IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

IEEE VEHICULAR TECHNOLOGY SOCIETY SECTION

Received 14 August 2023, accepted 28 August 2023, date of publication 1 September 2023, date of current version 7 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3311136

RESEARCH ARTICLE

Increasing Safety of Automated Driving by Infrastructure-Based Sensors

THIAGO DE BORBA[1,2], ONDŘEJ VACULÍN[1], HORMOZ MARZBANI[2], AND REZA NAKHAIE JAZAR[2]

[1]CARISSMA Institute of Safety in Future Mobility (C-ISAFE), Technische Hochschule Ingolstadt, 85049 Ingolstadt, Germany
[2]Department of Mechanical and Automotive Engineering, School of Engineering, Royal Melbourne Institute of Technology (RMIT), Melbourne, VIC 3000, Australia

Corresponding author: Thiago De Borba (thiago.deborba@thi.de)

This work was supported in part by the German Research Foundation; in part by the Open Access Publication Fund of Technische Hochschule Ingolstadt; and in part by the Project IN2Lab through the Bayerisches Staatsministerium für Wirtschaft, Landesentwicklung und Energie under Grant DIK-2003-0011 and Grant DIK0186/01.

ABSTRACT This paper describes the development of an intelligent infrastructure, a test field, for the safety assurance of automated vehicles within the research project Ingolstadt Innovation Laboratory (IN^2Lab). It includes a description of the test field architecture, the RoadSide Units (RSU) concept based on infrastructure-based sensors, the environment perception system, and the mission control system. The study also proposes a global object fusion method to fuse objects detected by different RSUs and investigate the overall measurement accuracy obtained from the usage of different infrastructure-based sensors. Furthermore, it presents four use cases: traffic monitoring, assisted perception, collaborative perception, and extended perception. The traffic monitoring, based on the perception information provided by each roadside unit, generates a global fused object list and monitors the state of the traffic participants. The assisted perception, using vehicle-to-infrastructure communication, broadcasts the state information of the traffic participants to the connected vehicles. The collaborative perception creates a global fused object list with the local detections of connected vehicles and the detections provided by the roadside units, making it available for all connected vehicles. Lastly, the extended environment perception monitors specific locations, recognizes critical scenarios involving vulnerable road users and automated vehicles, and generates a suitable avoidance maneuver to avoid or mitigate the occurrence of collisions.

INDEX TERMS Automated vehicles, infrastructure-based sensors, safety, test field.

I. INTRODUCTION

The main contributions of Cooperative Connected and Automated Mobility (CCAM) are increasing safety and driving comfort. Besides that, CCAM collaborates to optimize the traffic flow, reducing congestion and CO_2 emissions, resulting in a more efficient means of transportation with lower stress and higher comfort for the occupants [1]. These benefits are possible due to the wide advancement of new technologies such as new sensors, communication protocols, computational power, and the mature development of algorithms for functional pieces of automation, e.g., perception, planning, decision, and control.

Automated Vehicles (AVs), also known as self-driving or driverless vehicles, have the potential to improve road safety by reducing human driving errors [2]. They are capable of sensing their environment and operating without human involvement [3]. Due to rapid technological development, automated driving is a reality on public roads today. However, the idea of having systems capable of assisting the driving tasks is centenary. The first attempt toward driverless vehicles dates back to the early 1920s. The driverless vehicles were called ''phantom autos.'' They were remote-controlled by tapping a telegraph key [4]. In 2022, Mercedes launched the S-Class with their newly developed Drive Pilot system,

The associate editor coordinating the review of this manuscript and approving it for publication was Jie Gao.

the first serial production SAE level 3 automated vehicle in Europe, allowing drivers to hand over control to the vehicle and not monitor the road all the time on certain public roads and at certain speeds [5]. After more than a hundred years of development, many changes and discoveries were made. Therefore, the objective has always been to increase driving safety and comfort.

Due to the considerable increase in the insertion of automated vehicles in the global market, the safety of automated vehicles has been widely discussed among academia, governmental organizations, stakeholders, and OEMs. Questions about how the road infrastructure should be improved for the arrival of this new technology must be clarified to enable full acceptance by the customers and society and for the preparation of the mobility of future cities. The fundamental architecture of automated vehicles consists of perception, planning, decision, and actuation. The perception system is responsible for understanding the environment in which the vehicle is inserted and relies mainly on the onboard sensors. The most used sensors for environment perception include ultrasonic sensors, cameras, LiDARs, and radars [6]. Despite the continuous improvement and optimization of the perception systems due to technological advancements, the system's performance is still limited by the sensors' parameters, such as range, accuracy, field of view, and target reflectivity, and also due to the occurrence of occlusion, and degradation of sensor data, under certain weather conditions [7]. Moreover, comprehending the surroundings is challenging once it is unpredictable and continuously changing. One solution to minimize the onboard sensors' limitations is using infrastructure-based sensors in the form of an intelligent infrastructure unit, also known as a roadside unit, which can be installed in specific locations and perceive the environment from a different perspective with a higher detection range and field of view.

Our study has four main research contributions:

- a detailed description of a smart infrastructure for the safety assurance of automated vehicles.
- a novel global object fusion method for fusing objects detected by different RSUs.
- an investigation of the overall measuring accuracy obtained from the usage of different infrastructure-based sensors.
- the presentation of relevant use cases and functionalities to ensure the safe operation of automated vehicles.

This paper is structured as follows: Section II presents the related works. Section III provides an overview of the test field architecture. Section IV describes the concept of a roadside unit. Section V provides information regarding environment perception. Section VI presents the mission control system and describes the use cases. Section VII presents the validation of the traffic monitoring use case, and section VIII concludes and gives an outlook on future work.

II. RELATED WORKS

In the last few years, many research institutions around the world have started to develop test fields for the assurance of automated driving safety and for the development and validation of automated driving functions. For example, the Martin Luther King (MLK) smart corridor presented in [8] deployed in the city of Chattanooga, USA; the ACCorD corridor for new mobility proposed by the University Aachen (RWTH) [9] set up in Aachen, Germany; test bed lower Saxony described in [10] deployed in different motorways in the north of Germany; and the test field autonomous driving Baden-Württemberg [11] deployed in Karlsruhe, Germany. The test field described in this study, developed within the project Ingolstadt Innovation Laboratory (IN^2Lab), was deployed in Ingolstadt, Germany [12]. In contrast to the previously presented studies, besides employing infrastructure-based sensors to monitor road traffic, the proposed infrastructure also provides redundant environment information to the connected vehicles and utilizes local detections of the vehicles to enhance the overall system perception capabilities.

The implementation of sensor-equipped RSUs in a real-world infrastructure has also been addressed by some authors in the literature. Correia et al. [13] described the Collective Perception Service (CPS) implementation, intending to provide additional perception information to connected vehicles and to a central road operator to reduce uncertainty in the road environment. In this study, the RSUs are equipped with cameras and radars. The Collective Perception Messages (CPMs) are generated based on the information provided by the radars only, and the messages are broadcasted locally and to a cloud Message Queuing Telemetry Transport (MQTT) broker through ITS-G5. One limitation is that the proposed system relies on one sensing modality. With the implementation of different sensors and sensor data fusion, the CPMs could be enhanced with additional detection information and a higher confidence level.

Tsukada et al. [1] proposed a system that combines RSUs equipped with LiDARs and APU4C4-embedded routers to create an infrastructure-based cooperative perception system. The cooperative perception is realized by a software called AutoC2X, a combination of Autoware, an open-source autonomous driving stack containing perception algorithms, and OpenC2X, responsible for generating Cooperative Awareness Messages (CAMs) and CPMs. The authors prioritize CPMs corresponding to areas closer to the receivers to avoid an overload of broadcast capacity. The experiments revealed that the proposed system presented low latency utilizing Wi-Fi communication, even in the worst cases.

Shan et al. [14] investigated the usage of Collective Perception (CP) service within intelligent infrastructure to improve awareness of vulnerable road users (VRUs) and increase safety for connected AVs in different traffic scenarios. The RSU consisted of a tripod with sensors, e.g., cameras, LiDAR, and radar, a processing unit, and a Cohda Wireless MK5 RSU. After sensor data fusion, the perceived

objects were transformed to local coordinates, encoded into European Telecommunications Standard Institute (ETSI) CPMs, and broadcast by a Cohda V2X unit at 10 Hz. The proposed RSU was deployed in an intersection. During the investigation, the received perception data from the RSU was used by the AV as the only source of information for multiple road users' detection. According to the experiments, the AV could perceive the ongoing traffic activity far beyond the reach of its onboard sensors, even occluded objects, which demonstrates the improvement of the sensing capabilities of the AV. However, the authors deployed a very limited coverage system with only a single RSU, and no strategy for global object fusion was presented.

III. TEST FIELD ARCHITECTURE

The test field "First Mile" in Ingolstadt, Germany, consists of public road infrastructure, which enables the automated driving vehicles to operate in a real-world mixed traffic environment. The test field connects the exit from Highway A9 with the technology park IN-Campus. It consists of a bidirectional road, approximately 2 km long, including intersections, and urban features, such as bus stops, pedestrian crossing with traffic lights, parking lots, bike paths, and sidewalks. A top view is shown in Figure 1. The test field is equipped with eleven roadside units with an approximate distance of 200 m from each other as indicated in Figure 1 by the yellow dots.

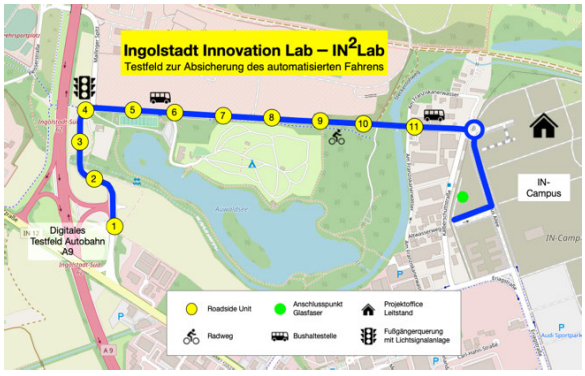

FIGURE 1. IN²LAB test field.

Each roadside unit is equipped with a specific combination of sensors connected to a switch through Ethernet or CAN bus. Similarly, Ethernet wiring enables the connection between the application unit and the switch. The communication between each sensor driver and the local data processing unit takes place in Robot Operating System (ROS) environment employing ROS messages. ROS messages are the primary container for exchanging data in ROS environment. They are data structures comprising typed fields. Here, standard primitive types and arrays of primitive types are supported, including integer, floating point, and Boolean, among others [15]. Afterward, the sensors' raw data are processed locally in the application unit by the environment perception. As a result, a local object list, including the detected traffic participants, is provided. An overview of the test field architecture is illustrated in Figure 2.

The application units are physically connected to the mission control server with an optic fiber network, and the communication between them occurs based on SENSORIS messages. SENSORIS (Sensor Interface Specification) is a standardized interface to exchange information between sensors and a dedicated cloud, which implements Google's protocols buffer for message serialization [16]. In mission control, the global data processing of the local object list provided by each roadside unit takes place. Here, the same objects detected by different RSUs are fused, and a global object list is compiled. Lastly, the global object list is made available to the use cases and data storage through ROS messages or Collective Perception Messages (CPM), an advanced service to distribute safety information between vehicles and infrastructure using vehicle-to-x or vehicle-to-everything communication units [17].

