Dennis Hospach
(Universität Tübingen)

Functional Safety of Sensor Systems in a Virtual Driving Environment

Functional Safety of Sensor Systems in a Virtual Driving Environment

Dennis Hospach, Stefan Müller, Oliver Bringmann

Universität Tübingen

Lehrstuhl für Eingebettete Systeme

Sand 13

72076 Tübingen

Telefon: 07071 29 – 75865

Email: dennis.hospach@uni-tuebingen.de

Abstract

Recently, optical sensors have become a standard item in modern cars, raising questions with respect to the testing under various ambient effects. With the raising automation in cars and communication between cars even more obstacles for safety have been added. Therefore, the focus needs to be on the functional safety of highly automated vehicles and the sharing of sensing information between road users. In order to achieve a high test coverage of vision-based surround sensing systems, a lot of different environmental conditions need to be tested. Unfortunately, it is too time-consuming to build test sets of all relevant environmental conditions by recording real video data. We present an approach for ambient- aware virtual prototyping and robustness testing that is capable of using real and artificial data. We aim at testing the robustness of sensor hard- and software under various environmental conditions and find the borders of safe operation. Sharing of information between vehicles can help to improve safety, but shared data needs to be validated to prevent spreading of erroneous or unverified data. We show how sharing and rating of sensor data under environmental conditions can be tested in a virtual environment.

Introduction

In recent years, testing and validating the sensor based assistance systems in vehicles has been one of the driving questions for research. Some of the questions are still lacking convincing answers but more and more the aspect of sharing the sensed data and combining them into a global image of the world becomes important.

Advanced Driver Assistance Systems (ADAS) heavily rely on sensor perception, many of them are vision- based. These systems significantly increase driving safety and comfort, but also raise new problems and challenges concerning the functional safety. This is also reflected by the road map of various automotive companies, where one of the biggest problems to address is the amount of $10^7 - 10^9$ h (Weitzel & Geyer, 2014), (Nordbusch, 2014) of on-road recordings necessary to validate a new ADAS. The limits of classical testing approaches have been reached and new solutions are needed. One possible solution is to use simulation of relevant conditions instead of comprehensive recording.

When data sharing comes in, the validity of data of a single vehicle is the basis for the validity of the common dataset. Therefore, we first discuss possibilities of robustness testing using simulation approaches of environmental conditions on real and artificial records of a single vehicle. A measure for the validity can be derived from the distance of the sensor state from its functional borders. Each road user can rate the trustworthiness of foreign data and integrate the creditable information into his world view, discarding the bad ones. An approach for simulating the sensing, testing and distribution of such information using the virtual driving environment Virtual Test Drive (VTD®) of VIRES is discussed below.

Related Work

The challenges in safety evaluation of automotive electronic using virtual prototypes are summarized in (Bannow, et al., 2014). As stated by the authors, the obstacles are the validation of early state virtual prototypes against real world conditions and to handle the complexity of stress simulation in these virtual prototypes. The contributions of (Neumann-Cosel, Roth, Lehmann, Speth, & Knoll, 2009), (Nentwig & Stamminger, 2011) and (Nentwig, Miegler, & Stamminger, 2012) use fully artificial input data from computer simulations to test and improve the underlying hardware and algorithms. Environmental influences, if part of the simulation approach, are simulated together with the rest of the scene. In (Coskun, Tuncer, Karsligil, & Guvenc, 2010) the authors generate artificial input for a lane detection assistant. The influence of sensor characteristics on ADAS in combination with the simulation of the behavior of another sensor on image data is presented in (Hospach, Mueller, Gerlach, Bringmann, & Rosenstiel, 2014). In (Mueller, Hospach, Gerlach, Bringmann, & Rosenstiel, 2015) a method to search huge parameter spaces and to evaluate the behavior of systems under these parameters is given. A workflow for efficient and systematic generation of virtual testing environments is presented in (Schuldt, Saust, Lichte, Maurer, & Scholz, 2013).

