Mapping of a Low-Textured Environment Using Visual Simultaneous Localization and Mapping to Use Augmented Reality Simulation for Testing Advanced Driver Assistance Systems in Future Automotive Vehicles

Michael Weber
Institute of Energy Efficient Mobility
Hochschule Karlsruhe -
University of Applied Sciences, HKA
Karlsruhe, Germany
email: michael.weber@h-ka.de

Tobias Weiss
Institute of Energy Efficient Mobility
Hochschule Karlsruhe -
University of Applied Sciences, HKA
Karlsruhe, Germany
email: tobias.weiss@h-ka.de

Franck Gechter
CIAD (UMR 7533)
Univ. Bourgogne Franche-Comte, UTBM
Belfort, France

LORIA-MOSEL (UMR 7503)
Université de Lorraine
Nancy, France
email: franck.gechter@utbm.fr

Reiner Kriesten
Institute of Energy Efficient Mobility
Hochschule Karlsruhe -
University of Applied Sciences, HKA
Karlsruhe, Germany
email: reiner.kriesten@h-ka.de

***Abstract*—Taking advantage of Advanced Driver Assistance Systems (ADAS) testing in simulation and reality, this paper presents a new approach to using Augmented Reality (AR) to test ADAS. Our procedure creates a link between simulation and reality besides existing methods like Vehicle in the Loop (ViL) and should enable a faster development process for ADAS tests, which will become increasingly complex. High computer power is needed for complex automotive environmental conditions, such as high vehicle speed and fewer orientation points on a test track compared to AR applications inside a building. A three-dimensional model with accurate information about the urban test site is generated based on the combination of Image Segmentation (IS), Artificial Intelligence (AI) for object recognition, and Visual Simultaneous Localization and Mapping (vSLAM). The use of AI and IS aims to significantly improve performance, such as robustness, calculation speed, and accuracy for AR applications in complex automobiles. Another focus of this work is to make the relocalisation stable even in low-texture environments.**

***Index Terms*—Artificial Intelligence; Augmented Reality; Advanced Driver Assistance Systems; Visual Simultaneous Localization and Mapping; Oriented FAST and BRIEF**

I. Introduction

Advanced Driver Assistance Systems (ADAS), such as the active lane departure warning (LDW)-system and traffic sign recognition support the driver, offer comfort, and take responsibility for increasing road safety. These complex systems endure an extensive testing phase resulting in optimization potential regarding quality, reproducibility, and costs. ADAS of the future will support ever-larger proportions of driving situations in increasingly complex scenarios. Due to the increasing complexity of vehicle communication and the rising demands on these systems in terms of reliability to function safely even in a complex environment and to support the driver and increase safety, the test scenarios for ADAS are constantly further developed and adapted to higher requirements. European New Car Assessment Programme (Euro NCAP) has therefore introduced a series of new safety tests for ADAS into its program and created a road map until the year 2025 [2] [3].

Current testing methods for ADAS can be divided into simulation and reality. The core concept behind simulation is to replicate the vehicle behaviour as realistically as possible in virtual test drives. The goal of this approach is to leverage its benefits, such as reproducibility, flexibility, and cost reduction, by testing and evaluating specifications and solutions in the early stages of development. Suitable simulation methods allow for the efficient design, development, and implementation of vehicles and vehicle components. However, simulation cannot yet completely replace real-world tests. Physical conditions, such as weather, road surface, and other variables, play a crucial role in evaluating ADAS road tests and cannot be fully replicated in a virtual environment [4] [5].

However, the test and evaluation effort correlate with the complexity of an ADAS. The more complex the system, the greater the testing effort. The robustness, functional safety, and reliability of the ADAS must be proven in increasingly dynamic, complex, and chaotic traffic situations. This also includes the interaction with different road users, each with their natural movements, such as, e.g., the interaction of road users with each other. MAURER and STILLER say *"If testing and assessment methods cannot keep pace with this functional growth, they will become the bottleneck of the introduction of Advanced Driver Assistance Systems to the market."* [2] Therefore, new and efficient test methods are required to pave the way for future ADAS [6]. A new approach *Vehicle in the Loop (ViL)*, which is already being used in the industry today, combines the advantages of simulation and reality. The approach in this paper is a new method besides existing ViL-solutions.

This paper describes the usage of Augmented Reality (AR) for testing camera-based ADAS. Section II gives an introduction to the field of AR. Based on this, we describe the different single modules of Simultaneous Localization and Mapping (SLAM) to map the environment in Section III. Section IV gives a short Introduction to ADAS. Section V describes the current methods used for testing ADASs. In Section VI we describe a possible usage of AR to test ADAS and its challenges in detail. The paper concludes with Section VII, where the research, results, and lessons learned are discussed.

II. Augmented Reality

According to Azuma's proposal, AR can be defined as a combination of three fundamental characteristics: the combination of real and virtual worlds and the precise three-dimensional registration of real and virtual objects, both in a real-time interactive environment [7]. The basic principle of AR is mainly known from the mobile game Pokémon Go [8]. Within this game, users can interact with digital creatures through their smartphones. These creatures are placed virtually in the user's environment. One such AR application is shown in Figure 1. Figure 1 shows also a self-created AR-App for demonstrating a possible scenery with traffic signs and a pedestrian. The three parts of the algorithms behind AR are image analysis, 3D modeling, and augmentation [1].

Image analysis serves to identify points or areas of interest within the given image. Feature detection, such as corner detection is often used for this step [9]. A three-dimensional model of the environment is created using the results of the image analysis. The types of algorithms used for this step vary

Fig. 1. Pokémon Go-App on the left side of the figure [8] and a self-created Augmented Reality application showing a possible scenery on the right side of the figure [1].

depending on the type of AR application. SLAM or Structure-from-Motion (SfM) algorithms are often used for AR in unknown locations [9]. The augmentation is based on the results of 3D modelling. The scene model is typically provided as a positional description of a plane or coordinate system that represents the real world [9]. With this information, a virtual object can be placed on the plane or in the coordinate system with appropriate characteristics, such as size and orientation. After object placement, the virtual content is combined with the real image [1] [9].

There are different versions of applications for AR. These applications are very diverse in their fields, from the use of AR in psychology [7] to use in hospital operating rooms [9] to mobile games [9] to military applications [9]. What all these apps have in common is that human reality is expanded. With humans as users of AR, there are implications for the application. One is that, in most cases, the human user is forgiving of not accurately placed virtual objects, if the error lies within a small margin. In addition, the speed of human movement, and therefore the distance covered in a given time, is limited. Because of these limitations, localization, mapping, object placement, and runtime requirements are not as strict and demanding as in the automotive environment given in this paper [1].

In order to use AR, the entire system must first be able to orient itself in the environment in order to finally augment virtual objects on reality. SLAM is a possible approach for this and is described in more detail in the following section.

III. Simultaneous Localization and Mapping

SLAM, or Simultaneous Localization and Mapping, is a method for determining the 3D structure of an unknown environment and the movement of sensors within it. Originally developed for autonomous robot control, the use of SLAM has since expanded to various applications, such as online 3D modeling using computer vision, AR visualization, and self-driving cars [10]. Early SLAM algorithms utilized multiple sensor types, such as laser range sensors, rotary encoders, inertial sensors, GPS, and cameras. However, more recent developments focus on using only cameras due to the simplicity of the sensor setup and greater technical challenges. SLAM systems that still use visual information as input are known as visual SLAM, or vSLAM.