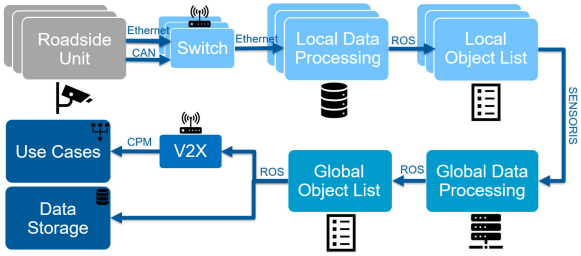

FIGURE 2. Test field architecture.

IV. ROADSIDE UNIT CONCEPT

The roadside units are responsible for road traffic surveillance, processing all infrastructure-based sensor data, and providing mission control with the local object lists. Each RSU comprises three main components: A pole structure of approximately 4.5 meters in height, providing mechanical support; the set of sensors, including different sensing modalities, e.g., vision and ranging sensors; and a control cabinet equipped with an application unit, for local data processing, switches, and power supplies.

In general, the sensors allow the infrastructure and the automated vehicle to detect their surroundings and, after processing, understand the environment they are located [18]. Individual sensing modalities present strengths and weaknesses, due to their physical measurement principle. Cameras are able to detect colors and textures and provide high-definition images, which are essential for object detection. However, they are sensitive to low light intensity and are affected by adverse weather conditions [19]. LiDARs provide higher robustness against unstable illumination and demand less computational power when compared to cameras [20]. However, the high cost and the performance degradation in adverse weather conditions, e.g., fog, snow, and rain, are some of the limitations faced by this technology [18]. Radars provide high-accuracy distance assessment, direct relative velocity measurement through Doppler shift, reduced

cost, and robustness in adverse weather conditions. However, the drawbacks include difficulty distinguishing stationary objects, receiver saturation if a large object is too close to the transmitter, and significantly lower spatial resolution compared to LiDARs [20].

Based on the sensor's strengths and weaknesses, the RSUs were designed with a flexible combination of different sensing modalities. Thus, the number of employed sensors varies according to the mast location and the required coverage. The most equipped RSU presents seven sensors: two cameras with 16 mm lens, one camera with a fish-eye lens with a field of view up to 180 degrees, two LiDARs, and two radars. Moreover, an overview of the employed sensors and some of their physical characteristics are presented in Table 1. As illustrated in Figure 3, each pair of sensors covers the right and left side of the mast, and the fish-eye camera covers its frontal area. Additionally, all roadside units are equipped with an application unit with parallel processing capabilities and some with a vehicle-to-x communication unit to enable direct communication between the infrastructure and the vehicles within the test field area. Moreover, the local application units can process one 2.35 MP camera operating at 30 frames per second, without overloading the graphics processing unit. As the most equipped RSUs have three cameras, a common frame rate of 10 Hz was set for all cameras in the test field.

TABLE 1. Infrastructure-based sensors and vehicle-to-x communication unit.

Device	Manufacture	Model	Field of View	Range	Frame Rate
Camera	IDS	UI-5260CP C-HQ Rev2	V 33.6° H 44.3°	-	10 Hz
LiDAR	Blickfeld	Cube Range 1	V 12° H 18°	250 m	4.5 Hz
Radar	Continental	ARS 408 LRR	H 18° far H 120° near	250 m 20 m	13.8 Hz
V2X	Commsignia	ITS-RS4	-	1000 m	-

FIGURE 3. Roadside unit concept.

V. ENVIRONMENT PERCEPTION CONCEPT

In automated driving, the environment perception module performs crucial tasks to guarantee the safe operation of the vehicle, including detecting traffic participants, interpreting traffic signalization, and comprehending any unexpected changes in the driving scenario. Similarly, the infrastructure environment perception collects and processes sensors' data to detect and understand the characteristics of the environment. Among its various functions, the object detection identifies the presence of traffic participants, measures their current state, and performs classification. It utilizes different detection methods according to the sensing modality employed. For instance, camera-based, LiDAR-based, and radar-based object detection can be implemented.

A. CAMERA-BASED OBJECT DETECTION

Cameras are devices equipped with image sensors capable of detecting the light reflected by objects, capturing wavelengths corresponding to the visible portion of the spectrum, such as the red, green, and blue wavelengths (RGB), creating high-resolution images with accurate color representation [21]. They have been widely used for object detection in automated driving because of their relatively lower costs, compared to other sensing modalities, and their ability to obtain shapes, textures, and colors. This enables the recognition of traffic participants, traffic lights and signs, lane and pavement markings, etcetera. From the camera's raw data, algorithms based on convolution neural networks can detect objects with high positioning and classification accuracy in real-time applications.

An approach called You Only Look Once (YOLO), first presented in 2016 by Redmon et al. [22] was implemented in its version 4. YOLOv4 was implemented because the algorithm features a simple and optimized pipeline that enables real-time processing at high frame rates with high-accuracy detection [23]. YOLO consists of 24 convolutional layers to extract features from the image and two fully connected layers to predict object labels. In one evaluation, a single trained neural network predicts bounding boxes and class probabilities to detect objects from image pixels. This network globally reasons the entire image and all objects and divides it into an $S \times S$ grid. Thus, if the object's center falls into a grid cell, that cell is used for detecting the respective object. Each cell predicts B bounding boxes and confidence scores, reflecting how confident the model is that the box contains an object and also how accurate the algorithm considers the box prediction. YOLO uses single-bounding box regression to predict the parameters of the box. The bounding boxes consist of 5 predictions: (x, y) coordinates of the center of the box relative to the bounds of the grid cell, (w, h) width and height, and the confidence prediction that represents the intersection-over-union between the predicted box and the ground truth box. Furthermore, each grid cell also predicts the conditional class probabilities. In operation, the conditional class probabilities and the individual box confidence predictions are multiplied, resulting in the class-specific scores, which encode both the likelihood of the class appearing in the box and how well the predicted box fits the object [24].

Once for the posterior sensor data fusion, a single point representing the object location is required, the image-based

detection delivers an object list consisting of all detected objects in the scene with their respective state and classes, considering the middle-bottom point of each bounding box as the object position. For example, the front or rear middle bottom point of a vehicle is defined as the vehicle's location, depending on the vehicle's direction relative to the image sensor.

In the first moment, we trained a YOLOv4 neural network based on the COCO, an open-source dataset containing more than 320.000 labeled images with 91 different classes [25]. However, the cameras installed in the infrastructure present a completely different perspective compared to the cameras' positions used for data collection in the COCO dataset. Moreover, in this project, the camera-based object detection mainly focuses on detecting only six classes: pedestrian, cyclist, motorcyclist, car, bus, and truck. Hence, the network was retrained to improve its results for these specific classes. After the data collection, 4875 images were labeled according to the class definition of COCO names and used as input for retraining. Comparing the performance of the previous and the retrained neural network with a new dataset showed a significant improvement in the class confidence level and a considerable reduction in misclassification. Another feature obtained was the detection of motorcyclists and cyclists as single objects instead of multiple detections, as presented in Figure 4. The upper part of Figure 4 presents the detection of a cyclist with the previous neural network in multiple objects, such as a bicycle, person, and backpack. However, in the lower image, the newly retrained network detects the cyclist as only one object with higher classification confidence. Once the COCO class definition does not include the class cyclist, the cyclists are classified, in the first moment, as bicycles and changed to a cyclist in the posterior processing steps.

FIGURE 4. YOLOv4 neural network retraining.

B. LiDAR-BASED OBJECT DETECTION

A comprehensive and detailed representation of the environment is fundamental for accurate object detection in automated driving. Unlike 2D object detection solely based on flat image data, sensors with spatial sensing ability, such as LiDARs, have the benefit of detecting objects with additional information. Thus, from LiDAR point clouds, obtained from the transmitting and receiving laser pulses in the scanning range, the 3D object detectors provide a reliable estimation of the objects' size and precise location [26]. The advances in deep learning with publicly accessible datasets, have a positive impact on the 3D object detection task, resulting in several emerged LiDAR-based 3D object detectors. A common ground among all LiDAR-based 3D object detectors with deep learning can be established, including LiDAR Sensor Data Representation (SDR), feature extraction, and core object detection [27].

In LiDAR SDR, the incoming point clouds, with unstructured form and non-fixed size, can be transformed into a structured and compact representation by utilizing mainly five distinct representations: point-based, voxel-based, pillar-based, graph-based, and projection-based [27]. Thus, the infrastructure LiDAR-based object detection implements the voxel-based representation. The voxel-based methods discretize the 3D space into fixed-size voxel grids [26]. Voxelization is the process of assigning points to voxels. This method partitions the 3D space according to a Cartesian or cylindrical coordinate frame, resulting in a voxel of a cuboid or cylindrical slice shape [28]. In this study, a voxel of a cuboid with a size of 125 cm^3 was implemented. Therefore, after its implementation, all points belonging to a grid are represented by one point located in the center of the voxel cuboid grid. This method samples the point cloud and significantly reduces its size. Before implementing the structured data representation, a background removal filter is utilized to remove all points that remain static after the first 200 frames. This method facilitates and speeds up the clustering of the remaining points. The background extraction is possible because the infrastructure-based LiDARs are static.

After transforming the point cloud into a structured and compact representation, the feature extraction module extracts rich and high-dimensional features. Here, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm interacts through the structured point cloud and clusters points belonging to the same objects. The main idea behind DBSCAN is that a point belongs to a cluster if its relative distance is smaller than a threshold [29]. There are two main parameters of DBSCAN: The distance threshold to specify the neighborhoods and the minimum number of data points to define a cluster. In this case, if the Euclidean

distance between two points is smaller than 2 meters, the points belong to the same cluster. If two objects are less than 2 meters apart, LiDAR-based object detection will merge the two objects. However, this will not occur in camera and radar-based detection. Thus, this is compensated in the sensor data fusion stage, and camera and radar detections are prioritized. Furthermore, three points are needed to constitute a cluster. In order to facilitate cluster representation, each cluster is represented by its center front bottom point. This point is later used to extract the state of the object.

After clustering, high-dimensional features can be extracted, such as regression values regarding the object class, size, and location of a 3D bounding box and classification confidence. However, once the classification task is performed in the camera-based object detection, only the cluster single point state information is required. The LiDAR-based object detection does not provide class and 3D bounding box information. Afterward, the object list containing all LiDAR-based detected objects is forwarded to sensor data fusion.

C. RADAR-BASED OBJECT DETECTION

Automotive radars play an important role in the environmental perception of automated vehicles. They have been widely applied on production vehicles with lower SAE automation levels, reaching a market penetration of millions of units sold. One of the main reasons for the success story of automotive radar is its physical principle that offers unique performance features at reasonable costs. Radar electromagnetic waves can penetrate smoke, fog, and dust, proportioning considerable robustness against adverse weather in different lighting conditions. Moreover, radars can detect long-range targets, up to 250 m, and directly measure the targets' relative speed, with a resolution of up to 0.1 m/s. These characteristics are indispensable for automated vehicles' motion prediction and driving decisions [30], [31].

According to related research dealing solely with radar-based object detection, there are mainly two approaches for object detection, including classification. One uses radar-based grid maps with deep learning approaches, and another employs clustering and classification of original radar point clouds with deep learning approaches [30], [32]. In the first method, radar-based grid maps are determined by accumulating multiple data frames. Afterward, segmentation networks are employed to process radar-based grid maps similar to image processing. Thus, from the grid maps, the classification and the definition of the orientation of static traffic targets become possible [30]. In the second method, the radar point cloud is processed in its original form and omits a mapping step. The radar's reflections are prefiltered by a Constant False Alarm Rate (CFAR) filter and clustered in space, time, and Doppler by using a DBSCAN clustering algorithm. Afterward, features are extracted from each cluster, and the feature vectors are used as input to a neural network for classification [32]. Hence, the here selected infrastructure-based radars provide a detection list as output instead of raw data, and the object classification is obtained from camera-based object detection after sensor data fusion. For these reasons, a simplified radar-based object detection method was implemented.

