Automated Sensor Performance Evaluation of Robot-Guided Vehicles for High Dynamic Tests

1st David Hermann
Dept. for Simulation
Porsche Engineering Services GmbH
Stuttgart, Germany
David.Hermann@porsche-engineering.de

2nd Granit Tejeci
Electromobility
Universität Stuttgart
Stuttgart, Germany
st175405@stud.uni-stuttgart.de

3rd Clara Marina Martínez
Dept. for Simulation
Porsche Engineering Services GmbH
Stuttgart, Germany
Clara.Martinez@porsche-engineering.de

4th Gereon Hinz
Robotics, Artificial Intelligence and Real-time Systems
Technische Universität München
Munich, Germany
gereon.hinz@tum.de

5th Alois Knoll
Robotics, Artificial Intelligence and Real-time Systems
Technische Universität München
Munich, Germany
knoll@mytum.de

***Abstract*—As the demand for automated vehicle testing on proving grounds grows, the need for comprehensive and reliable environment monitoring systems becomes increasingly important. In highly dynamic driving test scenarios, long-range perception is essential for detecting dangers and hazards, ensuring the safety of both the test vehicle and other people on the track. However, determining an appropriate sensor setup can be challenging due to the complexity of sensor perception limitations. Perception limitations depend on the sensor characteristics and the environment. In this work, we propose a new approach to automatically evaluate sensor performance for high dynamic driving to improve the safety and efficiency of automated testing on proving grounds. Our approach involves estimating the detection range of common sensor technologies and analyzing the performance of sensor systems under various environmental conditions. By evaluating sensor performance in advance and comparing different sensor setups on tracks with a high-speed profile, we are able to identify critical track sections with higher collision risks and safeguard tests accordingly. This study emphasizes the importance of advanced environmental monitoring and sensor analysis in ensuring the safety and efficiency of automated vehicle testing.**

***Index Terms*—Sensor Performance, Criticality Assessment, High Dynamic Test, Automated Vehicle Testing, Ray-cast**

I. Introduction

Automated vehicle testing on proving grounds is becoming increasingly popular as it allows for efficient and safe testing of vehicles [1]. Automated testing enables the repetitive execution of more complex and demanding tests, while also increasing safety by removing the need for passengers in test vehicles. It also reduces the risk of health issues for test drivers caused by high levels of physical and mental stress.

The vehicles can be controlled using robots with powerful pedal and steering wheel actuators. Thus, automated high-speed testing on proving grounds offers significant benefits, such as increased consistency, accuracy and efficiency, while minimizing human error [2]. The robotic systems integrated into the test vehicles are used to apply the required high steering torques and brake pedal forces, especially during highly dynamic tests. With the ability to program precise test scenarios, including velocity profiles and trajectories, accurate and reproducible results can be obtained.

For high-speed testing, the ability to perceive potential hazards from a distance is critical for ensuring safety. Pre-series sensors in test vehicles cannot be relied upon, as they are often not designed for such safeguards and have not been thoroughly tested. Selecting appropriate additional sensor equipment is crucial in minimizing costs and ensuring safe automated driving operations. Furthermore, an evaluation of sensor performance in relation to specific test execution is also important in selecting an appropriate sensor setup. This includes assessing factors such as sensor range, measurement inaccuracies and the ability to capture a picture of the relevant driving environment.

Considering the relevant factors can be a complex task as the limitations of sensors can vary. Different types of sensors can have vastly different ranges and capabilities [3]. Depending on the speed and the vehicle, the test specifications may also affect sensor performance. Therefore, it is necessary to thoroughly evaluate sensor limitations in order to ensure accurate data collection and to make accurate conclusions about the performance of the sensor systems during tests, which is still challenging.

This study contributes a new approach for evaluating and comparing the performance of sensor systems in different environmental conditions when testing robot-guided vehicles on proving grounds. By identifying critical test track sections, with a high risk of collision, we aim to improve the reproducibility and reliability of automated vehicle testing, while also reducing the risk of accidents.

These critical sections require further surveillance, either coming from infrastructure or test vehicles. Hereby, a method for automatically evaluating onboard sensors on proving grounds is presented, enabling a safe and effective environmental monitoring system for a wide range of scenarios.

II. Related Work

The performance of sensor systems is critical to the safety and reliability of autonomous vehicles (AV) on the road. To ensure the success of AVs, it is important to optimize the performance of sensor systems and to accurately detect and classify objects in the environment.

One of the main challenges in evaluating advanced driver-assistance systems (ADAS) sensor performance in highly dynamic driving situations is the need for high accuracy [4]. In these scenarios, the sensor may be exposed to rapid changes in speed, traffic, weather and other environmental factors, which can significantly affect sensor performance [5]. In order to accurately assess the sensor's capabilities in these situations, a detailed analysis of its response to a wide range of inputs is required [6]. This can be a time-consuming and labor-intensive process, which may not be practical for many applications.

