On the generation of Corner case examples for visual perception in autonomous driving[*]

Alaa DAOUD$^{1[0000-0002-3640-327X]}$

Normandie Univ, INSA Rouen,
UNIROUEN, LITIS, 76801 Saint Etienne du Rouvray, France
alaa.daoud@insa-rouen.fr

Abstract. Corner cases in visual perception for autonomous driving refer to difficult or unexpected situations that the vehicle's perception system may encounter while driving. An AV simulator used to generate corner case examples for training the visual perception system of an autonomous vehicle would need to meet several requirements such as: (1) Realism, (2) Flexibility, (3) Scalability, (4) Hardware compatibility, (5) Safety, (6) Reusability, and (7) Integration with other simulators. Overall, a good AV simulator for generating corner case examples should be able to accurately mimic real-world conditions, be flexible and scalable, be compatible with the hardware used in the autonomous vehicle, be safe and reusable.

Keywords: First keyword · Second keyword · Another keyword.

1 Intro

Corner cases in visual perception for autonomous driving refer to difficult or unexpected situations that the vehicle's perception system may encounter while driving. These can include:

- Low-light conditions (e.g. at night or in a tunnel)
- Adverse weather conditions (e.g. heavy rain, snow, or fog)
- Glare from the sun or other bright lights
- Objects with unusual appearance or reflectivity (e.g. shiny vehicles or construction cones)
- Shadows or reflections that can create confusion
- Unusual or unexpected road layouts (e.g. construction sites or temporary detours)
- Pedestrians or animals that may behave in unexpected ways
- Occlusions (e.g. when an object is obscured by another object)
- Dynamic objects (e.g. other vehicles or bicycles) moving at high speeds

Dealing with these corner cases is important for ensuring the safety and reliability of autonomous vehicles

[*] part of the ANR MultiTrans project

2 AV Simulation and corner cases

A common approach used in the development and testing visual perception systems of autonomous driving is to generate a vast variety of situation examples via simulation [11].

Simulation involves the creation of a virtual environment that can mimic real-world conditions, such as different lighting, weather and road conditions. This can be used to generate a large number of scenarios that would be difficult or impossible to replicate in the physical world. In addition, it allows engineers to easily control different scenario parameters, such as object position and event timing, to test the limits of the perception system.

Using simulations to generate training examples for the visual perception system can also be useful for creating extreme cases and challenging scenarios that may not be commonly encountered in real-world driving. This can help ensure that the system is robust and can handle a wide range of conditions [9]. It is important to note that while simulation can be a useful tool for training and testing visual perception systems, it is still important to also test the system on real-world data to ensure its performance on real-world scenarios.

An autonomous driving simulator employed to produce corner case examples to train the visual perception system of an autonomous vehicle should meet several requirements:

Realism: the simulator needs to precisely mimic real-world conditions, such as lighting, weather, and road layout. This would guarantee that the scenarios generated by the simulator are realistic and representative of situations that the vehicle might encounter in the real world.

Flexibility: The simulator must be able to produce a wide range of situations, including extreme cases and unfamiliar situations, to test the limitations of the perception system. It should also enable easy manipulation of parameters such as object position and timing of events.

Scalability: a simulator should be able to manage a variety of set-ups and generate high-frequency data to effectively train the perception system.

Hardware compatibility: The simulator must be able to interoperate with the hardware involved in the autonomous vehicle, such as sensors and actuators, to make sure that the perception system is tested under realistic conditions.

Safety: The simulator must be developed with safety in mind to provide assurance that it does not pose a risk to the users or the surrounding environment.

Reusability: The simulator should be built in such a way that it can be easily adapted for multiple projects, experiments and maintenance.

Integration with other simulators: The simulator needs to integrate well with other simulators such as physics, dynamics, and control simulators to create a comprehensive and accurate representation of vehicle behavior in the simulated environment.

Overall, a good AV simulator for generating critical case examples must be able to accurately mimic real-world conditions, be flexible and scalable, be compatible with the hardware used in the autonomous vehicle, and be safe and reusable.

2.1 Technical requirements

For autonomous driving visual perception, a simulation used to generate corner case examples must have the following key technical capabilities:

Scene generation: the simulator must be able to generate realistic 3D environments that emulate real-world scenes. This includes roads, buildings, trees and other objects that the vehicle may encounter. The simulator must also be able to generate different types of weather conditions and lighting situations, such as rain, snow, fog and sun glare.

Object generation: The simulator must be able to generate a wide range of objects, such as vehicles, pedestrians, bicycles, and animals, that may be encountered by the vehicle. The simulator should also be able to generate unusual or unexpected objects, such as construction cones, that may confuse the perception system.

