Automation of the UNICARagil Vehicles

Dr.-Ing. Michael **Buchholz**; Fabian **Gies**, M.Sc.; Andreas **Danzer**, M.Sc.;
Matti **Henning**, M.Sc.; Charlotte **Hermann**, M.Sc.; Manuel **Herzog**, M.Sc.;
Markus **Horn**, M.Sc.; Markus **Schön**, M.Sc.; Nils **Rexin**, M.Sc.;
Prof. Dr.-Ing. Klaus **Dietmayer**
Institute of Measurement, Control and Microtechnology, Ulm University, Ulm, Germany

Carlos **Fernandez**, Ph.D.; Johannes **Janosovits**, M.Sc.; Danial **Kamran**, M.Sc.;
Christian **Kinzig**, M.Sc.; Dr. Martin **Lauer**; Eduardo **Molinos**, Ph.D.;
Prof. Dr.-Ing. Christoph **Stiller**; Lingguang **Wang**, M.Sc.
Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany

Stefan **Ackermann**, M.Sc.; Tobias **Homolla** M. Sc.;
Prof. Dr. rer. nat. Hermann **Winner**
Institute of Automotive Engineering, Technical University of Darmstadt, Darmstadt, Germany

Grischa **Gottschalg**, M.Sc.; Dr.-Ing. Stefan **Leinen**; Prof. Dr.-Ing. Matthias **Becker**
Institute of Geodesy, Chair of Physical Geodesy and Satellite Geodesy, Technical University of Darmstadt, Darmstadt, Germany

Johannes **Feiler**, M.Sc.; Simon **Hoffmann**, M.Sc.; Dr.-Ing. Frank **Diermeyer**
Institute of Automotive Technology, Technical University of Munich, Munich, Germany

Bastian **Lampe**, M.Sc.; Till **Beemelmanns**, M.Sc.; Raphael **van Kempen**, M.Sc.;
Timo **Woopen**, M.Sc.; Prof. Dr.-Ing Lutz **Eckstein**
Institute for Automotive Engineering (ika), RWTH Aachen University, Aachen, Germany

Nicolai **Voget**, M.Sc. M.Sc.; Prof. Dr.-Ing. Dieter **Moormann**
Institute of Flight System Dynamics, RWTH Aachen University, Aachen, Germany

Inga **Jatzkowski**, M.Sc.; Dipl.-Ing. (FH) Torben **Stolte**, M.Sc.;
Prof. Dr.-Ing. Markus **Maurer**
Institute of Control Engineering, Technische Universität Braunschweig, Braunschweig, Germany

Dr.-Ing. Jürgen **Graf**; Dr.-Ing. Edgar **v. Hinüber**
iMAR Navigation GmbH, St. Ingbert, Germany

Dr.-Ing. Norbert **Siepenkötter**
flyXdrive GmbH, Aachen, Germany

Contact: michael.buchholz@uni-ulm.de

Summary

The German research project UNICAR*agil* is a collaboration between eight universities and six industrial partners funded by the Federal Ministry of Education and Research. It aims to develop innovative modular architectures and methods for new agile, automated vehicle concepts. This paper summarizes the automation approach of the driverless vehicle concept and its modular realization within the four demonstration vehicles to be built by the consortium. On-board each vehicle, this comprises sensor modules for environment perception and modelling, motion planning for normal driving and safe halts, as well as the respective control algorithms and base functionalities like precise localization. A control room and cloud functionalities provide off-board support to the vehicles, which are additionally addressed in this paper.

1 Introduction

The UNICAR*agil* project [1] aims to develop disruptive modular architectures and methods for new agile, automated vehicle concepts [2]. The modular concepts will be realized in four different demonstration vehicles to be built by the consortium. UNICAR*agil* is a collaboration of eight universities and six industrial partners and funded by the German Federal Ministry of Education and Research. Started in 2018, the project has reached mid-term now. Within the project, different research topics are addressed. The main research domains are:

- geometry, focusing on the modular construction and design of the vehicle concept;
- mechatronics, developing and realizing a modular and redundant electric/electronic architecture;
- software focusing on a service-oriented architecture for automotive use-cases;
- safety, researching on concepts to ensure safe operation with such a modular vehicle comprising service-oriented approaches; and
- automation, aiming for a SAE Level 4 driverless operation of the vehicles.

Numerous results from these domains have already been published, see [1] for a complete list of publications. An overview on the mid-term status of the project is given in the accompanying overview paper [3]. Further papers in this symposium present aspects from other research domains than automation:

- [4] describes the dynamics modules in detail, which provide the traction, braking, and steering abilities at each of the four wheels of the UNICAR*agil* vehicles;
- [5] gives insight into the service-oriented software architecture for automotive applications developed within the project; and
- [6] presents the interaction of the different safety approaches undertaken in the project.

The rest of the paper is structured as follows: Section 2 gives an overview on the domain of automation. Then, Sections 3 and 4 describe the on-board and off-board functions of the UNICAR*agil* automation concept, respectively. Finally, the paper closes with some conclusions and an outlook on the next steps in Section 5.

2 Overview on the UNICAR*agil* Automation Concept

The automation concept in UNICAR*agil* follows the overall project goal of modularity and redundancy. Additionally, it makes use of the concepts, components, etc. developed within the other domains to realize an SAE level 4 driverless functionality of the UNICAR*agil* vehicles. For example, the automation is embedded within the modular electronic structure of the mechatronic architecture. This architecture is inspired by a biological nervous system, as described in [2], [7]. Additionally, the automation provides its functionality by services within the automotive service-oriented architecture (ASOA) from the software domain [5]. Stemming from the safety domain, the self-perception collects available information of services and components and derives a system-wide picture of the vehicles current capabilities. The derived representation of the vehicle's capabilities is an important input for the vehicle automation in order to ensure safe vehicle behavior, cf. [6].

2.1 System Architecture

On-board the vehicle, the automation comprises of sensor modules, which redundantly surveil the vehicle's environment and calculate an environment model. A central processing unit, called cerebrum, then fuses this information from several sensor modules and realizes behavior and trajectory planning. This part of the systems architecture is depicted in Fig. 1. The trajectories are transmitted to another central electronic control unit (ECU), the so-called brainstem, which - besides other functions - controls the vehicles dynamics modules to follow the trajectories. As part of the safety concept, the brainstem additionally is able to realize a safe halt with own additional sensors, the so-called platform sensors. As a basis for trajectory planning and control, dynamic state of the vehicle and its localization with respect to a digital map are estimated.

In addition to the on-board functionalities, the UNICAR*agil* automation concept includes also off-board support for the vehicles. A control room provides the fleet management as well as possible support by a teleoperator, while cloud services represent the collective knowledge of all vehicles of the fleet. The so-called info-bee, a drone equipped with sensors, can support the vehicles and/or the control room when the vehicles' sensors have limited view on relevant parts of the environment.

Before these on-board and off-board components of the automation concept are described in more detail, the superior vehicle operation mode management is shortly introduced. It is jointly developed between the software architecture, the safety, and the automation domain to allow for safe testing and research of all new concepts within the four UNICAR*agil* vehicles that are currently built by the consortium.

2.2 Vehicle Operating Mode Management

Although the project UNICAR*agil* strives towards SAE level 4 automation [8], it quickly became obvious that there is a need for operating automated vehicles separate from the usual automated operation. This requirement arises from considering different aspects. On the hand, there is the need for maintenance in repairs shops. On the other hand, considerations regarding safety and functional limitations, which can arise due to the complexity of the vehicle automation task, must be taken into account. Thus, we implement a vehicle operating mode management that coordinates the four operating modes: *Automated mode*, *manual mode*, *safe halt mode*, and *remote control mode*.