An example of the requirements and challenges of sensing in high-speed situations are presented in the paper [7] during the Autonomous Challenge at CES 2022. These vehicles were equipped with LiDARs (light detection and ranging), RADARs (radio detection and ranging) and cameras for opponent detection, but static obstacles on the track were not considered.

There have been numerous studies on the evaluation and optimization of sensor systems for AVs, [4], [8], [9]. These studies have focused on a variety of sensor types, including LiDAR [10], RADAR [11] and camera [9]. In order to accurately and reliably detect and classify objects, it is important to consider the specific characteristics of the sensor modality, such as its range, resolution and field of view (FoV) [12].

With optimized sensor systems, AVs may still encounter situations where objects are not detected. In [7] it is shown that LiDAR sensors can be affected by occlusions, shadows and reflections, which can lead to missed detections or false positives in critical situations. Similarly [13] found that RADAR sensors can be affected by noise, interference and multipath effects, which can also lead to missed detections or false positives in critical scenarios.

To address these challenges, researchers have proposed a variety of methods for analyzing the limitations of sensor systems and improving their performance in critical situations, as shown in [9]. These methods include the use of simulations to evaluate the performance of sensor systems under different conditions [14], as well as the development of fault-tolerant and redundant sensor architectures [15].

In [16] phenomenological sensor models are used for analyzing sensor performance. Phenomenological sensor models are a class of models in which physical phenomena are modeled. Based on the physical description of the sensor characteristics, the signal-to-noise ratio is determined, which can be used to estimate the probability of detection using receiver operating characteristics curves. Different scenarios were analyzed with regard to the detection probability of objects in a simplified environment.

In [17], an optimization procedure is presented that evaluates the information content of point cloud-based sensor models based on the entropy description. A ray-cast sensor model is used to determine the detection performance of sensors in a virtual environment by tracing rays from a defined origin point and detecting intersections with objects within a specified range. Based on this approach, a score is calculated, which is used for optimizing the sensor position.

A method is proposed in [6] to fulfill the safety of the intended functionality (SOTIF) [18] guidelines to ensure the safety of the perception systems of AVs. In this work, the uncertainty of the system is evaluated by empirically modeling the relationship between the environment and the system itself using a Bayesian network. In contrast, our proposed method evaluates the uncertainty by estimating critical sections quantitatively, using vision limitations of specific sensor setups.

III. Methodology

In this chapter, we first present a criticality metric to identify critical situations on the track. Then, we demonstrate the approach based on a virtual environment and models to estimate the sensing range. Finally, we present the complete algorithm for the automatic analysis of sensor performance. This approach aims to overcome the limitations of previous analyses, which relies on simplifying assumptions that may not accurately reflect real-world situations.

A. Criticality Definition

In situations with high collision risk, braking without any avoidance maneuver is one option to avoid collisions in high-dynamic automated driving as it is simple, reliable, fast and safe. It does not require complex calculations of a new trajectory or communication delays and avoids additional risks due to an evasive maneuver. Therefore, for this application, the robot-guided vehicle initiates emergency braking while following the intended trajectory.

There are certain sections on the test track for robot-guided vehicles where the range of perception of the onboard sensors is not sufficient. In these sections, the distance the vehicle must stop d_{stop} is greater than the maximum distance d_{det} at which a static target object can be detected for the first time by the onboard sensors. In these situations, emergency braking is not sufficient to avoid a collision. The criticality metric of the situation can be calculated similar to the difference of space distance and stopping distance [19] with

$$C_{\text{crit}} = d_{\text{stop}} - d_{\text{det}} \tag{1}$$

where $C_{\text{crit}} > 0m$ indicates a collision with a static target object.

The stopping distance results in $d_{\text{stop}} = d_{\text{react}} + d_{\text{brake}}$. The reaction time distance d_{react} depends on the latency of the sensor data processing and the actuators, as well as the time until a braking torque is applied to the wheels. The braking distance d_{brake} is the distance the vehicle travels from the moment the brakes are applied to the moment it comes to a stop and is influenced by speed, road and braking conditions. According to [20] the braking distance can be determined with

$$d_{\text{brake}} = \frac{v^2}{2\mu g} \tag{2}$$

where v is the vehicle speed, g is the force of gravity and μ is the coefficient of friction of the tires. The coefficient μ depends on various factors such as tires, weather conditions and road surface and can vary significantly. On proving grounds, these factors can be measured and a more accurate estimate of the braking distance can be obtained.

The detection range d_{det} needs to be determined taking into account various influencing factors and is therefore usually complex to calculate. We present our method for determining d_{det} in the following.

