Environment-aware Development of Robust Vision-based Cooperative Perception Systems

Georg Volk[1], Alexander von Bernuth[1] and Oliver Bringmann[1]

Abstract— **Autonomous vehicles need a complete and robust perception of their environment to correctly understand the surrounding traffic scene and come to the right decisions. Making use of vehicle-to-vehicle (V2V) communication can improve the perception capabilities of autonomous vehicles by extending the range of their own local sensors.**

For the development of robust cooperative perception systems it is necessary to include varying environmental conditions to the scenarios used for validation. In this paper we present a new approach to investigate a cooperative perception pipeline within simulation under varying rain conditions. We demonstrate our approach on the example of a complete vision-based cooperative perception pipeline. Scenarios with a varying number of cooperative vehicles under different synthetically generated rain variations are used to show the influence of rain on local and cooperative perception.

I. INTRODUCTION

Fully autonomous vehicles aim for a safer, more efficient and more comfortable mobility. To achieve these goals those vehicles have to robustly perceive their environment. Only with a correct and complete knowledge about its environment an autonomous vehicle is able to operate correctly without harming its environment and plan safe maneuvers.

In order to get an accurate representation of the environment, a variety of sensors are installed in autonomous vehicles. Camera, radar, lidar and infrared sensors are used to complement each other and perceive relevant information of the surrounding environment. Nevertheless, vehicle-local perception has its limits. At intersections or near buildings, for example, the sensor's field of view gets obstructed. Not only obstructions but also adverse weather conditions like rain, snow and fog make perception more difficult. Rain and fog for example affect camera, lidar and radar sensors [1]–[3], but in a different extent. A report stated that snow proved to be one of the technology‘s greater challenges [4]. Adverse weather conditions do not only worsen perception of autonomous vehicles but are also responsible for 21% of current car crashes in the US, based on a ten year average from 2007 to 2016, according to the National Highway Traffic Safety Administration (NHTSA) [5]. Most weather-related accidents happen on wet pavement (70%) or during rainfall (46%), causing pavement friction and poor visibility [5]. In order to deal with these poor visual conditions it is necessary to integrate those weather conditions into the development process of autonomous vehicles.

Cooperative perception systems as presented in [6]–[9] can be used to cope with such adverse environmental conditions. These systems use V2V communication to extend the vehicle-local sensor range. Especially in critical environments such as complex intersections or during bad weather conditions these systems have benefits over local perception.

Due to the additional complexity of these cooperative perception systems, there is a need for extensive testing during the development phase. Real-world test drives are a possible solution but have one big disadvantage: it is impossible to reproduce the exact same scenario under different environmental conditions. Therefore, simulations in virtual environments are essential to complement the necessary real-world tests. Only in simulation it is possible to create reproducible scenarios. This allows a more detailed investigation of the same scenario leading to system failures under varying environmental conditions.

In this paper, we present a novel approach to simulate variations of synthetic rain in a multi-vehicle context to improve development and evaluation of robust cooperative perception algorithms. We demonstrate this approach on the example of a vision-based cooperative perception pipeline. Finally, we investigate the influence of the varying weather conditions on vehicle-local and cooperative perception.

II. RELATED WORK

Most approaches for evaluation and development of cooperative perception systems either use simplified or restricted real-world scenarios or simulated virtual environments. The used scenarios in virtual environment currently lack variation and especially realistic environmental influences like weather. Real-world scenarios have the problem that environmental influences can not be controlled or reproduced.

Real-world tests in an indoor testing facility have shown that weather conditions like fog affect camera sensors as well as lidar sensors [1], [3]. Additional tests have shown that rain affects camera, radar and lidar sensors [2]. Tests with artificially applied rain variations have also shown that the performance of camera-based object detection algorithms is strongly affected [10]. Hence, different weather conditions have to be integrated into the development process to enable the robustness of local and cooperative perception systems.

In [6] a cooperative perception system was developed and evaluated in a virtual environment. Vehicle-local perception was implemented using a probabilistic simulation of a radar sensor. However, the radar sensor simulation neglected

[1] University of Tübingen, Faculty of Science, Department of Computer Science, Embedded Systems Group, {georg.volk, alexander.von-bernuth, oliver.bringmann}@uni-tuebingen.de

*This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG) in the priority program 1835 under grant BR2321/5-1.

environmental influences. Another approach developed and compared different methods for cooperative perception [7]. The author evaluated his system both in simulation and in a real life scenario [7]. The simulative evaluation was similar to the one in [6] and also used a probabilistic sensor model without considering environmental effects on sensors. The real-world scenario in [7] consisted of two cars equipped by V2V communication on a highway. This scenario lacked variation in environmental influences, which is necessary to develop robust cooperative perception algorithms.

In [8], a multiple-object tracking system using V2V communication was developed. A different method for cooperative perception by using camera and lidar sensors was proposed in [9]. Both evaluations in [8], [9] were done on real-world scenarios. In [8], the scenario consisted of three cooperative vehicles following one leader vehicle, the authors in [9] used two cooperative vehicles. Both evaluation methods show the validity of the developed approach on a specific scenario with either two or three cooperative vehicles. Again these scenarios did not contain environmental variations.