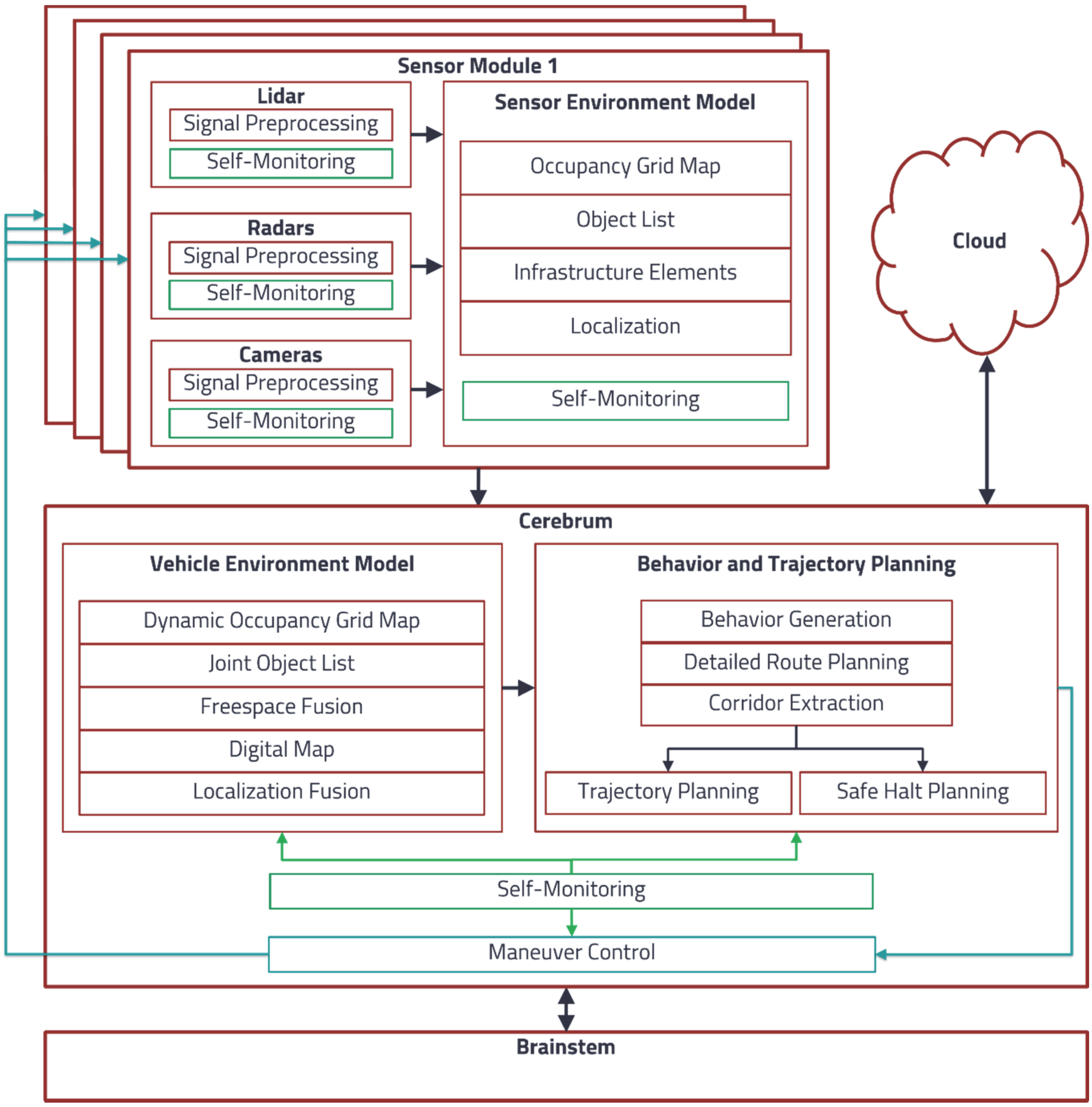

Fig. 1 System architecture of automation sensor modules and cerebrum.

The *manual mode* reflects the need for potential maintenance in repair shops and, thus, allows to move the vehicle by manual control of a human's inputs. In UNICAR*agil*, we use either a combination of a side stick and a brake pedal or an external control input. The combination of sidestick and brake pedal can be easily connected and mounted to an interface inside the cabin. For the use case *auto*CARGO, which does not feature seats, an external remote control will be available.

Safety considerations lead to the *Safe Halt mode*. Within this mode, the fallback system – the safe halt procedure – is activated. Still a level 4 functionality, it serves as a fail-safe procedure in case system parts superimposed on the trajectory tracking fail, e.g., in case of a complete loss of the perception system. It is running in hot-standby and stops the vehicle following a safe path while maintaining a certain degree of obstacle detection. After boot-up, during which all necessary services are started, the automated operation is only activated if all prerequisites for the automated operation are fulfilled. Transitions between operating modes are only possible in standstill. An exception is the transition to the Safe Halt mode. As the transition into the Safe Halt mode triggers the safe halt procedure, this transition can be also executed while the vehicle is moving. Further details regarding Safe Halt can be found in Subsection 3.2.3 as well as in [6] and [9].

In case of a Safe Halt maneuver or if the automated vehicle cannot handle a traffic situation itself, it comes to a stop and the *remote control mode* is activated. Then, an operator in the control room (cf. Section 4.3) can resolve the situation by remotely controlling the vehicle and contacting passengers.

For fulfilling its task, the vehicle operating mode management is fed by the self-perception (see Subsection 3.1.1) and other vehicle-internal inputs. In case of a transition, the orchestrator of the service-oriented software architecture reconfigures the services for the active operating mode. Services required in the Safe Halt mode are active at all times.

3 On-board Automation Functionality

This section describes the main components of the automation system on-board the UNICAR*agil* vehicles in detail. The structure of the section follows the common Sense-Plan-Act scheme.

3.1 Sensing and Modelling of the Vehicle's State and Environment

Automation of the vehicle starts with sensing the vehicle's own state. Compared to usual dynamic state estimation, this is enhanced by a self-awareness concept from the safety domain in UNICAR*agil*. In addition, the vehicle's environment is surveilled by several sensors and represented within environment models.

3.1.1 Self-perception and Vehicle Dynamic State Estimation

The vehicle’s dynamic state consisting of the three dimensional position, velocity, acceleration, orientation, and angular rate is estimated by the vehicle dynamic state estimation (VDSE). The high requirements for the VDSE regard not only the availability and accuracy of the VDSE’s outputs, but also include the demand for quality information, as pointed out in [2]. By continuously providing accuracy and integrity information, which are used among others by the self-perception system in UNICAR*agil*, these requirements are met. The quality information, the estimated state and the estimated side slip angle are made available to other services, e.g. the trajectory control, that run within the implemented automotive service-orientated architecture (ASOA).

In the hardware architecture of UNICAR*agil*, the VDSE is part of the brainstem [2], [7]. However, the VDSE uses a dedicated hardware platform (VDSE-ECU) with four integrated microcontroller boards (CPU: ARM Cortex-A8). Furthermore, the VDSE-ECU also integrates two different MEMS-IMUs (IMU A: Analog Devices ADIS 16465, IMU B: Bosch Sensortec BMI160) and a multi-frequency, multi-constellation, Real-Time Kinematic (RTK) global navigation satellite system (GNSS) receiver with two antennas (NovAtel OEM7720). Additionally, the VDSE-ECU receives odometry observations, in particular wheel speeds and steering angels of the wheel, from the four dynamic modules via the ASOA.

Fig. 2 depicts the structure of the VDSE's sensor data fusion. The fusion algorithms are developed in parallel by two independent teams, resulting in two dissimilar filter algorithms (filter 1 and 2). In this way, the risk of simultaneous error occurrences stemming from hidden design flaws is minimized. The implemented filters input subsets of the available sensor data. Filter 2a and 2b are identical apart from the parameters for the used sensor subset. All filters provide previously mentioned accuracy and integrity information to a so called voting algorithm which will decide about the VDSE's outputs. This dissimilar hardware design and fusion architecture enables the detection and handling of critical errors and, thus, increases the availability and reducing the risk for road users.

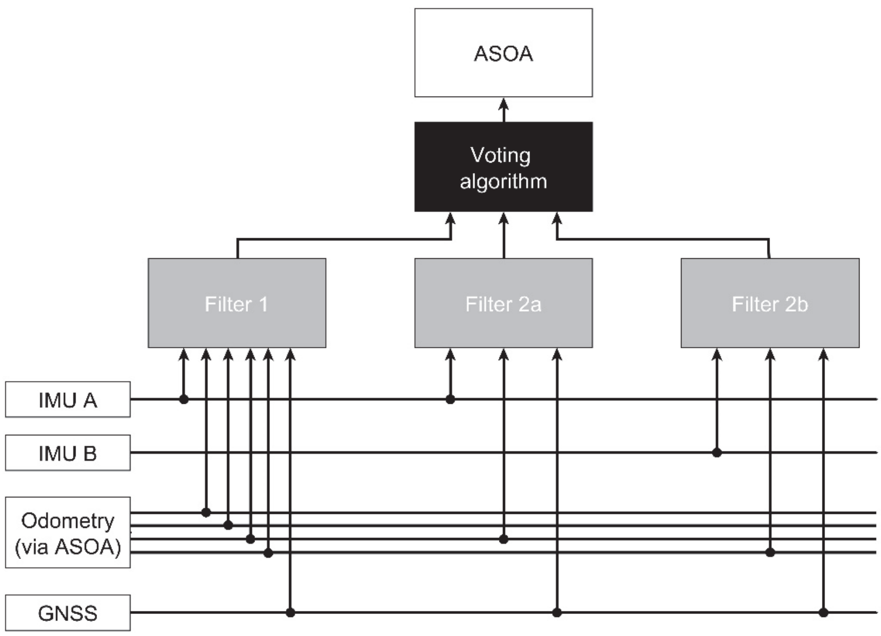

Fig. 2 Vehicle Dynamic State Estimation – Structure of sensor data fusion.

As the project UNICAR*agil* aims at SAE level 4 [8] automation, the vehicle system is responsible for all monitoring tasks additionally to the dynamic driving task. Thus, the system must monitor itself and must be able to make decisions that take the system's health and thereby its current capabilities into account. This self-monitoring is realized by implementing a self-perception system in the vehicle to provide a model-based view on the inner system state similar to how an environment perception system provides a view on the environment external to the vehicle [10][9]. In an automated vehicle, self-perception for the system includes a variety of different monitoring aspects. Among other tasks, such a self-perception system assesses the vehicle's current skills, detects degradations of these skills based on diagnosis data provided by the system's services and components and provides the information about the current vehicle skills to the vehicle's behavior generation as well as other services.

It follows that, additionally to the estimation of the vehicle state, self-perception of the vehicle system must be realized in order to ensure safe vehicle behavior. More information on the safety concepts pursued in the UNICAR*agil* project can be found in [6].

3.1.2 Sensor Modules

Environment perception and modeling are both crucial as well as heavily researched parts in the field of vehicle automation [11]. To realize the driverless concept of the UNICAR*agil* project, the respective need for a sensor setup providing high reliability and excellent sensing performance is therefore self-explanatory.

Accordingly, each UNICAR*agil* vehicle is equipped with four sensor modules, one at each corner of the vehicle. Each module contains a vertically stacked set of sensors chosen from the measurement principles lidar, radar, and camera, so that we can overcome shortcomings of one measurement principle using complementary sensors. A demonstration prototype as well as the first two assembled sensor modules are shown in Fig. 3.

Fig. 3 Demonstration prototype (left) and two assembled (right) sensor modules.

Using the specific sensors listed in Tab. 1, each sensor module has a combined perception area as depicted in Fig. 4. This field of view is roughly covering 270° horizontally, while the primary moving directions, forward and backward, are additionally covered by a high-resolution stereo camera system with a smaller field of view.

Tab. 1 Sensors used in each sensor module.