B. Detection Range

Object detection algorithms rely on the information content of sensor data to recognize objects. However, physical sensor limitations can affect not only the quality and quantity of data that a sensor can provide, but also the information content [21] of the data, which can affect the performance of these algorithms. These limitations include:

- Resolution: Resolution refers to the ability of a sensor to distinguish between closely spaced objects or details. A sensor with a higher resolution will be able to detect smaller or more closely spaced targets, while a sensor with a lower resolution will have a more limited detection range.
- Sensor FoV: Sensor FoV is the extent of the observable world that is seen by the sensor. A sensor with a wider FoV will be able to detect more of the environment, for example in a curve at one time, while a sensor with a narrower FoV may miss important details of the surroundings.
- Operational sensor range: Most sensors have a limited range within which they can detect or measure a target. The maximum range is often different from the operational range, which is the range over which the sensor can provide useful or accurate data. The operational range may be much shorter than the maximum range due to factors such as sensor noise, resolution, accuracy, or other limitations.
- Occlusion: Occlusion refers to the presence of an object or obstacle that blocks or partially blocks the view of a target. In the case of a detection system, occlusion can prevent the detector from detecting the target or can make it more difficult for the detector to locate the target.

All of these factors together can affect the overall range of a sensor and the sensor's ability to detect specific objects. Other factors, such as the wavelength of the energy being detected or the directivity of the sensor, can also have an impact on the range of a sensor. It is important to keep in mind that the specific limitations of a sensor will vary depending on the type of sensor and the environmental conditions.

In our approach, similar to [22], we use a virtual environment with a ray-cast sensor model to account for the various physical factors that can limit the range of a sensor. Using a ray-cast sensor model [23] allows us to simulate the propagation of the sensor's signal and to consider the limitations of the sensor's FoV and operational range by parametrizing the sensor model. The number of rays and the distance at which the ray hits the target can then be used to further analyze the detection performance. In addition, the ray-cast sensor model is suitable for simulating ADAS sensor types, such as LiDAR, RADAR and camera, due to its similar characteristics as described in [17].

Our approach includes also a 3-dimensional (3D) representation of the track. The 3D environment model contains relevant objects that may obstruct the line-of-sight of the ray-cast sensor representing the onboard sensors. These may include walls, buildings, partitions, elevations and changes in the course of the track. This approach allows for the testing of various scenarios and conditions in a simulated setting, which can provide a more accurate prediction of the sensor's performance.

The ray-cast sensor provides a point cloud $\mathcal{T}$ with 3D coordinates for each ray that encountered an object. The resolution of an object is determined by the points in the point cloud $\mathcal{O} \subseteq \mathcal{T}$ that correspond to the object. The occlusion or coverage t_{cov} of an object can be evaluated using bounding boxes.

A minimal bounding box, the ground truth bounding box, is placed around the reference object based on the ground truth data, and another bounding box is created around $\mathcal{O}$ generated by the ray-cast sensor model that corresponds to the reference object. Since a sensor can only see certain sides of an object, the others are in the invisible area of the sensor. To account for this, the area of the bounding box around the object that is visible to the sensor is compared to the corresponding area of the bounding box around the entire object. The visible and relevant area for detection is determined by selecting the largest area A_{PC} of the bounding box based on $\mathcal{O}$.

The coverage of an object t_{cov} by a sensor is then determined by comparing the size of the corresponding areas of the ground truth A_{GT} and the point cloud based bounding box

$$t_{\text{cov}} = \frac{A_{\text{PC}}}{A_{\text{GT}}} \tag{3}$$

The size and shape of the bounding boxes depend largely on the FoV and resolution of the sensor, as well as the position and orientation of the object relative to the sensor. Fig. 1 illustrates the coverage determination based on the point cloud of a target $\mathcal{O}$.

The detection performance can be estimated by analyzing the resolution of the target object and the level of coverage present in the data. High-resolution data provides more detailed information about the target object, increasing the chances of successful detection. However, occlusion can obstruct the view of the target object, making it more difficult for the algorithm to accurately identify it, thus reducing the detection performance.

In order to estimate an object detection performance, we introduce a detection score

$$\kappa = \frac{n_{\mathcal{O}}}{n_{\mathcal{T}}} \cdot t_{\text{cov}} \tag{4}$$

with the number of points corresponding to the target object

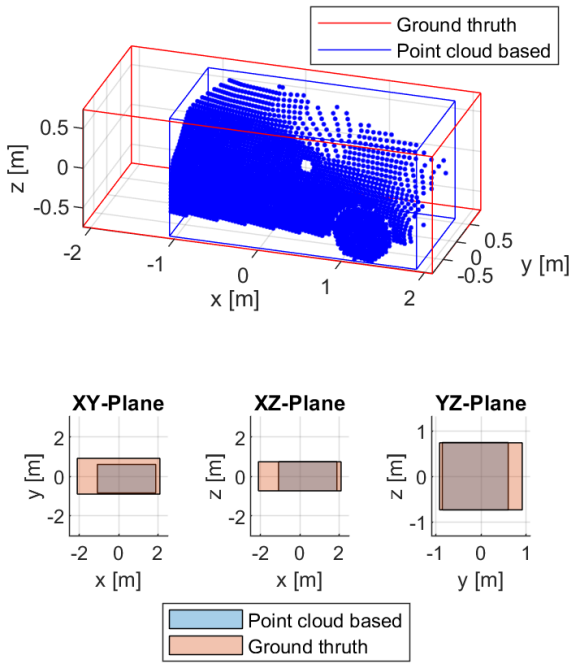


Fig. 1: The top graph presents a visual representation of the point cloud data $\mathcal{O}$ acquired by a ray-cast sensor, which corresponds to a target vehicle and its bounding boxes, as determined by ground truth data and the point cloud. The target vehicle is partially occluded by an obstacle. In the bottom portion of the figure, the 2D projected coverage areas of the bounding boxes for three relevant planes are depicted. Based on the comparison of the maximum areas, the coverage of the target is $t_{\text{cov}} = 80.17\%$.