Object dynamics: The simulator must be able to generate dynamic objects that move and interact with each other and with the environment. This includes other vehicles, bicycles, and pedestrians that may move unpredictably. This will help test the ability of the perception system to track and predict the movement of objects in the scene.

Sensor simulation: the simulator must be able to simulate the sensor data that the perception system would receive from real sensors, such as cameras, lidars and radars. This includes simulating the noise and distortions that may be present in real sensor data.

Annotation: the simulator must be able to generate annotation data for the generated scenes, such as object labels, bounding boxes, and semantic segmentation masks. This data can be used to train the perceptual system to recognize and understand the objects in the scene.

Reusability: the generated scenes, objects and sensor data must be stored in a format that allows them to be easily reused and shared between different teams and projects.

Randomization: the simulator must be able to randomly generate scenario variations to make the system more robust to different conditions.

Integration: The simulator should be able to integrate with other simulators such as physics, dynamics and control simulators to create a holistic and accurate representation of the vehicle's behavior in the simulated environment.

Overall, a good AV simulator for generating example corner cases for visual perception should be able to generate realistic and varied 3D environments, dynamic objects, sensor data, and annotation data, be reusable, and integrate with other simulators.

2.2 Existing simulators

There are several simulators capable of generating extreme case examples for AV visual perception, some of which include:

CARLA: An open-source urban driving simulator based on Unreal Engine that generates realistic environments and dynamic objects. It also supports simulation of various sensors, including cameras and lidar, and provides annotation tools for object detection and tracking [4].

ApolloScape: Developed by Baidu, this is a high-fidelity open source driving simulator that includes a diverse large-scale dataset of 3D environments and dynamic objects. It also includes annotation tools for object detection and tracking, as well as sensor simulation tools [1, 8].

OpenDS: an open source, realistic driving simulator that generates various weather conditions, lighting and traffic scenarios. It includes a physics engine to simulate vehicle dynamics and supports simulation of various sensors, including cameras and lidar [6].

PreScan: A commercial driving simulator developed by TASS International. It can generate a wide range of scenarios, including extreme cases and unusual situations. It also includes sensor simulation tools and supports simulation of various vehicle types and driving conditions [13].

AirSim: a high-fidelity open source simulator for drones and other aerial vehicles. It allows the generation of realistic environments and dynamic objects, and supports the simulation of various sensors, including cameras and lidar [12].

Webots: a commercial, open-source multi-robot simulator that allows the creation of realistic environments and dynamic objects. It includes physical engines to simulate robot dynamics and supports simulation of various sensors, including cameras and lidar [3].

These are some examples of existing simulators that can generate extreme case examples for AV visual perception, but it should be noted that there are many other options available and new ones under development. When selecting a simulator, it is important to consider factors such as the type of vehicle being simulated, the level of realism desired, and the specific requirements of the project.

What about “Foxglove” and “Mushr” in this context?

As far as I can determine, Foxglove [5] and MuSHR [10] are not simulators specifically designed to generate example corner cases for AV visual perception.

Foxglove is a platform that allows for the collection, annotation, and management of sensor data from autonomous vehicles. It can be useful in the development and testing of AV visual perception systems, as it allows the creation of large and diverse datasets of sensor data.

The Multi-agent System for non-Holonomic Racing (MuSHR) is an open-source, modular and portable framework for simulating robotic system. It is mainly used to test, evaluate and develop control algorithms for robotic systems. It can be useful for testing and evaluating AV control systems, but it does not include visual perception capabilities.

Foxglove and MuSHR can be useful tools in the development and testing of autonomous vehicle systems, but they may not be specifically designed to generate example corner cases for visual perception. Other simulators mentioned earlier could be better suited for this purpose.

3 Related works

A state of the art study is issential to come with a conclusion of what simulation tool we can use. A state of the art study will allow us to: Understand the current capabilities and limitations of existing simulators for generating corner case examples for AV visual perception. Identify the most promising simulators currently available, based on factors such as realism, flexibility, scalability, and sensor compatibility. Understand the research gaps and challenges in this area and find potential areas for future research. Understand the strengths and weaknesses of the different simulators and select the one that best suits our needs. It's worth noting that the field of autonomous driving and simulators are constantly evolving, so it's important to keep an eye on the latest developments and updates in the field and re-evaluate the state of the art periodically.