Measurement Principle	Sensor Model	Number of sensors	Sensor orientation
Radar	Continental ARS 408-21	2	1 x front 1 x side
Lidar	Velodyne VLP-32C Ultra Puck	1	-
Camera	FLIR Blackfly S BFS-U3-88S6C with Meike MK 6.5 mm f/2.0 Fisheye	2	1 x front 1 x side (same level)
	FLIR Blackfly S BFS-U3-88S6M with Meike MK 6.5 mm f/2.0 Fisheye	2	2 x front
IMU	Xsens MTi-30-2A8G4	1	-

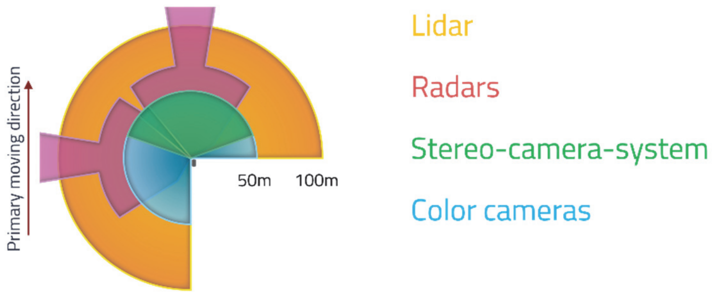

Fig. 4 Combined perception area of one sensor module.

With that, the combined sensor modules provide a robust 360° multi-covered perception area, where each module covers the blind spots of the two neighboring modules. If one module fully degrades or fails, the general 360° perception area persists. However, blind spot coverage of the neighboring modules cannot be guaranteed on automation level anymore. In this case, sensor components of the vehicle platform provide a fallback safety measure for safe halts.

The tasks of each sensor module consist of the detection of traffic participants, infrastructure, and free space, as well as the sensor module localization. Accordingly, a powerful computation unit is allocated to every sensor module containing high-end IT consumer products to process the sensor data in real-time. Time synchronicity between sensors is ensured by a set of four trigger boxes (one per sensor module), which are synchronized among each other.

Each sensor type individually contributes to the environment model as described in the following subsections. The processed sensor information is fused into an individual environment model per sensor module, which consist of an occupancy grid map [12], a list of tracked traffic participants using an LMB Filter [13], information about infrastructure elements, and a localization. The environment models from all sensor modules are transmitted to the vehicle's cerebrum, where a consolidated vehicle environment model, as described in Sec.3.1.4, is built.

3.1.2.1 Camera Data Processing

The processing steps for the undistorted camera images of each sensor module can be clustered in four separate subtopics. While the grayscale images are used to compute stereo, the color images are stitched to a panoramic view. The latter is utilized to perform both semantic segmentation and bounding box object detection in a large field of view. Finally, detected traffic lights are matched with the ones in the preexisting map.

3.1.2.1.1 Stereo Vision

The depth calculation (Fig. 5) is based on the grayscale cameras. The two cameras are mounted as a vertical stereo camera setup on each sensor module. The depth calculation is based on block matching by estimating planes in three-dimensional space following the approach of [14].

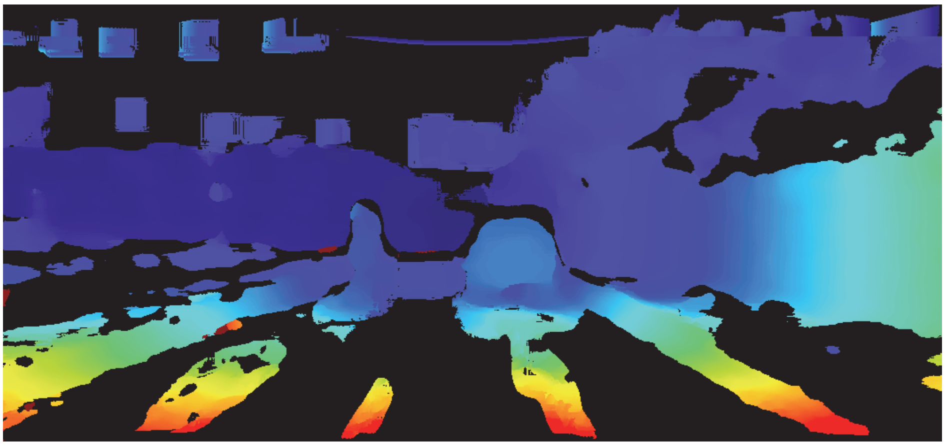

Fig. 5 Depth image from grayscale stereo camera system.

3.1.2.1.2 Image Stitching

Image stitching (Fig. 6, center) is performed in order to create a 270° panorama image using the synchronized color camera images. The main purpose of the stitching operation is the improvement of the pixelwise semantic segmentation (Fig. 5, left) as well as the bounding box object detection (Fig. 5, left, right) especially at the image borders.

To simplify the stitching process, we can assume in our case that the focal points of the cameras are approximately the same. Also, we use a spherical camera model. Since the orientation of the cameras is determined in the calibration process, the images can be superimposed along the matching azimuth angle. To avoid offsets between both images in the panoramic view seam carving is applied, which was presented in [15]. This method was beforehand successfully used for image blending and stitching, see [16].

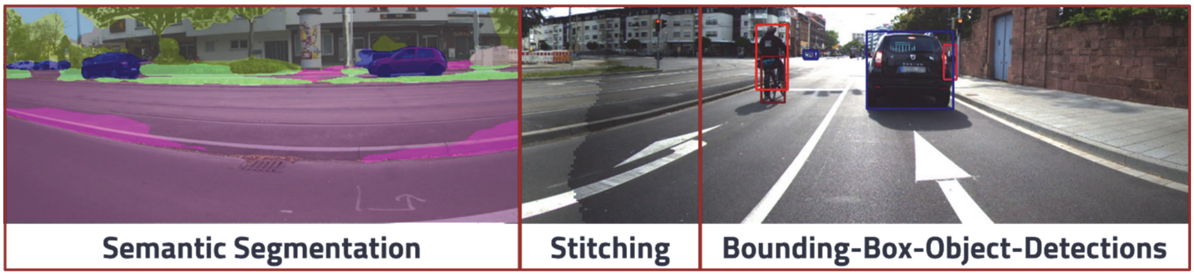

Fig. 6 Result of processed synchronized color camera images.

3.1.2.1.3 Pixelwise Semantic Segmentation and Bounding Box Detection

The pixelwise semantic segmentation and the bounding box detection is applied to the panorama image. The pixelwise semantic segmentation as well as the two-dimensional rectangular object bounding boxes are determined by the same neural network. The extracted labels correspond to the widely used Cityscapes dataset [17]. The used network architecture is based on the work of [18]. The network is built as an encoder-decoder structure with two outputs and consists mainly out of stacked ResNet blocks, which have the same structure as previously presented in [19]. The network is trained on the Cityscapes dataset [17] and was evaluated both on the Cityscapes dataset and on the KITTI benchmark [20].

3.1.2.1.4 Detection of Traffic Sings and Traffic Lights

The traffic sign as well as the traffic light detection is based on the bounding box object detection. Whereas the traffic signs do not change their state, the traffic lights need to be tracked over time. For this purpose, all possible lanes and traffic lights given the determined route are extracted in the map. After the classification, the traffic lights from the map are matched with the detections by projecting them into the image. If a correct association can be assured, their state is estimated and the result is used in planning. Detected traffic signs as well as traffic lights are extracted from the image and are separately classified by small convolutional neural networks (CNNs) to maintain computational efficiency. To provide robust estimation of the state for the tracked traffic lights, a filter is implemented. This reduces the impact of outliers, which are caused by misclassification. In return, this ensures a smooth acceleration profile.

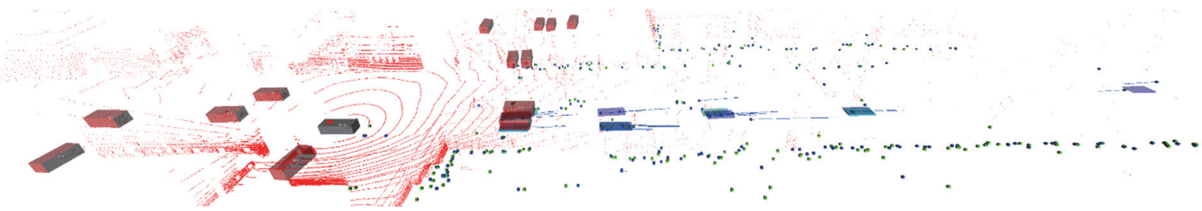

Fig. 7 Combined view of lidar data (red dots), radar data (blue and green dots) for both radar sensors and their respectively colored detected object bounding boxes. Measured and compensated radial velocities are depicted as arrows (blue, green).

3.1.2.2 Radar Data Processing

The radar data is used for 2D object bounding box estimation containing the objects position, orientation and size. In Fig. 7, the radar data of both radar sensors are shown in blue and green alongside their 2D object detections.

For the estimation, we built a patch of fixed spatial dimension around every point of the radar point cloud and then feed it into a three-stage PointNet [23] architecture. In the first stage, patches are classified as a traffic participant type or clutter. The second stage assigns segmentation probability scores to the radar points within the classified patches while masking contained clutter and the third stage then estimates the 2D bounding boxes based on classified and segmented radar points. Finally, we apply a heuristic to determine the most relevant objects for the output towards the later processing chain.

The corresponding architecture was presented in [21], while network training is based on real radar data. For the application in UNICAR*agil*, we extended the proposed architecture to incorporate additional classification granularity, namely the classes of car, truck, bike, pedestrian, and clutter.

3.1.2.3 Lidar Data Processing

The lidar data are also used for object bounding box estimation. Our approach is inspired by [22]: we first group the point cloud into fixed-size pillars. Then relevant pillars are fed to a network that is a combination of a PointNet [23] and a CNN. This network outputs the classification, position, orientation, and size of detected objects. Since the network may provide multiple detections for the same physical object, non-maximum suppression is applied to remove redundant objects. The lidar point cloud as well as the detected 3D object bounding boxes are shown in Fig. 7 in red.