The employed Continental ARS408 LRR is a long-range radar that provides positional measurements in 2D coordinates. The sensor gives a detection list as output, where each detection state is defined with range, azimuth (horizontal angle), Doppler speed, and radar cross-section (reflectivity of the object). As the provided radar sensor measurements are typically noisy, many undesired reflections from the ground and other static objects are available within the sensor's field of view. Hence, to remove these static detections, a velocity filter of 0.3 m/s (absolute value) is used to remove noise and ground points. After filtering, the DBSCAN clustering is applied. A minimum of 2 points per cluster and a distance threshold of 3.5 meters to cluster the detections belonging to the same object are selected. Then, for each generated cluster, the minimum value of x, the mean value of y, the mean value of Doppler speed, and the mean value of radar cross-section are calculated from the detections belonging to that cluster to represent the object state. In the end, the radar object list is forwarded to sensor data fusion.

D. OBJECT-LEVEL SENSOR DATA FUSION

Due to the previously mentioned sensors' limitation, the safe operation of automated vehicles, in all operational design domains and weather conditions, cannot be guaranteed with only one sensing modality. A reliable solution is the implementation of multiple sensors to generate a combined output that provides several benefits, such as higher measurement accuracy, by compensating errors and limited operating ranges of individual sensors, reliable operation in adverse weather conditions, and higher resolution output ideal for posterior feature extraction, among others [33]. Through sensor fusion, by combining the strengths of each sensor, the overall performance of the system is enhanced [19]. However, if one sensor provides less accurate measurements for a specific parameter, the more accurate measurements from another sensing modality may be degraded during sensor fusion. This will be discussed in more detail in the conclusions.

Sensor calibration is a requisite processing step before implementing sensor data fusion. There are three categories of sensor calibration: intrinsic, extrinsic, and temporal calibration [34]. The intrinsic calibration estimates the internal or intrinsic parameters of the sensor, e.g., the focal length of a camera, that correct systematic or deterministic errors. The extrinsic calibration estimates the position and orientation of the sensors relative to the three orthogonal axes of the 3D space with respect to an external frame of reference, usually the vehicle frame [20]. Each sensor has its specific coordinate frame. In order to fuse data from different sources, the data needs to be in a common frame. The extrinsic calibration process is a procedure to find the relationship between the coordinates of sensor frames, the calibration parameters

rotational matrix, and the translation vector, to enable the transformation from one frame to another [35].

Currently, three sensor combination forms are prevalent for environment perception, including camera-LiDAR (CL), camera-radar (CR), and camera-LiDAR-radar (CLR). The combination CR is the most employed for multi-sensor fusion for automated driving, followed by CLR and CL. The CR combination provides high-resolution images with additional accurate distance and velocity information of surrounding objects. This combination is widely employed by Tesla, which also includes ultrasonic sensors for short-range detections [36]. Similarly, the sensor combination CLR provides high resolution at a greater range, precise representation of the environment features through the point clouds, accurate velocity and position information, high-resolution images ideal for environment interpretation, etc. This combination was actually implemented in the Mercedes S-class level 3 automated vehicle [37]. Additionally, if two or more sensors are employed for the same task, the sensor fusion also improves the safety redundancy of automated systems [38].

In Multi-Sensors Data Fusion (MSDF), there are three main approaches to combine sensory data from different sensing modalities: High-level Fusion (HLF), Low-level Fusion (LLF), and Mid-level Fusion (MLF) [39]. In HLF or object-level fusion, each sensor's raw data is processed separately. The raw data passes through an object detection and coordinate transformation step and the sensor fusion is performed subsequently. The HLF approaches are often adopted due to lower relative complexity and less computational requirement than LLF and MLF approaches. However, it can provide incomplete information as classifications with a lower confidence value are discarded when there are several overlapping obstacles. In the LLF approach, the data from each sensor are fused at the lowest level of abstraction, raw data. In this case, all information is available and can potentially improve the object detection accuracy [20]. In practice, LLF is complex and comes with several challenges. It requires an accurate extrinsic calibration of the sensors, and the raw needs to be time-synchronized and compensated for vehicle motion [39]. As a result, LLF has the potential to improve the detection accuracy and reduce latency where the domain controller does not have to wait for the sensor to process the data before acting upon it. In contrast, the MLF or feature-level fusion fuses multi-target features extracted from the raw data, such as color information from images and location features from LiDARs and radars, and subsequently performs object detection based on the fused multi-sensor features. The MLF approach also provides a powerful feature selection technique that detects corresponding features and feature subsets to improve the recognition accuracy [20].

As mentioned, in object-level fusion, each sensor's raw data is processed separately. The perception results of single sensors are then matched and fused, improving the resulting confidence and accuracy for a further tracking step. In this context, data association is required to match the perception results of single sensors. Frequently used algorithms for data association include the Global Nearest-Neighbour (GNN), Probabilistic Data Association (PDA), and Joint Probabilistic Data Association (JPDA). Moreover, state filters such as Extended Kalman Filters (EKF) and Unscented Kalman Filters (UKF) are usually applied to solve the problem of multi-sensor multiple object tracking. Thus, through joint calibration, the conversion between the spatial relation of the two sensors is established [28].

In the IN2LAB project, the infrastructure-based sensors operate in different frame rates, as presented in Table 1. For instance, to have a reasonable number of scan lines, the LiDARs operate at 4.5Hz. To avoid the processing overload of graphic cards during camera-object detection, the camera operation is limited to 10Hz. Moreover, the radars operate at 13.8Hz. Since for the LLF or MLF the sensor synchronization is crucial, all sensors would have to operate at the same frame rate, or the frame rate would have to be reduced to the lowest value. By doing so, lots of relevant information would be discarded, and the system operation would be limited to 4.5 Hz, which is not enough for a safety-critical system. For this reason, the object-level sensor data fusion was implemented.

FIGURE 5. Object detection and sensor data fusion.

The implemented multi-sensor object-level sensor data fusion has an adaptable modular design [40]. It does not rely on the number and type of sensors, sensor synchronization is not required, and it has a straightforward operation. The perception results of single sensors are transformed for the same roadside unit reference frame, based on the rotational matrix and translation vector obtained from sensor calibration, and assigned to the Unscented Kalman filter. Afterward, the Hungarian association method associates incoming sensor measurements from different sensing modalities. This association method is an optimization process where the overall cost, the Euclidean distance between existing objects and new measurements, has to be minimized [40]. Thus, a new object is created when detected in the image for three consecutive frames and is not associated with any existing object. Moreover, the new measurements are assigned to the existing objects as soon as they arrive, enabling a maximum overall operating frequency of 28.3 Hz. Figure 5 shows an example of the sensor data fusion output. In Figure 5, two

classes are detected: car and bicyclist. The blue bounding boxes represent camera-based detections, and the green and grey dots represent LiDAR-based and radar-based detections. In the end, the local object list of each roadside unit, containing fused measurements in local coordinates, is provided to mission control for global processing.

VI. MISSION CONTROL SYSTEM CONCEPT

Mission Control System (MCS) is the central hub of the test field, in which all local object lists provided by the roadside units are fused, and the global object list is compiled [41]. The MCS is mainly responsible for processing the local object detections, performing the global object fusion, and providing environment perception information to connected automated vehicles within the test field. The MCS server presents a scalable architecture. Therefore, in case of further extension of the test field, it is possible to extend CPU, GPU, memory, and storage. Currently, all global processing occurs in an NVIDIA RTX Server ASUS with AMD 24 cores processors and 2 Nvidia GeForce RTX 3090 graphics cards. A scalable storage server currently offers 88 Terabytes of storage space. The main functions of MCS are subdivided into four use cases: traffic monitoring, assisted perception, collective perception, and extended perception.

From the perception information provided by each roadside unit, the traffic monitoring generates a global fused object list and monitors the state of the traffic participants. The assisted perception, using vehicle-to-infrastructure communication, broadcasts the state information of the traffic participants, obtained in the traffic monitoring, to the connected vehicles. The collaborative perception extends the traffic monitoring and generates a global fused object list based on the detections of connected vehicles and roadside units, making it available for all connected vehicles. Moreover, the extended environment perception monitors predefined locations, recognizes critical scenarios involving vulnerable road users and automated vehicles, and generates a suitable avoidance maneuver to avoid or mitigate the occurrence of collisions.

A. TRAFFIC MONITORING

One of the main functions of the MCS is traffic monitoring. The traffic monitoring allows system users to have an overview of vehicles and vulnerable road users, maintaining unique IDs within the entire test field. As mentioned, the MCS receives local object lists from individual roadside units. These local object lists contain state information about vehicles and vulnerable road users, e.g., pedestrians, bicycles, cars, buses, and trucks. Hence, each detected object is described by a set of parameters, including object ID, roadside unit ID, class, position with respect to the roadside unit in cartesian coordinates, longitudinal and lateral velocity, heading angle, and detection confidence, among others. This information is collected and fused to create a global object list. In the transition region between two masts, single objects can be detected by different roadside units simultaneously. Thus, the fusion of local detections allows tracking of the traffic participants along the whole test field and avoids double detections. The traffic monitoring presents three main tasks: coordinate transformation, global object fusion, and visualization of road traffic in a 2D map, an overview is presented in Figure 6.

1) COORDINATE TRANSFORMATION

Each sensor presents its reference coordinate system with a specific axis distribution. In order to have a common coordinate system among all sensors, a local roadside unit coordinate system was defined, and all sensors' measurements were transformed to this common reference. However, the test field comprises eleven roadside units, and tracking traffic participants between them is only possible by defining a unique reference coordinate system. Hence, the global coordinate system is based on a map. The map was obtained by exporting a delimited region of the OpenStreetMap containing the entire test field. Afterward, the SUMO net convert function generated the road networks to describe the traffic-related part of a map [42]. The road networks contain positional information of all assets in cartesian and geographic coordinates. Moreover, only road assets such as lanes, road limits, pedestrian crossing, traffic lights, stop lines, and sidewalks are maintained for posterior visualization.

In order to perform the coordinate transformation between two coordinate systems, an approach similar to the extrinsic sensor calibration process was implemented. The transformation from local to global coordinates is presented in (1). Thus, by finding the transformation parameters, rotational matrix, and translation vector, the state of a detected object can be transformed from one reference frame into another [35]. The transformation of coordinates ${r_{(x,y,z)}}^{local}$ from the local coordinate system to the coordinates ${r_{(x,y,z)}}^{global}$ in the global coordinate system can be written as follows:

$$ {r_{(x,y,z)}}^{global} = T * {r_{(x,y,z)}}^{local}, \tag{1} $$

where T is the transformation matrix consisting of a rotational submatrix $R = R_z(\gamma)R_y(\beta)R_x(\alpha)$ and translation vector ϱ:

$$ \begin{aligned} T &= \begin{bmatrix} R & \varrho \\ 0_{1\times 3} & 1 \end{bmatrix}, \\ &= \begin{bmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & x \\ \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & y \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta & z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{2} $$

where γ, β, and α are the yaw, pitch, and roll angles, and x, y, and z are the respective translations, all measured in the local coordinate frame.