A. Visual Simultaneous Localization and Mapping

SLAM techniques that rely solely on visual information are called visual Simultaneous Localization and Mapping (vSLAM). These algorithms are widely used in computer vision, robotics, and AR, and are particularly useful for camera pose estimation in AR systems [10]. Because AR systems often require real-time processing on light portable devices, various low-computational vSLAM algorithms have been developed. These algorithms have applications beyond AR systems and are also valuable for unmanned autonomous vehicles in robotics [11]. Most vSLAM approaches are comprised of five technical modules: three basic modules and two additional modules. These modules will be briefly described.

B. Basic Modules of Visual Simultaneous Localization and Mapping

The basic modules of vSLAM are *Initialization*, *Tracking*, and *Mapping* and are shortly presented in the following:

- *Initialization* is a crucial step in vSLAM that involves defining a coordinate system for estimating camera position and reconstructing a 3D environment. During initialization, a global coordinate system must be established, and a part of the environment is reconstructed as an initial map in this system [12].
- *Tracking* is a component of vSLAM that follows the reconstructed map in an image to continuously estimate the camera position relative to the map. That is achieved by determining distinctive matches between the captured image and the created map using feature matching or feature tracking [12].
- *Mapping* involves expanding the map by determining the 3D structure of an environment when the camera encounters previously unmapped regions. It involves understanding and calculating the unknown parts of the environment [12].

C. Additional Modules of Visual Simultaneous Localization and Mapping

The basic modules of vSLAM are supplemented by the additional modules *Relocalization* and *Global Map Optimization*:

- *Relocalization* refers to determining the current camera position in a reconstructed map when tracking has failed, which can occur due to fast camera movements. That allows the system to recompute the camera's location [12].
- *Global Map Optimization*, which includes *Loop Closing*, is a method to refine the map by eliminating cumulative estimation errors that accumulate with camera movement. The map is optimized by considering the consistency of all map information. If previously recorded map elements are recognized, loops are closed, and the estimation error is corrected from the beginning to the present. Loop Closing is used to acquire reference information by comparing the current image to previously acquired images. Relocalization is used to recover the camera position, and Pose Graph Optimization is employed to suppress cumulative error by optimizing the camera positions. Bundle Adjustment (BA) is also used to minimize the map reprojection error by optimizing both the map and camera positions. In large environments, this optimization method is used to effectively minimize estimation errors, while in small environments, BA can be performed without loop closure as the cumulative error is small [12].

Different sensors, and combinations of sensors, can be used in vSLAM. These sensors will be described in the following section.

D. Sensors for Visual Simultaneous Localization and Mapping

We discuss the primary sensors for visual SLAM in this section.

1) Monocular Camera: Monocular cameras consist primarily of an image sensor and a lens. After removing lens distortion, the camera can be modeled as a pinhole camera [13], allowing a 3D point in the camera's coordinate reference system to be projected into 2D pixel coordinates. Most industrial cameras are global shutter cameras, which capture the entire image at once, while consumer cameras tend to be rolling shutter cameras that capture pixel rows at different times, which can affect accuracy in visual SLAM if not adequately modeled. Due to the nature of monocular cameras, they cannot accurately determine the actual scale of the world and therefore monocular SLAM can only estimate the map and camera trajectory up to scale. To resolve this issue, additional information, such as an Inertial Measurement Unit (IMU) or known distances in the map, is necessary to scale the solution correctly [14].

2) Stereo Camera: Stereo cameras consist of two cameras that are attached rigidly to each other. Ideally, they have synchronized for simultaneously capturing the images. The depth of an object can be estimated from a single stereo frame by finding correspondences between pixels in the left and right cameras. To achieve this, both cameras must be calibrated internally and the rotation and translation between both cameras must be calibrated using several stereo frames with a calibration pattern. The distance between the cameras, known as the baseline, along with the focal length and image resolution, determine the range of depth where depth estimation is accurate. The projection function for a rectified stereo camera maps a 3D point in the camera coordinate system to another 3D point [14].

3) RGB-D Camera: Red Green Blue-Depth (RGB-D) cameras consist of a color camera and a depth sensor that uses structured light or time-of-flight technology. By calibrating the intrinsic parameters of the camera and the extrinsic parameters between the color camera and depth sensor, the depth measurements can be transformed into a depth map with a

1:1 relationship with the color image, eliminating the need for stereo matching as with a stereo camera. However, their use is limited to indoor environments due to the nature of the depth sensor [14].

4) Inertial Measurement Unit: IMUs detect an object's motion by combining a gyroscope that measures angular velocity and an accelerometer that measures linear acceleration. They provide information about the object's self-motion, complementing what is seen by vision. IMUs can be used to determine the motion between camera frames or calculate the scale of monocular SLAM, and they can also estimate gravity, allowing for the calculation of absolute pitch and roll. They typically measure acceleration and angular velocity hundreds of times per second. To effectively use IMUs with vision, they should be synchronized and calibrated to match the reference frame of the camera [15].

E. Solving Visual Simultaneous Localization and Mapping

Exploiting information from a stream of images captured by a vision sensor is crucial for visual SLAM. There are two main approaches to this problem: feature-based and direct methods. When applied to automotive vehicles, vSLAM faces challenges such as fast scene changes and low environmental texturing. Several vSLAM algorithms are available and listed in [16], where they are compared in terms of accuracy and robustness, among other factors. In the following chapter, a specific vSLAM approach relevant to our research work is described.

1) Direct and Feature-Based Simultaneous Localization and Mapping: Feature-based methods analyze images to identify and extract unique, recognizable *keypoints*. These keypoints can be detected repeatedly and consistently in images of the same scene, even under varying viewpoints and lighting conditions. Each keypoint is then assigned a descriptor, a numerical representation used to match keypoints across images by comparing the descriptors. The keypoint and its descriptor make up a *feature*. Once features have been extracted, the original image is no longer necessary as the features are used for all subsequent processing. That is beneficial as features are easier to match and manipulate for solving geometry problems in visual SLAM, such as triangulation, epipolar geometry, Perspective-n-Point (PnP) problem, and transformations between reference systems. The direct approach in SLAM involves utilizing sensor measurements, such as pixel intensity in an image directly. It can be categorized into dense (using all pixels) [17], semi-dense (using pixels with high gradients) [18], or sparse (using only a few pixels) [19] methods. These direct methods are considered more accurate and reliable when there is minimal texture or blur in the image, as they do not depend on keypoint detectors. The objective of direct SLAM is to determine the depth of each pixel in selected cameras by minimizing photometric error through optimization [14].

2) Oriented FAST and Rotated BRIEF (ORB)-SLAM: The ORB-SLAM algorithm was first presented in 2015 and is the current state of the art as it has higher accuracy than comparable SLAM algorithms [20]. ORB-SLAM represents a complete SLAM system for monocular, stereo, and RGB-D cameras. The system operates in real-time and achieves remarkable results in terms of accuracy and robustness in a variety of different environments. ORB-SLAM is used for indoor sequences, drones, and cars driving through a city. The ORB-SLAM consists of three main parallel threads: *Tracking*, *Local Mapping*, and *Loop Closing*. It is possible to create a fourth thread to execute the *BA* after a closed loop. This algorithm is a feature-based approach that represents the detected points in a three-dimensional *Point Cloud* [16]. ORB-SLAM seems to be the best algorithm for our approach [1]. To use ORB-SLAM for testing ADAS using AR, the following chapter first provides an introduction to the topic of ADAS.

