SAFER Pre-studies

Final Report

Reference	CP57
Project Title	Generating Synthetic Scenarios to Test an Al-Enabled Traffic Measurement System
Coordinator	RISE Research Institutes of Sweden
Project Manager	Markus Borg (markus.borg@ri.se)
Project Duration	2022-04-01 – 2022-08-31

Contents

Summary

Traffic is the sum of road design, road user behavior, and their interactions. Understanding the dynamics is central to making the transport system efficient, safe, and sustainable. Viscando offers AI-enabled data-driven solutions for traffic analysis, safety diagnostics, intelligent traffic control, naturalistic data collection, and extended perception at the very core of smart cities and autonomous driving. This is enabled by the infrastructure sensor OTUS3D, which is used for collection of accurate traffic movement data.

Software quality assurance (QA) must evolve as systems increasingly rely on AI. Building on initial work on simulation-based testing, we set up this pre-study to pave the way for evolutionary algorithms to generate test scenarios in CARLA that stress OTUS3D to its limits. We developed and published a digital model of an intersection in Lindholmen under an open-source software license. The model is available in two formats to allow importing into MathWorks RoadRunner and the open-source simulator CARLA. We also provide a set of trajectories for road agents.

Our project prepares for future search-based software testing of OTUS3D. Based on the trajectories and the environmental conditions, we propose key parameters to configure traffic scenario execution in CARLA. Furthermore, we share source code for test scenario generation using the NSGA-II algorithm. We present intermediate results as a proof-of-concept in this report. In the next two months, our work will continue and the final results will be published in a MSc thesis at Lund University.

Generating Synthetic Scenarios to Test an AI-Enabled Traffic Measurement System

1. Background

Traffic is complex! It is the sum of road design, road user behavior, and their interactions. Understanding this dynamic is central to making the transport system efficient, safe, and sustainable — in line with the UN Sustainable Development Goals 11 and 3.

Viscando offers AI-enabled data-driven solutions for traffic analysis, safety diagnostics, intelligent traffic control, naturalistic data collection, and extended perception at the very core of smart cities and autonomous driving. This is enabled by Viscando's proprietary AI- and stereovision-based infrastructure sensor OTUS3D, which is used for collection of accurate traffic movement data. The OTUS3D sensor relies on deep learning for object recognition and classification in traffic. Computer vision is used to extract trajectories for road users, e.g., cars, vans, pedestrians, and bicyclists.

Software quality assurance (QA) must be treated differently when system features rely on deep learning. No longer is all logic expressed in source code instructions, instead hundreds of millions of parameters are trained on huge datasets – thus, standard QA turns less effective. Simulation-based testing is often proposed to identify critical inputs that lead to system faults. Contemporary photorealistic virtual environments enable efficient, effective, and safe testing. Still, how to best create fault-revealing traffic scenarios is an open research question. Several research studies suggest using search-based software testing to find effective test scenarios in the huge space of possible inputs, i.e., relying on evolutionary approaches such as genetic algorithms. This pre-study developed a corresponding proof-of-concept.

2. Project set up

This section presents the purpose, objectives, project period, and project partners.

2.3 Purpose

In this project, we set out to create a digital model of an intersection in Lindholmen to allow simulation-based testing of OTUS3D. Testing in simulators allows advanced test generation techniques to be applied to stress OTUS3D to its limits, i.e., to explore the robustness limits of the system and guide future improvement work.

Using the open-source simulator CARLA, we aimed at using evolutionary algorithms to generate test scenarios. The project aimed at supporting the development of robust traffic measurement systems for smart city and autonomous driving applications. Moreover, the testing results could also guide subsequent training data collection campaigns – or possibly even complement training sets with synthetic imagery.

2.4 Objectives

In the application, we listed six expected results. For each item, we summarize the current status.