Another approach for the evaluation of V2V applications like cooperative perception is the use of existing simulation environments. Traffic simulators like SUMO [11] are focused on traffic flow simulation in whole areas rather than providing low-level sensor information. In combination with a network simulation, for example OMNeT++ [12], they allow the evaluation of V2V applications at an abstract level. This makes it difficult to investigate the influence of different weather conditions on the perception system.

Other vehicle simulations like Virtual Test Drive (VTD) [13], CarMaker [14] or CARLA [15] focus on the simulation of vehicle surroundings and provide sensor models and access to the raw camera sensor data. However, the used methods to apply different weather conditions like rain are not validated. It is not ensured that their rain model has the same effects on sensors and algorithms like real rain. When simulating rain on a camera sensor, these simulations also lack drops on the windshield like they occur in real life.

For the evaluation of cooperative perception systems an adequate model of the V2V communication is essential. Network simulations like OMNeT++ offer a complete model of the network stack, which makes them very complex and hard to integrate in the development process. The reception probability of a packet and the corresponding delay are the key factors of the communication [16]. An approach to model the reception probability in dependence to the distance between sender and receiver was presented in [17]. However, this approach lacks the influence of the surrounding environment on the reception probability. This is especially important on intersections and other situations where the line-of-sight (LOS) between sender and receiver is obstructed by buildings. Parameterizable models considering different environments were proposed in [18], [19]. Another model which makes use of this parametrization is presented in [20]. This model consists of two components a general model and the specialized intersection model as proposed in [19]. These two components are parameterized and validated with

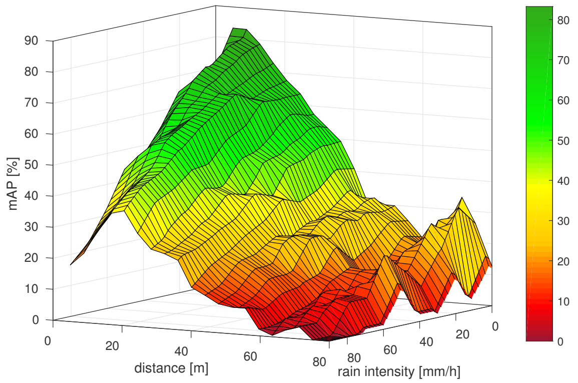


Fig. 1: Detection range of Faster-RCNN under the influence of different rain variations.

real-world measurements. To obtain channel access times for different communication ranges a simulation-based approach was presented in [17]. However, the channel access times do not represent the complete transmission time. A real-world evaluation on the communication delay of up to four vehicles and a packet frequency of $100\,\mathrm{Hz}$ was presented by the authors in [16].

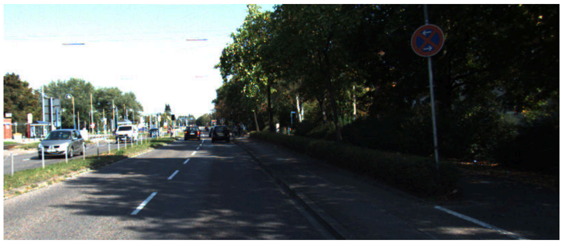
III. PRELIMINARY STUDY

Many simulation-based approaches used for development and validation of algorithms for autonomous driving such as traffic simulators like SUMO, simulators like VTD or Carmaker do not address the effects of varying weather conditions on sensor data and especially the subsequent object recognition. Also real-world tests include very little data with varying weather conditions and do not allow conclusions to be made about the influence of weather on object recognition. To show the influence of rain on image-based object detection we conducted a test with variations of synthetically generated rain.

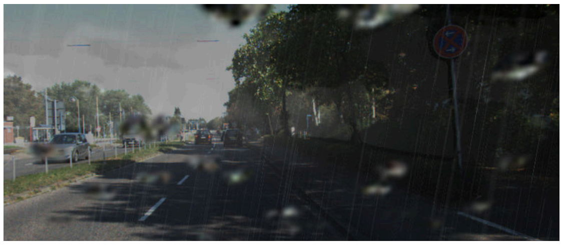
The result of the evaluation is shown in Figure 1. The mean average precision (mAP) is plotted over different object distances under varying rain conditions. It can be seen that even for close objects the probability of a correct object detection drastically decreases. Resulting in a much lower perception range under adverse weather conditions.

The evaluation was performed on Faster-RCNN presented by Ren et al. in [21] which is an artificial neural network for object detection and well established in the automotive community. For training the KITTI dataset from Geiger et al. [22] is employed. The data was split into two disjoint sets: a training set with 6800 images and a test set with the remaining 680 images. To evaluate the influence of rain variations on object detection the original KITTI images were augmented with variations of synthetically generated rain as described in section IV-B. The rain intensity was varied from 0-90 $\mathrm{mm/h}$. An example of the augmentation technique is illustrated in Figure 2.

Faster-RCNN then was evaluated on the three KITTI labels *car*, *pedestrian* and *cyclist*. To evaluate the quality of object detections the established average precision (AP)



(a) Original KITTI image.



(b) KITTI with our applied synthetic rain variant.