In the scope of the simulation of weather conditions in videos, only few work has been done until now. While recent work often dealt with visually convincing results for entertainment, we focus on the impact on sensors, hardware and software. In (Starik & Werman, 2003), the authors discuss how to simulate the visual appearance of rain in a video sequence without knowledge of the scene depth. They derived visual properties of rain streaks in videos and separated the rain streaks from the background to build a rain mask. Some more detailed results, including environmental lighting, are created by (Wang, Lin, & Yu, 2005). While not considering scene depth, these approaches are missing some of the effects of rain, e.g. depth-depending water vapor. A very detailed and physically convincing discussion of simulated rain, including depth information of the scene, is given by Garg and Nayar in (Garg & Nayar, 2006) and (Garg & Nayar, 2007). They extract important properties and examine their interactions with respect to a camera and present a method for rendering photo-realistic artificial rain. Other papers examine the physical properties of rain, such as drop distributions with size (Marshall & Palmer, 1948) and (Willis, 1984), terminal velocity (Gunn & Kinzer, 1949) and shape (Beard & Chuang, 1987).

The simulation of dust particles on a lens is first discussed in (Willson, Maimone, Johnson, & Scherr, 2005). The authors describe a method for the simulation of circular, monochrome dust artifacts on images, present a camera model for this use case and derive the necessary math. They also present a comparison of images of real dust particles and artificial ones to validate their models.

Sensors and the environment

There are two major approaches for the simulation of artificial ambient effects. First, the use of real images with application of artificial effects and second, the use of both artificial scenes and effects. To the author's knowledge, recent work has mainly covered the first approach, e.g. the authors in (Neumann-Cosel, Roth, Lehmann, Speth, & Knoll, 2009), (Nentwig & Stamminger, 2011) and (Nentwig, Miegler, & Stamminger, 2012) use artificially modified real data as input to the surround sensing devices. Our environmental effects can be applied to both options.

Using real data however, imposes some requirements on the recorded data. First, the source material needs to be recorded with a low noise level and at known exposure time and aperture settings. The lens should be focused to infinity as well as the aperture should be kept as small as possible, to yield a wide depth of field. Additionally, the recording environment should be close to the one that is to be simulated. The appearance of ambient effects often changes with the depth of the scene. Thus, a simulation without knowledge of the scene depth can only yield coarse results.

When using artificial image sources like VTD things are a lot easier. The parameters of the simulation are fully controllable, starting with the camera parameters and ranging to the setup of the whole scene. Even ground truth depth data can be acquired for subsequent simulation steps.

Framework for Varied Sensor Perception

For simulation of various effects our filter framework is used. It streams image data and applies the effects to them. To achieve maximum flexibility, the framework uses three types of plug-ins for preparing the input data: source, filter and sink. These can be connected to preparation chains. Sources read a recorded scenario and pass it to the first filter or the sink. Filters modify the incoming data and pass it to the next stage, which can be a filter or a sink. Subsequently, an evaluation chain can be connected to the Design Under Test (DUT) to measure and evaluate the impact of the effects on the image. Figure 1 illustrates such a filter chain. For the full details of this framework see (Mueller S. , Hospach, Gerlach, Bringmann, & Rosenstiel, 2015).

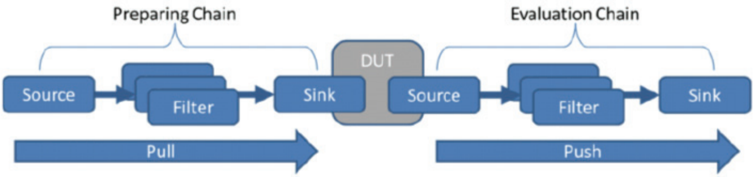

Figure 1: The environmental filtering chain framework

Simulation of environmental conditions

Camera modelling

The standard camera model used in many computer graphics applications is the so-called pinhole camera model. Its simple geometric relations allow fast and in most cases accurate projective imaging. However, some effects that occur with real cameras are not representable in this easy model. The most important and most obvious effect is the depth blur, resulting from the limited depth-of-field of a camera, which can be observed in real images. It has severe impact on visual realism for the human eye but also on the realism when simulating the imaging process of an optical sensor. Therefore, standard pinhole camera models like the one used by OpenGL are not sufficient for this simulation.