IV. Advanced Driver Assistance Systems

ADAS enhances the driving experience by offering assistance to the driver when operating a vehicle. Depending on the specific system, it can enhance comfort, improve safety, decrease energy usage, or streamline traffic flow. The technology uses sensors to monitor driving conditions, employs powerful computers to process the collected data, and provides feedback to the driver through visual, auditory, or touch signals. In some cases, ADAS can even take control of the vehicle by accelerating, braking, signaling, or steering, potentially leading to fully autonomous driving. For ADAS to be effective, it requires rapid data processing with near real-time response and a highly dependable system [21] [22]. The proliferation of environmental sensors, including radio detection and ranging (radar), cameras, ultrasonic sensors, and Light Detection and Ranging (LiDAR)-sensors, makes it possible to use ADAS and related autonomous driving functions in modern vehicles. However, each sensor has its limitations and cannot provide all of the necessary information about the vehicle's surroundings to guarantee safety. Only through the fusion of data from multiple sensors a complete environment model can be created, which is crucial for the dependability and safety of driver assistance systems and autonomous driving [23]. Both simulation and real-world testing are required for thoroughly evaluating individual ADAS functions [1]. The following section outlines the different testing options for ADAS.

V. Testing of Advanced Driver Assistance Systems

Simulative test procedures during the development process and test procedures, in reality, are used to evaluate the functionality of individual ADAS sensors and their joined interaction in ADAS-relevant scenarios. While in the early concept stage, all components of the road test are still virtual and characterized by test procedures such as:

- Model in the Loop (MiL)
- Software in the Loop (SiL) or
- Hardware in the Loop (HiL)

Through the various stages of development, a gradual exchange of virtual for the equivalent physical world test components enrolls. By the end, reality replaces the simulated elements completely [22]. Table I shows an overview of different stages for testing ADAS in simulation and in reality. In the following, an overview of the respective possibilities and the advantages and disadvantages in simulation and in reality is given.

A. Testing Advanced Driver Assistance Systems in Simulation

The objective of the virtual road test is to recreate the experience of an actual road test as closely as possible in a virtual environment. The goal is to take advantage of simulation, such as reproducibility, versatility, and ease of use, and to evaluate and test vehicle specifications and solutions early in the development process. By using appropriate simulation methods, vehicle and component design, development, and implementation can be made more efficient. These methods reduce the time it takes for real-world prototypes to become available. Optimizing simulation techniques based on actual driving tests and actual test results requires balancing modeling, parameterization, and simulation effort with the efficiency gained. The approach mainly employs methods from embedded mechatronic system development, including SiL, MiL, and HiL methods [4].

B. Testing Advanced Driver Assistance Systems in Reality

Validation of vehicle dynamic control systems, despite their complexity and wide range of variations, can be expensive to test in actual driving tests. However, this approach is not feasible for driver assistance systems with environmental perception due to the high-level system complexity, the complexity of test cases, and the extent of testing required. Even if the tests are performed identically, it is impossible to ensure that the same conditions are being tested, due to the numerous and unknown factors influencing the results. That makes the reproducibility of results uncertain. Function-relevant features may require the involvement of multiple road users and can also be affected by complex interactions of various conditions, such as glare from the sun and reflections on a wet road at a certain angle. Current ADAS access information about the environment, often gathered by multiple sensors with different functions and processed in an environment representation [5] [24]. As suggested in literature Euro NCAP is a European standard for actual ADAS driving tests, which focuses on the vehicle's behavior and response in safety-critical scenarios. Dangerous situations are simulated using dummy vehicles, pedestrians, and cyclists to evaluate the effectiveness of ADAS systems. To improve road safety, the requirements for ADAS are being increased, and as a result, Euro NCAP test procedures will become increasingly complex in the future. The roadmap for 2025 includes the inclusion of additional road users, such as scooters, motorcycles, and wild animals [26]. Figure 2 displays a test configuration for pedestrian detection. In the test, a pedestrian dummy (representing a child) is crossing the road behind a parked car, and the test vehicle must detect the pedestrian and apply the brakes to prevent personal injury or damage to property [25].

It is rarely possible to conduct tests with prototypes in real road traffic and test persons due to legal and safety restrictions. The validation of safety-critical functions such as Automatic Emergency Braking (AEB) for pedestrians cannot be replicated adequately, even within the framework of NCAP test scenarios on a test track. As the driving situation becomes more complex for the assistance system under test, it becomes increasingly challenging to realistically and reliably assess the interaction between system behavior and driver experience and behavior. ViL bridges the gap between driving simulation and real-world testing. By combining virtual visual representation with the experienced haptics, kinaesthetics, and acoustics from actual vehicle movement, ViL provides a new augmented reality-based approach to efficiently and safely develop and evaluate ADASs [5]. This approach is described in the following chapter.

C. Combining Virtual and Real Testing

To enable the testing of camera-based assistance systems in real environments earlier in the development phase and thus

TABLE I
OVERVIEW OF DIFFERENT STAGES FOR TESTING ADVANCED DRIVER ASSISTANCE SYSTEMS IN SIMULATION (VIRTUAL) AND IN REALITY. V: VIRTUAL, R: REALITY [22].

	MiL	SiL	HiL	Chassis Dynamometer	ViL	Road Test
Functional Code	V	R	R	R	R	R
ECU	V	V	R	R	R	R
System	V	V	R	R	R	R
Vehicle	V	V	V	R	R	R
Driver	V	V	V	V/R	V/R	R
Driving Dynamics	V	V	V	V	R	R
Perceptibility	V	V	V	V	R	R
Lane	V	V	V	V	R	R
Environment	V	V	V	V	V	R

Fig. 2. Test setup for pedestrian detection, where a pedestrian (child as a dummy) crosses the road behind a parked car in a scenario for testing Automatic Emergency Braking (AEB) based on Euro NCAP regulations [25].

increase the quality of the systems, the use of AR as a link between virtual and real testing lends itself to this. Using AR to test camera-based systems combines the advantages of a virtual environment and these of the real world: Reproducible, complex scenes with realistic environmental conditions. AR thus makes it possible to dispense with test dummies or second vehicles including drivers even in the initial phases of testing. This reduces the testing costs and increases the safety of the test engineers. The combination of different test situations is also possible: The display of several vehicles, lane markings, and road signs allow the simultaneous testing of all camera-based driver assistance systems. The unlimited variety of test scenarios allows a significant increase in the depth of testing at an early stage of development. This increases the quality of the testing and the overall system. In 2010, a Swedish team led by Jonas Nilsson presented a software framework at a conference that used AR to evaluate a pedestrian detection system. The framework was able to augment the images from the vehicle camera to include a walking pedestrian. The resulting detection system results were comparable to test results obtained with real obstacles. As summarized in this paper, deeper investigations are needed further advance an AR test system [10]. A different method than ours for combining virtual and reality testing is discussed in the subsection that follows.

D. *Vehicle in the Loop*

ViL stands for a newer method to meaningfully complement and improve the development of the V-Model for driver assistance systems. It addresses the need for many driver assistance functions for an elaborate driving test and a high demand for functional safety. This group of driver assistance functions will increasingly gain in importance and scope. One main reason is the ever-growing number of vehicle derivatives in which driver assistance-functions are offered, and the accompanying ever-higher level of automation and networking. The ViL method allows the operation of the test vehicle in a virtual environment. The coupling between the vehicle and the virtual environment can be done in two ways. One way is to replace the physical sensor system with an interface. At this interface, the simulation environment feeds in simulated sensor signals which correspond to the sensor response from a physical environment. The second way is to retain and artificially stimulate real sensor technology, as is feasible, for example, with ultrasonic sensor technology exposed to artificially generated response signals via ultrasonic transducers [27]. In both variants, the physical test vehicle reacts to features and events in the virtual environment. Critical driving manoeuvres towards obstacles or objects on a collision course can thus be tested safely and reproducibly. The interface created can also be used to generate the sensor signals in a way that would occur due to a changed position in a vehicle derivative or due to different tolerances. Thus, this method enables testing corresponding derivatives or tolerances with a test vehicle.

