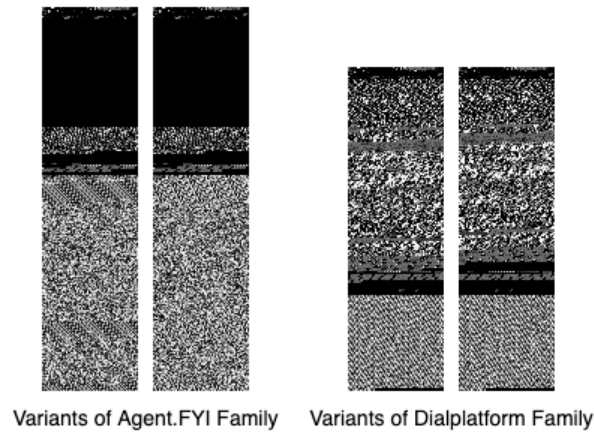


Fig 1. Variants of FYI Malware Family and Dialplatform Malware Family

To convert the malwares to images, we firstly convert the malware binaries to 8-bit vectors (bytecodes). After, we then convert the bytecodes into grayscale images with value ranged from 0 to 255, that is, each vector is converted to a pixel with value ranged from 0 to 255. In the next step we then convert the grayscale images into 3-channel RGB images by duplicating the grayscale channel for three times and then concatenate all of the three channels to form A RGB image. The procedure as Fig 2. shown.

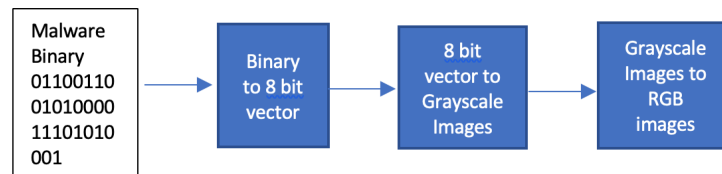


Fig 2. Converting Malware Binary Codes to RGB images

## 2.2 Deep Residual Network

Many complex visual recognition and image classification tasks show benefit from very deep models. However, when the deep model went deep, it is getting harder to train the model and the model's performance starts to decline, the degradation problem is starting to be exposed as well. To solve this problem, Kaiming He etc. proposed the deep residual learning framework, the deep residual network is originated from deep convolutional neural network, it uses identity mapping for shortcut connections between the input and weighted layer in the deep convolutional network. The deep residual network has led to a series of breakthroughs for image classification tasks, it won the ImageNet ILSVRC 2015 classification task, it achieved the state-of-the-art performances in many other computer vision and classification tasks as well, such as winning the 1<sup>st</sup> places in ImageNet localization, COCO detection, COCO segmentation tasks in ILSVRC and COCO 2015 competition. The strong evidence shows that the deep residual framework is generic and highly promising in computer vision problems.

In our proposed method we use an 18-layers deep residual net as the malware classifier. The architecture of the deep residual network is as shown in Fig 4. It consists 17 convolutional layers with filters in different sizes. Following each convolutional layer, there are a normalization layer (function), a rectified linear unit layer (function), and a pooling layer (function). At the last layer there is a fully connection layer and SoftMax layer (function) as the classifier. To avoid overfitting problem, drop out layers are added in between of some convolutional layers and normalization layers as Fig.3 shown.



Fig 3. The Architecture of 18-layers Deep Residual Network

## 2.3 Malimg Dataset

The dataset we used in our experiment is the Malimg Dataset, this dataset comprised 25 malware families with different number of samples in each class, the details are as shown in the Table 1. The largest family of in Malimg Dataset is the Allapple.A family, it consists 2949 malware samples; the smallest malware family in Malimg Dataset is the Skintrim.N malware family, it consists 80 malware samples. The samples in Malimg dataset are all converted to grayscale images with a size of 32 by 32 originally. We then furtherly convert the samples into 3-channel RGB images, the size of each sample in each malware family is 32 by 32 by 3 channels ultimately.