In addition to the object bounding box estimation, the lidar is used to create an occupancy grid as described in [12]. As a prerequisite, we remove the ground points from the lidar point cloud by comparing the elevation of every point to the elevation of the corresponding field in an elevation grid. This elevation grid estimates the ground level for every cell in the grid. It is based on a preliminary ground classification of points and is continuously improved when new lidar measurements become available.

3.1.2.4 Localization

The localization software computes the pose of the ego vehicle from sensor data. In this project, the localization uses the GNSS data, the acceleration and rotation from the IMU, and the images from the cameras to compute the pose. Using imagery that is matched to a previously computed 3D map is key to cope built-up areas of poor GNSS reception. The localization system combines the advantages of its individual components.

The IMU provides a high local precision with high data rates at low latencies. IMU data are robust to many environmental conditions. RTK-capable GNSS is complementary and provides absolute accuracy in open sky areas. Unlike many other localization systems, the implemented software uses map-relative accuracy even in GNSS-deprived areas. The result is robust, precise and accurate 6d-pose and velocity measurements in real time, even in environments with poor GNSS availability.

Fig. 8 shows the same scene recorded at different times. Images are taken several months apart. The trees in the background show leaves on the one image and none on the other. The image on the left is used to build the 3D map whereas the image on the right is used to localize the vehicle relative to that map. The 3D vector map is shown in both images for comparison.

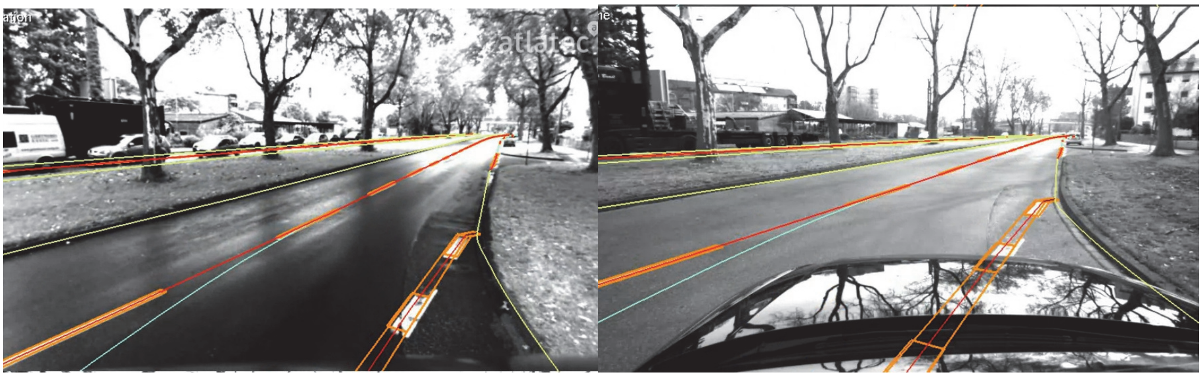

Fig. 8 Example of mapping and localization.

3.1.3 Platform Sensors

The safety activities in UNICAR*agil* project [6] lead to functional safety requirements for the automated vehicle. Even with significant degradations of the primary perception system (the sensor modules) and driving functions, the automated vehicle shall be enabled to be transferred into a risk-minimized state at all times. For this purpose, the automated fallback system Safe Halt [9] is introduced.

To provide the availability of the fallback system, the relevant hardware and communication architecture of the vehicle in UNICAR*agil* is fail-safe. Independent from the primary perception system, a secondary perception system (the platform sensors) is used by the fallback system so that the monitoring of the driving environment is possible even in case of severe failures of the primary perception system.

The vehicles in UNICAR*agil* can perform a 360° direction of movement. However, the maximum velocities are demonstrated only in the longitudinal direction of the vehicle: up to 20 m/s forward and 10 m/s backward. The fallback system uses embedded hardware and thus, does not provide the computing power known from high-performance computers with graphics cards. Hence, the raw sensor data are processed by the sensor hardware. Due to the required sensor range in the main driving directions, radar sensors are installed in the front and rear of the vehicle. To monitor the 360° motion direction, the two radar sensors are supplemented by a 360° perception system. Driving with large side slip angles is only permitted at low vehicle velocities, thus a comparatively small sensor range is required. Due to the required sensor range, ultrasonic sensors and cameras with fisheye lenses are used as 360° perception system.

3.1.4 Environment Models

Each sensor module produces a local environment model, which is represented in two different formats. First, an occupancy grid map is generated, which represents the environment in a local grid. The occupancy grid map is filled from the lidar sensor measurements by using an inverse sensor model and an occupancy mask to sum up the reflections at a single grid cell. Finally, each cell holds an occupancy probability, which indicates if the cell is occupied, free or unknown. Secondly, every detected traffic participant is represented in a list of objects, which holds the objects states, e.g. position, orientation and velocity, a classification type and an existence probability. To generate this object list, multiple subsequent steps are necessary. Modern deep learning approaches are developed to detect other traffic participants from single sensor measurements, as described in section 3.1.2. A multi-object tracking algorithm fuses these object detections to estimate a full object state by temporal filtering. Additionally, the vehicle pose is estimated relative to the high precision digital map using the camera system and the IMU (see Subsection 3.1.2.4). Finally, the data are transmitted to the cerebrum.

The cerebrum consists of two major components, the vehicle environment model and the behavior and trajectory planning. Based on the current state of the environment model, the vehicle's next driving maneuver is calculated in the behavior and trajectory planning module. An overview of the architecture is shown in Fig. 1. For safe automated driving it is of great importance that the data from the environment model are highly reliable. For that reason, the four independent sensor module environment models are fused to receive a redundant and accurate vehicle environment model. Due to the placement of sensors and sensor modules (see Subsection 3.1.2), a full 360° environment perception is guaranteed. If one sensor module fails, the environment model can still be calculated but with less reliability. This behavior enables a fail-safe deployment of the vehicle's environment model for the behavior and trajectory planning.

The environment model consists of multiple subparts for an object-based and grid-based representation of the environment. First, a dynamic occupancy grid map (DOGMa) [12] is calculated by fusing the occupancy grid maps from the sensor modules. In addition, the DOGMa estimates dynamic areas using a particle filter. Besides the occupancy probability, every grid cell holds an estimation of a local velocity. This extension enables a separation between dynamic and static obstacles, which is crucial for estimating the drivable area. Secondly, the individual object lists from the sensor modules are fused to a joint object list by a track fusion approach. Here, the redundancy of every tracked object is considered and besides the dynamic states, an existence probability is calculated. Every object exceeding a minimum required threshold will be added to the environment model's object list. This process can further be improved by the approach described in [24]. Besides estimating the dynamic environment, the static infrastructure is provided by a digital map. This map consists of stop lines, crosswalks, traffic lights, traffic signs and highly accurate mapped lanes (see Subsection 3.1.2.4) and is represented in the Lanelet2 format from [25]. After fusing the individual localization results from the sensor modules to an overall vehicle pose, the data from the digital map can be transformed from the vehicle coordinates to global coordinates and vice versa. A full depiction of the vehicle's environment model is shown in Fig. 9.

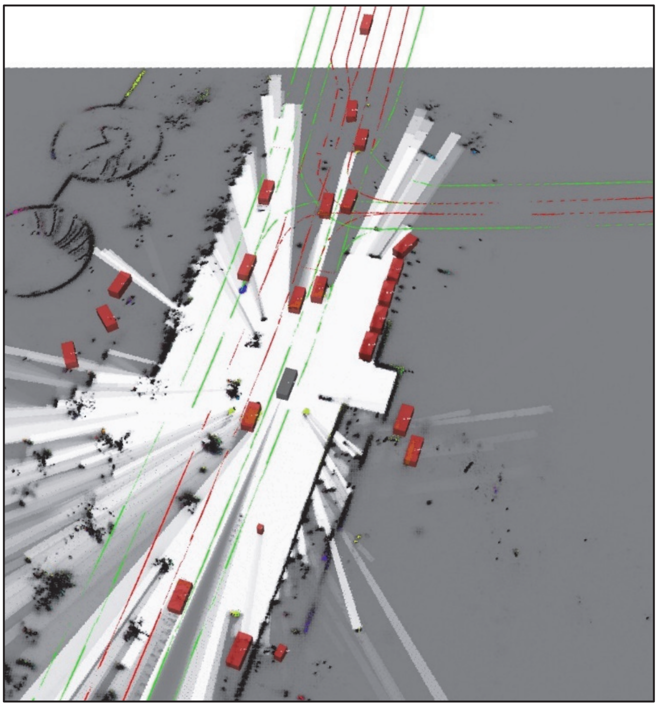

Fig. 9 Vehicle environment model with tracks (red boxes), dynamic occupancy grid map and digital map (green and red lines).

3.2 Planning and Behavior

Planning and behavior generation are done in two steps on the vehicle’s cerebrum. The decision making module selects a general maneuver, while the trajectory planner creates a trajectory for that maneuver.

3.2.1 High Level Decision Making

The decision making module is responsible to provide suitable behaviors for different situations that can occur during automated driving in an urban environment such as stopping for pedestrians near crosswalks, yielding to other vehicles at intersections or driving when the traffic light state turns to green. For each of these situations, specific sub behavior policies are implemented. Each sub behavior policy has specific activation conditions that are valid only when the ego vehicle is approaching a traffic situation that is related to that sub behavior.

The structure of the decision making module, its sub behavior policies, and its interface between perception and planning modules is depicted in Fig. 10.