$n_{\mathcal{O}}$, the total number of points $n_{\mathcal{T}}$ cast by the sensor model and the coverage (3) of the target. This method can also be applied to configurations with multiple sensors. When using multiple ray-cast models, the combined detection score value results from the superposition of the point cloud.

To determine detection of a target object, we impose a threshold ν_{ds} for the detection score. The target object is considered to be detected above this threshold $\kappa > \nu_{\text{ds}}$. Depending on the safety requirements, the threshold value can be chosen; a high threshold value results in high detection performance being considered.

In order to determine the detection range d_{det}, a target object is positioned in the virtual environment at different locations relative to the sensor and evaluated. At the locations where the detection score of the sensor is too low with $\kappa < \nu_{\text{ds}}$, the limit of the detection range is reached.

C. Automatic Sensor Performance Evaluation

Using the criticality metric (1), it is possible to identify points on the track along a given trajectory $\mathbf{p}$ and a velocity profile $\vec{v}_{\text{t}}$ where the onboard sensors have insufficient range. To thoroughly analyze a given trajectory, this process is repeated at different positions. For this purpose, N waypoints $\mathbf{W}_i$ with $i \in \{1, \cdots, N\}$ are created at intervals of distance d_{w} along

Algorithm 1: Automatic Sensor Performance Evaluation Algorithm

input : $\mathbf{W}_i$, N, t_{max}, $\vec{v}_{\text{t}}$, ν_{ds}, $\vec{d}_{\text{sensor,rel}}$
output: $\mathbf{C}_{\text{crit}}$
begin
  for $i \in \{1, \cdots, N\}$ **do**
    $t \leftarrow i + 1$;
    $\vec{p}_{\text{sensor}} \leftarrow \mathbf{W}_i + \vec{d}_{\text{sensor,rel}}$;
    Position target to $\mathbf{W}_t$;
    Get point cloud $\mathcal{T}$;
    Get $\mathcal{O}$;
    Calculate κ ;
    $t \leftarrow t + 1$;
    while $\kappa > \nu_{ds}$ *and* $t - i < t_{max}$ **do**
      Position target to $\mathbf{W}_t$;
      Get point cloud $\mathcal{T}$;
      Get $\mathcal{O}$;
      Calculate κ ;
      $t \leftarrow t + 1$;
    end
    Determine d_{det} from $\mathbf{W}_i$ to $\mathbf{W}_t$ along $\mathbf{p}$;
    Calculate $\mathbf{C}_{\text{crit},i}$ based on $\vec{v}_{\text{t,i}}$ along $\mathbf{p}$;
  end
  return $\mathbf{C}_{\text{crit}}$
end

the trajectory. These waypoints are used to position the sensor $\vec{p}_{\text{sensor}}$ with a relative position $\vec{d}_{\text{sensor,rel}}$ to the waypoint $\mathbf{W}_i$.

The parameter $\vec{d}_{\text{sensor,rel}}$ describes the sensor position and orientation on the robot-guided vehicle, with the origin of the vehicle coordinate system at $\mathbf{W}_i$. The position and orientation of a sensor influence its field of view and thus the performance of the sensor system, as [17] demonstrates. A sensor that is positioned higher, for example on the roof of a vehicle, will have a wider FoV compared to one that is positioned lower.

A target object is positioned along the preceding waypoints $\mathbf{W}_t$ with $t \in \{i+1, i+2, \cdots, i+t_{\text{max}}\}$. The parameter t_{max} describes the number of waypoints in the maximum FoV range of a sensor. At each step, κ of the sensor is determined and compared to the threshold ν_{ds}. From the waypoint, where the detection score is below the threshold, the maximum detection range of the sensor is reached. The procedure is described in pseudocode in Algorithm 1.

The procedure presented can measure the criticality metric along $\mathbf{p}$, but also for a specific object, which is used to calculate $\mathcal{O}$. Different objects with difficult geometries, such as vehicles, people, or detached vehicle parts on the lane, can be evaluated for sensor performance.

The velocity profile $\vec{v}_{\text{t}}$ for the robot-guided vehicle along the trajectory $\mathbf{p}$ is typically defined by the test specifications. The trajectory also has an impact and can significantly influence the criticality value. A trajectory that is reversed in the direction of travel or that passes closer to adjacent buildings may result in newly identified or relocated critical track sections.