1) Digital model of an intersection (Lindholmen and/or Kölliken, Schweiz)

   During the hearing meeting, the SAFER board expressed a preference for the Lindholmen intersection. This project provides a model of Lindholmsallén/Lindholmspiren (lat 57.7086727, long 11.9395434) which we have tested in two contemporary simulators.

2) Parametrized scenario execution in CARLA

   We provide a set of trajectories for road agents in the intersection. Based on these, we selected the following parameters to control traffic scenarios: 1) number of road agents following the pre-defined trajectories, 2) starting time for each road agent, 3) target velocities, 4) car colors (RGB), and 5) several weather parameters.

3) Proof-of-concept implementation of evolutionary test case generation

   The source code is publicly available on GitHub.

4) A set of videos containing critical (synthetic) traffic scenarios

   Four stereo video sequences are available on GitHub.

5) A joint academic publication

   Work on the MSc thesis report is ongoing. The introductory sections are in a mature shape, but we are still waiting for additional results. Cross-organizational experimental work got delayed during the vacation period (July-August 2022). Whether to submit a manuscript for peer-review at an academic event remains to be decided.

6) Embryo of a joint application for a Vinnova project

   Important networking between RISE, Lund University, and Viscando has taken part during the project. If the right call opens, a joint application is possible.

2.5 Project period

2022-04-01 – 2022-08-31

2.6 Partners

- Humanized Autonomy, Mobility and Systems, RISE Research Institutes of Sweden
- Viscando AB
- Department of Computer Science, Lund University

3. Method and activities

This section presents the work done in the pre-study.

3.1 Creation of the digital model

The digital model is based on a real-life intersection, one where Viscando had already gathered data. This intersection is located in Lindholmen, Gothenburg (lat 57.7086727, long 11.9395434). The intersection is shown in Figure 1. The model was created using the MathWorks package RoadRunner. RoadRunner is an interactive editor built with the purpose of creating 3D models of traffic environments. Roads can easily be created and customized and there exist a large number of blueprints of 3D models such as signs, guardrails, trees and buildings that can be inserted into the environment.

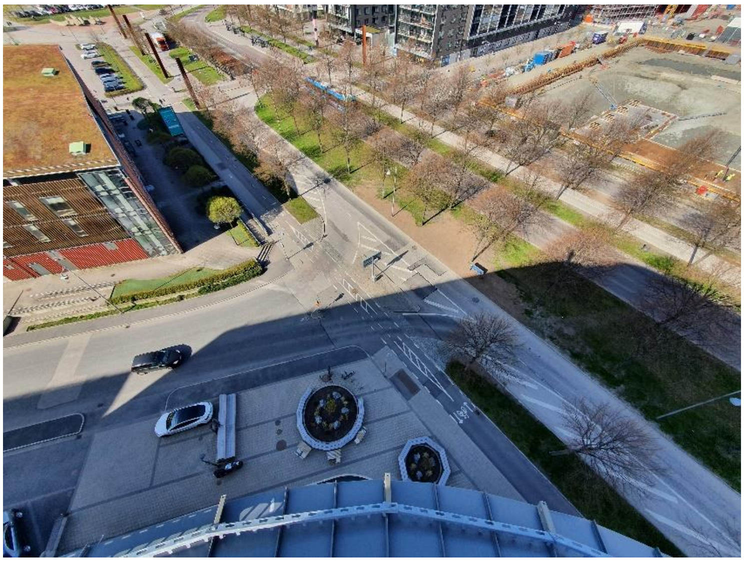

Figure 1: A photograph taken above the intersection in Lindholmen. Note the distinct shadow from the building in the morning sun (April 29th, 9:45 am).

When exporting the scene, the data was stored in one xodr file and one fbx file. The xodr extension indicates the OpenDRIVE file format. OpenDRIVE files describe road network data: the geometry of roads, lanes and markings as well as features along the road like stop signals are saved in such files. The fbx file contains 3D geometry

and animation data: in the case of our scene, it contained information about 3D objects like trees, bushes and street lights. With the data contained in the xodr and fbx files, importing a 3D scene into CARLA (a version built from source) was easily done by using methods that are provided with the installation. An image of the digital model built in RoadRunner is shown in Figure 2.