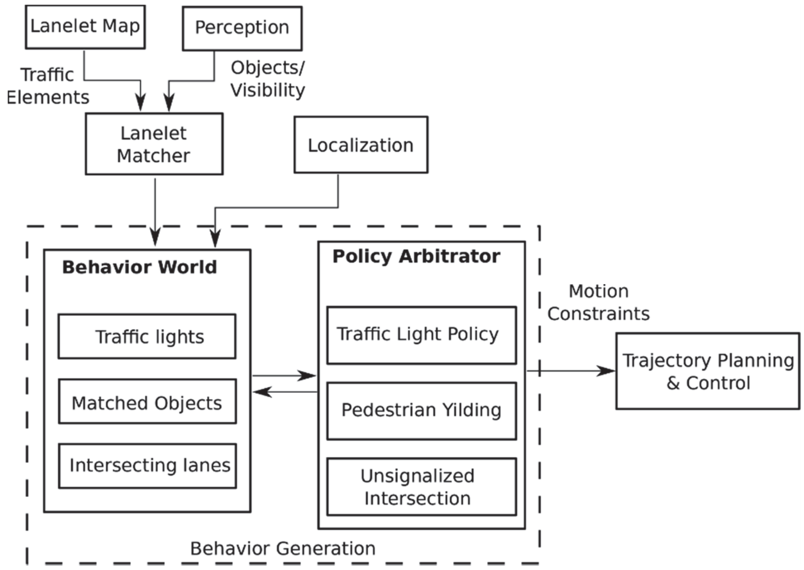

Fig. 10 Structure of the decision making module.

The module *behavior world* receives upcoming traffic elements that are existing on the ego vehicle route such as traffic lights, crosswalks, or non-signalized intersections. It also receives detected objects from the perception modules that are matched to their related road elements. According to the upcoming traffic elements, specific sub behavior policies are activated. In the following part of this section, we have a closer look to two sub behavior policies.

3.2.1.1 Pedestrian Yielding Policy

The pedestrian yielding policy requires positions and predicted paths of all pedestrians walking close to a crosswalk. If the predicted paths are headed toward the crosswalk, the policy sends a stop action to force the ego vehicle to stop at safe distance in front of the crosswalk. If any pedestrian is headed towards the crosswalk the policy sends a driving command meaning that the ego vehicle can drive and pass the crosswalk.

3.2.1.2 Intersection Yielding Policy

Fig. 11 depicts a situation that can occur at an intersection where the ego vehicle has to yield. The policy requires the distance and velocity of all other vehicles that are driving at the intersection which are represented as d_i and v_i. The perception module may be unable to detect vehicles occluded by obstacles. Therefore, the yielding policy also needs to know up to which distance d_{o_i} vehicles on each lane i could be observed. Finally, v_{o_i} denotes maximal permitted velocity for each lane.

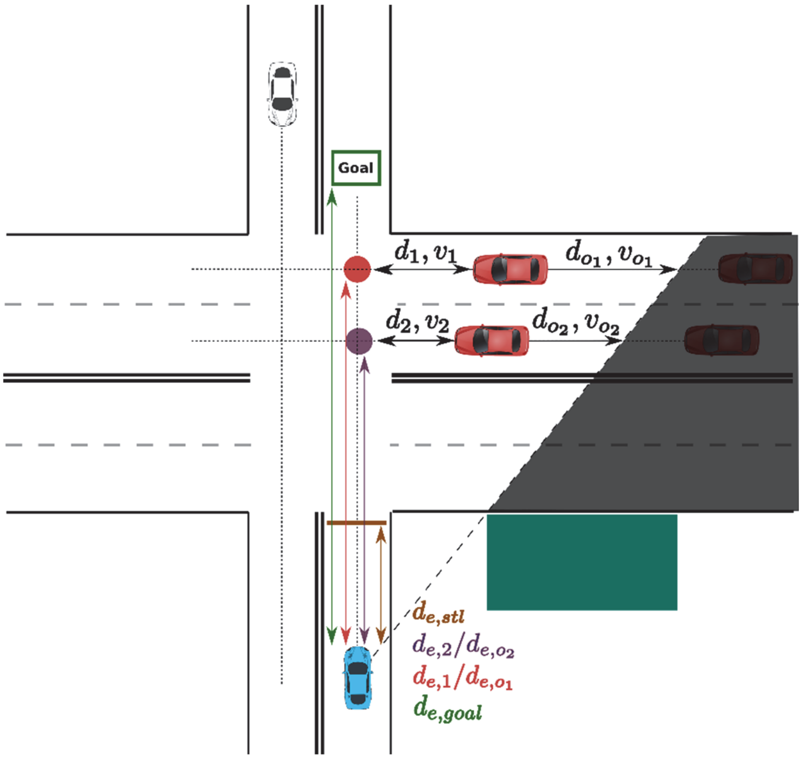

Fig. 11 Example of a non-signalized intersection. The ego vehicle at the bottom has to yield vehicles on the crossing lanes.

The output action of the yielding policy is the highest safe velocity for the ego vehicle that can be executed safely. For safety reasons we always consider whether it is possible to stop in front of the conflict area or whether it is possible to leave the intersection before other vehicles have reached the conflict area. If any of these two maneuvers is possible, we reduce the velocity.

3.2.2 Normal Driving

Planning a trajectory for normal driving is designed as an optimization problem with constrains given by the perception and decision making modules. This module plans a trajectory for the next 5 seconds that does not leave free space. It is important to note that the planner does not try to reach a goal but navigates on the free space with the references directions and speed recommendations obtained from the other modules.

The first step is to obtain a feasible global plan to reach the goal of the trip. As the navigation map is encoded using the lanelet2 library [25], this global plan can be obtained in an easy way, e.g. by searching the path with minimum distance that connects the starting point and goal.

Once the global plan has been obtained, it is necessary to extract the drivable area around the car. The global lanelet2 path is cut with a predefined number of meters in front and behind of the current position of the car.

We utilize virtual obstacles to represent the signals that could block the path, e.g. when a traffic light is red. These virtual obstacles are added together with the real obstacles obtained from perception.

Finally, the drivable area for each point in time up to the planning horizon is calculated considering the expected movement of other traffic participants. We distinguish three cases.

- The object is behind the current position: the drivable area is not modified as we do not plan to drive backwards.
- The obstacle partially occupies the drivable space partly but can be avoided inside the drivable area. Then, the drivable area is shrunk by a trapeze that models the object plus a safety margin.
- The obstacle occludes the drivable area entirely and cannot be avoided. The drivable area is completely cut at this place. All the obstacles that are farer away are neglected.

Once the free spaces along the planning horizon is extracted, an optimization problem is established and solved. The optimization problem considers the following criteria.

- Minimizing the acceleration.
- Minimizing the jerk.
- Staying inside the drivable area.
- Being close to the reference line.
- Being close to the reference speed.
- Respecting the velocity, acceleration and jerk limits of the vehicle.

All of these criteria can be weighted to achieve a more aggressive or more careful driving style. It is important to note that none of these criteria is considered a hard constraint. However he terms for the physical limits of the vehicle and staying inside the drivable area are weighted considerably higher in the cost function than the other criteria.

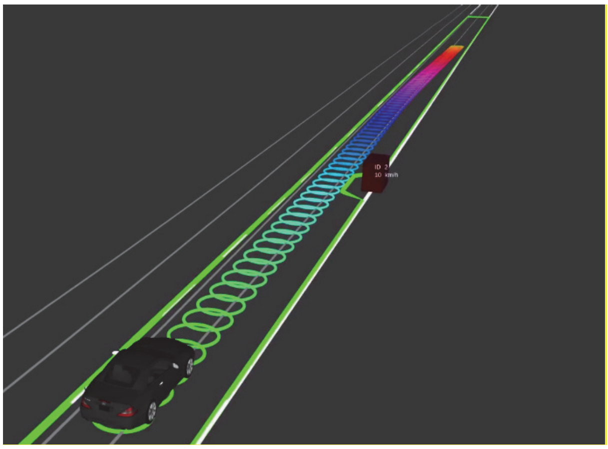

Fig. 12 Example of a planned overtaking trajectory. The trajectory is illustrated as a sequence of circles that model the path of the vehicle.

The planner is able to work at constant frequencies of 5 to 10 Hz depending of the number of points assigned to the trajectory. Fig. 12 shows an example of such a trajectory.

3.2.3 Safe Halt

A behavior planning module of a fully self-driving system should always have an alternative plan. If the self-driving vehicle can no longer proceed on a trajectory of the normal trajectory planning, it must be capable of performing a safe halt, known as a minimal risk condition or fallback. This might include situations when the self-driving system experiences a problem, when the vehicle is involved in a collision, or when environmental conditions change in a way that would affect safe driving within our operational design domain. After encountering those situations, the system should determine an appropriate response to keep the vehicle and its passengers safe, including pulling over or coming to a safe halt.

To guarantee a meaningful solution and be real-time capable, we propose to utilize path-velocity decomposition for the planner, which provides a non-globally optimal solution but reduces the computational burden.

Before computing the safe halt trajectory, a clear goal (position and orientation) is determined. The goal should not be occupied by any obstacle. Ideally, the goal should not be inside intersections, on sidewalks, or any other place where it is dangerous or illegal.

First, a large number of possible geometric curves from the start position to the goal position are sampled. The curves that leave the drivable space, that intersect with a static obstacles are removed. For obtaining comfort and safe behavior, curves that are less curvy and that keep large distance to static obstacles are preferred over other curves. Note that these calculations are done in the Frenet frame with respect to the centerline of the road. Afterwards, we can calculate which parts of the curve are occupied by other traffic participants during which time intervals.

Given the starting velocity, goal position, goal velocity (which in our case is 0 m/s), acceleration limits and velocity limits, we can check whether a drivable, collision-free trajectory on the selected curve exists using a modified A*-algorithm. An example trajectory is shown in Fig. 13.