Fig. 2: Real-world image from KITTI compared to the same image with synthetic rain variant simulating $30\,\mathrm{mm/h}$.

metric was used. The AP metric is also used by KITTI [22] or COCO [23] to evaluate the quality of object detections. The metric is described by Everingham et al. in [24]. To count as true positive (TP) the overlap threshold between ground truth (GT) and detected bounding box was set to 70% and duplicate detections count as false positive (FP). KITTI also offers the 3D world position of a labeled object. The KITTI labels were grouped into 16 bins for evaluation. Each bin holds all labeled objects from the 680 images, which are in a specified range from the observer. The first bin contains all objects within a distance of 0-5 m and the last bin contains objects within a distance of 75-80 m.

The plot shows a lower perception probability for objects with a distance of 60 m than compared to more distant objects at 70-80 m. This is due to the nature of the input data from KITTI. In the evaluation data there are 65 labeled objects at a distance between 55-60 m compared to only 19 objects at a distance of 70-75 m. Faster-RCNN detects 8 out of the 19 more distant objects compared to 13 out of 65 for the nearer ones. However, the small number of objects at the distance of 70-75 m does not allow to make a valid statement about better perception probability of more distant objects for Faster-RCNN in general.

This evaluation shows the need to increase the robustness of detection algorithms against environmental influences. The performance degradation is partly due to the training data. Hence, one possible solution would be to include these missing conditions to the training data and retrain algorithms again to gain a more robust perception. However, we aim to use cooperative perception to increase the sensor range of the vehicle-local perception. This allows to reduce the effects of adverse weather conditions and additionally to improve perception in case of occlusion. The results also reveal the need to consider these effects in the development of cooperative perception systems to make them more robust and able to handle adverse weather conditions correctly.

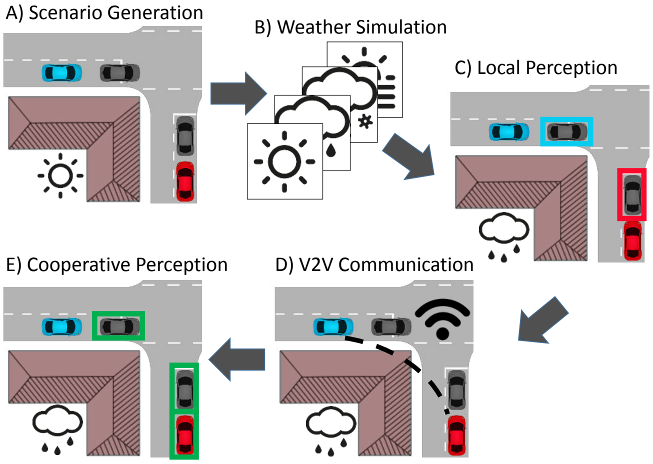


Fig. 3: Illustration of our cooperative perception pipeline. Red and blue bounding boxes represent the vehicle local perception of the ego vehicle (blue) and the cooperative one (red). Cooperative perception is illustrated in green.

IV. PROPOSED APPROACH

To enable the development of a robust cooperative perception system we define a cooperative perception pipeline which considers varying environmental conditions. The proposed pipeline consists of five steps as illustrated in Figure 3. First a scenario variant is generated specifying cooperative vehicles and the desired variations in environmental conditions. The next step is to realistically simulate these conditions followed by a vehicle local perception. The locally perceived objects have to be transmitted via V2V communication to finally perform a cooperative perception.

A. *Scenario Generation*

First, we generate a scenario variant to specify the desired variations of environmental conditions. We use the driving simulation VTD as it utilizes the open source standard OpenScenario for describing present vehicles and their behavior in the virtual environment. The simulator controls the vehicles according to their specified behavior. This simplifies the generation of scenarios as we do not have to deal with vehicle controls. OpenScenario allows the specification of a weather condition in a scenario definition. However, the present weather conditions do not offer the needed flexibility to cover the large variety of possible weather effects.

We defined a specific scenario variant $S = \{O, W, V\}$. O represents the original OpenScenario description and W specifies the desired environmental variation like e.g. rain. The parameters needed to specify a rain variant are defined in section IV-B. V is a subset of vehicles from scenario O. Each vehicle o in V contains a definition of attached sensors, their position and orientation within the vehicle and sensor specifications. The sensor specification c for a camera sensor includes field-of-view (FOV) in horizontal and vertical direction, the resolution of the camera in pixel, the far and near clipping plane in meter and the focal distance in millimeter. These sensor specifications will later be needed for our simulation of weather conditions. After a scenario variant is completely specified, the simulation will be automatically configured to attach all specified sensors in V to the corresponding vehicles defined in the scenario.

B. *Simulation of Weather Conditions – Rain Variations*

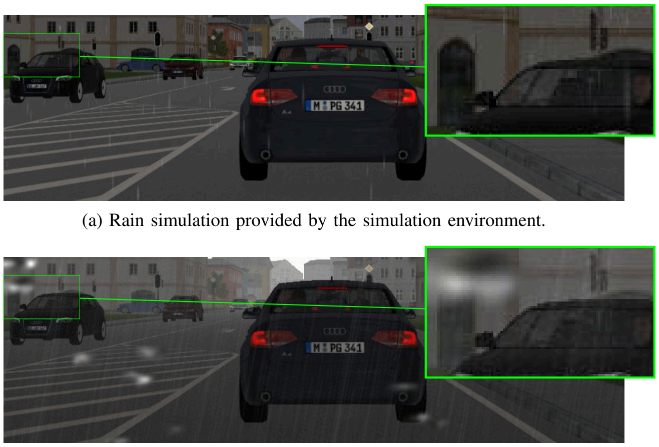
Our approach to simulate synthetically generated rain variations with realistic properties is composed of two steps. Both steps are applied on all camera sensors specified in each vehicle o. The rendered scene of the simulation environment in combination with the corresponding depth image is used as input for our augmentation approach.