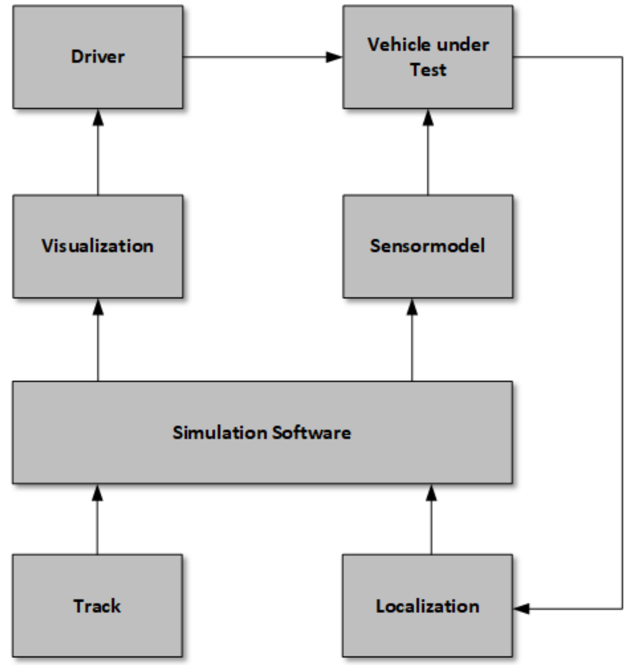

Fig. 3. ViL-architecture as well as the flow of information [22].

In addition to the considerably safer test operation, this allows effective testing and application of driver assistance-functions. That results in the considerable economic potential for driving tests in driver assistance. The use of virtual integration in conjunction with the ViL method allows efficient application of the customer function. The efficiency and reproducibility of the test cases required for this can thus increase significantly. Figure 3 illustrates the general operating principle of the ViL by the architecture as well as the flow of information [22]. The following section describes another approach besides existing ViL-Solutions using AR.

VI. Augmented Reality Simulation in Advanced Driver Assistance Systems

With a focus on the camera-based ADAS sensors, the area around the test field is recorded, as shown in Figure 4. The path between the sensor fusion module and the Electronic Control Unit for Advanced Driver Assistance Systems (ADAS-ECU), which causes the vehicle to intervene, for example by braking, has to be disrupted and a new path has to be found through the Augmented Reality Electronic Control Unit (AR-ECU). Within the AR-ECU, the captured environmental data is augmented with virtual objects, such as traffic signs or lane markings. The aim here, is a realistic and consistent behavior of the ADAS-ECU as in real object detection. For the final augmentation of the virtual objects on the real image of the sensor, a detailed 3D environment of the test environment must first be created. This section will describe the single steps as well as the results so far to use AR for testing ADAS.

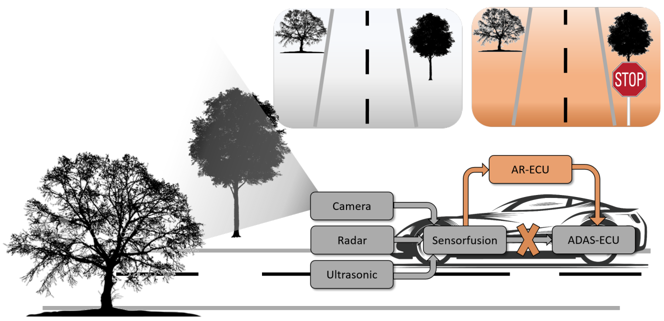

Fig. 4. Our approach for using augmented reality in advanced driver assistance systems.

A. Technical Steps to use Augmented Reality in Advanced Driver Assistance Systems

Figure 5 shows the technical steps of our approach to using AR in ADAS. The *Sensor Input* as a camera stream is transferred in greyscale to the element function block *ORB-SLAM3*. The keypoints are *tracked* and transferred to relevant *featurepoints*. These featurepoints are filtered by the Image Segmentation (IS) in the function *DS-SLAM*. The relevant featurepoints are merged into a *Point Cloud* in *Mapping*. *Loop Closing* and *Bundle Adjustment* are further elements of *ORB-SLAM3* to enable corrections of the Point Cloud and increase accuracy. The next function block is the *Point Cloud Preparation*. The first step is a *Preparation* of the Point Cloud to identify relevant points for *Plane Detection*. Based on these results, a plane is inserted on which virtual objects like e.g., a cyclist or traffic signs from the *Virtual Objects Library* can be placed. These objects can be static or animated. After the successful placement of the virtual elements, *rendering* is carried out in the function block *AR Viewer*, whereby the virtual objects are augmented with the real image stream. In the final step the *Output is Prepared* and adapted to the target medium as *Augmented Images*. The following subsections describe the single steps in a more detailed way.

B. Mapping of the Environment

The environment mapping is done using different stereo cameras such as the *ZED2i* equipped with a polar filter and a baseline of 120 mm, and the *Intel RealSense D455* with a baseline of 95 mm. The vSLAM approach ORB-SLAM3 is applied to these cameras. The results presented in the images were obtained using the *Intel Realsense D455*. The detected features are recorded in a three-dimensional Point Cloud. The test drives were performed on appropriate NCAP test areas, as shown in Figure 6. The impact of the low texture of the environment must be taken into consideration. To overcome the repetition of scene images, pylons are placed along the test track at intervals of 20 metres, alternating in number on either side of the track. That ensures that feature matching and proper orientation in the Point Cloud occur. The camera is mounted at the rear-view mirror height on the top of the windshield, as is typical for cars. Figure 6 displays the recorded test track scene. The rectangles represent the feature points detected

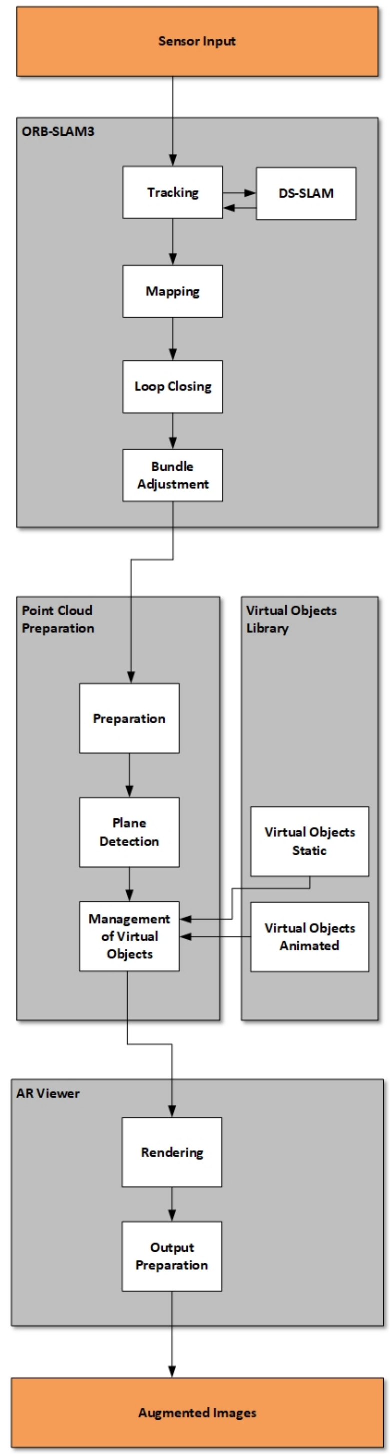

Fig. 5. Technical steps to use augmented reality in advanced driver assistance systems.

by the ORB-SLAM algorithm. The generated Point Cloud can be viewed in Figure 7 using the *RViz* tool. The Point Cloud depicts a straight road with pylons, road markings, and other objects such as trees on the left side or a hill on the right. Stereo-based cameras allow for accurate mapping of the environment with the correct scale. Approaches that

Fig. 6. Test area on the top of the figure with pylons and test vehicle as well as detected feature points using ORB-SLAM (rectangles) on the bottom.

utilize a combination of mono-based cameras and IMUs do not result in usable outcomes due to the lack of depth and scale information. As shown in Figure 6, feature points are detected in the sky as well as on the hood of the ego vehicle, which hinder the performance of our vSLAM algorithm. To address this issue, we decided to employ Image Segmentation (IS) to detect these false feature points.