According to (1) and (2), to transform the detected objects' position from local to global coordinates, the rotational

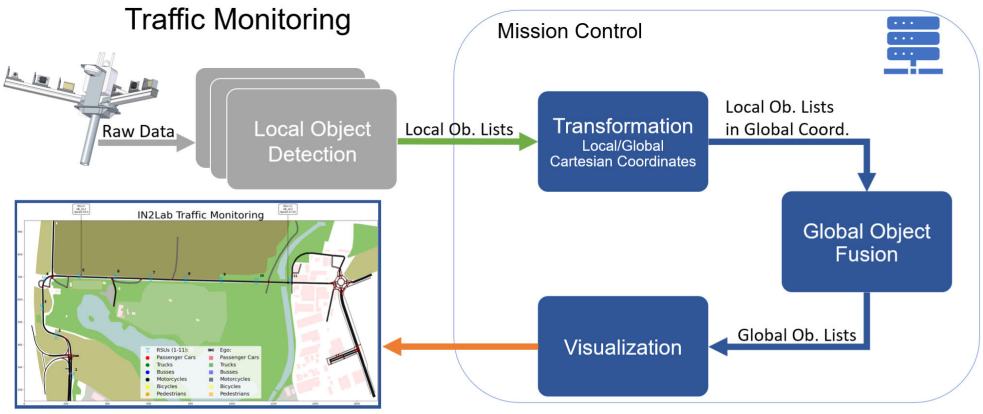

FIGURE 6. Traffic Monitoring.

submatrix R and the transnational vector ϱ must be defined. The transnational vectors include the x, y, and z coordinates of each RSU with respect to the map. In order to obtain the positional information of each RSU, a GNSS device with Real-Time Kinematics (RTK) correction was utilized. The ANavS Multi-Sensor RTK, composed of three antennas and one computational unit, was then used to define the positions in geographic coordinates with an accuracy of up to 1.5 cm. Once the map possesses both cartesian and geographic coordinates, the obtained geographic coordinates could be converted to Cartesian coordinates creating a total of eleven fixed translation vectors.

Moreover, to describe the rotational matrices, another three variables must be defined for each RSU. Figure 7 shows the RSU 6 and its local coordinates system, likewise the global map and its respective orthogonal axis. According to Figure 7, the local and global x-axis have the same direction but with the positive part pointing in opposite directions. The same occurs with the local and global y-axis. Once the environment perception system detects objects on the ground level, neglecting their respective heights, the z coordinates are always considered zero for transformation. By analyzing the arrangement of the x and y axes, it is possible to identify that no rotation around the x and y, roll and pitch, is necessary and can be considered zero. Nevertheless, the rotation around the z-axis (yaw) has to be defined for each RSU. In order to define those angles, the road asset, and lane limits, present in both the real test field and global map have been used. In this case, by detecting a pedestrian walking over the lane limits on both sides of the road near each RSU and plotting its positions in the global map, it was possible to adjust the yaw angles manually to match the pedestrian location with the road limits presented in the map. After analyzing the results, the data collection and manual adjustment were replicated to the others. The RSUs are aligned with the front part pointing to the geographic south. In this case, the RSUs have a heading angle of approximately $179° \pm 1°$ with respect to the geographic north.

After defining the transformation matrices, the positional information of each detected object could be transformed from local to global Cartesian coordinates. This information is crucial for the subsequent global object fusion and the other use cases dealing with the broadcast of the environment detections to the vehicles connected to the infrastructure.

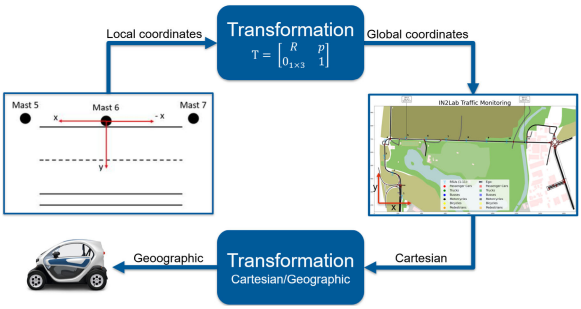

FIGURE 7. Coordinates transformation.

2) GLOBAL OBJECT FUSION

The global object fusion consists of two main tasks: tracking each object detected by the RSUs and their association with the same objects detected by the subsequent RSUs. Different authors have addressed object tracking and association in the literature. Some of these methods will be described next.

Lefévre et al. [43] presented a review of existing methods for tracking, motion prediction, and collision risk assessment addressed to automated vehicles. They segregated motion modeling and prediction into three different groups, the physics-based motion, the maneuver-based motion models, and the interaction-aware motion models, and introduced the Kalman filtering techniques for recursively estimating a vehicle's state. Guo et al. [44] presented a technique for real-time pedestrian tracking in an urban traffic environment with partial occlusions. This method integrated the Camshift algorithm, which detects and tracks objects in a color distribution map, with a Kalman filter to allow the real-time tracking of partially occluded road users. Ellis et al. [45] proposed a non-parametric model for pedestrian motion based on Gaussian process (GP) regression, in which trajectories are modeled by regressing relative motion against the current position. The main idea of the model is the use of Gaussian

processes to estimate the instantaneous velocity of an actor given its current position for long-term path prediction. The authors used a data set containing a set of positions and instantaneous velocity to create independent motion models. The position was assumed to be Gaussian at all time stamps. For this reason, the predictions were performed first by an extended Kalman filter. Thus, by recursively calculating the predicted mean and covariance, the position could be calculated in many steps in the future.

Keller et al. [46] presented an approach for pedestrian path prediction and action classification, focusing on scenarios where an approaching vehicle monitors a crossing pedestrian, who might present different moving models by standing still or continuing walking at the road curbside. Here, the conceit of the Interactive Multiple Model Kalman Filter (IMM-KF) was implemented for pedestrian position estimation. Toledo-Moreo et al. [47] implemented a concept of Interactive Multiple Model Extended Kalman Filter (IMM-EKF) for sensor fusion, combining the information from Global Navigation Satellite System (GNSS), Inertial Navigation System (INS), and odometry sensor to estimate the vehicles' future states. Tao et al. [48] designed a concept of object tracker that utilizes a family of unscented Kalman filters for tracking multiple, irregularly moving objects in 3D space. In this study, the objects were detected in a 2D image plane, an Unscented Kalman Filter (UKF) was implemented for each detected object separately, and the predicted state was used for data association.

The authors in [49] and [50] presented an empirical analysis evaluating the performances of the unscented Kalman filter and extended Kalman filter, utilized as a fusion method for navigation and estate estimation. The results have shown that the UKF presented a slightly more accurate state estimation when applied to nonlinear motions. Therefore, its computational time was higher than the computational time of the extended Kalman filter. Due to the accurate state estimation for linear and nonlinear motion, the unscented Kalman filter was here addressed for object tracking.

a: KALMAN FILTER

The Kalman filter is an optimal linear estimator introduced by Rudolf Emil Kalman in 1960 [51]. The proposed method uses the prior state information, sensor measurements, and kinematics' or transition equations to recursively estimate the optimal current state. It assumes that the sensors' measurements are noisy and the errors are random, following a normal (Gaussian) distribution [49]. A Kalman filter estimates the state of a linear stochastic process $x_t \in \mathbb{R}^n$ using the measurement $z_t \in \mathbb{R}^m$ as

$$\begin{aligned} x_t &= Fx_{t-1} + Bu_{t-1} + \eta_{t-1}, \\ z_t &= Hx_t + \zeta_t. \end{aligned} \tag{3}$$

where: t can be interpreted as the timestamp, $\mathrm{F} \in \mathbb{R}^{m\times n}$ denotes the linear state transition matrix, B is the control-input matrix applied to the control vector u_{t-1}, $\mathrm{H} \in \mathbb{R}^{m\times n}$ is the measurement matrix, and η_t and ζ_t are uncorrelated, zero-mean, and normally distributed process and measurement noises, i.e., both can be represented by a zero mean multivariate normal distribution $\mathcal{N}$ with covariance Q_t and R_t: $\eta_t \sim \mathcal{N}(0,Q_t)$ and $\zeta_t \sim \mathcal{N}(0,R_t)$. Kalman filter algorithm consists of two stages: prediction and update, also called propagation and correction [52]. Thus, in each computational step, the Kalman filter computes the predicted state estimate $\hat{x}_{t_p} \in \mathbb{R}^m$ and the predicted covariance $P_{t_p} \in \mathbb{R}^{m\times n}$

$$\begin{aligned} \hat{x}_{t_p} &= F\hat{x}_{t-1} + Bu_{t-1}, \\ P_{t_p} &= F\hat{x}_{t-1} + F^{\mathsf{T}} + Q_{t-1}. \end{aligned} \tag{4}$$

In the update stage, the measurement residual $\tilde{y}_t$ is computed first as the difference between the true measurement, z_t, and the estimated measurement, $H\hat{x}_{t_p}$. Afterwards, The residual, $\tilde{y}_t$, is multiplied by the Kalman gain, K_t, to provide the correction, $K_t\ \tilde{y}_t$, to the predicted estimate $\hat{x}_{t_p}$. Then, after obtaining the updated state estimate, the Kalman filter calculates the updated error covariance, P_t according to

$$\begin{aligned} \tilde{y}_t &= z_t - H\hat{x}_{t_p}, \\ K_t &= P_{t_p}H^{\mathsf{T}}(R_t + HP_{t_p}H^{\mathsf{T}})^{-1}, \\ \hat{x}_t &= \hat{x}_{t_p} + K_t\tilde{y}_t, \\ P_t &= (I - K_tH)P_{t_p}. \end{aligned} \tag{5}$$

The basic Kalman filter detailed above requires linear state transition and measurement. In order to perform state estimation of nonlinear systems, other variations of Kalman filtering were introduced, e.g., the extended Kalman filter [53] and the unscented Kalman filter [54]. In this work, the UKF was employed for object tracking. One of the reasons is that it captures the posterior mean and covariance more accurately than the third-order Taylor series expansion for any nonlinearity [49], [50].

b: UNSCENTED KALMAN FILTER

The Unscented Kalman filter is an extension of the basic Kalman filter for systems with nonlinear process or measurement equations [55]. This filter is based on the unscented transformation. The idea behind this transformation is that a small set of points is enough to reconstruct a distribution. Thus, the UKF computes a set of weighted samples, the sigma points, and propagates them through the nonlinear function. Afterward, a transformed Gaussian distribution, characterized by their mean and covariance, is reconstructed from the new sigma points [56]. Considering a nonlinear stochastic process, x_t, with uncorrelated, zero-mean, and normally distributed process and measurement noises, $\eta_t \sim \mathcal{N}(0,Q_t)$ and $\zeta_t \sim \mathcal{N}(0,R_t)$, dimension n_a, mean $\hat{x}_t^a$, and covariance P_t^a as

$$\begin{aligned} x_t &= f(x_{t-1}, \eta_{t-1}), \\ z_t &= h(x_t) + \zeta_t. \end{aligned} \tag{6}$$

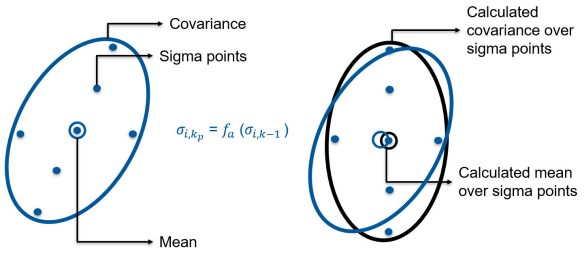

FIGURE 8. Unscented transform process: propagation of the sigma points through a nonlinear function to compute the Gaussian approximation.