We integrate an aperture into the model (see Figure 2) and gain the depth blur effect as a natural side-effect. If the environmental effect are static over time, we can use convolution of the aperture with the effects and finally blend it with the original image. If this is not possible due to fast dynamic changes in the effect scene and the high computational cost of the convolution, we need to use another approach. Knowing the camera parameters and the depth of an object in the scene, the blur effect can be calculated as the circle of confusion c on the sensor plane as

$$c = \frac{(o - s) * f^2}{o * (s - f) * N}$$

where o is the object distance, s the focused distance, f the focal distance and N the f-number.

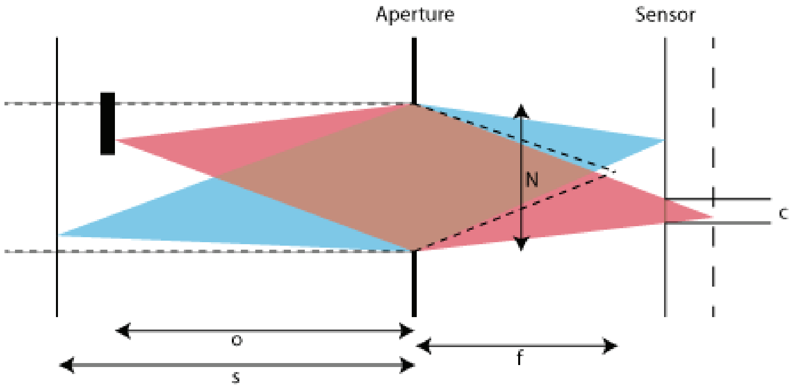

Figure 2: The camera model used within our simulations. o is the object distance, s the focused distance, f denotes the focal distance and N is the f-Number of the aperture.

Simulation of falling rain

The simulation of falling rain on images has already been published in (Hospach, Mueller, Rosenstiel, & Bringmann, 2016). So for the full details, we refer the reader to that paper and only include a brief description.

The first step of the simulation is the scene reconstruction. Using the depth information of the previously calculated depth map, the pixels of the original input image are projected into 3D space. In a second step, the scene is divided into a number of subsequent partitions with their own vertex buffer in depth direction to account for the huge amount of rain drop primitives. The rain streaks are randomly distributed in space. Following the Marshal-Palmer-Distribution which relates the amount of rain to the size distribution of the drops, the respective amounts of drops are generated. The final rendering step first blurs the rain drops using the circle of confusion described in the previous section and finally blends it with the input image.

An image showing the effect of the rain filter is depicted in Figure 3.

Figure 3: Simulated falling rain at a rain rate of 12 mm/hr, falling in an angle of zero degrees.

Simulation of dust particles on lenses

The simulation of dust on lenses has first been published in (Willson, Maimone, Johnson, & Scherr, 2005), using circular dust particles with monochrome color, placed perpendicular to the camera only. Our particle model supports several more properties: 3D-position, color (including alpha channel) and shape variance. The particles are randomly distributed on a user-defined plane that doesn't necessarily have to be perpendicular to the image plane. The shape is basically a circle with smooth deviations from it. The final shape is triangulated and the triangulation detail level can be controlled by specifying the number of edges of a particle.

The following steps are performed to calculate the final image. First, for each pixel $p_{x,y}$ on the sensor, the intersection point p_w of the light ray starting at the pixel and leaving through the center of the aperture towards the scene with the dust plane needs to be found. In case of perpendicular planes this can be calculated rather easy using similar triangles. If the plane can have arbitrary geometry, it is best calculated using ray-plane-intersection. Second, the projection of the collection cone A at the point p_w needs to be calculated. This is done by projecting the triangle edge points of the aperture onto the plane, gaining a projected polygon of the collection cone. The simulation geometry is depicted in Figure 4.