C. Image Segmentation

We have selected *Deeplabv3+* developed by Google as the model for IS and semantic segmentation after evaluating various options. The pixellib library, a python tool for training and testing deep neural networks (NN), was utilized in conjunction with Deeplabv3+. This choice was based on Deeplabv3+ delivering the best results for our ORB-SLAM implementation, as it outperformed most recent NN models for semantic segmentation. For a comprehensive overview of different models and their performance, we refer to [28]. The Xception-65 model, with pre-trained weights from the ADE20k-Dataset, was utilized as the backbone. The pre-trained model provided 120 labels for indoor and outdoor environments with a color mapping for each label. The model

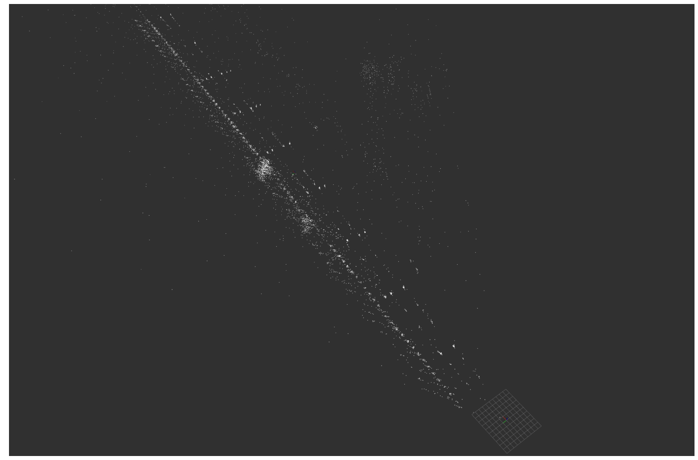

Fig. 7. Created three-dimensional Point Cloud based on detected feature points using ORB-SLAM.

performed so effectively on our test images and videos that additional custom training was not required, although it could be easily implemented for further development. As depicted in Figure 8, relevant feature points were successfully detected in the testing ground, while false feature points in the sky or on the car hood were disregarded. The use of this IS led to an improvement in the robustness of relocalization.

D. Map Preprocessing

The subsequent step is to fit a plane suitable for the Point Cloud. This plane ensures that virtual objects can be realistically inserted with the correct height and position in the scene image. As an initial attempt, we applied the RAndom SAmple Consensus (RANSAC)-algorithm to determine a plane on the road surface. Figure 9 illustrates that using the

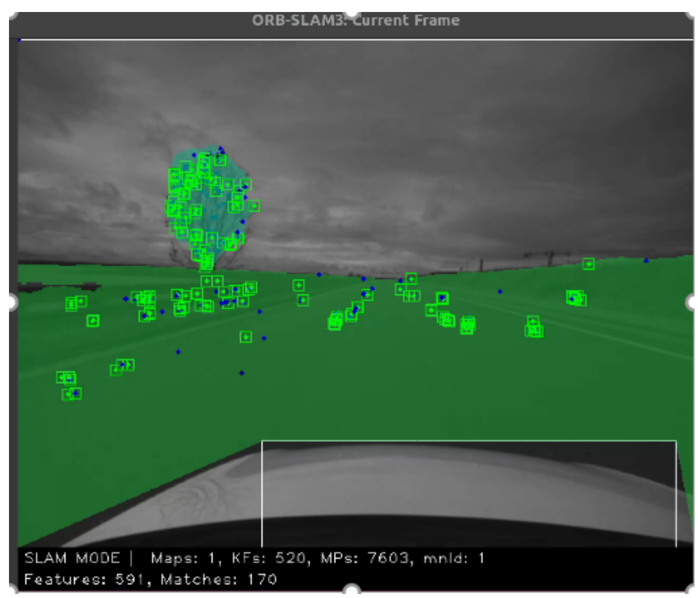

Fig. 8. Results of our image segmentation to detect feature points in relevant regions (green area).

entire Point Cloud results in an inclined plane due to the abundance of feature points on the hill and the scarcity of points on the street. To address this issue, we extract only the relevant feature points from the Point Cloud. That is achieved by using a cylindrical area around the camera trajectory to select the feature points on the street and eliminate irrelevant feature points outside the test track. We can define the cylinder by specifying 2 points along the axis, and the radius of the cylinder:

$A(x1, y1, z1)$
$B(x2, y2, z2)$
and radius = R

Consider the line coordinates with the direction

$$e = r_B - r_A \tag{1}$$

and moment

$$m = r_A \times r_B \tag{2}$$

These two vectors represent the infinite line between A and B. A point P with position rP lies in the cylinder between A and B and radius R if:

1. Distance of P to line AB is equal or less than R:

$$d = \frac{||m + e \times r_P||}{||e||} <= R \tag{3}$$

2. Closest point Q on line to P is:

$$r_Q = r_P + \frac{e \times (m + e \times r_P)}{||e||^2} \tag{4}$$

3. The barycentric coordinates of $Q(w_A w_B)$ such that r_Q = $w_A r_A$ + $w_B r_B$ are:

$$w_A = \frac{||r_Q \times r_B||}{||m||} \tag{5}$$

$$w_B = \frac{||r_Q \times r_A||}{||m||} \tag{6}$$

4. Check that point Q lies between A and B by making sure the barycentric coordinates are between 0 and 1:

inside = $(w_A >= 0)$ and $(w_A >= 1)$ and $(w_B >= 0)$ and $(w_B >= 1)$

After defining which feature points are included by our cylinder and which radius is to be used, we applied RANSAC again. This cylindrical approach is depicted in Figure 10.

Virtual NCAP-relevant objects such as traffic signs, pedestrians, and road signs are created using the Blender software. Figure 11 demonstrates a created cyclist for our approach. The use of Blender software enables us to design both static and dynamic objects with intricate details. The virtual objects can be placed on the plane as shown in Figure 12, with no restrictions on their placement or orientation. After the virtual objects have been placed, the next step is to render them into the camera stream.

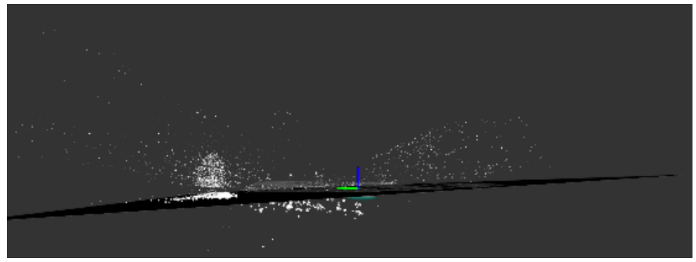

Fig. 9. Incorrectly placed inclined plane using the random sample consensus algorithm.

