Sensor Fusion Desynchronization Attacks (Artifact)

Andreas Finkenzeller
School of Computation, Information and Technology, Technical University of Munich, Germany

Andrew Roberts
FinEst Centre for Smart Cities, Tallinn University of Technology, Estonia

Mauro Bellone
FinEst Centre for Smart Cities, Tallinn University of Technology, Estonia

Olaf Maennel
School of Computer and Mathematical Sciences, The University of Adelaide, Australia

Mohammad Hamad
School of Computation, Information and Technology, Technical University of Munich, Germany

Sebastian Steinhorst
School of Computation, Information and Technology, Technical University of Munich, Germany

Abstract

Environmental perception and 3D object detection are key factors for advancing autonomous driving and require robust security measures to ensure optimal performance and safety. However, established methods often focus only on protecting the involved data and overlook synchronization and timing aspects, which are equally crucial for ensuring profound system security. For instance, multi-modal sensor fusion techniques for object detection can be affected by input desynchronization resulting from random communication delays or malicious cyber attacks, as these techniques combine various sensor inputs to extract shared features present in their data streams simultaneously. Current research acknowledges the importance of temporal alignment in this context. However, the presented studies typically assume genuine system behavior and neglect the potential threat of malicious attacks, as the suggested solutions lack strategies to prevent intentional data misalignment. Additionally, they do not adequately address how sensor input desynchronization affects fusion performance in depth. This paper investigates how desynchronization attacks impact sensor fusion algorithms for 3D object detection. We evaluate how varying sensor delays affect the detection performance and link our findings to the internal architecture of the sensor fusion algorithms and the influence of specific traffic scenarios and their dynamics. We compiled four datasets covering typical traffic scenarios for our empirical evaluation and tested them on four representative fusion algorithms. Our results show that all evaluated algorithms are vulnerable to input desynchronization, as the performance declines with increasing sensor delays, highlighting the existing lack of resilience to desynchronization attacks. Furthermore, we observe that the Light Detection and Ranging (LiDAR) sensor is significantly more susceptible to delays than the camera. Finally, our experiments indicate that the chosen fusion architecture correlates with the system's resilience against desynchronization, as our results demonstrate that the early fusion approach provides greater robustness than others.

2012 ACM Subject Classification Security and privacy → Systems security; Hardware → Safety critical systems

Keywords and phrases Security, Time Synchronization, Sensor Fusion, Autonomous Driving, Delay Attack

Digital Object Identifier 10.4230/DARTS.11.1.1

Funding The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the programme of “Souverän. Digital. Vernetzt.”. Joint project 6G-life, project identification number: 16KISK002 and the European Union-funded project CyberSecDome (Agreement No.: 101120779).

Dagstuhl Artifacts Series, Vol. 11, Issue 1, Artifact No. 1, pp. 1:1–1:3
DAGSTUHL ARTIFACTS SERIES
Dagstuhl Artifacts Series
Schloss Dagstuhl – Leibniz-Zentrum für Informatik,
Dagstuhl Publishing, Germany

Related Article Andreas Finkenzeller, Andrew Roberts, Mauro Bellone, Olaf Maennel, Mohammad Hamad, and Sebastian Steinhorst, "Sensor Fusion Desynchronization Attacks", in 37th Euromicro Conference on Real-Time Systems (ECRTS 2025), LIPIcs, Vol. 335, pp. 6:1–6:22, 2025.
https://doi.org/10.4230/LIPIcs.ECRTS.2025.6

Related Conference 37th Euromicro Conference on Real-Time Systems (ECRTS 2025), July 8–11, 2025, Brussels, Belgium

1 Scope

This document introduces the artifact for our related research paper, *"Sensor Fusion Desynchronization Attacks"* [1].

2 Content

The artifact repository is structured into several subfolders that will be further explained in the following:

- **code:** The *code* folder contains the Docker containers and bash scripts to run the sensor fusion algorithms and produce the raw results. The relevant files for each algorithm are bundled in a subfolder (*clocs*, *epnet*, *frustum-convnet*, *vir-conv*), which contains a *Dockerfile* for the container description and a *run_evaluation.sh* script to start the experiments. All containers can be started very conveniently at once with the accompanying *docker-compose.yaml* file.
- **datasets:** This folder contains the four compiled datasets (*scenario1*, *scenario2*, *scenario3*, *scenario4*). The *training* folder follows the KITTI dataset format and contains the raw sensor data, calibration info, and ground truth data. The *testing* folder is a simple copy of the *training* folder. Its only purpose is to avoid `FileNotFound` errors for algorithms that also expect a test dataset. Note that the data in this folder is not used for the evaluation and consequently does not impact the final results, nor does its existence. Also note that we do not train the ML model with our dataset but only use it for evaluation, so the wording "training" and "testing" could be misleading here. The *ImageSets* folder contains the index lists to select which data samples shall be considered for the evaluation. While the index files will be changed dynamically during algorithm execution, the default lists remain always available in *ImageSets/default* for initialization. *scnearioX_dynamics.png* depicts the scene dynamics and can be generated with the *scripts/create_plots.py* script. *scenarioX_video.mp4* shows the continuous camera stream for each scenario and can be generated with the *scripts/generate_video.py* script.
- **final_results:** This folder contains the final evaluation results as presented in the research paper. The results for each algorithm are stored in individual subfolders (*clocs*, *epnet*, *f-convnet*, *vir-conv*). If these folders are not shown, you first have to generate the results with the *scripts/evaluate_performance.py* script. The *latex* folder contains boilerplate code to create a minimal PDF file containing the result tables as shown in the paper.
- **results:** This folder contains the intermediary raw results from the algorithm evaluation. It is mapped as a volume into the Docker containers to easily export the result data.
- **scripts:** This folder contains several Python scripts:
  - *create_plots.py*: Create the scenario dynamics plots as shown in the paper.
  - *generate_video.py*: Generate a video from the continuous camera stream of a scenario.
  - *evaluate_performance.py*: Generate the final evaluation results, i.e., the raw data for the evaluation tables included in the paper.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the artifact is also available at: https://osf.io/x2yrq/files/osfstorage?view_only=f148ad01c4104699b214e76d3cad7388.

4 Tested platforms

This artifact has been tested on our evaluation platform with the following system specifications:

- **CPU:** Intel Xeon Gold 5220R (24 Cores)
- **GPU:** PNY NVIDIA RTX A5000 (24 GB VRAM)
- **RAM:** 128 GB DDR4
- **OS:** Ubuntu 20.04

5 License

The artifact is available under *Creative Commons Attribution-NonCommercial-ShareAlike 3.0* license.

6 MD5 sum of the artifact

70273eb9be624fd9fbc6c5c93eec69f5

7 Size of the artifact

1.35 GB

References

1 Andreas Finkenzeller, Andrew Roberts, Mauro Bellone, Olaf Maennel, Mohammad Hamad, and Sebastian Steinhorst. Sensor Fusion Desynchronization Attacks. In Renato Mancuso, editor, *37th Euromicro Conference on Real-Time Systems (ECRTS 2025)*, pages 6:1–6:22, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ECRTS.2025.6.