Table 1. Maling Walmare Dataset

No.	Family	Family Name	No of Variants
1.	Worm	Allaple.L	1591
2.	Worm	Allaple.A	2949
3.	Worm	Yuner.A	800
4.	PWS	Lolyda.AA 1	213
5.	PWS	Lolyda.AA 2	184
6.	PWS	Lolyda.AA 3	123
7.	Trojan	C2Lop.P	146
8.	Trojan	C2Lop.gen!G	200
9.	Dialer	Instantaccess	431
10.	Trojan Downloader	Swizzor.gen!l	132
11.	Trojan Downloader	Swizzor.gen!E	128
12.	Worm	VB.AT	408
13.	Rogue	Fakerean	381
14.	Trojan	Alueron.gen!J	198
15.	Trojan	Malex.gen!J	136
16.	PWS	Lolyda.AT	159
17.	Dialer	Adialer.C	125
18.	Trojan Downloader	Wintrim.BX	97
19.	Dialer	Dialplatform.B	177
20.	Trojan Downloader	Dontovo.A	162
21.	Trojan Downloader	Obfuscator.AD	142
22.	Backdoor	Agent.FYI	116
23.	Worm:AutoIT	Autorun.K	106
24.	Backdoor	Rbot!gen	158
25.	Trojan	Skintrim.N	80

## 2.4 Model Training

As Table 1. shown, the Maling dataset is extremely unbalanced. To test our model, we balanced out the dataset at first : we take the first 30 samples of each malware family as the testing data. If the sample size of the malware family is more than 200, we randomly select 130 samples in that family as the training data. If the sample size is smaller than 200 in a malware family, we chose all data samples left in that family as



the training data. We use a smaller batch size of 16 when training the residual network. The training is carried on GPU Tesla V100, the time for each training epoch is about 15 seconds.

### 3 RESULTS

#### 3.1 5-Fold Cross Validation

We train the 18-layers deep residual network using the datasets described in last section with 5-fold cross validation. After 50 epochs of model training for each fold of the data, we got a mean of classification validation accuracy of 86.54%.

#### 3.2 Apply the Trained Model to Testing Data

Then we applied the trained model on the unseen testing data, we use precision, recall and f1-score as metrics to evaluate the results. The way to calculate them are as the following formulas shown.

The precision talks about how precise/accurate of the deep residual network model, the recall calculates how many of the actual positives the model catches by labeling it as positive. F1-score function is a function of precision and recall, the f1-score is generally used to seek of balance between precision and recall when there is a uneven class distribution.

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$f1 - score = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

After we applied the trained model on the testing data, we got an average of testing accuracy of 84%. The Precisions, Recalls and f1-Scores of the classification results of each classes are as Table 2 shown.

Table 2. Malware Classification Result by Deep Residual Network

	precision	recall	f1-score	support
Allaple.L	1.00	1.00	1.00	30
Allaple.A	1.00	1.00	1.00	30
Yuner.A	0.54	0.63	0.58	30
Lolyda.AA 1	0.68	0.87	0.76	30
Lolyda.AA 2	0.94	1.00	0.97	30
Lolyda.AA 3	1.00	1.00	1.00	30
C2Lop.P	0.42	0.60	0.49	30
C2Lop.gen!G	0.55	0.20	0.29	30
Instantaccess	1.00	0.93	0.97	30
Swizzor.gen!I	1.00	1.00	1.00	30
Swizzor.gen!E	0.86	1.00	0.92	30
VB.AT	1.00	1.00	1.00	30
Fakerean	0.88	0.97	0.92	30
Alueron.gen!J	0.97	0.97	0.97	30
Malex.gen!J	0.97	1.00	0.98	30