First, the methodology already presented in [25] is used to simulate falling rain based on a reconstructed 3D scene using depth images. Falling rain streaks are distributed in the space between camera and background respecting physical properties and the sensor specification c. The second step is to simulate resting raindrops and water spray on the windshield. This is done by using the approach in [26]. This approach uses depth images as well to reconstruct the 3D scene. Then raindrops are distributed on a virtual windshield and ray tracing is used to render the drops physically correct onto this virtual windshield.

The two approaches in [25], [26] have been used for synthetic rain generation and variation as they are validated to generate physically correct results. The authors in [25] validated their rain model using metrics of basic image processing algorithms represented by sobel filtering, canny edge detection, harris corner detection and SURF features. These algorithms have been applied to images with real rain and with simulated rain. Afterwards the detected features on the images containing real rain were compared to the ones with simulated rain. It was shown that simulated rain produces a similar drop of correctly detected features like real rain. This proves that the used rain model not only produces realistic visual effects as illustrated in Figure 4 but also has similar affects on objects detection algorithms as real rain. Furthermore, these approaches are scalable and allow the generation of varying rain conditions in contrast to other approaches like [2], [27] which apply static rain filter masks. The combination of [25] and [26] allow variations on rain angles, different rain intensities, specification of the number and size of drops on the windshield to test perception systems on a large variety of possible rain conditions. An illustration of our synthetic rain is shown in Figure 4. A comparison to the provided rain generation of the used simulation environment shows the improvement of our approach.

C. *Vehicle-local Perception Pipeline*

The third step after augmenting camera sensor data is to perform a vehicle-local object detection. First, objects are detected on the image plane and need to be converted from 2D image coordinates to 3D world coordinates to be able to use them for cooperative perception. Finally, the perceived 3D objects are tracked with a multiple-object tracker for a more robust perception.



(a) Rain simulation provided by the simulation environment.

(b) Proposed method for generating rain variations.

Fig. 4: Comparison of VTD rain simulation with maximal supported rain intensity vs. our method for synthetic rain variant generation simulating $30\,\mathrm{mm/h}$ on an example frame of our intersection scenario.

1) Image based object detection: For image-based object detection the TensorRT [28] implementation of Faster-RCNN was used to enable real-time inference. Our approach currently focuses on applying weather conditions on image-based sensors and hence our perception is solely built on image-based object recognition.

2) Estimation of 3D object location: After 2D bounding boxes in image space are obtained, the 3D location and dimension of the detected object needs to be estimated. Therefore, the provided depth image of the simulation environment is used. Our focus is on image-based object detection. Hence, depth values are only needed to obtain realistically estimated 3D positions of the detected objects. The center of the detected image plane bounding box for object i is defined as $bb_i = (u_i, b_i)$, u_i and b_i represent 2D pixel coordinates. Starting from bb_i, pixel coordinates with the corresponding depth values are defined to belong to the detected object i as long as neighboring depth values have a distance below the defined threshold of $1\,\mathrm{m}$. The resulting point-cloud of object i consists of 3D camera coordinates containing u and b in pixel coordinates and their corresponding depth value in world coordinates. With the sensor specification c the intrinsic camera matrix K gets calculated to convert the point-cloud from 3D camera coordinates to 3D world coordinates. This lidar like point-cloud allows to estimate the rectangular shape of the detected object using the l-shape fitting algorithm presented by Zhang et al. in [29]. The position of a detected object is defined by the center of its detected shape. In comparison to [29] no clustering of the point-cloud is needed as each point-cloud i only contains points from the corresponding object i.

3) Multiple-object tracking (MOT): As last step in the vehicle-local perception pipeline a MOT for perceived objects was implemented using a constant velocity (CV) model and a Kalman filter (KF) [30]. The state space of the KF was defined as $[x, y, \dot{x}, \dot{y}]$, where x is defines the longitudinal and y the lateral distance of an object relative within the vehicle-local coordinate system. $\dot{x}$ and $\dot{y}$ are the derivatives in time representing the absolute velocity v. A new track for an object is initialized if at least three observations are present. It is deleted if no new observation within a time span of $0.5\,\mathrm{s}$ was observed since the last observation. The association of an object to a track is performed by a nearest-neighbor (NN) matching with the euclidean distance. Additionally the estimated object dimension in x (length) and y (width) gets associated to the object track. The mean absolute error of the ego vehicle location within the simulation was set to $1\,\mathrm{m}$, the error of velocity was set to $0.17\,\mathrm{m/s}$ and the error in observing the yaw angle of the ego itself was set to $0.12°$ as observed by real-world measurements in [7]. To improve the estimation of the ego vehicle state, it is tracked with a constant turn rate and acceleration (CTRA) model and an extended Kalman filter (EKF) [30]. The state space of the EKF is defined as $[x, y, v, \psi, a, \dot{\psi}]$. x and y represent the location of the ego vehicle within the global world coordinate system, v is the absolute velocity and a represents the acceleration in moving direction. ψ is the yaw angle and $\dot{\psi}$ is the yaw rate. The corresponding state transition for a time step t can be found in [7].