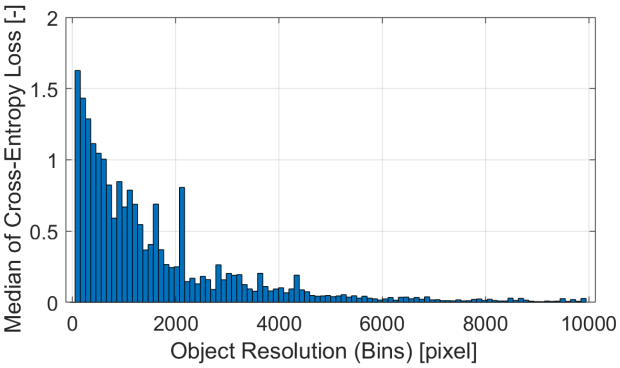


Fig. 2: Median values of cross-entropy losses over all detected objects using YoloV4 as a function of resolution with $100pixel$ bins. Over 85000 vehicles were evaluated based on the PandaSet dataset. With increasing resolution, the cross-entropy loss becomes smaller and thus demonstrates better performance in terms of detection rate.

Fig. 3: Image of the PandaSet dataset with bounding boxes for cars based on the YoloV4 algorithm. For simplicity, 3 out of a total of 26 recognized cars are shown. The CS of YoloV4 and κ (DS) are given for each of these objects. The detection score here correlates with the CS depending on the resolution of the respective object.

IV. Experiments

In the first step, we show experimentally that the resolution of objects in an image and the confidence score (CS) of an image recognition algorithm are correlated. Then we evaluate the recognition score for two distinct recognition algorithms using the PandaSet dataset [24]. In the final experiment, we apply the automatic sensor performance evaluation to a virtual race track from a real proving ground.

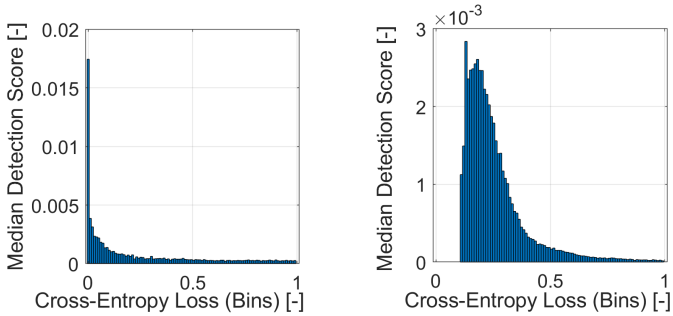
A. Detection Score Evaluation

The performance of a sensor system with an object recognition algorithm is evaluated by the cross-entropy loss function [25] $E_t = -p_t \cdot \log(q_t)$ for each object t based on the ground truth probability $p_t = 1$ and the CS q_t from the detection algorithm. Then the median for the cross-entropy losses was calculated for certain resolution ranges (bins).

The relationship between resolution and the cross-entropy loss is illustrated in Fig. 2. Here we evaluate the performance of the image recognition algorithm YOLOv4 [26] using the PandaSet dataset. The illustration shows that as the resolution of the object increases, the median of the cross-entropy losses decreases, which improves the ability of the algorithm to detect objects. Since image recognition performance also depends on other factors, such as non-uniform lighting and low contrast, there are variations in detection.

For an evaluation of (4), we compare the detection score results with different detection algorithms, such as YoloV4 for a camera and the point cloud segmentation SalsaNext [27] algorithm for a LiDAR sensor. We use also the PandaSet dataset for the comparison. The detection score is determined independently of the cross-entropy loss of the detection algorithms.

Fig. 3 shows the result for a camera image and illustrates the relationship between detection score and CS depending on the object resolution.



(a) Image recognition: YoloV4

(b) Point cloud recognition: SalsaNext

Fig. 4: Relationship of the median distribution of κ and the cross-entropy losses for vehicle targets. Two distinct recognition algorithms are applied for image recognition and LiDAR point cloud detection based on the PandaSet dataset. As the cross-entropy loss decreases, indicating better recognition performance, κ increases. The detection score thus reflects the performance of the recognition algorithm.

The analysis for the entire PandaSet dataset is shown in Fig. 4. The graphs Fig. 4a and Fig. 4b illustrate the correlation between κ and cross-entropy loss, demonstrating the suitability of the equation (4) as a method for calculating detection performance for different sensor systems and detection algorithms.

B. Setup Environment

In our experiment for evaluating the automatic sensor performance analysis based on the table Algorithm 1, we are using CARLA 0.9.12 [28] and MATLAB R2021a. CARLA is an open-source simulation environment for autonomous vehicle testing. It provides a ray-cast model, simulation ground truth data and allows for the creation of detailed 3D environments based on the Unreal game engine [29]. Our algorithm is

TABLE I: Sensor Parameters

Sensor	α_H	α_V	Resolution	Max. physical range
LiDAR	$25°$	$12°$	$833332pps$[a]	$300m$
Camera	$86.6°$	$86.6°$	$2304000pixel$	–

[a]points per seconds

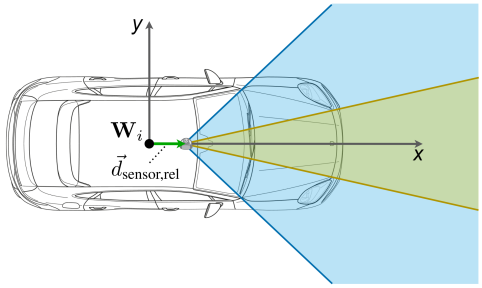


Fig. 5: Sensor positions in (x,y) coordinates relative to the vehicle coordinate system. LiDAR (FoV in yellow) and camera (FoV in blue) are positioned on the roof of the robot-guided vehicle.

implemented in Matlab and has an interface to interact with CARLA, which is used to control the simulation environment and collect sensor data.