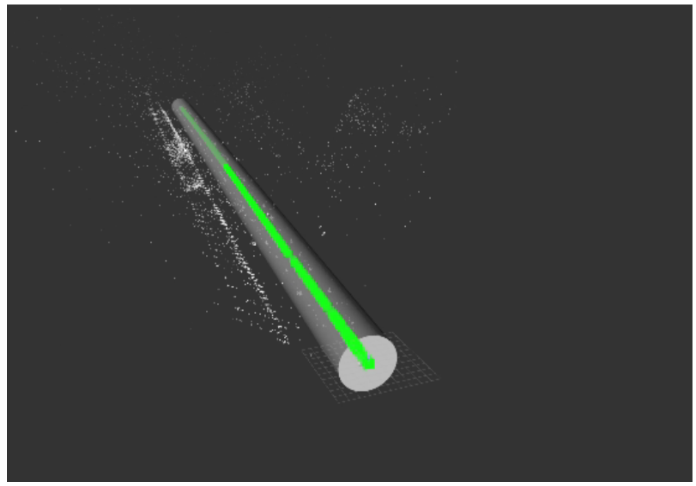

Fig. 10. Approach with a cylinder to select the relevant feature points to place a correct plane on the street.

E. Augmentation-Process

Open Graphics Library (OpenGL) is utilized to integrate virtual objects from Blender into a scene. This tool enables the rendering process. As depicted in Figure 13, the virtual cyclist has been added to the scene without material properties such as color or shadow. The grid displayed on the floor demonstrates that the cyclist is positioned on a recognized plane. The final step in our AR-ADAS pipeline involves

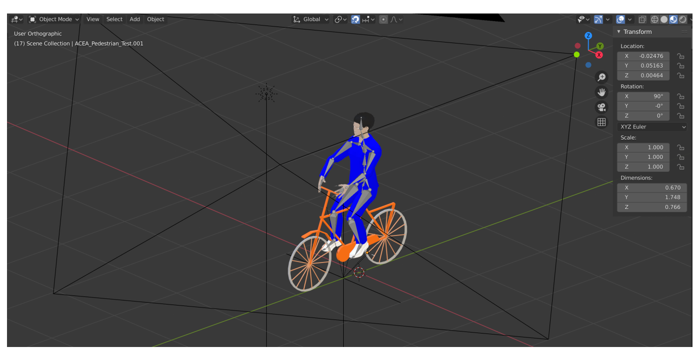

Fig. 11. NCAP-testobject cyclist using software *Blender*.

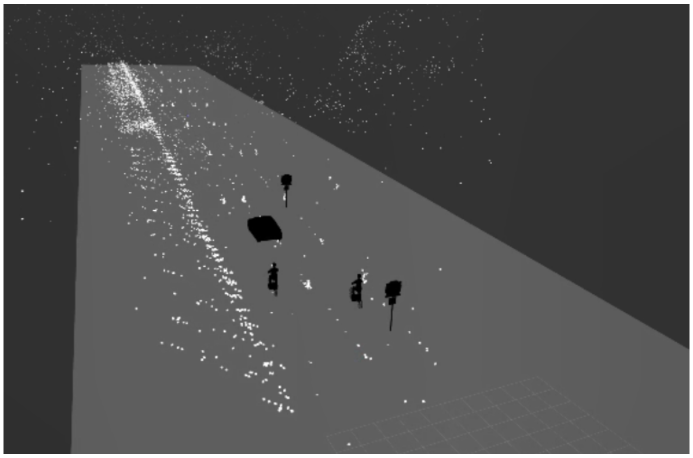

Fig. 12. Object placement on the surface in the Point Cloud.

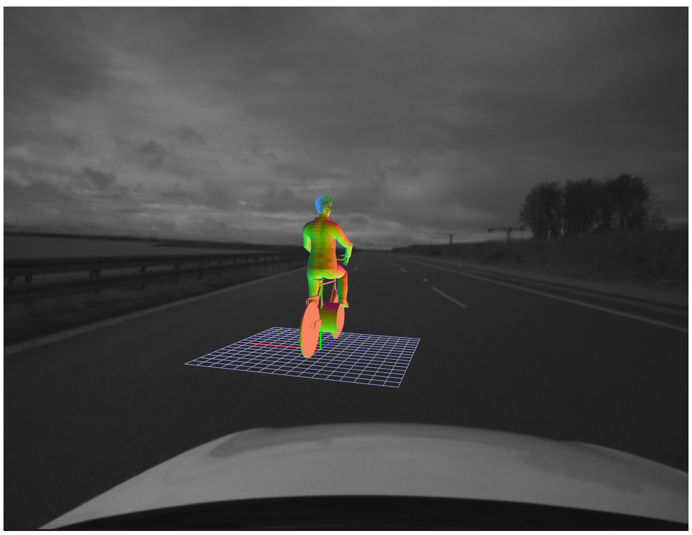

Fig. 13. Augmented cyclist in NCAP-scene.

driving the physical vehicle through the test scene once more. This time, ORB-SLAM is not utilized in mapping-mode, but rather in localization mode, which results in a faster vehicle speed due to reduced computational requirements. The real camera images are now augmented with virtual objects.

F. Advantages of our Approach

Our approach aims to merge the benefits of testing in simulation, such as reproducibility, flexibility, and efficiency, with the complexities of the real-world vehicle and environmental conditions. This approach seeks to bridge the gap between testing methods and enable the testing of more intricate scenarios, thus enhancing road safety.

Industry-based ViL-approaches demonstrate the need for such a method. Unlike conventional ViL-methods, which simulate a virtual environment to stimulate vehicle sensors while maintaining the vehicle's realistic dynamics, our approach represents a step towards more realistic vehicle testing. By utilizing AR, our approach allows us to use the real environment with all its characteristics, eliminating any simplifications made in simulation. Figure 14 contrasts existing ViL-methods, represented at the top, with our approach, depicted at the bottom.

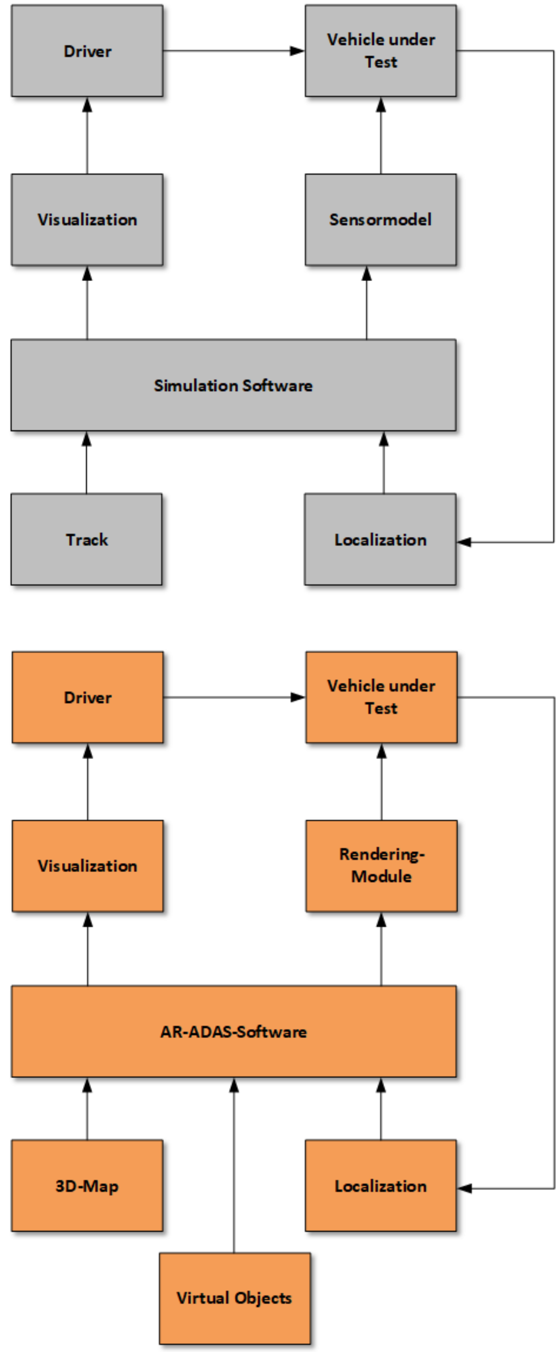

Fig. 14. Existing ViL-approach on top [22] and our approach of using augmented reality in advanced driver assistance systems on the bottom.