Lolyda.AT	0.93	0.90	0.92	30
Adialer.C	0.67	0.87	0.75	30
Wintrim.BX	1.00	1.00	1.00	30
Dialplatform.B	0.94	0.97	0.95	30
Dontovo.A	1.00	0.97	0.98	30
Obfuscator.AD	0.44	0.37	0.40	30
Agent.FYI	0.37	0.23	0.29	30
Autorun.K	0.97	0.97	0.97	30
Rbot!gen	0.91	0.67	0.77	30
Skintrim.N	1.00	1.00	1.00	30
micro avg	0.84	0.84	0.84	750
macro avg	0.84	0.84	0.84	750
weighted avg	0.84	0.84	0.84	750
samples avg	0.84	0.84	0.84	750

#### 4 CONCLUSION

Our experiment results show that the deep residual network performed very well on the Maling malware dataset for the malware classification. By the 5-fold cross validation, the average classification validation accuracy achieves 86%. By converting the malwares raw bytecodes into images, and then applying to the deep learning model, our proposed deep learning based malware classification methods significantly reduced the domain expert knowledge needed in the malware detection and classification tasks, the deep learning model makes it a lot easier for anyone without the domain knowledge to perform the malwares classification tasks. Besides, our proposed method also significantly reduced the computation or software resources needed in the malware detection and classification tasks, greatly simplified the procedure.

However, from the classification report, we could see that the precision and recall scores of some classes such as malware family of Agent.FYI or Yuner are a lot lower than the other classes. The possible reasons of the lower scores of these classes might be caused by the unbalanced training data issue among different classes, or the small training sample issue within that class when training the model.

In the future, we will furtherly investigate in improving the performance our proposed deep residual model and solving the small data or unbalanced data issue. GAN (Generative Adversarial Network) is a possible solution. Besides, further tuning of the hyper-parameters of the deep residual network model will possibly improve the performance of the model as well.

#### REFERENCES

- Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, 2015, “Deep Residual Learning for Image Recognition”, arXiv preprint arXiv:1512.03385.
- L.Nataraj, S.Karthikeyan. 2011. “Malware images: visualization and automatic classification”, Proceeding of the 8<sup>th</sup> International Symposium on Visualization for Cyber Security Article No. 4, Pittsburgh, Pennsylvania, USA, July 20-21,2011.
- Quan Le, Oisín Boydell, Brian Mac Namee, 2018, “Deep Learning at the shallow end: Malware classification for non-domain expert”, Digital Investigation 26 (2018) S118-S126.
- Daniel Gilbert, 2016, “Convolutional Neural Network for Malware Classification” , Master thesis, Department of Computer Science, University of Barcelona.

# MSVSCC 2019

## PROCEEDINGS



### Business, Industry & Transportation

Page 133	<b>Stocks Selection Using Machine Learning</b> Md Reshad Hoque and Jiang Li
Page 135	<b>An Agent Based Reinforcement Learning Approach for Dynamic Taxi Ride Sharing</b> Zhenyu Wang, Hong Yang and Qingyu Ma
Page 137	<b>A Multi-user Online Driving Simulator with Virtual Reality and Microscopic Traffic Simulation</b> Mahmud Hasan, Daniel Perez, Yuzhong Shen and Hong Yang

# STOCKS SELECTION USING MACHINE LEARNING

Md. Reshad Ul Hoque, Jiang Li  
Department of Electrical & Computer Engineering  
Old Dominion University  
Norfolk, Virginia, USA  
E-mail address: [mhoqu001@odu.edu](mailto:mhoqu001@odu.edu), [JLi@odu.edu](mailto:JLi@odu.edu)

## ABSTRACT

In the world of finance, stock market analysis is the heart of the effective strategy development scheme for stock exchange transactions. Stock price fluctuations and uncertainties are the biggest challenges for stock analysis. Fundamental and technical are the two approaches for stock analysis where fundamental analysis is performed using sentimental analysis using social media data whereas technical analysis is done using stock features. In this paper, we performed technical analysis using machine learning in order to select the best stock. We developed lasso-regression, genetic algorithm (GA) and found our genetic algorithm based model perform well for stock selection.

