Detection and Classification of Obstacles in Off-Road Autonomous Vehicles with LiDAR and Camera Data Fusion

Ana Luiza Santos Coura
Edson Caoru Kitani
FATEC – Santo André.

Antonio Marcos Silva
Cirilo de Paula Silva
Nouriandres Liborio Silva
Mercedes-Benz do Brasil Ltda.

Francisco Zampirolli
Luiz Antonio Celiberto Junior
Renato de Avila Lopes
Ugo Ibusuki
UFABC.

Leopoldo Yoshioka
USP.

ABSTRACT

Ensuring operational safety for off-road autonomous vehicles requires high accuracy in obstacle detection for rapid and precise decision-making. Recent studies indicate that integrating different sensors enhances this reliability. The method proposed in this work clusters 2D LiDAR points to detect objects, defines risk zones based on distance and position, and classifies obstacles using the YOLOv8 algorithm. The data fusion strategy for the 2D LiDAR and camera utilized an occupancy matrix, combining detections from both sensors to enable more accurate detection and classification than isolated solutions. Tests conducted on the off-road vehicle with objects of various dimensions in static and dynamic scenarios demonstrated the proposal's capability to identify and classify obstacles at different distances with low latency. Experimental results highlighted improvements in obstacle detection and classification accuracy, as well as low response latency. The proposed approach optimizes safety by integrating data from multiple sensors and enabling the vehicle's planning and control algorithm to make agile and accurate decisions.

I. INTRODUCTION

The safety of autonomous off-road vehicles depends on the ability to identify obstacles with high accuracy. Unlike urban situations, where predictable structures and rules facilitate locomotion, off-road environments present challenges such as uneven terrain, sudden changes in lighting, and unexpected obstacles, ranging from rocks to dense forest [1]. In this scenario, perception systems that utilize a single sensor, such as cameras or LiDAR, often have limitations: cameras exhibit lower metric accuracy and are sensitive to weather conditions, while LiDARs sometimes struggle to classify objects semantically [2].

Recent studies have demonstrated that integrating data from various sensor types, including LiDAR, cameras, and radar, enhances the robustness and accuracy of detection systems [2, 3]. However, the efficient integration of these sensors into embedded platforms with computational constraints is still a challenge. Works such as those by Elfes [4] and Matthies [5] have explored probabilistic representations for sensory fusion, while current approaches, as suggested by Yabroudi et al. [1], utilize adaptive clustering algorithms to handle point clouds. Despite these advances, gaps persist in the harmonization between latency, metric accuracy, and real-time semantic classification for off-road applications.

This work proposes an integrated approach that combines 2D LiDAR point clustering with classification via YOLOv8, combined with a fusion strategy based on occupancy grids. The method introduces three main contributions: (i) An optimized LiDAR clustering algorithm, which uses Euclidean distance, to reduce computational complexity, (ii) definition of risk zones, to suit the off-road environment, and (iii) a LiDAR-camera fusion matrix, linking visual detections to spatial data, to improve metric and semantic reliability. Experimental results in both static and dynamic scenarios demonstrate accuracy in obstacle detection and a reduction in false positives, even in areas with multiple obstacles.

The article's structure is as follows: Section II explains the methodology used, while Section III discusses the tests and the results obtained. Finally, Section IV presents the final considerations, along with possible future directions.

II. DEVELOPMENT

In this section, the steps of the proposed architecture, developed with a Camera for visual detection, a 2D LiDAR SICK LMS141-15100 (40 m range, 270° scan) for mapping, an Intel NUC computer (embedded processing) and a Mercedes-Benz Arocs 8X4 truck as an integration and testing platform, will be discussed.

A. YOLOv8

For object detection within the image, the YOLOv8 (You Only Look Once, version 8) architecture one of the most recent models in the YOLO class, was utilized. This release introduces new features that significantly enhance performance in object detection and recognition, enabling the identification and classification of multiple objects, even in highly complex images. YOLOv8 utilizes optimized layers and refined training strategies, enhancing the detection of objects of varying dimensions, shapes, and in challenging conditions [6].