This perception pipeline is executed for all vehicles o in V. The observed objects are time-stamped with the global time from within the simulation environment when the image data was captured. This ensures the same time base for later processing within cooperative perception. For the most realistic representation of the perception pipeline the observed objects are delayed by the needed processing time of our perception pipeline.

D. V2V Communication Model

The next step in our cooperative perception pipeline is represented by broadcasting the locally perceived objects via V2V communication. In order to represent the communication as realistically as possible we need to define a V2V communication model to accurately estimate the reception probability and delay of communicated messages.

An important message type for V2V communication is the cooperative awareness message (CAM) as defined in the ETSI ITS-G5 standard. This message includes the position, velocity, information about the sender vehicle and the current time stamp. However, this message does not allow for accurate cooperative perception, as it only contains information about the ego vehicle and does not include its perceived objects. Therefore the cooperative perception message (CPM) is used to include the perceived dynamic objects. This message type was defined by the research project Ko-PER [31]. It contains information about the motion state of the ego vehicle as well as its perceived objects.

The V2V model presented in [20] will be used in our approach as communication model as it is parametrized and validated with real-world measurements and can be used in a large variety of different scenarios considering the current environment. The proposed model consists of a general part which fits most situations like highways, rural and suburban areas except intersections. For intersections the model of Mangel et al. [19] was used. Both models use the Nakagami probability density function (PDF) as shown in Equation 1 as basis for calculating the reception probability. $\Gamma(m)$ is the Euler's Gamma function, m the shape parameter of the Nakagami function and Ω the average power of the transmitted signal at a given distance.

TABLE I: Communication parametrization from [20].

Transmission frequency	$5.9\,\mathrm{GHz}$
Height of transmitting & receiving antenna	$1.5\,\mathrm{m}$
Receiver sensitivity P_S	$-98\,\mathrm{dB}$
Transmitting power P_{TX}	$25\,\mathrm{dB}$
System loss L_S	$3\,\mathrm{dB}$

$$PDF_{nak}(r; m, \Omega) = \frac{2m^m r^{2m-1}}{\Gamma(m)\Omega^m} exp(-\frac{mr^2}{\Omega}) \qquad (1)$$
$$r \geq 0, \Omega > 0, m \geq 0.5$$

The Nakagami-m distribution describes the amplitude r of the transmitted signal according to the shape parameter m at a given distance in the wireless channel [32]. With the receiving power $p \propto r^2$ the reception probability of a transmitted packet can be calculated by the cumulative distribution function (CDF) of the Nakagami distribution (Equation 2) as defined in [33].

$$CDF_{nak}(p; m, P_{RX}) = \frac{\gamma(m, \frac{mp}{P_{RX}})}{\Gamma(m)} \qquad (2)$$

For evaluating the reception probability of a transmitted signal, the average receiving power $P_{RX} = \Omega$ gets calculated according to Equation 3.

$$P_{RX} = P_{TX} - L_S - L_P - L_W \qquad (3)$$

P_{TX} describes the transmitting power of the sender, L_S is the system loss, L_P represents the path loss and L_W the loss for a specific weather condition. Equation 3 is an extension of [20] by additional weather-specific loss.

The path loss is calculated by the adapted two-ray ground model as presented in [20]. It is calculated depending on the present environment between sender and receiver, the area and the used communication model. The area can be urban, suburban, rural or highway. The environment between sender and receiver can be defined by LOS without or with obstructions like wood, hills, buildings or walls. The environment and area between sender and receiver is extracted from underlying OpenStreetMap representation of the currently used scenario variant S.

L_W is defined as the rain specific attenuation for the wireless signal. According to [34] the rain specific attenuation L_W (dB/km) is obtained by the rain rate R (mm/h) with the power-law relationship:

$$L_W = kR^\alpha \qquad (4)$$

The parameters k and α are determined by functions of frequency, f(GHz). With the formula and parameters provided in [34] k and α are calculated for the used communication frequency of $5.9\,\mathrm{GHz}$:

$$k = 0.00044103, \alpha = 1.5797$$

With the specified parameters in Table I the reception probability can now be calculated for the receiving power $p = P_S$ for a given scenario variant S. In Figure 5 the reception probability in a LOS environment on a highway is illustrated with and without rain specific attenuation. The attenuation by rain has little influence on the overall reception probability. However, if we aim to improve the robustness under varying rain conditions we can not neglect rain effects. Therefore, we include the rain specific attenuation in our communication model to have a more accurate and environmental aware representation of the communication channel.

The second important part of the V2V communication model is the delay of transmitted packets. A real-world investigation of the transmission latencies was conducted in [16]. The latencies ranged from $5\,\mathrm{ms}$ to $22\,\mathrm{ms}$. The investigated messages consisted of CPMs and not solely of CAMs, which is consistent with our application case. Additionally the results were obtained from real-world measurements. Hence, our communication model uses the observed latency distribution from [16].

