

## Work Experience

<b>Sensor Software Engineer</b>	<b>MORAI</b>	<b>Oct. 2021- Dec. 2024</b>
Virtual Sensor / Map & Test Automation	Seoul, Rep Korea	
<ul style="list-style-type: none"><li>Developed a <b>shader-based radar simulation</b> using shader programming (.usf) in Unreal Engine<ul style="list-style-type: none"><li>utilized low-level radar parameters; the number of chirps, samples, receive antenna, and antenna pattern to simulate radar returns based on those parameters; created Python script to process radar data, received low-level output data from ROS, and generated range-doppler maps.</li></ul></li><li>Created the <b>Bounding Box Labeler</b> with the following features:<ul style="list-style-type: none"><li>implemented JSON file output for all object types: vehicle, vehicle parts, pedestrians with animation, and obstacles.</li><li>supported multiple coordinate systems for use cases (Camera, LiDAR, vehicle, and ENU), and developed 8-corner and center point-based representations.</li></ul></li><li><b>Developed Coordinate Converter Plugin for Unreal Engine 5:</b><ul style="list-style-type: none"><li>implemented bidirectional transformation between Unreal Engine's left-handed and right-handed systems (NED, ENU, AER) with unit tests, enabling cross-team compatibility and project integration.</li></ul></li><li><b>Developed a Comprehensive Scenario Runner Application</b> from scratch, implementing the ASAM OpenSCENARIO standards:<ul style="list-style-type: none"><li>created functionality to load, edit, and save OpenSCENARIO files, adhering to the standard's defined elements and attributes, and implemented a user interface for scenario editing and batch simulation management.</li><li>implemented various OpenSCENARIO actions, including TrafficSpawnAction and custom PedestrianSpawnAction.</li><li>integrated gRPC protocol for communication with the simulator; developed an adaptor class to handle responses/requests, with the runner client performing all conditions and action evaluations for improved maintainability.</li><li>optimize collision detection using an Oriented Bounding Box (OBB) and Separating Axis Theorem.</li><li>architected scalable batch simulation functionality with Python APIs for simulation controls (start, skip, stop, load map, simulation time)</li><li>ported existing functionality of Scenario Runner to Unreal Engine 5; migrated core features and system from previous Python scripts to Unreal Engine 5.</li></ul></li></ul>		
<b>Graduate Teaching Assistant</b>	<b>University of Missouri St. Louis</b>	<b>2019-2021</b>
	St. Louis, USA	
<ul style="list-style-type: none"><li>Developed and delivered hands-on C++ programming labs covering OOP, memory management, STL, and graph algorithms.</li></ul>		
<b>Teaching Assistant</b>	<b>Washington University in St. Louis</b>	<b>2017-2019</b>
	St. Louis, USA	
<ul style="list-style-type: none"><li>Provided supplemental educational services for undergraduate students studying Signal &amp; System / Engineering Mathematics Course.</li></ul>		

## Education

- |                                                                       |                  |
|-----------------------------------------------------------------------|------------------|
| • <b>M.Sc. Computer Science</b>                                       | <b>2019–2021</b> |
| University of Missouri - St. Louis. GPA: 4.0                          |                  |
| • <b>B.Sc. Electrical &amp; Electronic Engineering</b>                | <b>2017–2019</b> |
| University of Missouri / Washington University in St. Louis. GPA: 3.8 |                  |

## Technologies and Languages

- Languages: C, C++, C#, Java, Python, JavaScript
- Technologies: Git, OpenCV, Open3D, CUDA, DirectX11, DirectX12, Vulkan, HLSL, Unreal Engine, Unity, PyTorch, TensorFlow, PyQt, PyTest, Jenkins, GitLab CI/CD
- Other: Data structures and algorithms, Computer Vision, Deep Learning, Computer Graphics, ROS

## Projects

---

### **Hyundai UAM: Radar Point Cloud Visualization in Unreal Engine 5.**

- applied Gaussian Random Distribution to generate realistic radar returns and utilized the raycasting method to capture aerial objects exclusively, improving point cloud fidelity and target isolation.

### **Samsung Data Generation:**

#### **Scenario Data Acquisition**

- utilized **DataGen** to build complex scenario datasets, ensuring 15+ objects per batch and detailed descriptions of dynamic situations. Managed video acquisition and detail modifications to enhance data quality and alignment with project requirements.

### **Enhanced LiDAR Simulation with Motion Distortion**

- developed a post-processing technique to apply motion distortion to MORAI Simulator's LiDAR output, implementing coordinate transformation and spherical linear interpolation(SLERP) for rotation modeling, and integrated distortion effects into the pipeline.

### **Project ROKA:**

- implemented a flexible architecture for Scenario Runner, incorporating a default mapping system for offline loading vehicles, pedestrians, and miscellaneous objects, while maintaining compatibility with existing RestAPI-based retrieval, ensuring operation across security environments.

### **Project NIA:**

- developed, modified, and tested 1200 Edge Case Scenarios across 6 Korean maps in .xosc format, ensuring dataset accuracy and alignment with project requirements.  
- conducted simulations, identified necessary modifications, and updated to improve scenario quality and reliability

### **Root Detection Algorithm for Plant Phenotyping**

- Developed an advanced C++ algorithm using OpenCV for automatic plant root detection in 3D reconstructed images. The solution implements a multi-stage pipeline including image pre-processing, background subtraction, container detection, and root segmentation. Key techniques utilized include Canny Edge Detection, contour filtering, and morphological operations for effective noise reduction and feature extraction, resulting in accurate root system boundary detection.

## Others

---

- **Volunteer work**

**Brain Dynamics and Control Research Group**, Electrical Engineering Department, Washington University in St. Louis.

- Developed a portable Brain-Computer Interface (BCI) system using Raspberry PI3; built a portable device for anesthesia monitoring, storing real-time EEG data, participated in Conic Method research; implemented data preprocessing pipeline for noise reduction and artifact removal.

**3D Point Cloud Data Processing**, Spatial AI KR

- I presented computer graphics-related topics, focusing on the book '3D Point Cloud Analysis: Traditional, Deep Learning, and Explainable Machine Learning Methods'. Provided a brief explanation of related Differential Geometry concepts on YouTube.

- **Patents**

- Eungback Kim, Seungho Jang, Seongyeon Park, Hoseup Lee, Hein Jo, 2024. SCENARIO-BASED AUTONOMOUS DRIVING VEHICLE SIMULATION METHOD AND SYSTEM. WIPO Patent WO2024/117564, filed November 1, 2023, and published June, 6, 2024 Patent Approved & Registered.

- Heecheol Yoo, Seungho Jang, Hojun Lim, 2024. ELECTRONIC DEVICE AND METHOD FOR PROCESSING POINT CLOUD DATA, KR 10-2024-0076717, filed June 24, 2024. Patent Approved.

- **Awards**

- Sweeney Memorial Scholarship, Issued by Washington University in St. Louis, Engineering Department.
- Robert Heider Engineering Scholarship, Issued by Washington University in St. Louis, Engineering Department.

- **Certifications and Courses**

- Graduate Certificate in Artificial Intelligence - University of Missouri in Saint Louis / NSA / DHS National Center
- Robotic Software Engineering – Udacity
- Computer Vision – Udacity
- Introduction to Computer Graphics with DirectX 11 – Part 2. Realtime Pipeline
- AWS Machine Learning – Udacity
- Python for Computer Vision with OpenCV and Deep Learning – Udemy