A symmetric sigma sampling strategy can be used to calculate a set of $2n_a + 1$ sigma points σ^i as follows:

$$\begin{aligned} \sigma_{i,t} &= \hat{x}_t^a & i &= 0, \\ \sigma_{i,t} &= \hat{x}_t^a + (\sqrt{(n_a+\lambda)P_t^a})_i & i &= 1,\ldots,n_a, \\ \sigma_{i,t} &= \hat{x}_t^a - (\sqrt{(n_a+\lambda)P_t^a})_{i-n_a} & i &= n_a+1,\ldots,2n_a. \end{aligned} \tag{7}$$

In the above equations, λ is a scaling factor defined as $\lambda = \alpha^2(n_a + k) - n_a$. The parameter α determines the spread of the sigma points around the mean state value $\hat{x}_t$ and is normally set to a small positive value [57]. The second scaling parameter k should be set to value ≥ 0 to ensure the positive definiteness of the covariance matrix. Smaller values of k correspond to sigma points closer to the mean. $(\sqrt{(n_a+\lambda)P_t^a})_i$ is the $i-th$ column of the matrix square root of $(n_a+\lambda)P_t^a$). Once the covariance matrix is positive semi-definite, it is possible to apply the Cholesky factorization to calculate the matrix square root [55]. Afterward, a weight is assigned to each sigma point according to

$$\begin{aligned} w_m^0 &= \lambda/(n_a+\lambda), \\ w_c^0 &= \lambda/(n_a+\lambda) + (1-\alpha^2+\beta), \\ w_m^i &= w_c^i = 1/2(n_a+\lambda) \quad i = 1,\ldots,2n_a. \end{aligned} \tag{8}$$

In the above equations, β incorporates prior knowledge of the distribution of the state. For Gaussian distributions, $\beta = 2$ is optimal [57]. Finally, in the UKF prediction step, the $\sigma_{i,t}$ sigma points are propagated through the nonlinear function f^a as illustrated in Figure 8.

$$\sigma_{i,t_p} = f^a(\sigma_{i,t-1}). \tag{9}$$

The resulting sigma points and their corresponding weights are used to approximate the resulting values for the expectation and the covariance as

$$\begin{aligned} \hat{x}_{t_p}^a &= \sum_{i=0}^{2n_a} w_m^i \sigma_{i,t_p}, \\ P_{t_p}^a &= \sum_{i=0}^{2n_a} w_c^i (\sigma_{i,t_p} - \hat{x}_{t_p}^a)(\sigma_{i,t_p} - \hat{x}_{t_p}^a)^\mathsf{T}. \end{aligned} \tag{10}$$

Once a measurement z_t is obtained, the update step initiates, and a predicted measurement is computed for each sigma point. Based on the predicted measurements, the measurement mean and covariance are computed as

$$\begin{aligned} \hat{z}_{t_p} &= \sum_{i=0}^{2n_a} w_m^i z_{i,t_p}, \\ P_{yy} &= \sum_{i=0}^{2n_a} w_c^i (z_{i,t_p} - \hat{z}_{t_p})(z_{i,t_p} - \hat{z}_{t_p})^\mathsf{T} + R_t, \end{aligned} \tag{11}$$

approximating the cross-covariance as

$$P_{xy} = \sum_{i=0}^{2n_a} w_c^i (\sigma_{i,t_p} - \hat{x}_{t_p}^a)(z_{i,t_p} - \hat{z}_{t_p})^\mathsf{T}, \tag{12}$$

the Kalman gain, update state, and covariance estimates can be calculated according to

$$\begin{aligned} K &= P_{xy}P_{yy}^{-1}, \\ \hat{x}_t^a &= \hat{x}_{t_p}^a + K(z_t - \hat{z}_{t_p}), \\ P_t^a &= P_{t_p}^a + KP_{yy}K^\mathsf{T}. \end{aligned} \tag{13}$$

c: MOTION MODEL

A motion or transition model is utilized to predict the state of the detected objects from the previous to the current time step. Several motion models are available in the literature, such as the constant velocity (CV), constant acceleration, constant turn rate and velocity (CTRV), and constant turn rate and acceleration (CTRA) models, all of them making assumptions and simplifications to better describe the motion of an object [43]. Due to the low computational afford, the CV model has been selected to describe the objects' displacement. Therefore, it cannot describe the movement of a traffic participant while following a curved path. For this reason, two motion models have been selected. The 1D CV model was implemented in the very straight part of the test field, and the 2D Constant Turn (CT) model, with a higher level of complexity, for the non-straight and curved regions. The 1D CV transition model assumes that the target moves with nearly constant velocity, and its acceleration is modeled as white noise [58], being described by the following equation:

$$x_t = F_t x_{t-1} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, Q_t), \tag{14}$$

where:

$$\begin{aligned} x &= \begin{bmatrix} x_{pos} \\ x_{vel} \end{bmatrix}, \\ F_t &= \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}, \\ Q_t &= \begin{bmatrix} \frac{dt^3}{3} & \frac{dt^2}{2} \\ \frac{dt^2}{2} & dt \end{bmatrix} q. \end{aligned} \tag{15}$$

In (15), q is the velocity noise diffusion coefficient. Moreover, the 1D CV model can be combined with a linear Gaussian model of arbitrary dimension, D, to describe the

traffic participants' movements in x and y coordinates according to

$$F_t^D = \begin{bmatrix} F_t^1 & & 0 \\ & \ddots & \\ 0 & & F_t^d \end{bmatrix},$$
$$Q_t^D = \begin{bmatrix} Q_t^1 & & 0 \\ & \ddots & \\ 0 & & Q_t^d \end{bmatrix}. \tag{16}$$

The 2D constant turn model assumes that the objects move with nearly constant velocity with an unknown and constant turn rate ω [58], [59], and it is described as

$$x_t = F_t x_{t-1} + w_t, \; w_t \sim \mathcal{N}(0, Q_t), \tag{17}$$

where:

$$x = \begin{bmatrix} x_{pos} \\ x_{vel} \\ y_{pos} \\ y_{vel} \\ \omega \end{bmatrix},$$
$$F_t = \begin{bmatrix} 1 & \frac{\sin \omega dt}{\omega} & 0 & -\frac{1-\cos \omega dt}{\omega} & 0 \\ 0 & \cos \omega dt & 0 & -\sin \omega dt & 0 \\ 0 & \frac{1-\cos \omega dt}{\omega} & 1 & \frac{\sin \omega dt}{\omega} & 0 \\ 0 & \sin \omega dt & 0 & \cos \omega dt & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$
$$Q_t = \begin{bmatrix} \frac{dt^4 q_x^2}{4} & \frac{dt^3 q_x^2}{2} & \frac{dt^4 q_x q_y}{4} & \frac{dt^3 q_x q_y}{2} & \frac{dt^2 q_x q_\omega}{2} \\ \frac{dt^3 q_x^2}{2} & dt^2 q_x^2 & \frac{dt^3 q_x q_y}{2} & dt^2 q_x q_y & dt q_x q_\omega \\ \frac{dt^4 q_x q_y}{4} & \frac{dt^3 q_x q_y}{2} & \frac{dt^4 q_y^2}{4} & \frac{dt^3 q_y^2}{2} & \frac{dt^2 q_y q_\omega}{2} \\ \frac{dt^3 q_x q_y}{2} & dt^2 q_x q_y & \frac{dt^3 q_y^2}{2} & dt^2 q_y^2 & dt q_y q_\omega \\ \frac{dt^2 q_x q_\omega}{2} & dt q_x q_\omega & \frac{dt^2 q_y q_\omega}{2} & dt q_y q_\omega & q_\omega^2 \end{bmatrix}. \tag{18}$$

In (18), q_x and q_y are the acceleration noise diffusion coefficients, and q_w is the turn rate noise coefficient.

d: MEASUREMENT MODEL

The measurement model equations vary according to the number of physical constraints measured by each sensing modality [59]. Therefore, once the proposed system operates from object lists, we assume that only position can be measured, once positioning or ranging measurements are available for all sensing modalities previously mentioned. Thus, the measurement equation can be written as follows:

$$z_t = H_t x_t + \zeta_t, \quad \zeta_t \sim \mathcal{N}(0, R_t), \tag{19}$$

where:

$$H_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$
$$R_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \Omega. \tag{20}$$

In the initial step, the measurement noise covariance matrix R_t is defined with values of $\Omega < 0$, which means we strongly trust the measurements. Hence, the Kalman filter automatically adjusts the values of Ω at each processing step.

e: DATA ASSOCIATION

After updating the object state vector employing (14) and (15) or (17) and (18), according to the adopted transition model, a data associator is used to associate tracks, their predicted states, and new measurements. Data associators are able to combine state estimation and measurements in multi-target environments, e.g., Global Nearest Neighbour Associator (GNNA), Probabilistic Data Associator (PDA), and Joint Probabilistic Data Association (JPDA) [57], [58]. Due to the low computational efforts required for the association process, the Global Nearest Neighbour Associator was implemented. The GNNA is a single hypothesis tracking method that sequentially handles the input data. The goal is to assign the most likely observations to existing or new tracks so that the total cost of all associations is minimized, thus obtaining an optimal solution to the association problem. The first step in the association consists of gating the measurements, which means that for each track, only sufficiently close measurements are considered, based on the current estate estimation and a predefined distance value. This reduces computational costs and considerably diminishes unlikely observations-to-track pairings. In this study, the gate formed about the predicted object state and all observations that satisfy the gating relationship considered for track update utilizes the Euclidean distance as presented in (21) [60].

$$d_t^2 = (x_{\hat{x}_{t_p}} - x_{z_t})^2 + (y_{\hat{x}_{t_p}} - y_{z_t})^2 \leq \gamma_G, \tag{21}$$

where: $x_{\hat{x}_{t_p}}$ and $y_{\hat{x}_{t_p}}$ are the x and y coordinates of the position estimation, y_{z_t} and y_{z_t} are the coordinates of measurement, and γ_G defines the size of the gate. The associations in conflict situations, when there is more than one observation in a track gate or one observation is in the gate of more than one track, are addressed by the GNNA through the formation and solution of an association matrix. Thus, a generalized statistical distance for the assignment of observation j to track i is utilized to define the matrix elements.

$$d_{G_{ij}}^2 = d_{ij}^2 + \ln[|S_{ij}|]. \tag{22}$$

In (22), d_{ij}^2 is defined in (21), $\ln[|S_ij|]$ is the logarithm of the determinant of the residual covariance matrix, which has the effect of penalizing tracks with greater prediction uncertainty. The $d_{G_{ij}}$ is the cost for those observation-to-track assignments that satisfy the predefined gate. Those pairings that do not satisfy the gate are ignored. An example of an assignment matrix is presented in Table 2. In Table 2, the observation with minimum cost, shortest Euclidean distance, is assigned to the respective track. The non-allowed assignments, that failed the gate test, are represented by X. If a measurement is not associated with an existing track, it is assigned to a new track. Due to the presence of blind spots, a track is only deleted if it is not associated with any new measurement within a

TABLE 2. Assignment matrix.

Tracks	Observations 01	02	03
T1	$d_{G_{11}}$	$d_{G_{12}}$	$d_{G_{13}}$
T2	$d_{G_{21}}$	$d_{G_{22}}$	$d_{G_{23}}$
T3	$d_{G_{31}}$	$d_{G_{32}}$	$d_{G_{33}}$
NewTrack 1	0	X	X
NewTrack 2	X	0	X
NewTrack 3	X	X	0

ten-second interval. Moreover, to avoid the false association of two objects moving close to each other in opposite road lanes, the objects are filtered by the velocity direction. Thus, measurements with positive velocity are only associated with objects in the positive velocity range, even if an object with negative velocity is closer.