E. Cooperative Perception

The cooperative perception is realized through a track-to-track fusion of the cooperatively perceived objects (CPOs) to the local tracks of the ego vehicle. The perceived objects contain a time stamp t_{cap} characterizing the capture time of the camera image where these objects belong to. This time stamp t_{cap} lies in the past while t_{coop} represents the current time at the cooperative perception. The difference in the time stamps is given by $\Delta t = t_{coop} - t_{cap} = d_{lp} + d_{comm}$, where d_{lp} is defined as the delay of the local perception and d_{comm} is the communication delay.

Before processing the CPOs, they have to be aligned to the time $t_{cap,ego}$. This is the time stamp where the image data of the ego vehicle was captured and represents the corresponding time to the locally perceived objects (LPOs). As local perception included MOT this can be achieved by predicting the movement of the objects to the time $t_{cap,ego}$. The prediction of the CPOs is only done to $t_{cap,ego}$ and not until t_{coop} as the tracks shall first be updated with new observations and then predicted to the current time t_{coop} to achieve a more accurate tracking result. This prediction step is done for all CPOs resulting in the same time stamp $t_{cap,ego}$ for all present objects CPOs and LPOs .

After temporal alignment, the objects have to be matched to already existing tracks. The parameters for MOT for cooperative perception are similar to the local perception from section IV-C. A new track is created if at least three observations for an object are present. The track will be deleted if no update within $0.5\,\mathrm{s}$ after the last observation was performed. To associate the objects to tracks like in IV-C a nearest neighbor matching with the euclidean distance was performed. For the objects which got associated to an already existing track an update of the KF is performed with the new observation. Additionally the object dimensions associated to the corresponding track are updated as in IV-C. For unassociated observations new tracks are initialized, but are not valid until the previously mentioned three observations are perceived. After the locally and cooperatively perceived objects are processed, all valid tracks are predicted to the current time t_{coop}.

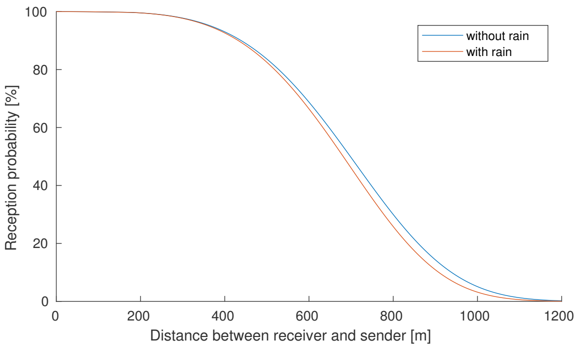


Fig. 5: Reception probability of the general V2V model on a highway. Comparison of scenario without rain and a rain intensity of $90\,\mathrm{mm/h}$.

V. RESULTS

Finally the presented approach for development and evaluation of robust cooperative perception systems is evaluated by comparing the performance of the implemented cooperative against a vehicle-local perception. We focus on showing the influence of synthetic rain variations on local as well on cooperative perception. We do not aim to prove the correctness of the cooperative perception algorithm itself. The realism of vehicle movements depends on the vehicle behavior models of the driving simulation. As introduced in section IV-A we base our approach on the driving simulation VTD. However, our approach can also be used in combination with other driving simulators. The only limitation is that the driving simulator has to utilize the OpenScenario standard to fully support the scenario generation step.

Three different scenarios with varying rain conditions are evaluated. This results in a more realistic evaluation of the perception system as compared to the evaluation for example in [6]–[9]. The evaluation presented in these approaches were either only restricted real-world scenarios with few variation tested or the sensor data was generated with a probabilistic model within a simulation. In the following the performance results of the perception system on three different scenarios will be presented. All scenarios are evaluated on the local perception (LP) of the ego vehicle compared to the ego vehicles cooperative perception (CP) using the CPMs of cooperative vehicles in its environment. For evaluation of perception performance the multiple-object tracking performance (MOTP) or the multiple-object tracking accuracy (MOTA) like presented in [35] could be used. This metric has the disadvantage that a tracked object hypothesis does not necessarily has to have an overlap with the underlying GT. The object hypothesis is a TP detection if the

TABLE II: Comparison of mean average precision of local perception (LP) against cooperative perception (CP) over different rain rates on the rural-road, intersection and highway scenario. CP20 and CP40 refer to different equipment rates.

Rain-rate	Rural road scenario		Intersection scenario			Highway scenario		
in $\mathrm{mm/h}$	LP in %	CP in %	LP in %	CP20 in %	CP40 in %	LP in %	CP20 in%	CP40 in %
0	32.17	37.83	15.86	22.30	32.98	10.63	22.69	28.27
10	23.50	39.50	15.60	21.12	31.57	7.86	20.59	26.44
20	20.83	39.83	15.44	21.21	31.80	5.49	19.81	25.70
30	12.50	38.83	15.46	20.77	31.28	4.21	19.06	25.40
40	11.50	37.17	14.99	19.74	30.46	3.21	18.22	24.73
50	11.33	38.50	14.72	18.89	29.72	1.51	17.39	24.48
60	4.00	37.83	12.99	17.35	28.56	0.58	16.61	24.36
70	3.67	35.50	11.65	14.62	25.85	0.07	16.54	24.30
80	2.00	32.50	10.25	12.55	23.58	0.00	16.35	24.00
90	1.67	31.33	8.64	11.45	22.59	0.00	16.15	23.75

distance of the objects is below a defined distance threshold. To evaluate the perception we therefore used the average precision metric as already introduced in section III. We calculate the overlap between object hypothesis and GT on 3D world coordinates with neglected height information and not on 2D pixel coordinates. As the 3D estimation of an object solely from image data represents a difficult task the required overlap threshold was set to 30%. This overlap threshold was chosen as it represents a more significant estimation of the perception performance than using a distance threshold between object hypothesis and GT as in [35]. Table II contains the results of the three different evaluated scenarios which will be discussed in the following.