Additionally, the architecture presents significant advances in tasks such as instance segmentation, tracking, and edge detection, which are fundamental elements for accurately recognizing partially hidden or overlapping objects. Another advantage is that it already supports transfer learning right at the facility, making it easier to adapt to different tasks with less need for labelled data. These characteristics make the YOLOv8 especially suitable for embedded applications such as driving assistance systems and mobile robotics.

B. Grouping of points

To separate distinct objects in the point cloud obtained by the 2D lidar sensor, a grouping approach based on Euclidean distance was employed, to identify sets of points belonging to the same object. The sensor used was configured to scan only the front of the truck used in the tests.

Given an ordered set of points $P = \{p1, p2, \ldots, pn\}$ acquired by the LiDAR sensor, where each point $p_i = (x_i, y_i)$ represents a coordinate in the plane P, the Euclidean distance between two consecutive points p_i e $p_{(i+1)}$ is given by:

$$d(p_i, p_{i+1}) = \sqrt{((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2)}. \qquad (2.1)$$

A threshold is defined as $\delta \epsilon R^+$, which represents the maximum distance between two points to be considered part of the same grouping (or object). The algorithm cycles through the points sequentially, grouping all the points p_i end $p_{(i+1)}$ such that:

$$d(p_i, p_{i+1}) \leq \delta. \qquad (2.2)$$

When the distance between two consecutive points exceeds this threshold, the current cluster is terminated, and a new one is started from $p_{(i+1)}$.

This method assumes that distinct objects in the environment are separated by regions that are sufficiently empty (or spaced greater than δ) so that they can be differentiated based on distance alone. Choosing the value of δ is critical and can be empirically calibrated according to the type of environment and the resolution of the LiDAR sensor.

While using the raw point cloud from the sensor preserves as much spatial information as possible, working directly with this large amount of data can be computationally costly, especially in real-time or embedded applications. In contexts like this, the amount of points generated by high-resolution 2D or even 3D sensors can make it challenging to analyse promptly.

Several approaches have been proposed in the literature to mitigate this problem, reducing the complexity of the data while preserving its most relevant geometric properties. An example is the work of Li et al. [7], who propose a geometry-based point cloud reduction method for mobile augmented reality systems, demonstrating that simplifications guided by geometric criteria can maintain essential spatial structure with lower computational cost.

Based on this approach, the proposed method returns only three representative points per detected object: the starting point, the end point, and the central point. This minimal representation, based on Euclidean relationships, is sufficient for spatial tracking and analysis applications, while significantly reducing the volume of data to be processed later.

In addition to the position of the points, the size of the object is estimated based on the distance between the extreme points. Considering that the LiDAR sensor performs an angular scan from 0 to 270 degrees every 0.5 degrees, it is possible to estimate the length of the object from the equation of the string in a circumference:

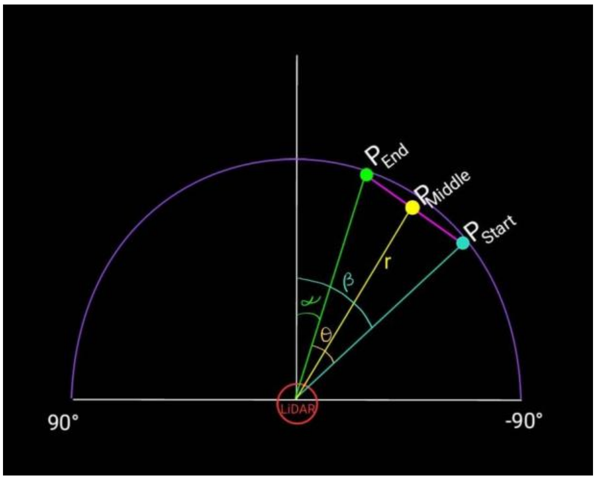
$$L = 2.r.sin\left(\frac{\theta}{2}\right), \tag{2.3}$$

where:

- L is the length of the string (straight distance between the ends of the object),
- r is the radial distance from the object's centre point,
- θ is the angle between the position vectors connecting the sensor to the first and last points of the object.

This approximation assumes that the points belong to a roughly circular contour in the scan plane, which is reasonable in many scenarios.