The overlay of lanes allows for independent testing of a lane departure warning system, regardless of the testing ground. Scenarios such as the presence of temporary lane markings or missing sections can be tested in the same area. Variations in lane width or international differences in lane markings can be represented. The camera image can also be augmented with superimposed vehicles ahead to test congestion assistance systems. That eliminates the need for second vehicles and drivers in the initial testing phase, reducing costs and increasing safety for test engineers. Test cases with traffic signs, pedestrians, and cyclists can be situational and quickly added, and a combination of different test scenarios is also possible. The limitless variety of test scenarios allows for a significant increase in the depth of testing during the early stages of development, leading to improved testing quality and overall system. With the increasing number of driver assistance systems and the move towards autonomous driving, the application area of the software can be expanded as needed.

G. Challenges for our Approach

Further advancements and optimizations regarding accuracy, robustness, and runtime can be seen in developments based on the ORB-SLAM approach, such as ORB2-SLAM [20] and ORB3-SLAM [16]. While ORB-SLAM demonstrates impressive performance in well-structured sequences, it can encounter errors in poorly structured sequences, such as those found in Euro NCAP test scenarios, or when feature points temporarily disappear due to factors like motion blur [29]. Along with accuracy, the runtime of the algorithm is also a crucial factor. Today, camera systems operate at a frame rate of 30 to 60 frames per second (fps), and the maximum overall runtime for processing a single frame can be found in Table II.

TABLE II
SEVERAL FRAMERATES AND THE ACCORDING MAXIMUM RUNTIME

Framerate	Maximum Runtime
10 fps	$\frac{1}{10}\,s = 0.1000\,s$
30 fps	$\frac{1}{30}\,s = 0.0333\,s$
40 fps	$\frac{1}{40}\,s = 0.0250\,s$
45 fps	$\frac{1}{45}\,s = 0.0222\,s$
50 fps	$\frac{1}{50}\,s = 0.0200\,s$
60 fps	$\frac{1}{60}\,s = 0.0167\,s$

For a successful evaluation of ADAS-test scenarios, the AR system must be able to orient itself in the environment very accurately [20]. One cause is the missing feedback about the impact intensity of test dummies when crashing them. For this reason, it is necessary to know the exact position of the car on the test track to calculate the intensity of the impact based on the braking distance. When using Euro NCAP test scenarios, velocities up to

$$130\,\frac{km}{h} \mathrel{\widehat{=}} 36.111\,\frac{m}{s} \tag{7}$$

are tested [26]. The AR algorithm must have a faster runtime compared to the speed of the camera system. The distance d the vehicle covers within a frame at any given velocity and framerate can be calculated by:

$$d = \frac{v_{Vehicle}\,[\frac{m}{s}]}{Framerate\,[\frac{frames}{s}]} \tag{8}$$

At a speed of 130 $\frac{km}{h}$ and a camera framerate of 30 fps, the vehicle travels

$$d = \frac{36.111\,[\frac{m}{s}]}{30\,[\frac{frames}{s}]} = 1.204\,\frac{m}{frame}. \tag{9}$$

Accordingly, for a framerate of 60 fps at the same speed, a distance of

$$d = \frac{36.111\,[\frac{m}{s}]}{60\,[\frac{frames}{s}]} = 0.602\,\frac{m}{frame} \tag{10}$$

is covered. A deceleration of one frame means a deviation of the test results of 0.602 to 1.204 metres. Based on the high speed of the car and the camera, and the high need for precision in object placement, it is clear that the requirements for this application of AR are far more strict than for the usual application for human users. Another task we have to deal with is the low texture of the environment and the uniformity that comes with it, so that *Image Matching* can occur. The use of multiple objects such as pylons to enrich the texture of the environment or approaches to *Optical Flow* can support this.

H. Visualization for the Testdriver

The main task of the Human-Machine-Interface (HMI) is to make the AR perceptible to the test driver in real time that the ADAS functions of the test vehicle, as well as the human interaction, can be evaluated. The acceptance of the HMI as an interface for the experience plays an important role. This depends for the most part on the quality of the display, interaction, and haptics [30]. For our approach, the selection of a suitable HMI concept focuses on visualization and interaction. To display AR visibly, the use of a suitable HMI or a corresponding display is necessary. Possible screen approaches are classified into feature classes based on their properties. Displays that use a medium-direct view through to the real environment in 3D belong to the class of See-Through (ST)-displays. Monitor-Based (MB)-displays only allow an indirect view of the real environment. Live or stored videos (2D) are used for this technology. Indirect displays (3D objects: video ST), which visualize AR in 3D using video, also belong to the group of ST displays. The three-dimensional concept is crucial here. The processing of the 2D-camera data of the real environment used, through 3D-scene modeling, makes it possible in the first place to integrate the virtual objects in the correct perspective (2D). Video-based ST displays (video ST) are used if the recording and playback of this same AR on an indirect display take place almost simultaneously. Optical ST-displays (3D-Objects:optical ST) are used when the reproduction of the virtual objects in combination with the direct view of the real environment is correctly integrated. The visualization of AR according to Azuma limits the AR-capable displays to those that can display virtual three-dimensional objects correctly oriented in perspective [7]. For the identification of suitable HMI approaches for testing camera-based ADAS, only these ST displays fulfill the necessary criteria. Figure 15 shows a summary of the different categories for visualization.

HMI approaches in which stationary displays are mechanically fixed to the vehicle for the duration of the test belong to the Head-Up-Display (HUD) group. Head-Up-Displays used in automotive vehicles to show the driver the actual speed or using the display of a smartphone or tablet belong to this category. Those in which the display is attached to the head like when using Virtual Reality (VR)-glasses or AR glasses belong to the Head-Mounted-Display (HMD) group [31]. In both HMD and HUD, HMI approaches of optical and video-based ST displays are identified. In the further progress of the approach to use AR as a visualization for the driver, different evaluations must be carried out [1].

I. Further Thoughts about Using Augmented Reality Simulation for Testing Advanced Driver Assistance Systems in Future Automotive Vehicles

In the first step, our approach will be transferred to camera-based sensors. As already highlighted in the previous chapters, only a few ADAS functions, such as traffic sign recognition or LDW, only access the camera. To evaluate further tests and achieve the equal behavior of the ADAS-ECU (cf. Figure 4) in reality as in using AR, the integration of further sensor technology such as radar is needed. It should also be mentioned that Euro NCAP test scenarios according to the current state only take place under ideal conditions (sun position noon - no or only a few shadows and reflections, no other road users, no rain, etc.) [26]. Using our approach is intended to further increase the complexity and realism of Euro NCAP test scenarios. Another aspect is the visualization of the Augmentation for the driver. Here, one considerable aspect is the acceptance of the user by AR. Further investigations into a visualization for the user are being pursued as part of this project.

VII. Conclusion

In this paper, we have proposed an approach using AR in automotive vehicles. The use of AR in ADAS is intended to combine the advantages of simulative test procedures, such as reproducibility and cost savings, with the advantages of test procedures in reality (complexity of the entire vehicle and the environment). We modeled the problem of creating an urban environment to use AR for testing in high-speed ADAS. Our approach is based on combining vSLAM-algorithms with Artificial Intelligence (AI) to use Object Detection. That should help generate a better overall performance concerning computing speed and accuracy. Creating a virtual three-dimensional environment with a superior understanding of the individual objects should, in a further step, make it possible to augment other sensors such as the car's radar and LiDAR with objects in addition to the camera data. That should once again increase the overall performance of the entire system. In addition to providing a link between virtual and real test procedures, this approach intends to increase the complexity of potential test procedures, accelerate the development speed of ADAS functions, and improve safety for future mobility solutions.