B. ASSISTED PERCEPTION

As mentioned in the previous chapters, automated vehicles are not fully capable of detecting the environment in all weather and driving scenarios due to onboard sensors' limitations. To overcome this drawback, the second use case provides redundant environment perception information for all automated vehicles connected to the infrastructure. The environment perception information includes all detected traffic participants' states, such as position, velocity, and heading angle. In the initial step, the assisted perception uses the same functionalities developed for traffic monitoring as shown in Figure 9. Once the global object list is created, two new tasks are performed: coordinate transformation, from Cartesian to geographic, and broadcast of global object lists via V2X communication. The global object lists are created in global "map-based" Cartesian coordinates. However, this information is only relevant for the clients with access to the map. For this reason, the coordinates of the detected objects have to be defined in geographic coordinates, which is common for all connected users. The road network utilized has both Cartesian and geographic coordinates. For this reason, the x and y coordinates can be easily transformed to longitude and latitude through the Sumolib "net.convertLonLat2XY" module [42]. Afterward, the global object lists in geographic coordinates are employed to generate collaborative perception messages. Then, the CPM messages are broadcasted in a frequency of 10 Hz, ten messages per second, from the Commsignia V2X units installed in RSUs 11, 5, and 1, to all connected vehicles equipped with V2X units within a range of up to 1000 meters from the transmitting RSU.

C. COLLABORATIVE PERCEPTION

The test field has eleven RSUs to constantly monitor the traffic participants. Therefore, mainly due to occlusion, caused by the obstruction of the sensor's field of view, and the limited field of view of the sensors, the infrastructure is subject to temporary and fixed blind spot areas. One solution to reduce the occurrence of blind spots is to use the local object lists, local detections, performed by the automated vehicles

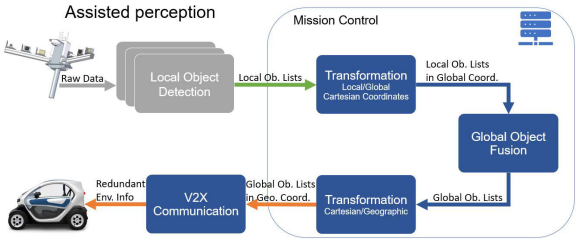

FIGURE 9. Assisted perception.

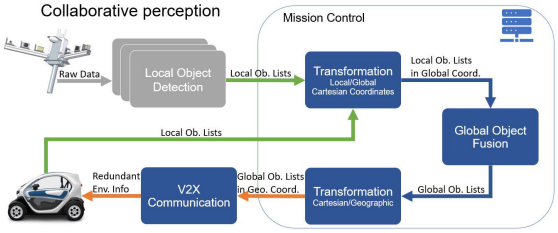

FIGURE 10. Collaborative perception.

connected to the infrastructure since they constantly move along the test field. The collaborative perception employs the same functionalities developed for assisted perception use case. However, in this case, not only the local detections of each RSU are used as input for the local/global transformation, but also the connected vehicles' detections as presented in Figure 10. In order to be used as a "moving RSU", the automated vehicle has to be equipped with onboard sensors with environment perception capabilities, an accurate positioning system, e.g., GNSS with real-time kinematics correction, a computational unit to generate the local object lists, and a V2X unit to enable the vehicle/infrastructure communication.

To integrate the connected vehicles' detections to the transformation and posterior global fusion, three pieces of information are essential: the position of the transmitting vehicle in geographic coordinates, the heading angle, and the position of the detected traffic participants in local Cartesian coordinates. Once the position of the vehicle in geographic coordinates is received, it is transformed to local Cartesian coordinates, and together with the heading information, the state transformation matrix is created according to (2). Unlike the RSUs, the transmitting vehicles constantly move along the test field. For this reason, a new transformation matrix must be generated for each received local object list. Afterward, the position of each local detect traffic participant is converted to global Cartesian coordinates for the posterior global fusion. The confidence level of the detections generated by the infrastructure is considered superior to the one generated by the connected vehicles. Thus, if the same traffic participant is detected by infrastructure and connected vehicles, during the global fusion, only the state information generated by the infrastructure is considered for the posterior steps.

D. EXTENDED PERCEPTION

The automated vehicles cannot guarantee a safe operation in all road traffic scenarios if the detection of the traffic

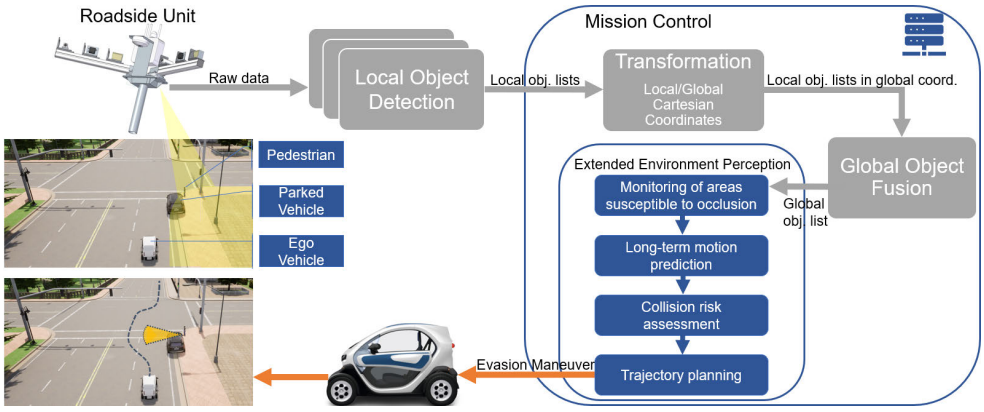

FIGURE 11. Extended perception.

participants relies only on the onboard sensors, once they are subject to occlusion. In a critical scenario, e.g., a pedestrian in an occluded area, obstructed by a parked car, with the intention to cross the road some meters in front of an automated vehicle driving straight, as illustrated in Figure 11. The in-vehicle sensors may not be able to detect the presence of this individual, which has a high chance of resulting in a fatal collision. Thus, the AVs require a method to maintain a safe operation even if the local perception system has limitations and cannot fully monitor the environment.

The extended environment utilizes different RSUs to monitor specific regions on the road. Suppose the system detects the occurrence of a critical scenario resulting in a possible collision between a connected automated vehicle and a vulnerable road user (VRU). In that case, it generates an avoidance maneuver and transmits this information as a high priority to the automated vehicle. The proposed approach allows the AVs to react to a critical situation with a larger time to collision when compared to other systems relying only on onboard sensors as presented in [61], [62], [63], and [64]. Thus, increasing the chance of successful avoidance even when implementing smoother maneuvers, once comfort is one of the target criteria.

The extended environment perception system for vulnerable road users' collision avoidance is composed of 5 parts. An overview is illustrated in Figure 11. The first part of the system is responsible for input generation. It comprises different infrastructure-based sensors, including range and visual sensors such as LiDARs, radars, and cameras. The sensors and a local processing unit provide a local object list, containing information on all detected objects, to a central computer, which can also receive data from other road users' sensors, such as automated vehicles driving within a specific range. The local object lists are transformed from local to global cartesian coordinates, and the same objects detected from different sources are fused. Then, the global object lists are compiled. Based on the global object lists, the system monitors specific areas on the road, e.g., pedestrian crossings and sidewalks nearby side parking areas, through the monitoring module. Thus, all VRUs within these areas are tracked, and the motion prediction module predicts their movement. The collision risk assessment module compares all future trajectories of the automated vehicles and possible crossing VRUs and evaluates the collision likelihood. The risk is determined at different levels. According to the risk level, the trajectory planning module generates the evasion maneuver, either a braking maneuver or a collision-free trajectory, balancing the actuation intensity with the comfort of the passengers. Hence, the automated vehicle receives the evasion maneuver as a high priority and activates the by-wire actuators to perform the avoidance maneuver.

VII. VALIDATION OF TRAFFIC MONITORING

Validation is the common term used to represent a sequence of test procedures to confirm and document that a process or component has met particular requirements for a specific use to ensure its safe operation [65]. In this context, validating the traffic monitoring and global fusion concepts requires a vehicle capable of recording its precise positioning information. Therefore, the automated vehicle platform, "ANTON," has been employed. ANTON is an experimental platform for developing and testing connected and automated driving functions [66]. The vehicle platform consists of a Renault Twizy Life, extended with a drive-by-wire system to control longitudinal and lateral dynamics, and a modified car body to allocate all necessary hardware. The vehicle has flexible sensor mounts to accommodate eight cameras, one LiDAR, one radar, two GNSS devices, a vehicle-to-x communication unit, and a router. The modified car body also provided additional space for the processing unit, an in-vehicle computer for local sensor data processing. Moreover, it is equipped with an autonomous driving stack, ROS 1-based, including software for environment perception, planning, decision, and actuation. Another essential feature is the approval for public roads, valid in the entire German territory. Figure 12 presents an illustration of all components.

A GNSS with high positioning accuracy is required to obtain ground truth measurements for system validation. For this reason, the vehicle platform is equipped with an ANavS Multi-Sensor RTK, a GNSS device with real-time kinematics

FIGURE 12. Automated driving platform.

correction. In addition, the comparison between ground truth and infrastructure positioning measurements requires a common time reference. The solution was to synchronize both systems with the same time source. In this case, a Network Time Protocol (NTP) server synchronized with Global Positioning System (GPS) time has been employed to establish a common time source for all RSUs. Likewise, the vehicle platform utilized an NTP server synchronized with a similar GPS device. Thus, the measurements generated in two independent and unconnected systems could be compared.

For validation purposes, the measurements obtained with the GNSS device have been defined as ground truth. From now on, the measurements are the quantities measured by the infrastructure. The distances between measurements and ground truth have been calculated using two approaches. The first employed the haversine formula, which determines the distance between two points (A and B) on a sphere given their longitudes and latitudes, as follows

$$\begin{aligned} dlon &= lonB - lonA, \\ dlat &= latB - latA, \\ h &= \sin^2\left(\frac{dlat}{2}\right) + \cos(latA)\cos(latB)\sin^2\left(\frac{dlon}{2}\right), \\ Distance(AB) &= 2\, r \arcsin(\sqrt{h}). \end{aligned} \tag{23}$$

where: $lonA$ and $lonB$ are the longitudes of points A and B in radians, $latA$ and $latB$ are the latitudes of the points A and B in radians, and r is the radius of the earth, which is approximately 6371000 m. The second approach consists of converting the positions of both ground truth and measurements to global Cartesian coordinates. Afterward, the shortest distance between them is obtained based on the Euclidean distance as presented in (21).

In order to simplify the analysis, the RSUs 5 and 6 have been selected for validation. In the validation scenario, ANTON stands still on the side of the road, around 70 m before the RSU 6. After 2 seconds, it is accelerated until it reaches a speed of 40 km/h, and it returns to a standstill condition, on the side of the road, around 70 m after

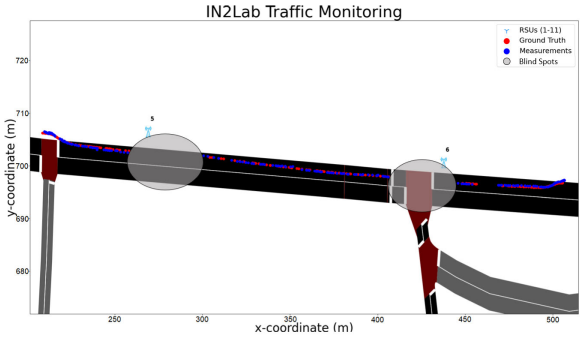

FIGURE 13. Preliminary positioning comparison.