A. Rural road scenario

The first scenario is a simple one containing three vehicles on a rural road in total, the ego vehicle and two preceding vehicles. For the CP scenario the rear vehicle of the two preceding ones is cooperative. The increase of CP without rain is only 5%. This is due to the fact that only two vehicles different from the ego vehicle are present within the scenario and the LP is able to detect the preceding vehicles in approximately $1/3$ of the cases. However, the robustness of the cooperative perception with increasing rain intesity can be clearly seen. With only 1.67% of correctly perceived vehicles for LP at a rain rate of $90\,\mathrm{mm/h}$, CP is able to achieve almost the same performance as LP without rain. Still, rain also affects CP and not only LP. The advantage of CP is also due to the fact the CPM includes the position of the cooperative vehicle and not only its perceived objects.

B. Urban intersection scenario

The second scenario to be evaluated is a complex intersection scenario containing the ego vehicle and 15 others. An extract of this scenario is illustrated in Figure 3. For CP two different equipment rates of 20% and 40% are evaluted, denoted as CP20 and CP40. This means that either 20% or 40% out of the 15 vehicles will be cooperative and sending CPMs. The evaluation is repeated 10 times and either 20% or 40% of the other 15 vehicles are randomly assigned as cooperative vehicles. The scenario consists of one complete passage through the intersection of the ego vehicle. Only the vehicles within a radius of 100m around the ego vehicle are evaluated. These represent the most relevant objects to have sufficient situational awareness.

The LP as well as the CP is affected by increasing rain intensities. It can be seen that the CP20 improves the perception. The improvement is not that big as the cooperative vehicles are randomly selected and some of the cooperative vehicles are driving away from the intersection and do either not perceive relevant objects or are outside the evaluation range. CP40 adds an additional 10% compared to CP20. However, the improvement of CP can be seen and especially with higher rain rates the detection rate of vehicles stays clearly above LP.

C. Highway scenario

The last scenario evaluated is a three-laned highway consisting of the ego vehicle and 20 others. The 21 vehicles are spread over a distance of 400m along the highway. The ego vehicle is approximately in the center of this 400m wide area. For the evaluation the vehicles 100m in front and 100m behind are considered. Similar to the intersection scenario the equipment rate for CP of this scenario will be set to 20% and 40%. Again the scenario is repeated 10 times and either 20% or 40% of the present 20 vehicles without the ego vehicle are randomly selected as cooperative.

This scenario shows the most significant difference between LP and CP. The LP is strongly affected by increasing rain rates. This may be caused due to higher number of vehicles as well as the greater distances between the vehicles, which make detection more difficult at higher rain rates. Both equipment rates CP20 and CP40 are much more robust against induced rain and have a less drastic decrease in perception rates. On average CP20 achieves a 15% higher detection rate than LP and CP40 performs on average 21.8% better than LP. However, the CP is also influenced by rain, although not so strongly as LP.

VI. Conclusion & Outlook

In this paper we presented a new approach to test the influence of adverse weather conditions on a whole cooperative perception chain on the example of rain variations. It has been shown that local perception as well as cooperative perception are affected by rain in different scenarios, whereas cooperative perception has clearly been more robust due to its extended perception range. The evaluation results have demonstrated the necessity to consider adverse environmental conditions in the development phase of local and cooperative perception systems.

In the future we are going to use the presented approach to investigate existing algorithms for cooperative perception regarding their environmental robustness. Furthermore this approach is going to be used to develop new robust algorithms for cooperative perception. We aim to use a more diverse set of local perception algorithms such that not all vehicles execute similar algorithms to further increase the robustness of our approach as well as the to be implemented cooperative perception algorithms. We are going to include the simulation of additional weather conditions like snow and fog to the camera sensor and further improve the rain simulation for stereo vision. In addition, more environmental aware sensor models such as radar will be added to our approach to perform a more realistic robustness investigation of perception systems as autonomous vehicles rely on multiple sensors.

References

[1] M. Bijelic, T. Gruber, and W. Ritter, "Benchmarking Image Sensors Under Adverse Weather Conditions for Autonomous Driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 1773–1779.

[2] S. Hasirlioglu and A. Riener, "A Model-Based Approach to Simulate Rain Effects on Automotive Surround Sensor Data," in *2018 21th International IEEE Conference on Intelligent Transportation Systems - (ITSC 2018)*. Maui, Hawaii, USA: IEEE, Nov. 2018, pp. 2609–2615.

[3] M. Bijelic, T. Gruber, and W. Ritter, "A Benchmark for Lidar Sensors in Fog: Is Detection Breaking Down?" in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 760–767.