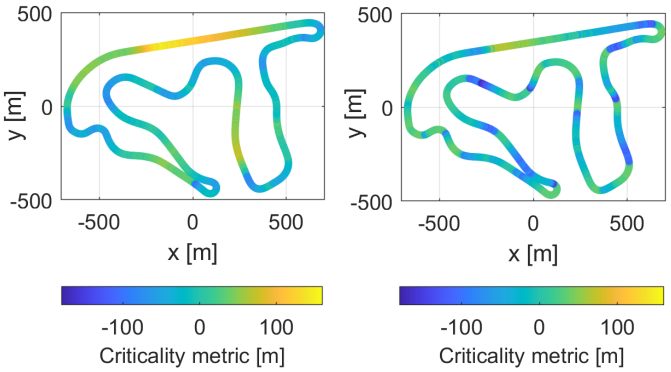
In our simulation environment, a 3D model of a real race track from a proving ground is used for analysis. The model of the track contains high-speed sections, different types of curves, an elevation profile, guard rails and buildings to replicate the conditions of the actual track. The 3D model is created based on an openDRIVE description [30].

For the evaluation of different sensor types and parameterizations, a long-range LiDAR with narrow horizontal FoV is combined with a high-resolution camera with wide azimuth and elevation angles, α_H and α_V respectively, as included in Tab. I. The camera has a resolution of 1920×1200 pixels. Both sensors are positioned at the same location on the roof of the robot-guided vehicle for criticality comparison, as shown in Fig. 5.

The trajectory $\mathbf{p}$ and velocity profile $\vec{v}_{\text{t}}$ are based on a measurement taken during a test on the track for future automation. The test was driven counterclockwise with a maximum velocity of $200\,\frac{km}{h}$ and high lateral and longitudinal accelerations of up to $1g$. The waypoint distance is set to $d_{\text{w}} = 8m$ for the analysis. To determine the stopping distance, a dry road with a friction coefficient of $\mu = 0.96122$ according to [20] is assumed. The detection score threshold $\nu_{\text{ds}} = 0.001$ is set to ensure robust detection.

C. Results and Discussion

In this chapter, we present the results of our sensor performance analysis Algorithm 1 applied on a virtual race track with a high dynamic driving profile. For this purpose, we



(a) Continuous criticality based on the camera sensor

(b) Continuous criticality based on the LiDAR sensor

Fig. 6: Top view of the race track and the criticality C_{crit} along the trajectory based on a high dynamic drive. The criticality is determined using sensors positioned on the vehicle. Based on the criticality metric, LiDAR performs better on straight sections. In contrast, the camera performs better on curves.

compare the detection performance of two different sensor types for monitoring the environment of robot-guided vehicles separately. Then we show the advantages of sensor fusion based on the previous results.

The results of the analysis of the two individually considered sensors are shown in Fig. 6 along the race track. These figures illustrate the camera Fig. 6a and LiDAR Fig. 6b criticality metrics.

In curves, the camera sensor has a higher detection range along the trajectory than the LiDAR sensor, due to the wide FoV α_H, which reduces the occlusion caused by objects and buildings. Furthermore, since the speed is lower in curves, the limited camera detection range is sufficient. Nonetheless, the LiDAR sensor cannot see around the corners due to narrower FoV in curves.

The plots indicate the significant influence of velocity profile on criticality. The criticality here depends on the speed and thus, according to (2), on the quadratically increasing braking distance. On long straight sections at high speeds, the detection range is insufficient to recognize objects or hazards in time. Compared to the camera in Fig. 6a, the LiDAR sensor in Fig. 6b has lower criticality on straight sections and higher velocities, due to the relatively higher point density for more distant targets.

In the middle of the race track on short straights at low velocities, the criticality metric remains high for both sensor types due to the negative elevation of the track. The negative elevation with -6%, starting on a plateau, poses a challenge for the sensor systems as it makes it difficult to detect objects that are obscured by the terrain. This highlights the importance of considering the impact of elevation on the sensor performance analysis of a given scenario.