**Keywords:** stock, regression, genetic-algorithm, lasso regression.

## 1 INTRODUCTION

Stock market selection is an enticing prospect to machine learning technologist and data scientist. Though there are lots of challenges, but it is possible due to machine learning techniques. In the stock market, price always ups and down. For a certain time in a week or month, the price goes up or down. So, there must be some kind of patterns of price change, from which the machine learning model can learn that pattern and make a prediction of future prices to select good stocks. Based on the selection, people can invest money on those particular stocks and can save themselves from loss.

There are a lot of research on stock market prediction but very few research on stock selection. This paper addresses the stock market data analysis challenges with machine learning approaches and selects good stocks from thousands of stock.

## 2 METHODS

### 2.1 REGRESSION

Regression is a statistical model from which the relationships among the dependent variables and independent variables can be measured. Lasso regression based on linear regression is one the robust regression model for the financial market behavior prediction [1]. Lasso uses L1 regularization and select operation ( $\alpha$ ) for optimization. When  $\alpha$  equal to 0, it simply consider as a linear regression model,  $\alpha$  equals to  $\infty$  forces all coefficients to zero but  $0 < \alpha < \infty$  is the optimal value for alpha.

### 2.2 GENETIC ALGORITHM

Genetic algorithm (GA) is similar to biological evolution which is a good alternative for stock selection. Population, selection, crossover, and mutation are the main steps of GA. We used the Sharpe ratio as the fitness function. At first, GA uses random weights as initialization, but after iterations, weights are updated using mutation and crossover. Higher weight value stocks treated as good stock.

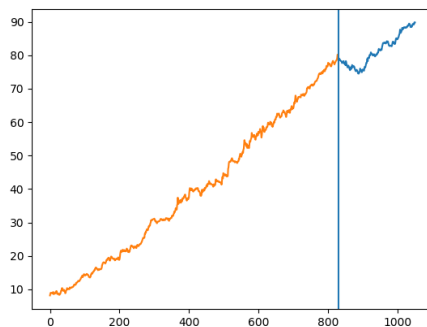
### 3 DATASETS AND PREPROCESSING

We used Chinese stock data set which are in excel files. All are time series data ranging from 2013 to 2018. Data-set contains data, strategy Id which is the stock identification number, Pnl (profit and loss statement), commission (profit bonus), slippage (unexpected price change).

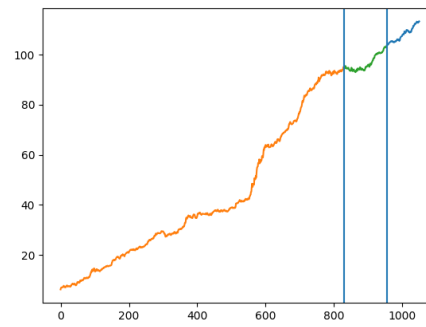
For stock analysis, we need to do some preprocessing on Pnl. First, we calculated percentage gain of profit-loss statement using:  $(\text{Pnl} / \text{total investment}) * 100$ ; total investment is 1 billion dollars. After that, we accumulated the percentage gain of Pnl and calculated the Sharpe ratio.

### 4 RESULT AND DISCUSSION

After preprocessing, we selected the top 20 based on the Sharpe ratio. For training, we used data from 2014 to 2017, May and rest for testing. But for GA, we used the next six months for validation and rest for testing. For fitness function, we used the Sharpe ratio, and for evolution, we select the top 5 from the population of validation dataset and select one which have the highest Sharpe ratio. For the fitness score, we divided 2014-2017 training data into twelve parts. Then measured the Sharpe ratio of each of the region. Earliest parts are multiplied by  $1/2$ ,  $1/4$ ,  $1/8$ , so on and add them all. This strategy capture the recent trend of stock data.