Figure 1- Calculation of the generated rope between the LiDAR beams.

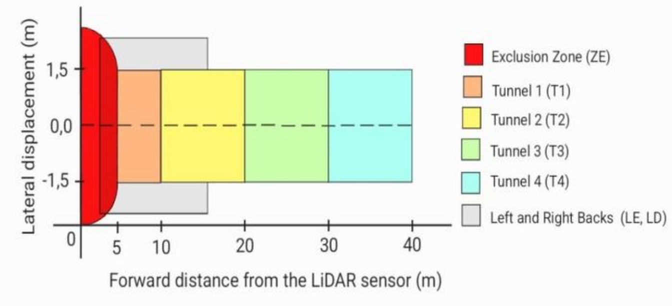


C. Tunnel

With the previously identified point groupings, this step establishes the region of interest for risk classification, using exclusively the data from the 2D LiDAR sensor.

The proposed approach is based on a frontal geometric segmentation adapted to the vehicular context, drawing on the concept of "conical tunnel" introduced by Elfes [4]. In this original concept, an area in front of the agent is probabilistically analysed in terms of occupation. In this work, the concept is reinterpreted as a fixed rectangular front tunnel, 3 meters wide and 40 meters long, aligned with the vehicle's longitudinal axis.

Figure 2 Schematic representation of the frontal tunnel segmented into zones



This region is subdivided into zones that represent different levels of risk, according to the distance of the object from the vehicle and its position relative to the center of the lane. The objective is to associate each detected cluster with a corresponding risk zone, enabling more accurate real-time decisions. Figure 1 illustrates the spatial segmentation adopted, which will be detailed in the following subsections.

1) <u>Exclusion Zone</u> (ZE)

It corresponds to the first 6 meters in front of the vehicle, covering the entire width of the tunnel (±1.5 m). This is the region of most significant risk, associated with the need for immediate response, such as emergency braking. To avoid sudden discontinuities, the rear limit of the ZE incorporates a smoothed oval transition, formed by the union of arches with a radius of 2 m at the lateral ends of the rectangle. Formally, this region is defined by:

- Length: $0 \leq x < 6m,$ (2.4)
- Width: $-1,5 \leq y < 1,5m,$ (2.5)
- Curved side contours forming the oval back with $r = 2m.$

Objects within this region are classified as risky 6.

2) <u>Progressive Risk Tunnel</u> (T1 a T4)

The remaining region of the tunnel (from 6 m to 40 m) is divided into four consecutive longitudinal zones, with decreasing risk levels as the distance from the object increases:

- T1: de 6 m a 12 m — risk 5
- T2: de 12 m a 20 m — risk 4
- T3: de 20 m a 30 m — risk 3
- T4: de 30 m a 40 m — risk 2

All these zones maintain a width of 3 m, centered on the vehicle's axle.

3) Left and Right Backs (LE e LD)

Regions beyond the lateral limits of the main tunnel ($y \vee 1{,}5m$), but still within the field of view of the LiDAR sensor, are designated as LE (left side) and LD (right side). Although they are outside the vehicle's direct trajectory, these zones may contain relevant information for overtaking analysis, evasive maneuvers, or lane edge monitoring.

Objects detected in these areas are classified as risk 1.

Based on the segmentation described, each grouping of points is analyzed by considering the position of its central point in the frontal Cartesian plane. The polar coordinates (r, θ) of the center point are converted to Cartesian coordinates relative to the vehicle, allowing the determination of:

- The longitudinal distance $x = r.cos(\theta)$, (2.6)
- The side position $y = r.sin(\theta)$. (2.7)

Classification logic can be formally defined as:

$$P(Z_i \vee x, y), \tag{2.8}$$

in which $Z_i \in \{ZE, T1, T2, T3, T4\}$ for $i = 0,1,2,3,4,5,6$ and x, y are Cartesian coordinates of the object in the tunnel plane.