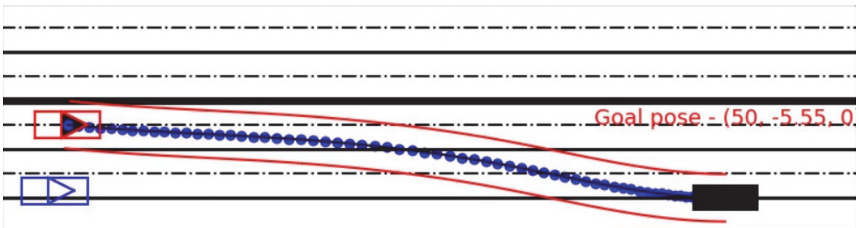

Fig. 13 Example of a safe halt trajectory. The red rectangle is the ego vehicle. The black one is the goal position. The blue rectangle is another traffic participant. The blue line with bullet points is the planned safe halt trajectory.

The automated fallback system Safe Halt [9] aims to enable the vehicle to be transferred into a risk-minimized state at all times. The fallback system is activated if the abilities of the primary driving functions do not suffice to fulfil the dynamic driving task. The emergency maneuver performed during the Safe Halt procedure relies on the pre-calculated emergency path presented above.

The emergency path is planned based on the current situation of the vehicle environment. However, since the vehicle is operating in a dynamic environment, the occupancy of the emergency path shall be monitored for a collision avoiding emergency maneuver.

The object information from the secondary perception system (the platform sensors, c.f. Sec. 3.1.3) are fused by the fallback system. With the knowledge of the course of the emergency path and the intended speed profile, future vehicle poses can be calculated. Together with the vehicle dimensions (width and height, supplemented by a safety margin), the vehicle pose is described in a spatio-temporal dimension. Based on the object states of the secondary perception systems, this dimension is monitored for collisions [26]. If a collision is predicted, the reference velocity is adjusted to avoid or at least mitigate the collision.

The trajectory generation of the fallback system generates an emergency trajectory for the trajectory controller based on the course of the emergency path, the included speed profile and the collision avoiding velocity input of the emergency path monitoring. The specifications of the emergency trajectory and the reference trajectory for automated driving are identical. It is therefore not necessary to develop a separate trajectory controller for the fallback system.

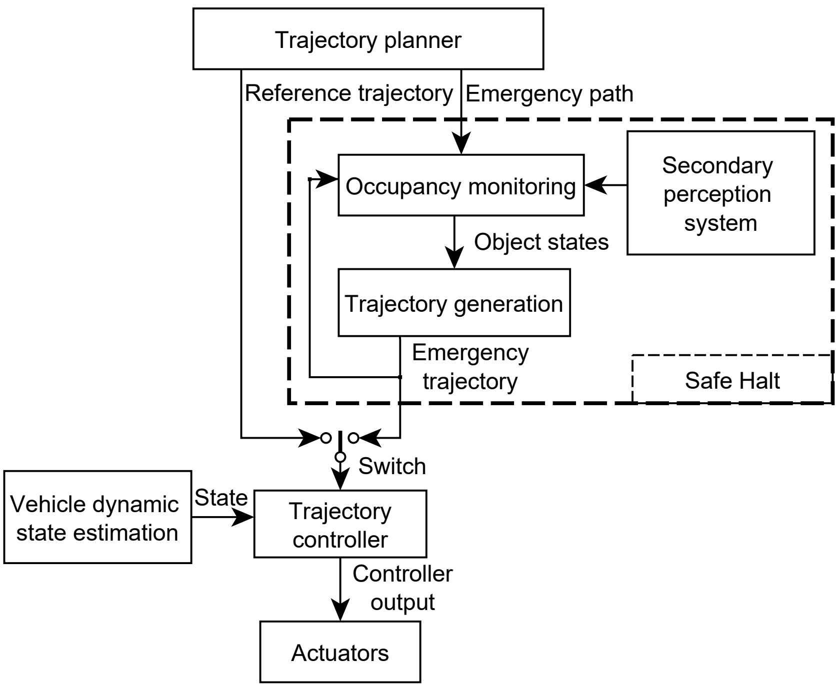

Fig. 14 Relevant interfaces and system architecture of the fallback system Safe Halt.

Fig. 14 illustrates the interfaces and system architecture of the fallback system Safe Halt. Even if the fallback system is not activated, the secondary perception system monitors the occupancy of the emergency path and generates a collision-avoiding trajectory. The modular software architecture of UNICARagil allows switching (c.f. switch in Fig. 14) between the reference trajectory and the emergency trajectory. The software and hardware of the fallback system are active even without severe degradations of the primary driving functions. When the fallback system is requested the input of the trajectory controller is altered by toggling the switch. With this architecture it is possible to switch to the fallback system with minimal latency and to monitor the abilities of the fallback system at all times.

3.3 Trajectory Control

The UNICAR*agil* vehicles possess single wheel actuators that can be controlled individually, enabling the sideslip angle of the vehicles as a fully independent degree of freedom. Combined with wheel steering angles of up to 90°, unconventional maneuvers, like sideways parking, narrow turning and bidirectional driving, are possible.

The trajectory tracking control within the brainstem of the vehicle is tasked with realizing the planned behavior and to ensure the vehicle stability by deriving setpoint values for the actuators in each wheel (drive, brake, steering). It is separated from services planning target trajectories and can therefore process trajectories from different sources, coordinated by the vehicle's orchestrator based on the current operating mode (e.g. automation, safe halt or remote control). The controller therefore does not need to know the operating mode or the source of received trajectories.

Target trajectories include a sequence of intended vehicle poses, velocities and accelerations for all three degrees of freedom of the vehicle. They are processed within a two degree of freedom control structure, consisting of an acceleration based feedforward term and a time-constant-weighted state feedback accounting for external disturbances and model uncertainties. If control deviations occur, target trajectories are not adapted until certain thresholds are exceeded.

Based on the input trajectory and the current vehicle state, estimated by vehicle dynamics state estimation, the trajectory control derives setpoint values for each actuator. No predefined distribution of wheel steering angles is needed, as all necessary information are derived from the target trajectory. The over-actuated vehicle structure is used for secondary goals like energy optimization and can also be utilized for compensation of actuator faults. While the single wheel actuators possess a conventional anti lock braking system, traction control is achieved by calculating limits for the wheel rotational velocity within the trajectory controller. These limits are then considered during electric drive control.

Due to the automated driving architecture within the project, additional support functions are needed. The target trajectory is planned based on an independent localization service (see Section 3.1.2.4), while the current vehicle state is provided by vehicle dynamics state estimation. To ensure consistent coordinate reference frames while dealing with systematic sensor errors, a pose offset correction is needed. The service compensates any systematic offsets between the two localization functions and therefore prevents unwanted control action [27]. Additionally, a service estimating the current kinematic and dynamic limits to be considered during trajectory planning is proposed. This ensures drivable trajectories, even with separated planning modules.

Apart from the trajectory control, a second controller is being provided for direct control of the vehicle by an operator during remote control. While most of the control structure is similar, the controller processes setpoint speed and curvature values as input instead of a target trajectory.

4 Off-board Support

The off-board support comprises of the so-called Info Bee as well as collective services and a control room for the vehicle fleet, which are shortly described in the following.

4.1 Info Bee

In some situations, a view from above can help to identify objects concealed from the vehicle. The UNICAR*agil* concept therefore plans to use UAVs that can be requested in order to supply additional data from another perspective. The data in question shall be gathered on demand as fast as possible. As the distance between different requested areas of interest can be significant, the used sensor platform should be able to move fast between these areas. Therefore, in the UNICAR*agil* project, tiltwing aircraft are employed.

These so-called Info Bees can fly with a cruise speed of 25 m/s while being able to take off and land vertically [28]. Due to the tiltable wing, the Info Bee resembles a fixed-wing airplane during cruise flight, thereby attaining a high efficiency. During transition to hover flight both the main wing and the elevator are tilted upwards, eventually resulting in a tricopter configuration. In contrast to ground-based sensor platforms, the Info Bees are not subject to restrictions from traffic lights, speed limits or congestions. Moreover, when encountering a blocked street a UAV can simply continue its observation route whereas a ground-based sensor platform would have to take a detour.

However, UAVs impose high limitations to the overall weight of the sensor equipment to be carried. This restricts both the sensor itself and the computing unit used to process the sensor data on board. Yet, processing of the sensor data has to be done on board at least to some extent as the amount of data produced by the sensor will be too much to be transmitted unprocessed to computing units on ground. In this project, an RGB camera is used to take pictures of the areas of interest. These pictures are then preprocessed by an algorithm based on a neuronal network on-board the aircraft. This algorithm extracts objects and determines their position in the global frame. Also, regions of the image that need further processing on ground are identified. Subsequently, these data are transmitted to the cloud from where it can be relayed to vehicles in question.

4.2 Collective Environment Model, Collective Memory, Collective Behavior

The three cloud-based software components Collective Environment Model (CEM), Collective Memory (CM) and Collective Behavior (CB) as described in [29] and [30] are capable of directly and indirectly supporting the automated driving of vehicles. A visualization of a simplified architecture showing the composition and interaction of these components can be found in Fig. 15.

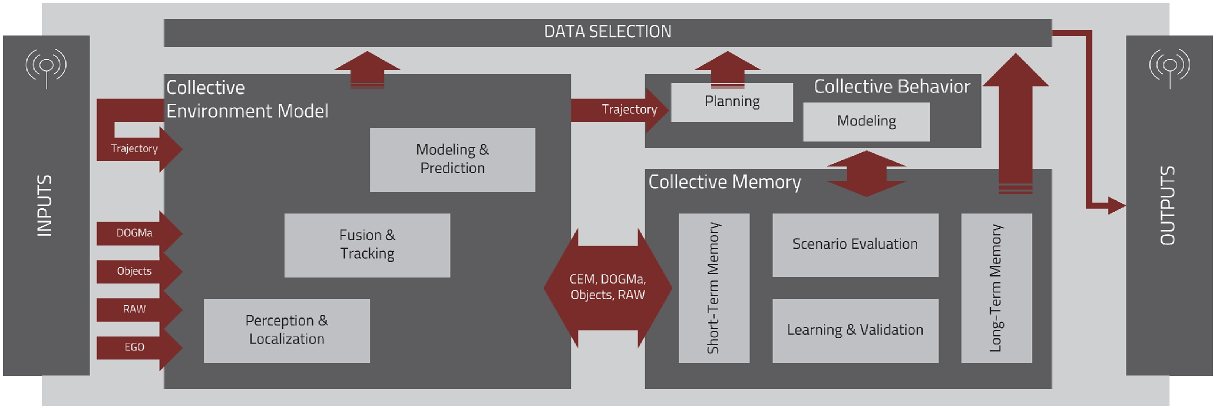

Fig. 15 Simplified architecture containing the cloud components Collective Environment Model, Collective Memory and Collective Behavior.