Lasso Regression



Genetic algorithm

Figure 1: Models output

Figure 1 showing the model output for lasso regression and genetic algorithm. Lasso regression's predicted return for 11 months is 9.469 whereas GA algorithm's predicted return for 5 months is 9.36 which indicates that GA captures good stocks. If the optimized weights are negatively correlated or zero, then we consider those stocks as bad stock. Higher value means better stocks. Experimentally, we found that GA captures more good stocks than Lasso and that's why GA 5 months predicted return is very close to LASSO 11 months predicted returns.

### 5 CONCLUSION

Our developed genetic algorithm based stock selection model shows promising result to select good stocks. In future, we want to tune our GA algorithm and will explore the deep learning realm for stock selection.

### REFERENCES

[1] Roy, Sanjiban Sekhar, et al. "Stock market forecasting using LASSO linear regression model." Afro-European Conference for Industrial Advancement. Springer, Cham, 2015.

# AN AGENT BASED REINFORCEMENT LEARNING APPROACH FOR DYNAMIC TAXI RIDE SHARING

Zhenyu Wang, Hong Yang, Qingyu Ma  
Department of Modulation, Simulation, Visualization and Engineering  
Old Dominion University  
4700 Elkhorn Ave, Norfolk, VA 23529, USA  
E-mail address: zwang002@odu.edu

## ABSTRACT

The ridesharing economy is experiencing rapid growth and innovation. Companies such as Uber and Lyft are continuing to grow at a considerable pace. However, many challenges remain in dynamic taxi ridesharing services, and are not currently under wide consideration. In this paper, an agent-based reinforcement learning approach is developed to address dynamic ride service. An unity based taxi simulation platform is further developed to evaluate the performance of proposed approach. The result on metrics such as total miles traveled by taxis and average waiting time of riders prove the efficiency of proposed approach compared with conventional greedy approaches.

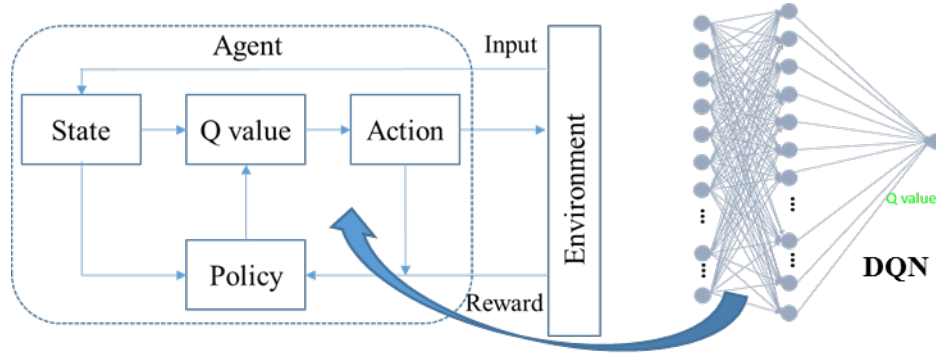
**Keywords:** Reinforcement Learning, Dynamic Ride Sharing, Taxi, Simulation

## 1 INTRODUCTION

Real-time ridesharing refers to the task of helping to arrange shared rides. Such a technique has been embedded in popular platforms including Uber, Lyft, and DiDi Chuxing, and significantly transformed the way people travel. Given the redundancy of individual trajectories in both space and time dimensions, rider sharing can be promising to take advantage of such redundancy, and offers more efficiency on traffic management. Over the years, the research community has made great efforts to investigate conventional approaches such as linear regression in static taxi ride sharing management. However, such static approaches cannot address large-scale real-time ridesharing systems, and there still lacks an efficient approach to dynamically dispatch orders, i.e., to assign orders to a set of active drivers automatically (Fagnant and Kockelman 2018). Since the quality of order dispatching will directly affect the utility of transportation capacity, the amount of service income, and the level of customer satisfaction, therefore, solving the problem of order dispatching is the key to any successful ride-sharing platform. In this paper, our goal is to develop a reinforcement learning based approach to maximize the system performance with the ability to scale up to a large number of drivers and robust to potential hardware or connectivity failures.