If we have a detection event such that:

$$E = \{\vee\, y \vee \leq 1{,}5m\} \tag{2.9}$$

then the event E indicates that there is object in the tunnel if:

$$P(E|y) = \begin{cases} 1, if \vee y \vee \leq 1{,}5m \\ 0, otherwise \end{cases} \tag{2.10}$$

Thus, if there is an event E_j, there is a $objeto^J$ and the determination of the zone Z_i will be given by:

$$P(Z_i|x, E_j) = \begin{cases} 1, if\ x \epsilon Z_i \\ 0, otherwise \end{cases} \tag{2.11}$$

$$P(Z_E|x, E_j) = \begin{cases} 1, if\ x \leq 6m \\ 0, otherwise \end{cases} \tag{2.12}$$

$$P(Z_1|x, E_j) = \begin{cases} 1, if\ 6 < x \leq 12m \\ 0, otherwise \end{cases} \tag{2.13}$$

$$P(Z_2|x, E_j) = \begin{cases} 1, if\ 12 < x \leq 20m \\ 0, otherwise \end{cases} \tag{2.14}$$

$$P(Z_3|x, E_j) = \begin{cases} 1, if\ 20 < x \leq 30m \\ 0, otherwise \end{cases} \tag{2.15}$$

$$P(Z_4|x, E_j) = \begin{cases} 1, if\ 30 < x \leq 40m \\ 0, otherwise \end{cases} \tag{2.16}$$

The logic was implemented in a parameterizable manner, utilizing configuration data stored in spreadsheets, which enables dynamic adjustments to the width, the limits of each tunnel, and the exclusion distance. The classification, based on the center point, offers robustness even in the face of objects with multiple echoes or irregular geometries, thanks to the clustering method. This method allows you to adjust the distance δ between points, determining when to consider a set of points as a single object.

D. Calibration

To enable data fusion between the 2D LiDAR and the camera, precise calibration was required between the two sensors. The camera was positioned 1.05 meters above the lidar sensor, which required adjusting the camera's tilt angle to align its field of view to the plane of the laser beam. The calibration process aimed to align the center of the image with the height of the Lidar scan line, thereby establishing a standard reference. This correspondence was calibrated for a reference distance of 20 meters and a ground clearance of 1.10 meters, ensuring adequate overlap between visual and spatial data.

Figure 3: Front view of AROCS showing the position of the LiDAR and Camera sensors.

E. Fusion LiDAR – Camera

The fusion of data from the LiDAR sensor and the camera aimed to associate the visual detection (performed by YOLOv8) with the spatial measurements (provided by the LiDAR), thereby promoting a more robust and accurate identification of objects in front of the vehicle. This association enabled the complementing of the semantic classification provided by the camera with the distance and angle metric information provided by LiDAR.

Based on the grid representation approach proposed by Elfes and Matthies [5], a matrix was constructed $M \epsilon R^{16x16}$, representing a discretization of the camera image plane (1280 × 960 pixels). Each cell $M_{i,j}$ was associated with a distance interval $[d_{min}(i,j), d_{max}(i,j)]$, defined by geometric calibration between the camera's field of view and the Lidar scan plane.

The concept of this fusion is based on Bayes' Theorem, which combines information from different sources to update the degree of belief about the state of the environment. The correspondence between an object detected by LiDAR and a visual detection was performed based on the overlap of the distance intervals and angular consistency. Conceptually, this association can be interpreted as a Bayesian update of trust in the existence of a specific object.

Given that the sensors operate independently, the probability that both are detecting the same object is updated based on:

- The proximity between the distance measured by LiDAR and the distance range estimated by the camera array;
- The angular coherence between the object's position in LiDAR space and its position in the image.

Formally, considering two sensors A (LiDAR) and B (camera), B being dependent on A to detect event E, we have:

$$P(E|A) = \frac{P(A|E).P(E)}{P(A)}, \quad (2.17)$$

And when entering the camera information:

$$P(E|A,B) = \frac{P(B|E).P(E \vee A)}{P(B \vee A)}. \quad (2.18)$$

In this context, LiDAR (sensor A) is the primary source of distance measurement, while the camera (sensor B) contributes with visual information. When the camera detects an object that LiDAR does not perceive, the visual position is published with the shortest estimated distance from the corresponding cell in the array. However, this distance has low reliability, since the confidence in the metric estimation comes entirely from LiDAR.