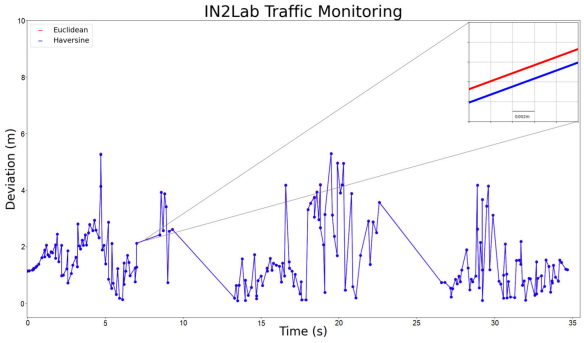

FIGURE 14. Preliminary positioning deviation.

the RSU 5. During the scenario execution, 210 infrastructure positioning and speed measurements were recorded and compared with the ground truth. The preliminary positioning results are illustrated in Figure 10. In Figure 13, the blue dots represent the infrastructure positioning measurements, and the red dots the GNSS positioning.

The 210 positioning measurements presented a deviation from the ground truth with a mean of 1.5 m and a standard deviation of 1.13 m. In order to evaluate the precision of the map-based transformations, the measurements, and ground truth have been compared using the Euclidean distance, calculated from the Cartesian coordinates of the measurements, and the map-based transformation from geographic to Cartesian coordinates of the ground truth. As well as using the haversine formula, utilizing the map-based transformed Cartesian to geographic coordinates of the measurements and the geographic coordinates of the ground truth. Both methods deviated in the mean by 0.002 m. The positioning deviation calculated with both methods is shown in Figure 14, where: the blue and red curves represent the deviation between measurements and ground truth calculated with the haversine formula and Euclidean distance, respectively.

Moreover, the velocity profile executed during validation has been recorded for analysis. A total of 210 measurements performed by the infrastructure have been compared with the velocity obtained by the positioning system. The velocity deviation from the ground truth presented a mean of 3.70 km/h and a standard deviation of 3.99 km/h. In Figure 15, the blue curve represents the velocity measured by the

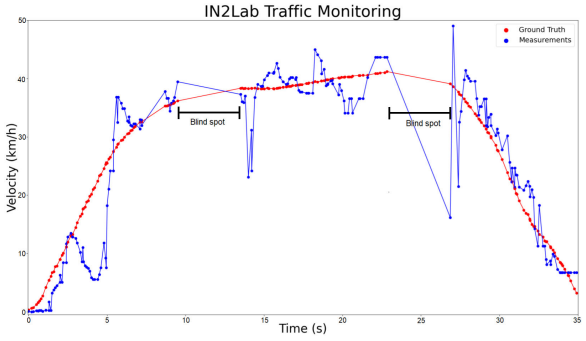

FIGURE 15. Preliminary velocity comparison.

infrastructure, and the red curve represents the velocity profile obtained with the GNSS device.

A preliminary performance evaluation of the test field considered two aspects: processing resources and latencies. A benchmark was used to analyze the processing capabilities of the application units and mission control server and the processing and communication latencies. At first, the computation of the local perception software stressed the Central Processing Unit (CPU) and the Graphics Processing Unit (GPU). Once the camera-based detection requires most of the processing capabilities, the camera frame rate was periodically increased until the application unit achieved its processing limits. In an overload operation, the application unit can process one camera capturing up to 40 frames per second, or in a normal load operation, under 90% of the processing resources, it can process one camera at 30 frames per second. As the most equipped RSUs have three cameras, a common frame rate of 10 Hz was set for all cameras in the test field. In addition, the local processing time required to process the sensor's raw data and generate a local object list, considering the most quipped RSU, is, on average, 10 ms. Moreover, the time for encoding, transmitting, and decoding messages through SENSORIS is, on average, 2,5 ms. The time needed to process the traffic monitoring functionalities on the mission control server, considering the tracking of 30 road users, is, on average, 50 ms. Thus, the total elapsed time from the instant the raw sensor data reaches the application unit until the generation of the global object list is approximately 62,5 ms.

VIII. CONCLUSION

This paper introduces the architecture of the test field "First Mile," currently being implemented in the framework of the project IN2lab in the city of Ingolstadt. The presented infrastructure includes eleven roadside units equipped with distinct vision and range sensors, an environment perception software stack, data processing units, and communications hubs. In this concept, the application units process the sensors' raw data locally in each RSU. Then, the local object lists are transmitted to the mission control responsible for the local/global coordinate transformation and the global object fusion, providing the global object list to each use case.

The use cases presented the functionalities of the test field and their benefits to the safety of connected automated vehicles, which include monitoring traffic participants, reception of local detections, and transmission of redundant environment perception information to connected vehicles. This information is indispensable for overcoming safety uncertainties caused by the limitation of onboard sensors due to temporary occlusion and adverse weather conditions. Another essential function being developed has the potential to identify critical scenarios in advance and actuate the connected automated vehicles to avoid or mitigate the occurrence of accidents.

The automated vehicle platform ANTON has been utilized to validate the traffic monitoring use case. The positioning comparison between the infrastructure measurements and ground truth, illustrated in Figure 13, presented a deviation in the mean of 1.58 m with a standard deviation of 1.13 m. Considering that ANTON has a length of 2.338 m and a width of 1.396 m, the majority of the measurements are inside the limits of the vehicle body. As the camera-based detection cannot guarantee the generation of the bounding boxes at the same location on the vehicle body, thus varying the measurement point, the implementation of 3D bounding boxes is a possible solution to reduce the positioning deviation. Moreover, the positioning deviation has been calculated with both haversine and Euclidean functions, presenting a variation of less than 0.002 m in the mean, demonstrating that the map-based transformation does not generate additional errors.

The velocity profiles in Figure 15 presented a deviation from the ground truth in the mean of 3.70 km/h and a standard deviation of 3.99 km/h. Despite the high accuracy of the velocity measurements provided by the radars, the deviation obtained is mainly generated during sensor data fusion, once the camera-based detection and LiDAR-based detection provide less accurate velocity information to be associated in the sensor data fusion. In Figure 15, the velocity bias present at 5 s is mainly caused by the limitation of the cameras to estimate the ego's velocity at a large distance from the sensor. When the road users are significantly far from the RSU, the speed estimation can be improved by fusing the camera with the radar detections. However, the Hungarian data association failed to associate the detections in this specific period. Lastly, Figure 15 also highlighted that after the blind spot regions, where the vehicle is not monitored by any sensors, the speed measurements deviate largely. The size of the blind spot in front of the RSUs 5 and 6 is also visible in Figure 13. It emphasizes the need to implement new sensors to monitor such areas and improve the overall system measurement accuracy. Moreover, processing and communication latencies were considered sufficiently low.

The future work will focus on continuing the implementation and optimization of the aforementioned functionalities and integrating new sensors. Moreover, a new camera-based detection approach based on 3D bounding boxes will be implemented to improve road users' position and velocity estimation and guarantee the same position reference point.

REFERENCES

[1] M. Tsukada, T. Oi, M. Kitazawa, and H. Esaki, "Networked roadside perception units for autonomous driving," *Sensors*, vol. 20, no. 18, p. 5320, Sep. 2020, doi: 10.3390/s20185320.

[2] N. Goberville, M. El-Yabroudi, M. Omwanas, and J. Rojas, "Analysis of LiDAR and camera data in real-world weather conditions for autonomous vehicle operations," *SAE Int. J. Adv. Current Pract. Mobility*, vol. 2, no. 5, pp. 2428–2434, 2020, doi: 10.4271/2020-01-0093.

[3] ZF. *Autonomous Driving: An Overview*. Accessed: Apr. 17, 2023. [Online]. Available: https://www.zf.com/mobile/en/technologies/domains/autonomous_driving/autonomous_ driving.html

[4] A. LaFrance. (2016). *Your Grandmother's Driverless Car*. Accessed: Apr. 17, 2023. [Online]. Available: https://www.theatlantic.com/technology/archive/2016/06/beep-beep/489029/

[5] J. Ramey. (2022). *Mercedes Launches SAE Level 3 Drive Pilot System*. Accessed: Apr. 17, 2023. [Online]. Available: https://www.autoweek.com/news/technology/a39943287/mercedes-drive-pilot-level-3-autonomous/

[6] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2018, pp. 287–296, doi: 10.1109/ICCPS.2018.00035.

[7] M. Kutila, P. Pyykönen, W. Ritter, O. Sawade, and B. Schäufele, "Automotive LiDAR sensor development scenarios for harsh weather conditions," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 265–270, doi: 10.1109/ITSC.2016.7795565.

[8] A. Harris, J. Stovall, and M. Sartipi, "MLK smart corridor: An urban testbed for smart city applications," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Los Angeles, CA, USA, Dec. 2019, pp. 3506–3511, doi: 10.1109/BigData47090.2019.9006382.

[9] ACCorD. *Corridor for New Mobility Aachen-Düsseldorf*. Accessed: Apr. 17, 2023. [Online]. Available: https://www.accord-testfeld.de

[10] DLR Transport. *Test Bed Lower Saxony for Automated and Connected Mobility*. Accessed: Apr. 17, 2023. [Online]. Available: https://verkehrsforschung.dlr.de/en/projects/test-bed-lower-saxony-automated-and-connected-mobility

[11] Testfeld Autonomes Fahren. *Test Field Autonomous Driving Baden-Württemberg*. Accessed: Apr. 17, 2023. [Online]. Available: https://taf-bw.de/en/the-test-field

[12] Ingolstadt Innovation Lab. *Technische Hochschule Ingolstadt (THI)*. Accessed: Apr. 17, 2023. [Online]. Available: http://in2lab.thi.de/

[13] M. Correia, J. Almeida, P. C. Bartolomeu, J. A. Fonseca, and J. Ferreira, "Performance assessment of collective perception service supported by the roadside infrastructure," *Electronics*, vol. 11, no. 3, p. 347, 2022, doi: 10.3390/electronics11030347.

[14] M. Shan, K. Narula, Y. F. Wong, S. Worrall, M. Khan, P. Alexander, and E. Nebot, "Demonstrations of cooperative perception: Safety and robustness in connected and automated vehicle operations," *Sensors*, vol. 21, no. 1, p. 200, Dec. 2020, doi: 10.3390/s21010200.

[15] *ROS, Camera Messages. Robot Operating System Documentation 2023*. Accessed: Jan. 31, 2023. [Online]. Available: http://wiki.ros.org/Messages

[16] SENSORIS. *Sensor Interface Specification*. Accessed: Jan. 31, 2023. [Online]. Available: https://sensoris.org/

[17] Commsignia. *Collective Perception Message*. Accessed: Jan. 31, 2023. [Online]. Available: https://www.commsignia.com/expertise/cpm/

[18] H. A. Ignatious and M. Khan, "An overview of sensors in autonomous vehicles," *Proc. Comput. Sci.*, vol. 198, pp. 736–741, Dec. 2021, doi: 10.1016/j.procs.2021.12.315.

[19] S. Campbell, N. O'Mahony, L. Krpalcova, D. Riordan, J. Walsh, A. Murphy, and C. Ryan, "Sensor technology in autonomous vehicles: A review," in *Proc. 29th Irish Signals Syst. Conf. (ISSC)*, Jun. 2018, pp. 1–4, doi: 10.1109/ISSC.2018.8585340.