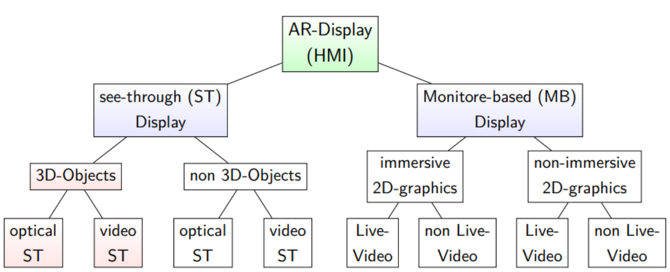

Fig. 15. Augmented reality display classes inspired by Milgram [7].

References

[1] M. Weber, T. Weiß, F. Gechter, R. Kriesten, "Augmented Reality Simulation for Testing Advanced Driver Assistance Systems in Future Automotive Vehicles," in: SIMUL 2022, The Fourteenth International Conference on Advances in System Simulation, Lisbon, Portugal, October 2022.

[2] K. Bengler et al., "Three decades of driver assistance systems: Review and future perspectives," in: IEEE Intelligent Transportation Systems Magazine vol. 6, no.4, pp. 6–22, Winter 2014.

[3] F. Schuldt, F. Saust, B. Lichte, M. Maurer and S. Scholz, "Efficient systematic test generation for driver assistance systems in virtual environments - Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen," 2013, [Online], Available from: https://publikationsserver.tu-braunschweig.de/servlets/MCRFileNodeServlet/dbbs-derivate-00031187/AAET-Schuldt-Saust-Lichte-Maurer-Scholz.pdf, accessed 2023.02.09.

[4] B.-J. Kim and S.-B. Lee, "A study on the evaluation method of autonomous emergency vehicle braking for pedestrians test using monocular cameras," Applied Sciences 10, no. 13: 4683, July 2020, doi: 10.3390/sapp10134683.

[5] C. Miquet et al., "New test method for reproducible real-time tests of ADAS ECUs: "Vehicle-in-the-loop" connects real-world vehicles with the virtual world," in: 5th International Munich Chassis Symposium 2014, pp. 575-589, July 2014.

[6] J.E. Stellet et al., "Testing of Advanced Driver Assistance towards automated driving: A survey and taxonomy on existing approaches and open questions," in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 1455–1462, September 2015.

[7] R.T. Azuma, "A survey of augmented reality," in: Teleoperators and Virtual Environments, pp. 355–385, August 1997.

[8] Pokémon GO, "Developer Niantic is working on a game for tourists - Pokémon GO: Entwickler Niantic arbeitet an einem Spiel für Touristen," [Online], available from: https://mein-mmo.de/pokemon-go-entwickler-app-touristen/, accessed 2023.02.09.

[9] A. State, G. Hirota, D. Chen, W. Garrett and M. Livingston, "Superior augmented reality registration by integrating landmark tracking and magnetic tracking," in: SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 429-438, August 1996.

[10] R. Chatila and J-P. Laumond, "Position referencing and consistent world modeling for mobile robots," in: IEEE International Conference on Proceedings Robotics and Automation, pp. 138-145, 1985.

[11] J. Engel, J. Sturm and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2815-2821, 2012.

[12] T. Taketomi, H. Uchiyama and S. Ikeda, "Visual SLAM algorithms: a survey from 2010 to 2016," in: IPSJ Transactions on Computer Vision and Applications, doi: 10.1186/s41074-017-0027-2, 2017.

[13] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," in: Cambridge University Press, second edition, ISBN: 0521540518, 2004.

[14] R. Mur Artal, "Real-Time Accurate Visual SLAM with Place Recognition," Ph.D.-thesis, Universidad de Zaragoza, Prensas de la Universidad, Zaragoza, Spain, 2017.

[15] P. Furgale, J. Rehder and R. Siegwart, " Unified temporal and spatial calibration for multi-sensor systems," in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1280–1286, Tokyo, Japan, 2013.

[16] C. Campos et al., "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," in: IEEE Transactions on Robotics, pp. 1-17, 2021.

[17] R. A. Newcombe, S. J. Lovegrove and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in: IEEE International Conference on Computer Vision (ICCV), pp. 2320–2327, Barcelona, Spain, 2011.

[18] J. Engel, T. Schoeps and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in: European Conference on Computer Vision (ECCV), pages 834–849. Zurich, Switzerland, 2014.

[19] J. Engel, V. Koltun and D. Cremers, "Direct sparse odometry," in: arXiv:1607.02565, July 2016.

[20] R. Mur-Artal, J. Montiel and J. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," in: IEEE Transactions on Robotics, pp. 1147-1163, 2015.

[21] M. Nagai, "Research into ADAS with autonomous driving intelligence for future innovation," in: 5th International Munich Chassis Symposium 2014, pp. 779–793, January 2014.

[22] H. Winner, S. Hakuli, F. Lotz and C. Singer, "Manual driver assistance systems - basics, components and systems for active safety and comfort - Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und Systeme fuer Aktive Sicherheit und Komfort," in: Springer Vieweg, Wiesbaden, March 2015, [Online], Available from: https://link.springer.com/content/pdf/10.1007/978-3-658-05734-3.pdf, accessed 2023.02.09.

[23] M. Darms, "A basic system architecture for sensor data fusion of environmental sensors for driver assistance systems - Eine basis-systemarchitektur zur Sensordatenfusion von Umfeldsensoren fuer Fahrerassistenzsysteme," Ph.D.-thesis, Technische Universität Darmstadt, 2007, [Online], available from: https://tuprints.ulb.tu-darmstadt.de/914/ accessed 2023.02.09.

[24] P. Seiniger and A. Weitzel, "Testing procedures for consumer protection and legislation - Testverfahren fuer Verbraucherschutz und Gesetzgebung," in: Manual driver assistance systems - Basics, components and systems for active safety and comfort - Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und Systeme fuer Aktive Sicherheit und Komfort, pp. 167–182, Springer Vieweg, Wiesbaden, March 2015. [Online]. Available from: https://link.springer.com/content/pdf/10.1007/978-3-658-05734-3.pdf, accessed 2023.02.09.

[25] Euro NCAP, "AEB Pedestrian," [Online], available from: https://www.euroncap.com/en/vehicle-safety/the-ratings-explained/vulnerable-road-user-vru-protection/aeb-pedestrian/, accessed 2023.02.09.

[26] R. Fredriksson, M.G. Lenné, S. van Montfort and C. Grover, "European NCAP program developments to address driver distraction, drowsiness and sudden sickness," November 2021, [Online], available from: https://www.frontiersin.org/articles/10.3389/fnrgo.2021.786674/full, accessed 2023.02.09.

[27] M. Sieber et al., "Validation of driving behavior in the vehicle in the loop: Steering responses in critical situations," in: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), 2013.

[28] C. Kamann and C.Rother, "Benchmarking the Robustness of Semantic Segmentation Models," in: IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2020.

[29] Yu et al., "DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments," in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018), 2018.

[30] J. Brade and A. Koegel, "Presence in virtual reality - Key to acceptance and transferability?!," in: 5. Fachkonferenz zu VR/AR-Technologien in Anwendung und Forschung, VAR² 2019, pp. 59-71, December 2019.

[31] R. Doerner, "Fundamentals and methods of virtual and augmented reality - Grundlagen und Methoden der Virtuellen und Augmentierten Realitaet," in: Virtual and Augmented Reality (VR/AR), Springer Viweg, pp. 1-143, 2019.