The CEM, CM and CB are addressing two main goals with respect to automated driving. First, an automated vehicle shall not only be capable of handling critical situations well, it should also be able to prevent them in the first place. Second, a fleet of vehicles shall be able to gather data such that algorithms performing the tasks of perception and planning can improve over time based on a smart combination of the gathered data.

The CEM primarily aims at the first goal, namely preventing critical situations. The perception of individual vehicles is extended by combining data received by multiple vehicles and potential other data sources such as the Info-Bee. Objects that are occluded for some traffic participants may not be occluded to others. Here, a combination of the perception data in the CEM's multi-object tracking module can help detect potential hazards early such that they can be incorporated into behavior planning. When partial occlusions prevent the correct detection of objects for all traffic participants, a combination of sensor data such as point clouds can allow a correct detection in the CEM. Among other challenges, transmitting sensor data to the CEM requires higher bandwidth compared to object data. The capabilities of 5G might be able to alleviate this challenge.

The CM primarily serves the second goal described above, namely collective learning using data gathered by the vehicle fleet. There are multiple tasks which can be learned. Since the UNICAR*agil* Cloud can gather large amounts of data and combine these to gain additional information, machine learning can be used extensively for learning the respective tasks.

Since cloud storage is limited as well and not all data are suited for learning, we develop a filtering mechanism which reduces the amount of data stored in the UNICAR*agil* Cloud by analyzing data for their potential to contribute to the learning mechanism. Only if ascribed with this potential, data are transferred from the so-called Short-Term Memory to the Long-Term Memory. This concept allows for scaling the methodology to larger fleets where unfeasibly much data would be available for storage.

For the task of planning, the CM gathers trajectory data and perception data to learn models for a scenario-dependent computation of optimized trajectories. The UNICAR*agil* vehicles can use these trajectories as additional input for behavior planning. Learning models for trajectory planning is possible in the CM by storing large amounts of trajectory data and analyzing them with respect to safety, efficiency and comfort. Those trajectories considered desirable can be used to train prediction models, which can in turn be used for planning as well.

The CB uses these models as well as current input provided by the CEM to generate trajectories that are suited to prevent critical situations. The extended perception provided by the CEM is particularly important to achieve this. It allows the incorporation of elements of the environment not visible to an individual vehicle in the trajectory planning.

For the task of perception, the CM can generate training data from a combination of multiple vehicle's object detection output. A mismatch between the perception output of multiple vehicles at the same location in the environment can trigger further analysis of the vehicles' data. This subsequent process provides an estimate of the correct environment representation which in turn can be used as a label for data processed by the learning module.

4.3 Control Room

The vehicles are connected to a control center, which is an essential part of the overall automated mobility concept. Control centers are known from other domains like space mission control or industrial processes [31]. Regardless of the application, control centers have three essential tasks. First, control centers operate automated systems and therefore take responsibility. Second, control centers take action during unforeseen events. And finally, control centers increase the efficiency of automated systems by reducing delays and avoiding system failures.

Within the UNICAR*agil* project, these tasks are applied to support the autonomous vehicle fleet and increase the benefit of the autonomous mobility concept. Therefore, a concept of a control center for autonomous vehicles was proposed in [32]. According to [32] the control center can be divided into three different service categories which are emergency services, fleet services and teleoperation services.

Teleoperation is often applied to operate in hazardous environments, but also to assist automated vehicles in certain situations [33]. The control center can solve certain problems without approaching the vehicle. Expected problems are technical defects in hardware or software, human interaction or the vehicle leaving its intended operational design domain. The vehicle is responsible for detecting those situations, and for coming to a save stop. Instead of stranding the passengers and causing delays, the control center can connect to the vehicle and operate the vehicle temporarily until the autonomous operation can continue. The concept of teleoperation introduces new challenges like a reduced situational awareness of the operator or additional latencies due to the communication link. Depending on the remaining vehicle capabilities different concepts of teleoperation can be applied to overcome those problems. Apart from directly controlling the vehicles actuators with a remote steering wheel, the operator can provide a path, the vehicle has to follow. This reduces latency issues by closing a delay sensitive control loop inside the vehicle. Another high level intervention for the operator would be to modify the vehicles environmental model by changing falsely classified objects. Apart from different teleoperation concepts the operator HMI needs to be considered to increase the operator's situational awareness. Solutions like head mounted displays [33] or 3D sensor fusion [34] can be applied. An example of a visual interface is shown in Fig. 16.

Fig. 16 Visual interface to increase situation awareness of the operator for teleoperation.

The fleet services are responsible for operating the vehicle fleet. It has an overview of the vehicles condition. All vehicles send information like state of charge, current route or health information of sensors and actuators to the cloud. Fleet services access this information and provide the control center with the necessary information to detect severe conditions and take the appropriate action. If necessary, it can assign routes to the vehicles or modify current routes to prevent navigation through temporarily blocked roads. In addition, it can modify the vehicle schedules or send a technician to a vehicle. The visual interface for performing these tasks is shown in Fig. 17.

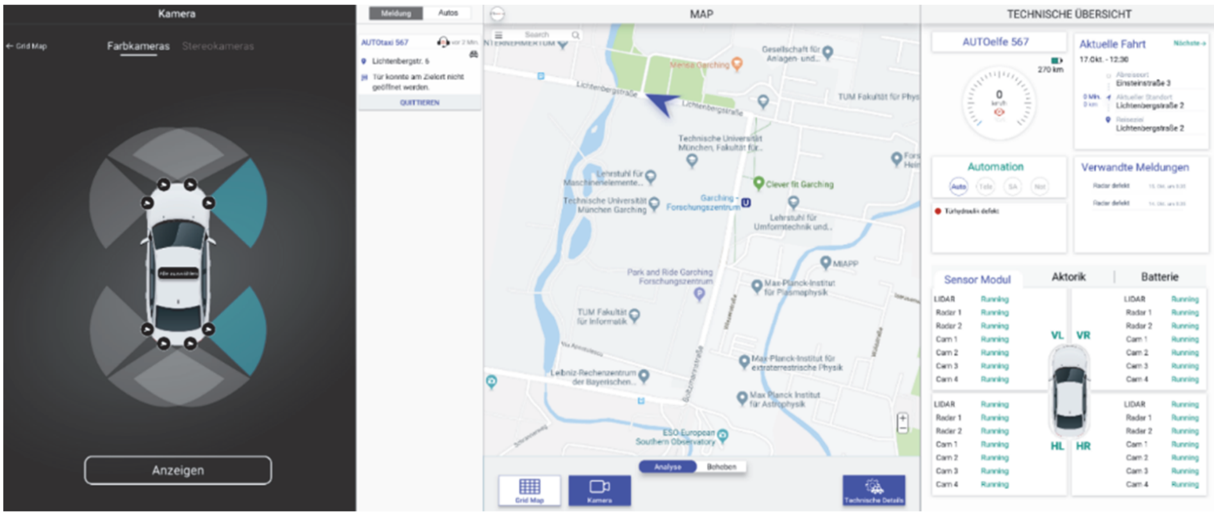

Fig. 17 Visual interface of the operator at the service desk.

The third service category is called emergency services. This service was introduced as a separate service category to ensure permanent availability. It is responsible for receiving calls from passengers or other control centers. Vehicle passengers can reach the emergency service through the vehicles interior HMI.

5 Conclusions

This paper presented the automation concept in the German research project UNICAR*agil*. Based on previous work from the partners as well as concepts and results from other research domains within the project, the automation will allow for driverless operation of the UNICAR*agil* vehicles in urban area. The concept includes measures for fail-operational (e.g. redundancy in the sensor modules) and fail-safe (e.g., Safe Halt mechanism) behavior within the vehicles and is complemented by off-board services.

The next steps include the implementation and evaluation of the presented concept within four different prototype vehicles that are currently built up by the consortium. Although each vehicle will serve a different purpose (e.g., delivery of goods or as a taxi), they will share the presented, modular automation concept, which is currently being implemented in a service-oriented manner.

6 References

[1] The UNICAR*agil* project, 2020.
Website: https://www.unicaragil.de.

[2] Woopen, Timo Lampe, Bastian, Böddeker, Torben, Eckstein, Lutz, Kampmann, Alexandru, Alrifaee, Bassam, Kowalewski, Stefan, Moormann, Dieter, Stolte, Torben, Jatzkowski, Inga, Maurer, Markus, Möstl, Mischa, Ernst, Rolf, Ackermann, Stefan, Amersbach, Christian, Leinen, Stefan, Winner, Hermann, Püllen, Dominik, Katzenbeisser, Stefan, Becker, Matthias, Stiller, Christoph, Furmans, Kai, Bengler, Klaus, Diermeyer, Frank, Lienkamp, Markus, Keilhoff, Dan, Reuss, Hans-Christian, Buchholz, Michael, Dietmayer, Klaus, Lategahn, Henning, Siepenkötter, Norbert, Elbs, Martin, v. Hinüber, Edgar, Dupuis, Marius, Hecker, Christian, 2018.
UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts.
In: 27. Aachener Kolloquium Fahrzeug- und Motorentechnik: October 8th - 10th, 2018 - Eurogress Aachen, Germany, pp. 663-694.