[4] "Reshaping Urban Mobility with Autonomous Vehicles Lessons from the City of Boston," World Economic Forum, Tech. Rep. REF 140518, Jun. 2018.

[5] U.S. Department of Transportation - Federal Highway Administration (FHA), *How Do Weather Events Impact Roads?* [Online]. Available: https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm

[6] K. Tischler, "Informationsfusion fuer die kooperative Umfeldwahrnehmung vernetzter Fahrzeuge," Dissertation, Karlsruhe Institute of Technology, 2013.

[7] A. Rauch, "Entwicklung von Methoden fuer die fahrzeugueber-greifende Umfelderfassung," Dissertation, University Ulm, 2016.

[8] M. Obst, L. Hobert, and P. Reisdorf, "Multi-sensor data fusion for checking plausibility of V2v communications by vision-based multiple-object tracking," in *2014 IEEE Vehicular Networking Conference (VNC)*. Paderborn, Germany: IEEE, Dec. 2014, pp. 143–150.

[9] S.-W. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "Multivehicle Cooperative Driving Using Cooperative Perception: Design and Experimental Validation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 663–680, Apr. 2015.

[10] S. Mueller, D. Hospach, O. Bringmann, J. Gerlach, and W. Rosenstiel, "Robustness evaluation and improvement for vision-based advanced driver assistance systems," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sep. 2015, pp. 2659–2664.

[11] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo - simulation of urban mobility an overview," 2011.

[12] [Online]. Available: https://omnetpp.org/

[13] [Online]. Available: https://vires.com/vtd-vires-virtual-test-drive/

[14] [Online]. Available: https://ipg-automotive.com/products-services/simulation-software/carmaker/

[15] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: an open urban driving simulator," *CoRR*, vol. abs/1711.03938, 2017.

[16] A. Rauch, F. Klanner, and K. Dietmayer, "Analysis of V2x communication parameters for the development of a fusion architecture for cooperative perception systems," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. Baden-Baden, Germany: IEEE, Jun. 2011, pp. 685–690.

[17] M. Torrent-Moreno, D. Jiang, and H. Hartenstein, "Broadcast reception rates and effects of priority access in 802.11-based vehicular ad-hoc networks," in *Proceedings of the first ACM workshop on Vehicular ad hoc networks - VANET '04*. Philadelphia, PA, USA: ACM Press, 2004, p. 10.

[18] T. Abbas, K. Sjoeberg, J. Karedal, and F. Tufvesson, "A Measurement Based Shadow Fading Model for Vehicle-to-Vehicle Network Simulations," *International Journal of Antennas and Propagation*, vol. 2015, pp. 1–12, 2015.

[19] T. Mangel, O. Klemp, and H. Hartenstein, "A validated 5.9 GHz Non-Line-of-Sight path-loss and fading model for inter-vehicle communication," in *2011 11th International Conference on ITS Telecommunications*, Aug. 2011, pp. 75–80.

[20] A. Eichberger, G. Markovic, Z. Magosi, B. Rogic, C. Lex, and S. Samiee, "A Car2x sensor model for virtual development of automated driving," *International Journal of Advanced Robotic Systems*, vol. 14, no. 5, p. 172988141772562, Sep. 2017.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[22] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset." *I. J. Robotic Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[23] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Computer Vision - ECCV 2014*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 740–755.

[24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[25] D. Hospach, S. Mueller, W. Rosenstiel, and O. Bringmann, "Simulation of falling rain for robustness testing of video-based surround sensing systems," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 233–236.

[26] A. von Bernuth, G. Volk, and O. Bringmann, "Rendering physically correct raindrops on windshields for robustness verification of camera-based object recognition," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 922–927.

[27] Y. Tian, K. Pei, S. Jana, and B. Ray, "DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars," *Proceedings of the 40th International Conference on Software Engineering*, pp. 303–314, 2018.

[28] [Online]. Available: https://developer.nvidia.com/tensorrt

[29] X. Zhang, W. Xu, C. Dong, and J. M. Dolan, "Efficient L-shape fitting for vehicle detection using laser scanners," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. Los Angeles, CA, USA: IEEE, Jun. 2017, pp. 54–59.

[30] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. The MIT Press, 2005.

[31] [Online]. Available: http://ko-fas.de/files/abschluss/ko-fas_a2_3_communication_for_cooperative_perception.pdf

[32] M. Nakagami, "The m-Distribution-A General Formula of Intensity Distribution of Rapid Fading," in *Statistical Methods in Radio Wave Propagation*, W. C. Hoffman, Ed. Pergamon, Jan. 1960, pp. 3–36.

[33] A. Kuntz, "Dienstbasierte Kommunikation ueber unzuverlaessige drahtlose Verbindungen fuer selbstorganisierende Sensor-Aktor-Netze," Dissertation, Karlsruhe Institute of Technology, Karlsruhe, 2011.

[34] "Specific attenuation model for rain for use in prediction methods," International Telecommunications Union (ITU), Recommendation P.838-3, Mar. 2005. [Online]. Available: https://www.itu.int/rec/R-REC-P.838/en

[35] K. Bernardin, A. Elbs, and R. Stiefelhagen, "Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment," 2006.