LEOVISTA: A Cesium-Based 3D Visualization Tool for Satellite-Enabled Vehicle Simulations

Darinela Andronovici*, Giuseppe Avino†, Mario Franke‡, Damien Nicolas†, Ion Turcanu†, Christoph Sommer‡
*University of Luxembourg
†Luxembourg Institute of Science and Technology (LIST), Luxembourg
‡TU Dresden, Faculty of Computer Science, Germany
darinela.andronovici.001@student.uni.lu
{ giuseppe.avino, damien.nicolas, ion.turcanu }@list.lu
https://www.cms-labs.org/people/ { franke , sommer }

***Abstract*—We present LEOVISTA, a Cesium-based 3D visualization tool for post-simulation analysis of Vehicle-to-Satellite (V2S) communications. Unlike existing solutions, LEOVISTA offers high-fidelity, geospatially accurate rendering of vehicles, urban environments, and Low Earth Orbit (LEO) satellite constellations. The proposed solution allows users to interactively explore how satellite coverage and environmental factors impact Connected and Automated Vehicle (CAV) connectivity. LEOVISTA supports playback control, parameter inspection, and side-by-side scenario comparison, making it a powerful tool for both visualization and debugging of complex CAV simulations.**

I. Introduction

Simulating Connected and Automated Vehicles (CAVs) has become increasingly complex, reflecting the growing intricacy of real-world deployment scenarios. Modern simulation setups often involve a combination of road traffic simulators, network simulators, and additional modules to capture the interactions between vehicles, pedestrians, cyclists, and even satellite systems. As CAV technology evolves, so does the need to account for the surrounding environment – not just as a passive backdrop, but as an active participant that can significantly affect performance. Elements such as buildings, traffic lights, foliage, and other physical obstructions can impact both vehicle sensor perception and communication links, including direct inter-vehicle communication as well as infrastructure-assisted terrestrial and non-terrestrial links, such as those relying on Low Earth Orbit (LEO) satellite constellations. These expanded simulation requirements result in a fragmented and intricate toolchain, where tracing and understanding system behavior becomes increasingly difficult.

Widely used open-source simulators like OMNeT++ and ns-3 offer limited native support for advanced visualization. OMNeT++ – which supports vehicle communications through frameworks like Veins [1] and space_Veins [2] – provides a graphical runtime environment based on its native Qtenv GUI. It allows for real-time 2D/3D visualization of network node mobility, network topologies, connections, and message exchanges. Yet, it only allows visualizing the current simulation state of an executing simulation – without the ability to rewind, e.g., to backtrack from a failed handover to the moment when the handover was initiated, visualizing the state of the network and its environment at that time.

In contrast, ns-3 – with its modules for simulating connected vehicle scenarios such as VaN3Twin [3] – does not provide any native GUI for 2D or 3D visualization during simulation runtime. Instead, it fully relies on external tools like NetAnim, which offers post-simulation 2D animation, or PyViz, which supports limited online 2D and 3D visualization. More fully-featured post-simulation 3D visualization in ns-3 can be enabled with NetSimulyzer [4], an open-source rendering tool specifically designed for integration with ns-3. NetSimulyzer allows users to navigate through simulation history, visualize scenarios in 3D, and choose which metrics and statistics of interest to display. However, this tool is tied to the ns-3 simulation environment and does not natively support the rendering of high-fidelity urban environments – such as buildings, roads and traffic infrastructure – and it has limited extensibility and data integration with external sources such as orbital parameters or high-resolution 3D city models.

To fill this gap, in this demo paper we propose LEOVISTA,[1] an open-source, post-simulation visualization tool able to support realistic, scalable, and geospatially accurate 3D visualization of complex mobility and communication scenarios. LEOVISTA is based on Cesium[2] – a powerful, open-source JavaScript library for 3D geospatial visualization in a web browser. The proposed solution enables scalable, high-fidelity rendering of vehicles, road infrastructure, and natural elements in a georeferenced 3D environment, while also supporting dynamic visualization of LEO satellites based on real orbital parameters. This integration of network and environmental context offers a more comprehensive and intuitive understanding of the factors affecting CAV performance.

II. Architecture and Visualization Features

Figure 1 depicts the complete toolchain illustrating all the necessary steps to visualize a post-simulation result with our proposed LEOVISTA tool. A simulation framework that is suitable for producing the necessary input information for LEOVISTA is space_Veins, which supports both direct Vehicle-to-Vehicle (V2V) and Vehicle-to-Satellite (V2S) communication.

[1]Full source code available from https://www.project-leone.eu/
[2]Cesium website at https://cesium.com/.

DS-RT 2025 – Demos

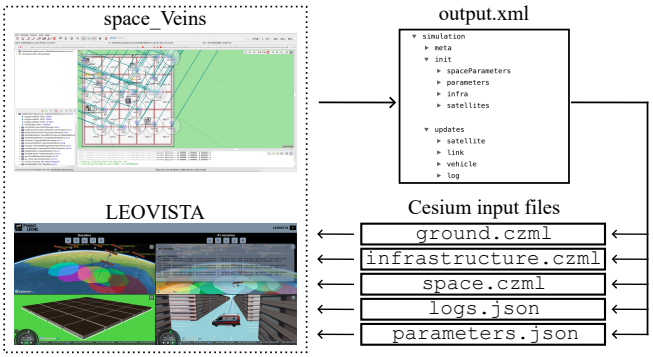

Figure 1. Reference architecture of LEOVISTA.