For a performance comparison, the proportion of non-critical sections to the total track is considered, as well as

TABLE II: Sensor System Performance Metric

Sensor	Non-critical sections	Max. velocity	Max. C_{crit}
LiDAR	54.62%	$185.5\frac{km}{h}$	$74.74m$
Camera	52.88%	$129.0\frac{km}{h}$	$152.15m$
LiDAR & Camera	85.75%	$185.5\frac{km}{h}$	$74.74m$

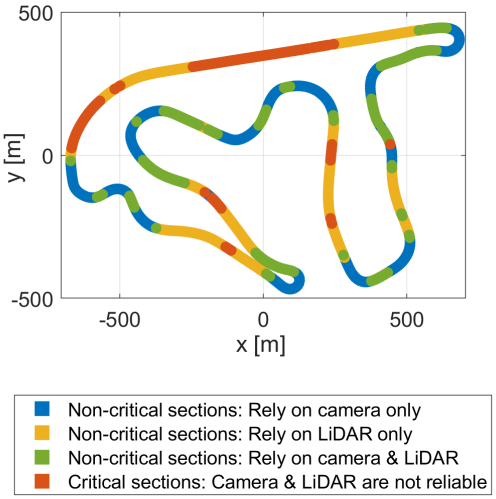


Fig. 7: Top view of the race track with the critical and non-critical sections. The non-critical sections are divided into sufficient sensor performance for the camera and LiDAR separately and in combination. Both sensors secure a large portion of the track. The critical sections without sufficient monitoring can be accurately identified.

the maximum possible vehicle speed in non-critical sections and the maximum value of the criticality metric over the race track. The lower the criticality metric in the case of a collision for $C_{\text{crit}} > 0m$ and under constant environmental conditions, the lower the impact energy. The corresponding values are listed in table Tab. II for the respective sensors.

The LiDAR sensor has performance advantages compared to the camera for this race track and conditions. The higher proportion of non-critical sections, a higher maximum speed in non-critical sections and a lower maximum criticality distinguish it here.

Considering both sensors combined for perception, e.g. by sensor data fusion, higher detection performance can be achieved as the sensors complement each other on different sections of the race track. For the combination of both sensors the criticality is determined by $C^{LC}_{\text{crit},i} = \min(C^{\text{LiDAR}}_{\text{crit},i}, C^{\text{Camera}}_{\text{crit},i})$. This combination describes the upper bound of the criticality and represents the minimum demand for a fusion of a sensor system. A significantly higher proportion of non-critical sections, a higher maximum speed and a lower criticality metric indicate that the sensor combination is preferable.

In Fig. 7, we illustrate the critical $C_{\text{crit}} > 0m$ and non-critical $C_{\text{crit}} \leqq 0m$ sections of the race track. The non-critical sections are divided into sections where only one or both sensors individually enable timely target detection. This allows us to identify track sections where one sensor outperforms the other, as mentioned earlier. As the figure demonstrates, using a camera in curves and LiDAR on straight sections effectively ensures safety. Combining the two sensors results in a significant improvement in sensor performance in terms of track coverage. Additionally, we have identified and precisely located the critical sections.

Our analysis of sensor data for the race track allows us to highlight the different strengths and weaknesses of the sensors. By comparing the results of the two sensors separately and in combination, we obtain reliable information about the performance of the sensor systems, which can be used to optimize and ensure the safety of the automated vehicle tests. Critical sections can be systematically covered by additional sensors, for example in the infrastructure, at a targeted location and with low effort.

V. Conclusion and Future Work

The analysis conducted in this study provides valuable insights into the performance of sensor systems for monitoring the environment of robot-guided vehicles. We proposed a new approach to automatically evaluate different types of sensors and by estimating the detection range based on the detection score under different conditions, we can identify critical and non-critical sections of a track. We have shown the strengths and weaknesses of individual sensors for a race track from a proving ground. Furthermore, our approach can be used to easily assess the performance of combined sensor systems.

The results of the analysis demonstrate the importance of considering factors such as speed, elevation and occlusion when assessing the criticality of a given scenario. This research provides a basis for the design and implementation of automated testing systems in terms of safeguarding.

This study is part of a work to optimize automated testing and increase the level of automation of test execution on proving grounds, where more test variations with different conditions, such as the impact of weather conditions, will be investigated. For further research, the performance analysis can be integrated into an automatic optimizer to find an optimal sensor setup. Furthermore, our approach could also be applied to worldwide scenarios, such as public roads, to analyze the limitations of ADAS sensor systems in different environments. In addition, the proposed method can be extended to calculate the criticality for dynamic obstacles. Various methods of fusing multiple sensors can also be explored for further analysis.

References

[1] Alessia Knauss et al. “Proving Ground Support for Automation of Testing of Active Safety Systems and Automated Vehicles”. In: *Fourth International Symposium on Future Active Safety Technology towards Zero-Traffic-Accidents (FASTzero) 2017*. Sept. 2017.

[2] Sebastian Siegl and Edgar L. v. Hinüber. “Testen auf dem Prüfgelände bis SAE Level 5, basierend auf dem Fundament des PEGASUS-Projektes”. In: *35. VDI Tagung Fahrerassistenzsysteme und automatisiertes Fahren*. May 2022.