## 2 METHODOLOGY

Configuration of a deep reinforcement learning model requires specification of three important aspects: the state representation, the reward function, the model structure, and the action space definition. As shown in Figure 1, the state of an agent is taxi's occupancy (vacant, 1 passenger occupied, 2 passengers occupied), its distance to other potential riders. The policy is learnt via Q network to determine whether or not to pick up a rider. Actions and environments are what a taxi will do and its surrounding riders.



**Figure 1. Framework of Reinforcement Learning.**

### 3 RESULTS AND DISCUSSIONS

Sioux Fall city network is used as one hour case study for simulation and verification. In total, 500 vehicles are simulated with a rider request of 3,631. The simulation step is one minute, for comparison, the conventional greedy search algorithm (GH) is used for comparison (Santos and Xavier 2013). As shown in Table 1, the average travel time of RL is longer compared with that of GH method. This is due to each rider does not select the shortest path with only one passenger. Meanwhile, total travel time of RL is less compared with that of GH algorithm. In addition, the average travel time in minute of RL algorithm is relatively longer compared with GH due to the driving distance between different passengers.

**Table 1. Comparison of RL and GH Performance**

Approach	Total time (Akermi <i>et al.</i> )	Total distance (mile)	Total wait time (min)
RL	124.5	4829.5	2736
GH	173.2	6898.9	4824
	Average time (min)	Average distance (mile)	Average wait time (min)
RL	4.05	2.7	0.76
GH	3.15	2.1	1.34

The agent based reinforcement learning approach enables taxi agents to learn the nonlinear relationships between surrounding potential riders and optimal policy to achieve a low waiting time for riders and small total travel distance. The simulation results prove its efficiency compared with GH method. Other than the simple network with Sioux Fall city, we plan to further simulate taxi ride sharing in New York city as a case study with real data, and examine the performance of our proposed approach.

### REFERENCES

- Fagnant, D.J., Kockelman, K.M., 2018. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in austin, texas. *Transportation* 45 (1), 143-158.
- Santos, D.O., Xavier, E.C., Year. Dynamic taxi and ridesharing: A framework and heuristics for the optimization problem. In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*.

# A MULTI-USER ONLINE DRIVING SIMULATOR WITH VIRTUAL REALITY AND MICROSCOPIC TRAFFIC SIMULATION

Mahmud Hasan, Daniel Perez, Yuzhong Shen, Hong Yang  
Modeling, Simulation and Visualization Engineering  
Old Dominion University  
1300 ECSB, Norfolk, VA, USA  
E-mail: mhasa002@odu.edu

## ABSTRACT

Driving simulators (DS) are systems consisting of driving wheels with pedals and shifters, having one or more monitors as the display where the user drives a virtual vehicle in a virtual environment. Most DS systems are single user and have low fidelity. This paper presents a framework and a prototype for online multi-user driving simulator with virtual reality and microscopic traffic simulation to alleviate the limitations of general DS systems.

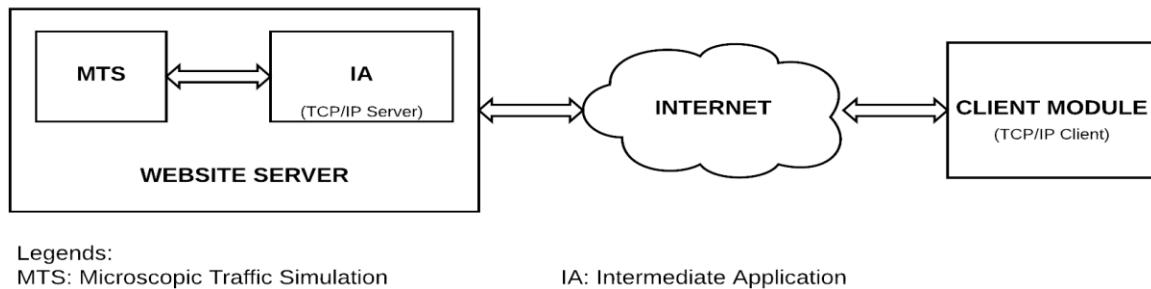
**Keywords:** Driving Simulator, Virtual Reality, Multi-user, Microscopic Traffic Simulation.