We extended space_Veins with a custom module that outputs an XML file containing the wide range of information required by LEOVISTA. This information includes: (i) environmental data, such as the position and height of buildings, (ii) ground mobility data, such as vehicle kinematics, (iii) satellite mobility data, including their orbital dynamics, and (iv) network-related data, such as link availability.

This output file from space_Veins is processed by a conversion tool that we developed and which includes a set of Python-based scripts. The tool converts the XML file into three CZML files – the specific format supported by Cesium – and two JSON files. Specifically, the CZML files are `ground.czml`, `infrastructure.czml`, and `space.czml`, which provide information about the ground node mobility, environmental infrastructure, and satellite orbital dynamics, respectively. The two JSON files, `logs.json` and `parameter.json`, store network-related metrics, such as the number of established satellite connections and received packets for each ground node, as well as specific simulation parameters (e.g., the satellite antenna beamwidth that determines the coverage area).

These five files are used by LEOVISTA to visualize post-simulation results. An overview of the LEOVISTA GUI – including key features such as toggling satellite links, enabling full-screen mode, accessing log or parameter menus, and controlling playback – is shown in Figure 2. LEOVISTA supports two main visualization modes: *ground view*, which lets users observe the simulation from the perspective of vehicles and their surrounding environment, and *space view*, which displays the simulated satellites and their orbital movements. Users can choose to view either mode independently or display both simultaneously using a horizontally split screen. The log menu provides access to all network-related events recorded in the `logs.json` file, while the parameter menu displays simulation settings extracted from the `parameters.json` file. A dedicated playback widget allows users to navigate the simulation timeline, supporting forward and reverse playback as well as adjustable playback speed. As illustrated in Figure 2, LEOVISTA also supports side-by-side comparisons between different simulation scenarios, with a one-click feature for temporal synchronization between the two simulations.

In the shown use case, vehicles are connected via direct LEO satellite links, making satellite coverage imperative

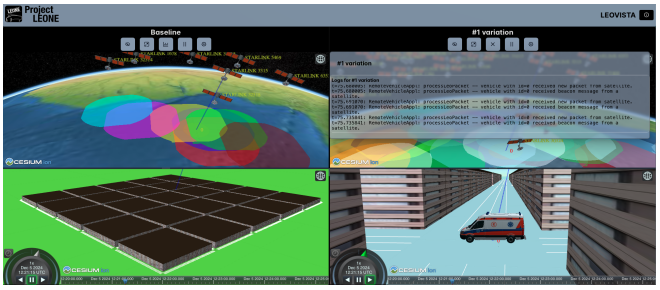

Figure 2. Web interface of the LEOVISTA visualization tool.

for maintaining connectivity. LEOVISTA visualizes this by displaying a colored link between a vehicle and the connected satellite. If a vehicle moves out of coverage without a successful handover, or if line-of-sight is obstructed by a building, the connectivity status – recorded in the `logs.json` file – is updated, and the visual link disappears, clearly indicating a loss of connection. When connectivity is restored, the link reappears, providing an intuitive visual cue that the connection has been reestablished.

III. Demo Description

This demo showcases a vehicle teleoperation use case enabled by LEO satellite communications. The simulated scenario is a 1 km x 1 km Manhattan Grid (5 x 5 blocks), within which vehicles are in motion. During the demo, participants can interact with LEOVISTA and explore its full range of features. In particular, users can visually examine how key parameters, such as satellite coverage, impact V2S communication performance. This interactive experience highlights LEOVISTA's capabilities both as a high-fidelity 3D visualization tool and as a powerful debugging aid, especially valuable in complex simulation scenarios where identifying the root cause of unexpected behaviors or communication failures can be challenging.

Acknowledgments

This research was funded in whole, or in part, by the Luxembourg National Research Fund (FNR), grant reference DEFENCE22/IS/17800623/LEONE. For the purpose of open access, and in fulfillment of the obligations arising from the grant agreement, the authors have applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

References

[1] C. Sommer, D. Eckhoff, A. Brummer, D. S. Buse, F. Hagenauer, S. Joerer, and M. Segata, "Veins – the open source vehicular network simulation framework," in *Recent Advances in Network Simulation*, A. Virdis and M. Kirsche, Eds., Springer, 2019, pp. 215–252.

[2] M. Franke and C. Sommer, "Toward Space-Air-Ground Integrated Network Simulation with 4D Topologies," in *19th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2024)*, Chamonix, France: IEEE, Jan. 2024, pp. 61–68.

[3] F. Raviglione, C. R. Carletti, M. Malinverno, C. Casetti, and C. Chiasserini, "ms-van3t: An integrated multi-stack framework for virtual validation of V2X communication and services," *Elsevier Computer Communications*, vol. 217, pp. 70–86, Mar. 2024.

[4] E. Black, S. Gamboa, and R. Rouil, "NetSimulyzer: a 3D network simulation analyzer for ns-3," in *Proceedings of the Workshop on ns-3*, Virtual Event: ACM, Jun. 2021, pp. 65–72.