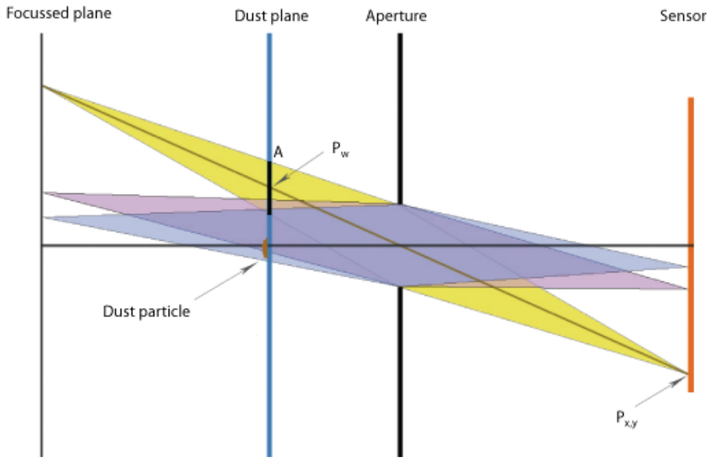

Figure 4: The geometry of the dust simulation. $p_{x,y}$ denotes the pixel at position x,y at the sensor. p_w is the intersection point of the ray starting at $p_{x,y}$ with the dust plane and A is the projection of the collection cone, visualized as black line.

To determine the influence of the dust particles, all surrounding particles close to p_w are retrieved. These particles potentially have an influence on the final pixel color. To calculate the amount of that influence, in the next step the intersection area A_j of the near particle j and A is calculated. If we assume that the particles are not mutually overlapping the sum of the intersection areas A_j are always less than the overall area A. The fraction $\propto_j = \frac{A_j}{A}$ is the alpha-blending factor of particle j and determines the amount of its color contributing to the final pixel color. This is clarified in Figure 5. By blending the input image with the dust particle mask we obtain the output. Figure 6 depicts an output image of the dust simulation in a road scenario.

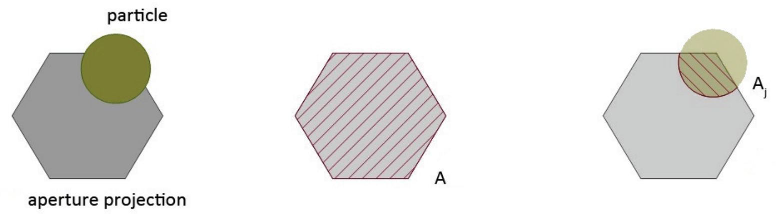

Figure 5: Calculation of the blending factor using the intersection of the aperture projection area A and the particle area A_j.

Figure 6: Dust particle simulation on a real traffic scenario image. The particle shape is varied slightly, the particle color is light yellow.

Fitness Landscapes for the Evaluation of Robustness

Fitness landscaping is a concept to evaluate the degree of adaption to a given environment of several samples, that originates from evolutionary biology and was first presented in (Wright, 1932). It has been adapted by computer science in fields of genetic algorithms and test-data generation. An overview is given by Harman and McMinn in (Harman & McMinn, 2010). Robustness metrics are a special kind of target function. They describe the distance from a given input to a given target. A similar function is also required for the creation of fitness landscapes. In this context such functions are called fitness functions. This allows to use these robustness metrics also as basis for the generation of fitness landscapes over the considered parameter space. An example is shown in Figure 7. It shows the fitness landscape of a weakly trained traffic sign recognition on different brightness values and rain densities.

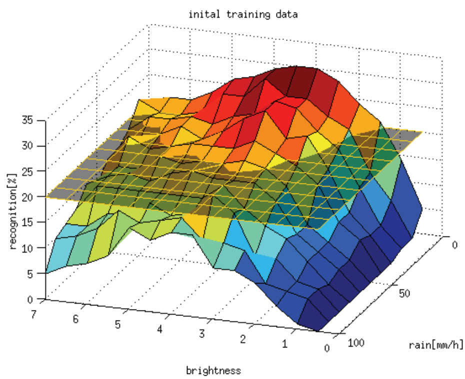

Figure 7: Fitness landscape of a traffic sign recognition with varied brightness and rain densities and a water-level hyperplane.

Robustness metrics define the minimum a system needs to achieve to be considered as robust. This minimum is described by a hyperplane inserted at this level in the robustness landscape, the so-called water-level. The water-level allows us to find the areas in the parameter space where all requirements are fulfilled. A water-level is also shown in Figure 8. For illustration purposes all examples in this paper are demonstrated on a two-dimensional parameter space. But it can also be applied to higher dimensional parameter spaces.

In this approach we use a support vector machine (SVM) to speed up this process. The SVM predicts the system boundary upon a subset of Monte Carlo distributed samples. Empirical tests show that 40- 50% of samples are enough to approximate the quality of a full grid search.