## 1 INTRODUCTION

Driving simulators are widely used to train novice drivers, perform research studies, assess a traffic situation, perform safety measures, collect data and so on. They offer many advantages, e.g., DS environments have high controllability and reproducibility, data collection is easy, experiments can be performed without physically endangering a participant. They have some limitations as well like (a) most DSs are single user so studies with multiple user interacting with each other is not possible, (b) the virtual environment seen on display(s) is not highly realistic, i.e., low-fidelity. Our proposed system presented in this paper completely addresses issues (a) and (b). In this paper, we created a framework where online multiplayer architecture solves issue (a) while incorporation of VR headsets makes the user-experience more immersive and realistic thus solving issue (b).

## 2 SYSTEM ARCHITECTURE

The framework consists of several integral components as shown in **Error! Reference source not found..** One important requirement of this framework is that the user needs an active internet connection. The arrow signs show the transfer of data.



**Figure 1. Component Architecture.**



## 2.1 Client Module

The Client module (CM) is the application that the user uses to visualize and interact with the virtual environment. This application can be executed w/o VR headsets and w/o driving wheels. The user uses the computer browser to access the server website.

## 2.2 Website Server

In this framework, this server hosts the webpage, stores the data, hosts the Intermediate Application (IA) and the Microscopic Traffic Simulation (MTS) application.

### 2.2.1 Microscopic Traffic Simulation (MTS) Package:

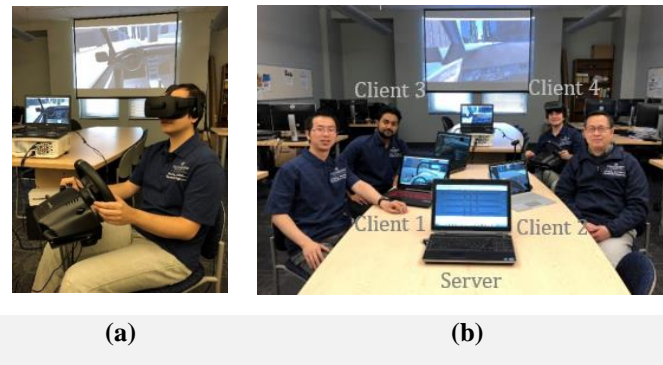
MTS is the emulation of real-world traffic movements in a virtual world with various traffic entities. By using the MTS API, information of the traffic entities like vehicles, pedestrians, traffic signals etc. can be accessed.

### 2.2.2 Intermediate Application:

Intermediate Application (IA) is the executable application that has the MTS API plugin codes. This IA is used to exchange information to and from the MTS. The IA is also used as the TCP/IP server to which multiple client modules connect and sends/ receives information.

## 3 IMPLEMENTATION

A prototype has been developed based on the above-mentioned framework. The CM who works as the TCP/IP clients was created with Unity3D, the game development environment. The MTS package used was Vissim, a commercial traffic simulation software. The IA was written in C++ because of the Vissim API. XAMPP, which uses Apache servers and MySQL databases was used as the server. For testing purposes, the website server is hosted in one of our laptop computers where IP tunneling tools were used to go around the firewalls and make that computer IP visible to the outside world.



**Figure 2. System Deployment Demonstration: (a) Driving Simulator with the VR Headset; and (b) Wirelessly Connected Users in the Same Environment.**

## 4 CONCLUSION

The prototype developed proves the feasibility of the framework. By incorporating virtual reality and online multi-user capability in the driving simulator, this framework has opened the door to more immersive, accurate and realistic user study. The limitation of this framework would be dependent on the internet speed. So, far five simultaneous connections were tested without any significant delay or loss of data and with further optimization, this framework can support a lot more.