Therefore, fusion can be understood as a probabilistic process in which prior knowledge (LiDAR measurement) is refined by additional evidence (camera detection). This results in greater confidence in the presence, location, and classification of objects. In this way, the system can mitigate individual sensor limitations such as the absence of LiDAR detection at high heights or incomplete visual detections, providing a more reliable representation of the environment for truck decision-making.

III. TESTS AND RESULTS

The experiments were conducted in a test yard with two distinct scenarios (Figure 4):

1. Oval course: consisting of two 80 m straights and curves with a radius of 20 m.
2. Straight path: 100 m linear path.

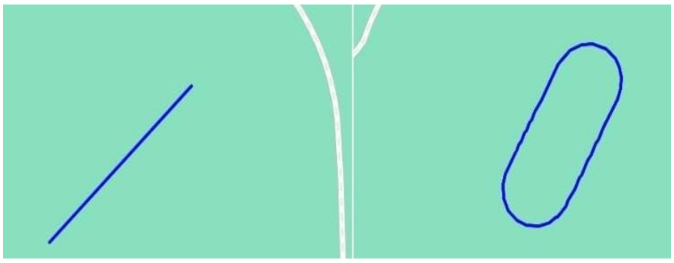


Figure 4: Routes used for testing

The autonomous vehicle was equipped with the proposed algorithm and subjected to a speed of 10 km/h, with five repetitions per condition for statistical validation. Two *dummy* objects were used for validation (Figure 5)

I. Balloon Car: simulates the rear of a vehicle (1350 mm × 1600 mm).

II. Puppet: represents a pedestrian (1154 mm × 298 mm).

Figure 5: Objects used in the tests

The first round of tests was conducted on the oval circuit, where the Balloon Car was placed on the second straight of the course between 35 and 40 meters after the corner. By analyzing the data from the five laps, as shown in Table 1, which outlines the data collection process, the detection maintained a consistent pattern at the angle, with a standard deviation of 0.44° and an average initial distance of 36.176 meters as the vehicle approached the object. The 'type' column displays the classification of the object, in other words, what it is. When an object appears as 'unknown', it means it was detected but could not be classified. This happens because, for some reason, it was not recognized by the YOLOv8 database. The consistency in the distance value holds more consistently, with a standard deviation between 0.05 and 0.01 meters from 30 meters away from the object, highlighting the fact that the closer to the target, the more accurate the measurement. This same pattern, of the closer to the object, the more robust the values acquired can be observed in Table 2 which shows the mean and standard deviation of the other test batteries performed, namely: the same route as the first test battery with the speed differential that was done at 15 km/h, the third that followed on the same route containing two objects for detour, one of them being the dummy and the other the Balloon Car and the fourth battery of tests that was carried out on the straight, using the Balloon Car as an object, positioned 50 meters from the beginning (center of the straight). When the system detects the object for the first time at T4 (figure 2), the average standard deviation of the distance is 1.42 meters. This value decreases as the truck approaches the object, reaching an average standard deviation of 0.007 meters at T1. For the dummy, this behavior was similar. However, unlike the Balloon Car, the standard deviation of the distance remained above 0.3 meters at T4 and T3, as shown in Table 1.

For all other routes, the results showed that the angle of detection of objects by LiDAR remained constant, with low standard deviations. The conclusion is that these deviations in the values are related to the vehicle's position on the route. If the vehicle were to start its journey a little further to the right or left, relative to the angle of the object, the angle detected by LiDAR would also change.

The estimated size of the two objects used in the tests remained constant with a mean standard deviation of 0.08 meters in T4 and 0.0016 meters in T3, T2 and T1 for the Balloon Car, where a difference of 0.3 meters in relation to the actual size of the object was obtained as estimated by the algorithm. For the dummy, the standard deviation of the size in T4 did not obtain results because only a LiDAR beam was able to capture the object, as it is smaller in width, an estimate of its size was only possible from T3 where an average standard deviation of 0.04 meters was obtained between T3 and T1, with an approximate difference of 0.04 meters in relation to its actual size.