As DUT for these fitness evaluations a lane detection algorithm (Aly, 2008) is used. All four accompanying test sets were evaluated and we were able to find a design weakness of the algorithm for rain that is falling straight downwards. One of the result is shown in **Fehler! Verweisquelle konnte nicht gefunden werden.** which depicts a noteworthy drop in the recognition rate of this algorithm.

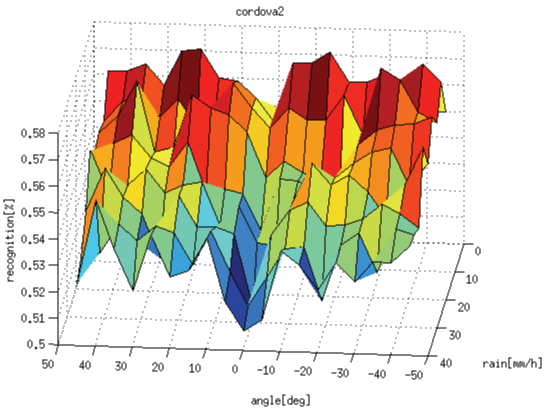

Figure 8: Fitness landscape of a lane detection with varied falling angle and rain densities.

A more detailed description and the results can be found in (Mueller S. , Hospach, Gerlach, Bringmann, & Rosenstiel, 2015).

VIRES Virtual Test Drive

VIRES VTD® is a simulation environment for test driving scenarios. It provides road and environment simulation and the ability to simulate the view of virtual sensors on the scene. The simulation software provides interfaces for third party components, enabling them to connect and communicate with the simulation core.

VTD already includes various environment simulation possibilities like wet road, rain, fog and snow. We enhance VTDs environmental effects with the ones developed by the authors, which allows us to test the behaviour of algorithms and sensors in virtual environments and also provides new paths to investigate and understand cooperative aspects in non-perfect simulation conditions.

Connection to the simulation framework

The connection between our simulation framework with the environmental filters and the VTD test bench is done via a source that connects to a shared memory segment provided by VTD. This interface streams per-frame images of camera sensors to our filters, where we can then apply environmental effects and camera modifications. Each vehicle can provide its own sensors with a unique view on the environment. Further a second connection via network is used to receive information about the objects and the course of the road in the current frame using the Runtime Data Bus (RDB).

The modified images are then processed by detection algorithms that try to detect interesting objects in the modified sensor images. Depending on the amount of modifications and the view of the sensor, this detection step will result in more or less good estimations of relevant objects with respect to the ground-truth.

The detection results are finally sent back to VTD via RDB, first giving each vehicle the chance to react on detected objects with an adequate reaction (e.g. lowering the velocity if a speed limit has been detected). In a second step, the detection results of the vehicles can be shared among each other and each vehicle can decide how to react to sensed data of foreign vehicles.

Cooperative Vehicles in VTD

In VTD, in most cases each vehicle behaves independent of the other vehicles. But VTD offers methods to distribute state information during runtime over RDB. So generally the vehicles can use the RDB to inform other road users of sensed objects and obstacles.

To investigate the sharing of sensed data between vehicles, each vehicle needs to receive and interpret sensor data from other vehicles and probably wants to have some kind of measure for the trustworthiness of the received information. If we assume to have information about the current environmental conditions and the sensor states, we may define such a measure as the distance of the sensors operating state to the boundaries of valid operation in the fitness landscape as follows:

Let $S = S_1 \times S_2 \times \ldots \times S_n$ be the cartesian product of the parameter domains of the fitness function and $F \subseteq S$ the boundary of safe operation (water level) of the fitness function. Let further $s \in S$ be the current sensor state. Then the measure of the validity of a vehicle's sensor data can be defined as

$$v := \min_{x} \|s - x\|, \forall x \in F$$

We incorporate our previous work on single vehicle's sensors and the operational states to conclude how much sensor data can be trusted. Shared sensor data can be integrated into a vehicles world model if the data are better than the vehicles own data or if it covers parts of the world that the vehicle itself cannot currently sense.