[20] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, Mar. 2021, doi: 10.3390/s21062140.

[21] Infiniti Electro-Optics. (2020). *Infiniti, Visible Imaging Sensor (RGB Color Camera)*. Accessed: Jul. 10, 2022. [Online]. Available: https://www.infinitioptics.com/glossary/visible-imaging-sensor-400700nm-colour-cameras

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015, *arXiv:1506.02640*.

[23] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," *Sensors*, vol. 22, no. 2, p. 464, Jan. 2022, doi: 10.3390/s22020464.

[24] J. Kim, J. Kim, and J. Cho, "An advanced object classification strategy using YOLO through camera and LiDAR sensor fusion," in *Proc. 13th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Dec. 2019, pp. 1–5, doi: 10.1109/ICSPCS47537.2019.9008742.

[25] COCO. *Common Objects in Context*. Accessed: Aug. 8, 2023. [Online]. Available: https://cocodataset.org/#home

[26] Y. Wu, Y. Wang, S. Zhang, and H. Ogai, "Deep 3D object detection networks using LiDAR data: A review," *IEEE Sensors J.*, vol. 21, no. 2, pp. 1152–1171, Jan. 2021, doi: 10.1109/JSEN.2020.3020626.

[27] G. Zamanakos, L. Tsochatzidis, A. Amanatiadis, and I. Pratikakis, "A comprehensive survey of LiDAR-based 3D object detection methods with deep learning for autonomous driving," *Comput. Graph.*, vol. 99, pp. 153–181, Oct. 2021, doi: 10.1016/j.cag.2021.07.003.

[28] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499, doi: 10.1109/CVPR.2018.00472.

[29] S. Yildirim. *DBSCAN Clustering—Explained, Detailed Theoretical Explanation and Scikit-Learn Implementation*. Accessed: Feb. 6, 2023. [Online]. Available: https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556

[30] T. Zhou, M. Yang, K. Jiang, H. Wong, and D. Yang, "MMW radar-based technologies in autonomous driving: A review," *Sensors*, vol. 20, no. 24, p. 7283, Dec. 2020, doi: 10.3390/s20247283.

[31] J. Dickmann, J. Klappstein, M. Hahn, N. Appenrodt, H.-L. Bloecher, K. Werber, and A. Sailer, "Automotive radar the key technology for autonomous driving: From detection and ranging to environmental understanding," in *Proc. IEEE Radar Conf. (RadarConf)*, May 2016, pp. 1–6, doi: 10.1109/RADAR.2016.7485214.

[32] N. Scheiner, N. Appenrodt, J. Dickmann, and B. Sick, "Radar-based feature design and multiclass classification for road user recognition," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 779–786, doi: 10.1109/IVS.2018.8500607.

[33] G. L. Foresti and C. S. Regazzoni, "Multisensor data fusion for autonomous vehicle navigation in risky environments," *IEEE Trans. Veh. Technol.*, vol. 51, no. 5, pp. 1165–1185, Sep. 2002, doi: 10.1109/TVT.2002.800629.

[34] Coursera. (2019). *Lesson 3: Sensor Calibration—A Necessary Evil*. Accessed: Jun. 6, 2022. [Online]. Available: https://www.coursera.org/lecture/state-estimation-localization-self-driving-cars/lesson-3-sensorcalibration-a-necessary-evil-jPb2Y

[35] M. Bouain, K. M. A. Ali, D. Berdjag, N. Fakhfakh, and R. B. Atitallah, "An embedded multi-sensor data fusion design for vehicle perception tasks," *J. Commun.*, vol. 13, no. 1, pp. 1–7, 2018, doi: 10.12720/jcm.13.1.8-14.

[36] Tesla. *Tesla Motors*. Accessed: Aug. 8, 2023. [Online]. Available: https://www.tesla.com/

[37] Mercedes. *Mercedes-Benz*. Accessed: Aug. 8, 2023. [Online]. Available: https://www.mercedes-benz.de/passengercars/models/saloon/s-class.html

[38] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *IEEE Access*, vol. 8, pp. 2847–2868, 2020, doi: 10.1109/ACCESS.2019.2962554.

[39] K. Banerjee, D. Notz, J. Windelen, S. Gavarraju, and M. He, "Online camera LiDAR fusion and object detection on hybrid data for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1632–1638, doi: 10.1109/IVS.2018.8500699.

[40] N. Senel, K. Kefferpütz, K. Doycheva, and G. Elger, "Multi-sensor data fusion for real-time multi-object tracking," *Processes*, vol. 11, no. 2, p. 501, Feb. 2023, doi: 10.3390/pr11020501.

[41] M. Slavík and O. Vaculín, "Concept of mission control system for IN2Lab testing field for automated driving," in *Proc. FISITA World Congr.*, Sep. 2021, doi: 10.46720/f2021-acm-119.

[42] SUMO. *Sumo's Documentation 2023*. Accessed: Mar. 31, 2023. [Online]. Available: https://sumo.dlr.de/docs/Tools/Sumolib.html.locatenearby.edges.based.on.the.geo-coordinate

[43] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, pp. 1–14, Dec. 2014, doi: 10.1186/s40648-014-0001-z.

[44] L. Guo, L. Li, Y. Zhao, and Z. Zhao, "Pedestrian tracking based on camshift with Kalman prediction for autonomous vehicles," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 3, p. 120, May 2016, doi: 10.5772/62758.

[45] D. Ellis, E. Sommerlade, and I. Reid, "Modelling pedestrian trajectory patterns with Gaussian processes," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Sep. 2009, pp. 1229–1234, doi: 10.1109/ICCVW.2009.5457470.
[46] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? A study on pedestrian path prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 494–506, Apr. 2014, doi: 10.1109/TITS.2013.2280766.
[47] R. Toledo-Moreo, M. Zamora-izquierdo, and A. Gomez-skarmeta, "IMM-EKF based road vehicle navigation with low cost GPS/INS," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst.*, Sep. 2006, pp. 433–438, doi: 10.1109/MFI.2006.265590.
[48] J. Tao and R. Klette, "Tracking of 2D or 3D irregular movement by a family of unscented Kalman filters," *J. Inf. Commun. Converg. Eng.*, vol. 10, no. 3, pp. 307–314, Sep. 2012, doi: 10.6109/jicce.2012.10.3.307.
[49] M. St-Pierre and D. Gingras, "Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 831–835, doi: 10.1109/IVS.2004.1336492.
[50] C. Yang, W. Shi, and W. Chen, "Comparison of unscented and extended Kalman filters with application in vehicle navigation," *J. Navigat.*, vol. 70, no. 2, pp. 411–431, Mar. 2017, doi: 10.1017/S0373463316000655.
[51] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960, doi: 10.1115/1.3662552.
[52] Y. Kim and H. Bang, "Introduction to Kalman filter and its applications," in *Introduction and Implementations of the Kalman Filter*. London, U.K.: IntechOpen, May 2019, doi: 10.5772/intechopen.80600.
[53] K. Fujii. (2003). Extended Kalman filter. The ACFA-Sim-J Group. Accessed: Jan. 30, 2023. [Online]. Available: https://www-jlc.kek.jp/2004sep/subg/offl/kaltest/doc/ReferenceManual.pdf
[54] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Proc. SPIE*, vol. 3068, pp. 182–193, Jul. 1997, doi: 10.1117/12.280797.
[55] M. Meuter, U. Iurgel, S.-B. Park, and A. Kummert, "The unscented Kalman filter for pedestrian tracking from a moving host," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 37–42, doi: 10.1109/IVS.2008.4621191.
[56] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000, doi: 10.1109/9.847726.
[57] Mathworks. *Mathworks' Documentation 2023*. Accessed: Mar. 31, 2023. [Online]. Available: https://www.mathworks.com/help/ident/ug/extended-and-unscented-kalman-filter-algorithms-for-online-state-estimation.htmlbvgiw03
[58] Stonesoup. *Stone Soup's Documentation 2023*. Accessed: Mar. 31, 2023. [Online]. Available: https://stonesoup.readthedocs.io/en/v0.1b3/stonesoup.models.transition.htmlstonesoup.models.transition.linear.ConstantVelocity
[59] X. Yuan, C. Han, Z. Duan, and M. Lei, "Comparison and choice of models in tracking target with coordinated turn motion," in *Proc. 7th Int. Conf. Inf. Fusion*, Philadelphia, PA, USA, 2005, p. 6, doi: 10.1109/ICIF.2005.1592032.
[60] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Boston, MA, USA: Artech House, 1999.
[61] S. Zhang, D. Chen, J. Yang, and B. Schiele, "Guided attention in CNNs for occluded pedestrian detection and re-identification," *Int. J. Comput. Vis.*, vol. 129, no. 6, pp. 1875–1892, Jun. 2021, doi: 10.1007/s11263-021-01461-z.
[62] M. Heuer, A. Al-Hamadi, A. Rain, M.-M. Meinecke, and H. Rohling, "Pedestrian tracking with occlusion using a 24 GHz automotive radar," in *Proc. 15th Int. Radar Symp. (IRS)*, Jun. 2014, pp. 1–4, doi: 10.1109/IRS.2014.6869181.
[63] C. Flores, P. Merdrignac, R. de Charette, F. Navas, V. Milanés, and F. Nashashibi, "A cooperative car-following/emergency braking system with prediction-based pedestrian avoidance capabilities," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1837–1846, May 2019, doi: 10.1109/TITS.2018.2841644.
[64] H. Al-Qassab, S. Pang, M. Al-Qizwini, and H. Radha, "Visual sensor fusion and data sharing across connected vehicles for active safety," SAE Tech. Paper 2018-01-0026, 2018, doi: 10.4271/2018-01-0026.
[65] J. D. McCaffrey. *Validation vs. Verification*. Accessed: Apr. 24, 2023. [Online]. Available: https://jamesmccaffrey.wordpress.com/2006/04/28/validation-vs-verification/
[66] T. De Borba, O. Vaculín, and P. Patel, "Concept of a vehicle platform for development and testing of low-speed automated driving functions," in *Proc. FISITA World Congr.*, Prague, Czech Republic, Sep. 2021, doi: 10.46720/F2021-ACM-118.

THIAGO DE BORBA received the bachelor's degree in automotive engineering from the Federal University of Santa Catarina, Brazil, and the master's degree in automotive engineering from the University of Applied Sciences Ingolstadt, Germany. He is currently pursuing the Ph.D. degree with the Royal Melbourne Institute of Technology (RMIT), Melbourne, VIC, Australia. His research interests include automated and connected vehicles, smart infrastructures, and road and vehicle safety.

ONDŘEJ VACULÍN was a Research Professor of passive vehicle safety and mechatronic systems with Technische Hochschule Ingolstadt, Germany, with co-appointment with the CARISSMA Institute of Safety in Future Mobility (C-ISAFE), Automotive Safety Research Centre, in 2018. His research interests include vehicle safety, in particular assurance of automated driving and application of AI methods in the passive safety.

HORMOZ MARZBANI received the Ph.D. degree in mechanical engineering from the Royal Melbourne Institute of Technology (RMIT), Melbourne, VIC, Australia, in 2015. Since 2015, he has been a Lecturer with RMIT. His main research interests include dynamics, vibration, vehicle dynamics, and autonomous vehicles.

REZA NAKHAIE JAZAR received the master's degree in robotics from Tehran Polytechnic, in 1990, and the Ph.D. degree in nonlinear dynamics and applied mathematics from the Sharif University of Technology, in 1997. He is currently a Professor in mechanical engineering with the Royal Melbourne Institute of Technology (RMIT).