Regarding the fusion of semantic information obtained by the camera, it can be observed that, in the ZE, all objects are unknown, as this zone is a blind spot for the camera due to the height of the vehicle, as for the parts of the tunnels and the sides, it depends on the ambient lighting, since if the sun hits the camera directly, it can cause difficulty in recognizing objects, the size of the object and the swaying of the truck when moving also interfere with the junction of LiDAR-Camera information.

Finally, an additional battery of tests was carried out with the Balloon Car just ahead of the peak of the curve on the oval course. As shown in Table 2, within the tunnels, the object was only detected in T1, very close to the ZE, where the truck almost failed to detect the object, forcing the driver to brake the vehicle manually due to the risk of collision. This failure to detect objects close to the vehicle on a curve was due to the fixed tunneling in front of the truck. The method proposed here assumes that objects will always be in straight sections, not curves, providing perspectives to enhance the fixed tunneling method.

IV. CONCLUSION

This work proposed an integrated obstacle detection and classification strategy for autonomous *off-road* vehicles, combining data from a LiDAR and a camera through sensory fusion based on an occupancy matrix. The approach demonstrated efficacy in grouping LiDAR points based on Euclidean distance, in defining progressive risk zones, and in classifying objects using YOLOv8, ensuring a good accuracy. The experimental results demonstrated robust detection and proved that fusion improves reliability, especially in dynamic scenarios, by harmonizing LiDAR metric information with camera visual semantics for a better understanding of what is in front of the vehicle.

The segmentation of the tunnel enabled prioritizing obstacles according to their distance and relative position about the vehicle, while geometric calibration between sensors ensured robust spatial matching. However, limitations were observed, such as in sharp curves, where the fixed orientation of the tunnel restricted early detection, and in conditions like vehicle swaying and high light, which temporarily affected the camera.

The solution developed represents a significant advancement in operational safety for off-road environments, striking a balance between accuracy, speed, and robustness in detecting objects in front of the vehicle.

REFERENCES

[1] M. E. Yabroudi, K. Awedat, R. C. Chabaan, O. Abudayyeh, and I. Abdel-Qader, "Adaptive DBSCAN LiDAR Point Cloud Clustering For Autonomous Driving Applications," *2022 IEEE International Conference on Electro Information Technology (eIT)*, Mankato, MN, USA, 2022, pp. 221-224, doi: 10.1109/eIT53891.2022.9814025.

[2] Y. Wang, B. Wang, X. Wang, Y. Tan, J. Qi and J. Gong, "A Fusion of Dynamic Occupancy Grid Mapping and Multi-object Tracking Based on Lidar and Camera Sensors," *2020 3rd International Conference on Unmanned Systems (ICUS)*, Harbin, China, 2020, pp. 107-112, doi: 10.1109/ICUS50048.2020.9274841.

[3] L. Zheng, P. Zhang, J. Tan and F. Li, "The Obstacle Detection Method of UAV Based on 2D Lidar," in *IEEE Access*, vol. 7, pp. 163437-163448, 2019, doi: 10.1109/ACCESS.2019.2952173.

[4] A. Elfes, "Sonar-based real-world mapping and navigation," in *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 249-265, June 1987, doi: 10.1109/JRA.1987.1087096.

[5] A. Elfes and L. Matthies, "Sensor integration for robot navigation: Combining sonar and stereo range data in a grid-based representation," *26th IEEE Conference on Decision and Control*, Los Angeles, CA, USA, 1987, pp. 1802-1807, doi: 10.1109/CDC.1987.272800.