Such a scenario could be as follows: The ability to recognize objects like traffic signs can be difficult under rainy conditions. False positives or false negatives can occur. The closer a vehicle is to the detectable object, the better the recognition should be due to less disturbances between the vehicle sensors and the sensed objects. A vehicle moving in front of another along a road could help the subsequent vehicle by sending its sensed information. The second vehicle can now integrate the data in its own sensor view early in time and thus yield reaction time. Figure 9 shows such a traffic scenario. In the first picture, the subsequent white vehicle (referred to as vehicle 2) cannot sense the traffic sign. In the second picture if there is little environmental effect the car might be able to see the sign but might still be in doubt which class of sign it is. In the third image, vehicle 2 is close enough to recognize the class of sign correctly but it could already be too late to stop the overtaking maneuver. The dark vehicle running ahead (referred to as vehicle 1) could help out by sending the sensed class of the sign at an earlier time stopping vehicle 1 from initiating the overtaking maneuver.

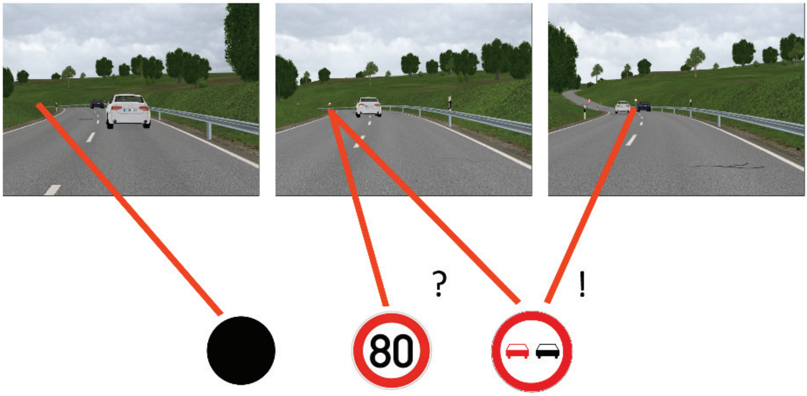

Figure 9: Cooperative traffic sign recognition scenario

In another scenario with the same scene setting, vehicle 1 could possibly misinterpret a traffic sign and share false information with other vehicles. Adding the annotation about the confidence, the second vehicle can carefully integrate the data and reassure the validity as soon as it gets close to the sensed position.
The cooperation of the vehicles could not only be limited to traffic signs. It could also incorporate lane/object detection, trajectory planning and danger warnings. All of these tasks are depending on good sensing and partially need data sharing.

Results

Respective results of our previous work have already been presented in the publications (Hospach, Mueller, Gerlach, Bringmann, & Rosenstiel, 2014), (Mueller, Hospach, Gerlach, Bringmann, & Rosenstiel, 2015), (Mueller S. , Hospach, Gerlach, Bringmann, & Rosenstiel, 2015), (Hospach, Mueller, Rosenstiel, & Bringmann, 2016) and have been briefly discussed in the chapters above. We are currently working on the evaluation of the environmental effects in cooperative scenarios and therefore refer to future publications in the scope of our currently running project RESIST.

Summary and Outlook

In this paper, we presented a novel approach to include and test environmental effects on sensor data in a virtual simulation environment to provide a feasible way for the testing of affected sensor data in cooperative traffic scenarios.
In a first step we are currently investigating the effects of environmental modifications in cooperative traffic planning and sharing of optical sensor data. In future work packages we plan to include further sensors like RADAR and LIDAR to supply a broad multisensory base for data sharing. VTD also supports these sensor types. In the field of our filter chain we need to develop modification steps for these sensor data, reflecting environmental effects on different types of sensor data than optical.
Concluding, interesting questions are: how severe are the implications due to weather effects on sensor data and how good can the sensor data quality be estimated? How can cooperation safely improve cooperative maneuvers? Is the multi-sensor approach able to help? Which environmental effects influence which sensor type more or less?

Acknowledgement

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

References

Aly, M. (2008). Real time detection of lane markers in urban streets. *2008 IEEE Intelligent Vehicles Symposium*, (S. 7-12).