[3] De Jong Yeong et al. “Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[4] Annkathrin Krämmer et al. “Providentia – A Large-Scale Sensor System for the Assistance of Autonomous Vehicles and Its Evaluation”. In: *Field Robotics*. June 2022.

[5] Chang-Gyun Roh, Jisoo Kim, and I-Jeong Im. “Analysis of Impact of Rain Conditions on ADAS”. In: *Sensors 2020*. Nov. 2020.

[6] Ahmad Adee, Roman Gansch, and Peter Liggesmeyer. “Systematic Modeling Approach for Environmental Perception Limitations in Automated Driving”. In: *2021 17th European Dependable Computing Conference (EDCC)*. Sept. 2021.

[7] Johannes Betz et al. “TUM Autonomous Motorsport: An Autonomous Racing Software for the Indy Autonomous Challenge”. In: May 2022.

[8] M. Nadeem Ahangar et al. “A Survey of Autonomous Vehicles: Enabling Communication Technologies and Challenges”. In: *Sensors 2021*. Jan. 2021.

[9] Eduardo Arnold et al. “A Survey on 3D Object Detection Methods for Autonomous Driving Applications”. In: *IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 10*. Oct. 2019.

[10] Thinal Raj et al. “A Survey on LiDAR Scanning Mechanisms”. In: *Electronics 2020*. Apr. 2020.

[11] Zoltan Ferenc Magosi et al. “A Survey on Modelling of Automotive Radar Sensors for Virtual Test and Validation of Automated Driving”. In: *Sensors 2022*. July 2022.

[12] Florian Alexander Schiegg, Ignacio Llatser, and Thomas Michalke. “Object Detection Probability for Highly Automated Vehicles: An Analytical Sensor Model”. In: *5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2019)*. Feb. 2019.

[13] Alper Cemil and Mehmet Ünlü. “Analysis of ADAS Radars with Electronic Warfare Perspective”. In: *Sensors 2022*. Aug. 2022.

[14] Sven Hallerbach et al. “Simulation-Based Identification of Critical Scenarios for Cooperative and Automated Vehicles”. In: SAE Technical Paper. Apr. 2018.

[15] B. Sari and H. Reuss. “Fail-Operational Safety Architecture for ADAS Systems Considering Domain ECUs”. In: SAE Technical Paper. Jan. 2018.

[16] Thomas Ponn, Fabian Müller, and Frank Diermeyer. “Systematic Analysis of the Sensor Coverage of Automated Vehicles Using Phenomenological Sensor Models”. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 1000–1006. DOI: 10.1109/IVS.2019.8813794.

[17] Stefan Roos et al. “A Framework for Simulative Evaluation and Optimization of Point Cloud-Based Automotive Sensor Sets”. In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. 2021, pp. 3231–3237. DOI: 10.1109/ITSC48978.2021.9564871.

[18] ISO Iso. “Pas 21448-road vehicles-safety of the intended functionality”. In: *International Organization for Standardization* (2019).

[19] S. M. Mahmud et al. “Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs”. In: *IATSS Research* 41 (Mar. 2017). DOI: 10.1016/j.iatssr.2017.02.001.

[20] Ali Abdi Kordani et al. “Effect of Adverse Weather Conditions on Vehicle Braking Distance of Highways”. In: *Civil Engineering Journal* 4 (2018), pp. 46–57.

[21] Yongjun Li et al. “YOLO-ACN: Focusing on Small Target and Occluded Object Detection”. In: *IEEE Access* 8 (2020), pp. 227288–227303. DOI: 10.1109/ACCESS.2020.3046515.

[22] David Hermann et al. “AutoSCOOP: Automated Road-Side Sensor Coverage Optimization for Robotic Vehicles on Proving Grounds”. In: *2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS)*. 2022, pp. 01–06. DOI: 10.1109/ICPS51978.2022.9816927.

[23] Guilherme Ferreira Gusmão, Carlos R. Hall Barbosa, and Alberto Barbosa Raposo. “Development and Validation of LiDAR Sensor Simulators Based on Parallel Raycasting”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[24] Pengchuan Xiao et al. “PandaSet: Advanced Sensor Suite Dataset for Autonomous Driving”. In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* (2021), pp. 3095–3101.

[25] John S Bridle. “Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition”. In: *Neurocomputing*. Springer, 1990, pp. 227–236.

[26] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. “YOLOv4: Optimal Speed and Accuracy of Object Detection”. In: *ArXiv* abs/2004.10934 (2020).

[27] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. *SalsaNext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving*. 2020. arXiv: 2003.03653 [cs.CV].

[28] Alexey Dosovitskiy et al. “CARLA: An Open Urban Driving Simulator”. In: (Nov. 2017).

[29] Epic Games. *Unreal Engine*. Version 4.27.2. Jan. 19, 2022. URL: https://www.unrealengine.com.

[30] Marius Dupuis, Martin Strobl, and Hans Grezlikowski. “Opendrive 2010 and beyond–status and future of the de facto standard for the description of road networks”. In: *Proc. of the Driving Simulation Conference Europe*. 2010, pp. 231–242.