Figure 2: The digital model of the Lindholmen intersection created in RoadRunner.

After the scene was imported into CARLA, some of the 3D assets had either disappeared or did not look particularly visually pleasing. This is shown in Figure 3. This problem was fixed by using the Unreal Engine editor to add blueprints developed by the CARLA team. The end result can be seen in Figure 4.

Figure 3: The digital model of the Lindholmen intersection after it was imported into CARLA. Note that some of the signs present in Figure 2 are missing. Some of the textures, particularly the leaves and the pavement material, also do not look particularly realistic.

Figure 4: The improved digital model of the Lindholmen intersection after it was imported into CARLA. Compared to Figure 3, the tree and bush textures look more realistic, some ground material has been changed and some signs have been added. The building to the left in the image has also been re-scaled.

3.2 Traffic scenario simulations in CARLA

The traffic scenarios were simulated using CARLA. Built upon Unreal Engine 4, CARLA provides high-end rendering quality and realistic physics. The CARLA server runs the simulation while receiving commands from the client. The client API is implemented in Python so that the server, and by extension the entire simulation, can be controlled through the execution of Python scripts. CARLA provides blueprint models of different types of vehicles (including bicycles), walkers and sensors that can be used to gather simulation data. Moreover, the weather can be customized through the tuning of different parameters.

3.3 Implementation of the evolutionary algorithm

The implementation of NSGA-II that was used in this project is based on the Python evolutionary computation framework DEAP. All parts of the algorithm, including the computation of the hypervolume used for the evaluation, was implemented using DEAP. The main part that we had to implement ourselves was the calculation of fitness of the individuals.

3.4 Creation of trajectories

When defining a traffic scenario, the paths that vehicles and walkers follow, i.e., their trajectories, are of great importance. In this project, all trajectories were kept constant over the course of the experimental testing that was made. In other words, a number of trajectories were manually created before any experiments were made, and these trajectories were not changed during or in between the experiments. A trajectory was defined as a vector of points in 2D space that specified the path to follow for the actor (vehicle or walker). Depending on the length of the trajectory, between 150 and 350 2D points were used. Suitable points were found with the help of data that had already been collected from the Lindholmen intersection by

Viscando. This data contained two pieces of information that were of interest to us: the estimated positions of passing vehicles and walkers in the form of 2D points and the corresponding estimated speeds of these objects, with a time interval of 0.05 between each data point. The different actors partaking in the simulation were set to follow the extracted trajectories and the corresponding speed estimations.

While simulating a scenario, images that could be used as input to the OTUS3D system needed to be captured. This was done by placing RGB cameras at the two spots in the simulated environment that corresponded to the placements of the two cameras used by the OTUS3D sensor in the real Lindholmen intersection. These cameras were set to capture 20 images per second, since this was an appropriate rate for OTUS3D to receive image input. Figure 5 shows an example of a captured simulated image that was used as input to OTUS3D.

Figure 5: A synthetic image from the simulation that was used as test input for OTUS3D.

3.5 Experimental setup

We parameterized the traffic scenario execution as follows:

- The starting time for each actor participating in the simulation. It ranged from a minimum of 0 seconds to a maximum of 10 seconds.
- The color of each vehicle. It consisted of a triplet of integers denoting the RGB-value.

- The level of cloudiness in the simulation, described by a float value between 0 and 100.
- The sun angle. It was set so that the sun altitude angle mimicked the position of the sun in the real Lindholmen intersection during early August.

These parameters were chosen partly because they were believed to affect the scenarios in ways relevant to the purpose of testing OTUS3D, and partly to keep the implementation of the scenarios as simple as possible.