Bannow, N., Becker, M., Bringmann, O., Burger, A., Chaari, M., Chakraborty, S., . . . Viehl, A. (2014). Safety Evaluation of Automotive Electronics Using Virtual Prototypes: State of the Art and Research Challenges. *DAC2014.*

Beard, K., & Chuang, C. (1987). A new model for the equilibrium shape of raindrops. *Journal of the Atmospheric sciences.*

Coskun, F., Tuncer, O., Karsligil, M. E., & Guvenc, L. (2010). Real time lane detection and tracking system evaluated in a hardware-in-the-loop simulator. *13th International IEEE Conference on Intelligent Transportation Systems.*

Garg, K., & Nayar, S. K. (2006). Photorealistic rendering of rain streaks. *ACM Transactions on Graphics, vol. 25, no. 3, p. 996.*

Garg, K., & Nayar, S. K. (2007). Vision and Rain. *International Journal of Computer Vision, vol. 75, no. 1, pp. 3–27.*

Gunn, R., & Kinzer, G. (1949). The terminal velocity of fall for water droplets in stagnant air. *Journal of Meteorology.*

Harman, M., & McMinn, P. (2010). A Theoretical and Empirical Study of Search-Based Testing: Local, Global, and Hybrid Search. *IEEE Transactions on Software Engineering*, (S. 226-247).

Hospach, D., Mueller, S., Gerlach, J., Bringmann, O., & Rosenstiel, W. (2014). Simulation and Evaluation of Sensor Characteristics in Vision Based Advanced Driver Assistance Systems. *17th International IEEE Conference on Intelligent Transportation Systems.*

Hospach, D., Mueller, S., Rosenstiel, W., & Bringmann, O. (2016). Simulation of Falling Rain for Robustness Testing of Video-Based Surround Sensing Systems. *Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE).*

Marshall, J., & Palmer, W. (1948). The distribution of raindrops with size. *Journal of Meteorology.*

Mueller, S., Hospach, D., Bringmann, O., Gerlach, J., & Rosenstiel, W. (2015). Robustness Evaluation and Improvement for Vision-based Advanced Driver Assistance Systems. *IEEE 18th International Conference on Intelligent Transportation Systems.*

Mueller, S., Hospach, D., Gerlach, J., Bringmann, O., & Rosenstiel, W. (2015). Framework for Varied Sensor Perception in Virtual Prototypes. *Methoden und Beschreibungssprachen zur*

Modellierung und Verifikation von Schaltungen und Systemen (MBMV).
Mueller, S., Hospach, D., Gerlach, J., Bringmann, O., & Rosenstiel, W. (2015). Robustness Evaluation and Improvement for Vision-based Advanced Driver Assistance Systems. *18th International IEEE Conference on Intelligent Transportation Systems.*
Nentwig, M., & Stamminger, M. (2011). Hardware-in-the-loop testing of computer vision based driver assistance systems. *Intelligent Vehicles Symposium (IV).*
Nentwig, M., Miegler, M., & Stamminger, M. (2012). Concerning the applicability of computer graphics for the evaluation of image processing algorithms. *IEEE International Conference on Vehicular Electronics and Safety.*
Neumann-Cosel, K. v., Roth, E., Lehmann, D., Speth, J., & Knoll, A. (2009). Testing of Image Processing Algorithms on Synthetic Data. *Fourth International Conference on Software Engineering Advances.*
Nordbusch, S. (2014). Vision oder Realität – Der Weg zum automatischen Fahren. *edaForum.*
Schuldt, F., Saust, F., Lichte, B., Maurer, M., & Scholz, S. (2013). Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen. *AAET2013 - Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel.*
Starik, S., & Werman, M. (2003). Simulation of rain in videos. *Texture Workshop, ICCV.*
Wang, L., Lin, Z., & Yu, X. (2005). Real-Time Rendering of Realistic Rain. *Microsoft Research, Tech. Rep.*
Weitzel, A., & Geyer, S. (2014). *Absicherungsstrategien für Fahrassistenzsysteme mit Umfeldwahrnehmung.* Bundesanstalt für Straßenwesen.
Willis, P. (1984). Functional fits to some observed drop size distributions and parameterization of rain. *Journal of the atmospheric sciences.*
Willson, R., Maimone, M., Johnson, A., & Scherr, L. (2005). An optical model for image artifacts produced by dust particles on lenses.
Wright, S. (1932). The rolles of mutation, inbreeding, crossbreeding and selection in evolution. *Proceedings of the Sixth International Congress of Genetics*, (S. 356-366).