[3] Woopen, Timo, van Kempen, Raphael, Eckstein, Lutz, 2020.
UNICARagil - Where we are and where we are going.
In: 29th Aachen Colloquium Automobile and Engine Technology 2020, Aachen.

[4] Martens, Timm, Pouansi Majiade, Lionel Brice, Li, Minglu, Henkel, Niclas, Eckstein, Lutz, Wielgos, Sebastian, Schlupek, Martin, 2020.
UNICARagil Dynamics Module
In: 29th Aachen Colloquium Automobile and Engine Technology 2020, Aachen.

[5] Mokhtarian, Armin, Kampmann, Alexandru, Alrifaee, Bassam, Kowalewski, Stefan, 2020.
The Dynamic Service-oriented Software Architecture for the UNICARagil Project.
In: 29th Aachen Colloquium Automobile and Engine Technology 2020, Aachen.

[6] Stolte, Torben, Graubohm, Robert, Jatzkowski, Inga, Maurer, Markus, Ackermann, Stefan, Klamann, Björn, Lippert, Moritz, Winner, Hermann, 2020.
Towards Safety Concepts for Automated Vehicles by the Example of the Project UNICARagil.
In: 29th Aachen Colloquium Automobile and Engine Technology 2020, Aachen.

[7] Keilhoff, Dan, Niedballa, Dennis, Reuss, Hans-Christian, Buchholz, Michael, Gies, Fabian, Dietmayer, Klaus, Lauer, Martin, Stiller, Christoph, Ackermann, Stefan, Winner, Hermann, Kampmann, Alexandru, Alrifaee, Bassam, Kowalewski, Stefan, Klein, Fabian, Struth, Michael, Woopen, Timo, Eckstein, Lutz, 2019.
UNICARagil - New Architectures for Disruptive Vehicle Concepts.
In: 19th Stuttgart International Symposium - Automotive and Engine Technology, Documentation, Nr. 2, Bd. 19.
Springer Fachmedien Wiesbaden.

[8] SAE, 2018.
Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, Society of Automotive Engineers, Standard J3016.

[9] Ackermann, Stefan, Winner, Hermann, 2020.
Systemarchitektur und Fahrmanöver zum sicheren Anhalten modularer automatisierter Fahrzeuge.
In: 13. Workshop Fahrerassistenzsysteme und automatisiertes Fahren. Walting, 16.-17.07.2020.
https://tubiblio.ulb.tu-darmstadt.de/120961/

[10] Nolte, Marcus, Jatzkowski, Inga, Ernst, Susanne, Maurer, Markus, 2020.
Supporting Safe Decision Making Through Holistic System-Level Representations & Monitoring – A Summary and Taxonomy of Self-Representation Concepts for Automated Vehicles.
To be submitted to: IEEE Transactions on Intelligent Vehicles.

[11] Gies, Fabian, Danzer, Andreas, Dietmayer, Klaus, 2018.
Environment Perception Framework Fusing Multi-Object Tracking, Dynamic Occupancy Grid Maps and Digital Maps.
In: 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, 2018, pp. 3859-3865.
DOI: 10.1109/ITSC.2018.8569235.

[12] Nuss, Dominik, Reuter, Stephan, Thom, Markus, Yuan, Ting, Krehl, Gunther, Maile, Michael, Gern, Axel, Dietmayer, Klaus, 2018.
A random finite set approach for dynamic occupancy grid maps with real-time application.
In: The International Journal of Robotics Research, vol. 37, no. 8, July 2018, pp. 841–866.
DOI: 10.1177/0278364918775523

[13] Reuter, Stephan, Vo, Ba-Tuong, Vo, Ba-Ngu, Dietmayer, Klaus, 2014.
The labeled multi-Bernoulli filter.
In: IEEE Transactions on Signal Processing, vol. 62, no. 12, pp. 3246 – 3260.

[14] Ranft, Benjamin, Strauß, Tobias, 2014.
Modeling arbitrarily oriented slanted planes for efficient stereo vision based on block matching.
In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 2014. IEEE, pp. 1941-1947.
DOI: 10.1109/ITSC.2014.6957990

[15] Avidan, Shai, Shamir, Ariel, 2007
Seam Carving for Content-Aware Image Resizing
In: ACM Transactions on Graphics. 26, 10
DOI: 10.1145/1276377.1276390

[16] Lee, Jun-Tae, Ahn, Jae-Kyun, Kim, Chang-Su, 2013.
Stitching of heterogeneous images using depth information.
In: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, Kaohsiung, Taiwan, 2013. IEEE, pp. 1-4.
DOI: 10.1109/APSIPA.2013.6694216

[17] Cordts, Marius, Omran, Mohamed, Ramos, Sebastian, Rehfeld, Timo, Enzweiler, Markus, Benenson, Rodrigo, Franke, Uwe, Roth, Stefan, Schile, Bernt, 2016.
The Cityscapes Dataset for Semantic Urban Scene Understanding.
In: IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. IEEE, pp. 3213-3223.
DOI: 10.1109/CVPR.2016.350

[18] Salscheider, Niels Ole, 2019.
Simultaneous Object Detection and Semantic Segmentation.
ArXiv:1905.02285

[19] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian, 2016.
Identity Mappings in Deep Residual Networks.
In: European Conference on Computer Vision, Amsterdam, The Netherlands. Springer, Cham, pp. 630-645.
ISBN: 978-3-319-46493-0.
DOI: 10.1007/978-3-319-46493-0_38

[20] Geiger, Andreas, Lenz, Philip, Urtasun, Raquel, 2012.
Are we ready for autonomous driving? The KITTI vision benchmark suite.
In: IEEE Conference on Computer Vision and Pattern Recognition. Providence, RI, USA. IEEE, pp. 3354-3361.
DOI: 10.1109/CVPR.2012.6248074

[21] Danzer, Andreas, Griebel, Thomas; Bach, Martin, Dietmayer, Klaus, 2019
2D Car Detection in Radar Data with PointNets
In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC).
IEEE, pp. 61 – 66.

[22] Lang, Alex H., Vora, Alex H., Caesar, Holger, Zhou, Lubing, Yang, Jiong, Beijbom, Oscar, 2018.
PointPillars: Fast Encoders for Object Detection from Point Clouds
In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12697-12705

[23] Qi, Charles R., Su, Hao, Mo, Kaichun, Guibas, Leonidas J., 2017.
PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation.
In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
DOI: 10.1109/CVPR.2017.16

[24] Gies, Fabian, Posselt, Joachim, Buchholz, Michael, Dietmayer, Klaus, 2020.
"Extended Existence Probability Using Digital Maps for Object Verification".
In: 23rd International Conference on Information Fusion (FUSION), Virtual Conference, July 2020, IEEE.
https://arxiv.org/abs/2003.10316.

[25] Poggenhans, Fabian, Pauls, Jan-Hendrik, Janosovits, Johannes, Orf, S, Naumann, Maximilian, Kuhnt, Florian, Mayr, M., 2018.
Lanelet2: A high-definition map framework for the future of automated driving.
In: 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, November 2018. IEEE, pp. 1672–1679.
DOI: 10.1109/ITSC.208.8569929.

[26] Ackermann, Stefan, Winner, Hermann, Buchholz, Michael, 2019.
Modul und Verfahren zur Absicherung von Solltrajektorien für automatisiertes Fahren.
German patent application, official file number: 10 2019 125 401.9.

[27] Homolla, Tobias, Gottschalg, Grischa, Winner, Hermann, 2020.
Verfahren zur Korrektur von inkonsistenten Lokalisierungsdaten in modularen technischen Systemen
In: 13. Workshop Fahrerassistenzsysteme und automatisiertes Fahren. Walting, 16.-17.07.2020.
https://tubiblio.ulb.tu-darmstadt.de/120962/

[28] Schütt, Marten, 2019.
Flugzustandsregler für Kippflügel-Fluggeräte mit hohen Flugleistungen
Aachen: Dissertation
DOI: 10.18154/RWTH-2019-06783

[29] Lampe, Bastian, Woopen, Timo, Eckstein, Lutz, 2019.
Collective Driving - Cloud Services for Automated Vehicles in UNICAR*agil*.
In: 28th Aachen Colloquium Automobile and Engine Technology 2019, Aachen, Edition 1, Volume 1, Pages 677-703.
DOI: 10.18154/RWTH-2019-10061

[30] Lampe, Bastian, van Kempen, Raphael, Woopen, Timo, Kampmann, Alexandru, Alrifaee, Bassam, Eckstein, Lutz, 2020.
Reducing Uncertainty by Fusing Dynamic Occupancy Grid Maps in a Cloud-based Collective Environment Model.
Forthcoming in: 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, USA.

[31] Johannsen, Gunnar, 1993.
Mensch-Maschine-Systeme.
Springer.
ISBN 978-3-642-46785-1.

[32] Feiler, Johannes, Hoffmann, Simon, Diermeyer, Frank, 2020.
Concept of a Control Center for an Automated Vehicle Fleet
Forthcoming in: 23rd IEEE International Conference on Intelligent Transportation Systems (ITSC). 2020

[33] Georg, Jean-Michael, Feiler, Johannes, Diermeyer, Frank, 2018.
Teleoperated Driving, a Key Technology for Automated Driving? Comparison of Actual Test Drives with a Head Mounted Display and Conventional Monitors.
In: 21st International Conference on Intelligent Transportation Systems (ITSC). Maui, Hawaii, USA, 2018.
DOI: 10.1109/ITSC.2018.8569408

[34] Georg, Jean-Michael, Diermeyer, Frank, 2019.
An Adaptable and Immersive Real Time Interface for Resolving System Limitations of Automated Vehicles with Teleoperation.
In: IEEE International Conference on Systems, Man and Cybernetics (SMC). Bari, Italy, 2019.
DOI: 10.1109/SMC.2019.8914306