3.1 Proposal for a research direction : A systematic literature review (SLR)

A systematic literature review on simulation tools for generating corner case examples of autonomous driving perception systems would be a valuable and informative research endeavor. Such a review would help to identify existing simulation tools, their capabilities and limitations, and how they have been used in the past to test and evaluate autonomous driving perception systems. Additionally, it may help to identify gaps in current simulation tool offerings and areas where further research is needed.

Here is a general overview of the process of conducting a systematic literature review (SLR) on simulation tools for generating corner case examples of autonomous driving perception systems:

1. Define the research question: Clearly define the research question you want to answer with the SLR. For example, "What are the available simulation tools for generating corner case examples of autonomous driving perception systems, and how have they been used in the past to test and evaluate such systems?"
2. Search for relevant literature: Conduct a comprehensive search of relevant databases, such as the IEEE Xplore Digital Library, Scopus, and the Web of Science, using appropriate keywords. You may also want to check the reference lists of relevant articles to find additional sources.
3. Selection of studies: Screen the titles and abstracts of the articles found in the literature search to identify those that are relevant to your research question. Then, read the full text of the selected articles to determine whether they meet the inclusion criteria for your SLR..

4. Data extraction: Extract relevant information from the selected articles, such as the simulation tools used, the methods used to test and evaluate autonomous driving perception systems, and the results and conclusions of the studies..
5. Data synthesis: Analyze and synthesize the extracted data to identify patterns, trends, and gaps in the literature. This may include creating tables and figures to summarize the data, and comparing and contrasting the findings of different studies..
6. Writing the review: Write the review, including an introduction, methods, results, discussion, and conclusion sections. The introduction should provide background information on the research question and explain the significance of the review. The methods section should describe the search strategy, inclusion and exclusion criteria, and data extraction and synthesis methods used. The results section should present the findings of the review, and the discussion should interpret the findings and discuss their implications for future research.

In the following a few papers that may be a good starting point for conducting an SLR on simulation tools for generating corner case examples of autonomous driving perception systems:

– A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research by Francisca Rosique, Pedro J. Navarro, Carlos Fernández and Antonio Padilla [11]. This paper provides a survey of simulation platforms for autonomous cars and their use in urban environments, including a discussion of the advantages and limitations of each platform.
– Data generation using simulation technology to improve perception mechanism of autonomous vehicles. by Minh Cao and Ramin Ramezani [2]. This paper demonstrates the effectiveness of combining data gathered from the real world with data generated in the simulated world to train perception systems on object detection and localization task
– Autonomous vehicles testing methods review. by WuLing Huang, Kunfeng Wang, Yisheng Lv and FengHua Zhu [7]. This paper provides a review of recent advances in simulation and testing methods for autonomous vehicles, including a discussion of the various types of simulations, the hardware and software used, and the challenges and opportunities for future research.

References

1. ApolloAuto. Apollo Scape simulator. “https://apolloscape.auto/” accessed : 2023-02-06.
2. M. Cao and R. Ramezani. Data generation using simulation technology to improve perception mechanism of autonomous vehicles, 2022.
3. CyberboticsLtd. Webots : Open source robot simulator. https://cyberbotics.com/#cyberbotics.
4. A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator, 2017.

5. Foxglove. The leading observability platform for robotics developers. https://foxglove.dev/.
6. W. Guo. OpenDS, Oct. 2022. "https://github.com/guowenbin90/OpenDS" accessed: 2023-02-06 original-date: 2017-02-23T19:37:17Z.
7. W. Huang, K. Wang, Y. Lv, and F. Zhu. Autonomous vehicles testing methods review. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 163–168, 2016.
8. W. Li, C. W. Pan, R. Zhang, J. P. Ren, Y. X. Ma, J. Fang, F. L. Yan, Q. C. Geng, X. Y. Huang, H. J. Gong, W. W. Xu, G. P. Wang, D. Manocha, and R. G. Yang. AADS: Augmented autonomous driving simulation using data-driven algorithms. *Science Robotics*, 4(28), mar 2019.
9. G. E. Mullins, P. G. Stankiewicz, R. C. Hawthorne, and S. K. Gupta. Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles. *Journal of Systems and Software*, 137:197–215, 2018.
10. MuSHR. (the multi-agent system for non-holonomic racing) an accessible platform for mobile, multi-agent robotics. https://mushr.io/.
11. F. Rosique, P. Navarro Lorente, C. Fernandez, and A. Padilla. A systematic review of perception system and simulators for autonomous vehicles research. *Sensors*, 19:648, 02 2019.
12. S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.
13. Simcenter. Prescan. "https://www.plm.automation.siemens.com/global/en/products /simcenter/prescan.html", accessed : 2023-02-06.