In order to investigate to what extent the genetic algorithm NSGA-II created critical traffic scenarios, we developed two different models: one that used NSGA-II to create scenario parameters and another baseline model that sampled scenario parameters from a uniform random distribution. Moreover, the two models were tested on two different sets of trajectories. The parameter values were used to simulate traffic scenarios, which in turn resulted in a set of images that were used as input to OTUS3D. OTUS3D in turn produced a set of 2D-points and estimated speed values at specific time points that could be compared to the ground truth that was obtained when running the simulation. A population size of 20 was used for both of the two sets of trajectories. The final analysis has not been completed as part of this pre-study, but we share intermediate results in the next section.

4. Results and Deliverables

The major contributions of the project is the digital model of the intersection, parameterized scenario execution, and a customized NSGA-II implementation. The implementation includes the definition of a fitness function that combines positional error, speed error, and misclassifications. We computed average errors for the first NSGA-II generation and the baseline. Unfortunately, we did not have the time to evolve subsequent generations of traffic scenario parameters during the pre-study - this will be done during October in collaboration with Viscando.

Table 1 shows the average errors of the output from OTUS3D when using synthetic scenarios as input. The errors are relative to the ground truth values provided by CARLA. We report them for three road agent categories, i.e., cars, bicycles, and pedestrians. The generated scenarios correspond to the first generation of parameter values from NSGA-II and the baseline. Mean positional error and mean speed error were compared against the ground truth. The fraction of misclassifications shows the percentage of incorrect road agent classifications. We conclude that the constituents of the fitness function appear promising for future work on search-based software testing.

Table 1: Average errors when testing OTUS3D on synthetic scenarios.

	Cars	Bicycles	Pedestrians
	Mean positional error (m)		
NSGA-II	2.86	1.93	1.84
Baseline	2.91	1.96	1.80
	Mean speed error (m/s)		
NSGA-II	1.00	0.43	1.84
Baseline	1.03	0.48	0.33
	Fraction of misclassifications (%)		
NSGA-II	11.9	11.6	39.7
Baseline	11.5	18.8	35.0

The prestudy resulted in three main deliverables that can be reused by others:

1) A digital model of the intersection in Lindholmen
2) A corresponding implementation of NSGA-II for traffic scenario generation
3) Four (stereo) video sequences that demonstrate synthetic traffic scenarios

5. Conclusions, Lessons Learnt and Next Steps

The prestudy initiated work on search-based software testing of OTUS3D. Several papers demonstrate that evolutionary approaches can be used to efficiently and effectively generate fault-revealing traffic scenarios for automated vehicles. Our work seeks to generalize the method to support testing of AI-based systems the smart city domain.

While we are not ready yet, we are positive that the approach can be successfully applied to the case under study. Two major challenges are often the generation of a representative digital model in the simulator and the design of a fitness function. We believe we have provided both as part of the pre-study. In the next steps, we will run iteratively run NSGA-II to evolve additional traffic scenarios. The goal will be to stress OTUS3D to its limits and guide future hardening cycles, i.e., engineering efforts to further improve the system. For a feature that relies on supervised deep learning, this might include additional data collection, amplification of existing training data or further ML model development.

More specifically, our next to milestones related to this project are:

1. Conclusion of the MSc thesis project at Lund University. This activity will resul in a publicly available report and improved source code and examples on GitHub.
2. Presentation of results at a SAFER Thursday seminar on November 10, 2022.

6. Dissemination and Publications

All source code and the digital model of the Lindholmen intersection is publicly available on GitHub: https://github.com/EliasSjoberg/rise-viscando-thesis

The digital model is available in two formats to allow importing into both CARLA and RoadRunner. The source code contains a proof-of-concept implementation of the NSGA-II algorithm for traffic scenario generation. NSGA-II (Non-dominated Sorting Genetic Algorithm) is a commonly used evolutionary algorithm that efficiently and effectively searches an input space.

An elaborated report will be published as a MSc thesis at Lund University. If the final results are promising, we plan to prepare a manuscript for submission to an academic venue for external peer-review.

7. Acknowledgement

Our thanks go to MathWorks for providing a trial version of RoadRunner.