[6] Sohan, M., Sai Ram, T., Rami Reddy, C.V. (2024). A Review on YOLOv8 and Its Advancements. In: Jacob, I.J., Piramuthu, S., Falkowski-Gilski, P. (eds) Data Intelligence and Cognitive Informatics. ICDICI 2023. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-99-7962-2_39

[7] LI, Yuhang; WU, Shuang; HU, Wei. A Geometry-Based Point Cloud Reduction Method for Mobile Augmented Reality System. *Journal of Computer Science and Technology*, 2018. DOI: 10.1007/s11390-018-1879-3

Table 1 Oval Circuit with 1 Object

Test Scenario	Object	Zone	Distance (m)	Angle (°)	Object width (m)	Type
Oval Circuit with 1 Object at 10 km/h	Balloon Car	**T4**	35.345	-2.0	1.233	car
			34.933	-2.0	1.219	car
			35.190	-3.0	1.535	car
			37.349	-2.0	1.303	unknown
			38.064	-2.0	1.328	unknown
		$\bar{\mu}$	36.176	-2.2	1.323	-
		$\boldsymbol{\sigma}$	1.427	0.4	0.126	-
		T3	29.961	-2.0	1.304	unknown
			29.949	-1.0	1.31	car
			29.978	-2.0	1.309	car
			29.954	0.0	1.309	car
			29.967	-1.0	1.311	car
		$\bar{\mu}$	29.961	-1.2	1.308	-
		σ	0.011	0.8	0.003	-
		T2	19.978	-1.0	1.389	car
			19.891	-2.0	1.395	car
			19.940	-2.0	1.389	unknown
			19.876	-2.0	1.395	car
			19.999	-3.0	1.385	car
		$\bar{\mu}$	19.936	-2.0	1.391	-
		σ	0.053	0.6	0.004	-
		T1	9.961	-2.0	1.394	unknown
			9.976	2.0	1.389	car
			9.981	-2.0	1.398	car
			9.986	-1.0	1.384	car
			9.994	-2.0	1.382	car
		$\bar{\mu}$	9.979	-1.0	1.389	-
		σ	0.012	1.7	0.006	-

Table 2 The table includes only the mean and standard deviation. To access the complete data, as shown in Table 1, visit: https://github.com/A2Nlu/Tables.git

Test Scenario	Object	Zone		Distance (m)	Angle (°)	Object width (m)
Straight Circuit with 1 Object at 10 km/h	Balloon Car	**T4**	$\bar{\mu}$	39.381	0.0	1.168
			σ	0.734	0.0	0.190
		T3	$\bar{\mu}$	31.976	-0.8	1.307
			σ	0.019	0.4	0.001
		T2	$\bar{\mu}$	19.963	-0.4	1.288
			σ	0.014	0.5	0.094
		T1	$\bar{\mu}$	9.987	0.2	1.392
			σ	0.005	0.8	0.003
Oval Circuit with 1 Object at 15 km/h	Balloon Car	**T4**	$\bar{\mu}$	35.214	-2.2	1.28
			σ	2.902	0.5	0.031
		T3	$\bar{\mu}$	29.021	-2.0	1.306
			σ	0.056	0.7	0.001
		T2	$\bar{\mu}$	19.967	-2.0	1.389
			σ	0.049	0.6	0.008
		T1	$\bar{\mu}$	9.984	-2.0	1.391
			σ	0.005	0.7	0.007
Oval Circuit with 2 Objects at 10 km/h	Puppet	**T4**	$\bar{\mu}$	35.866	-1.6	0.000
			σ	0.667	1.9	0.000
		T3	$\bar{\mu}$	29.453	-1.6	0.255
			σ	0.468	1.1	0.003
		T2	$\bar{\mu}$	19.950	-0.2	0.243
			σ	0.052	0.4	0.094
		T1	$\bar{\mu}$	9.986	-0.6	0.226
			σ	0.007	1.1	0.030
	Balloon Car	**T4**	$\bar{\mu}$	38.265	-2.2	1.330
			σ	0.609	0.4	0.003
		T3	$\bar{\mu}$	29.965	-1.0	1.306
			σ	0.018	0.0	0.001
		T2	$\bar{\mu}$	19.988	-0.8	1.393
			σ	0.009	0.4	0.003
		T1	$\bar{\mu}$	9.984	-0.6	1.389
			σ	0.005	0.5	0.005
Oval Circuit with 1 Object on the curve at 10 km/h	Balloon Car	**T1**	$\bar{\mu}$	5.012	-23.2	1.321
			σ	0.009	1.9	0.009
		ZE	$\bar{\mu}$	4.31	-27.2	1.298
			σ	0.784	6.8	0.015