2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

A Model-based Approach to Simulate Rain Effects on Automotive Surround Sensor Data

Sinan Hasirlioglu and Andreas Riener

Abstract— **Intelligent vehicles use surround sensors to perceive their local environment. However, already small errors in sensor data measurement and interpretation can lead to severe accidents, thus, testing the reliability of sensors and safety systems before market introduction is of high importance. The downside of real-world testing is that the increasing complexity of safety systems results in increased test effort and, as a consequence, longer development time. To counteract, simulation have to play a central role in future development and test of automated driving functions. It is common knowledge that adverse weather conditions such as rain degrade a sensor's performance due to absorption and scattering by water particles in the air. As a result, the effects of rain on sensor raw data can lead to incorrect environmental perception. This behavior is highly random but still needs to be replicated in the simulation environment to enable virtual tests with a high degree of realism. This work gives an overview on the related work, rain physics, and basic sensor theory. Further, we present a novel approach for simulating the effects of rain on data from camera, lidar, and radar sensors.**

I. Introduction

Active and integral vehicle safety systems rely on data given by surround sensors, such as camera, lidar, and radar. Using this information, safety systems can execute by-wire control actions, i.e., before an accident occurs. These systems could thus be regarded as precursors on the way to autonomous driving. In order to comply with functional safety standards such as ISO 26262 [1], car manufacturers need to perform endurance trials for millions of kilometers. With increasing system complexity, the number of test kilometers required to safeguard all vehicle functions will, according to [2], increase up to billions of kilometers. To counteract, simulations have to play a central role in the future development of automated driving functions. Virtual environments enable rapid execution of test runs in reproducible scenarios. Today, the main disadvantages are the simple surround sensor models, which output mainly error-free data under ideal weather conditions.

Hasirlioglu et al. showed that especially adverse weather conditions, such as rain or fog, can lead to degraded sensor performance [3], [4]. Therefore, it is of utmost importance to test automated driving function under adverse weather conditions. In this work, we aim to simulate the effects of rain on surround sensor data by applying novel noise filters in a post-processing step. The general process of noise filter generation and data modification is shown in Fig. 1.

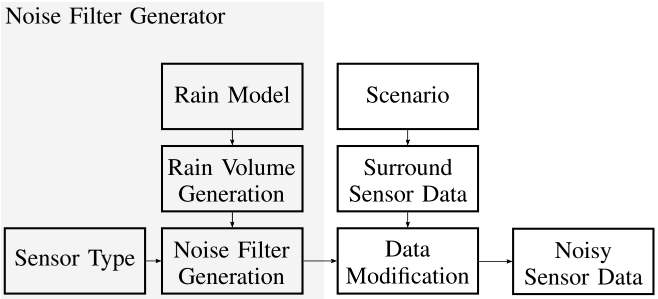

Fig. 1. General procedure for generating noise filters for surround sensor data. The effects of rain are added in a post-processing step.

Sensor specific noise filters are generated depending on the rain characteristics, which are described by the rain model. The noise filter generation is based on the ray tracing technique for optical sensors and cross sections from the Mie theory for radar sensors. Note that we neglect hardware-specific parameters for optical sensors and use empirical values for simulating general effects. Further, the filters will be applied to surround sensor data which can originate from real automotive surround sensors or surround sensor models. The data to be modified is hereinafter referred to as ground truth data. We focus on images for camera sensors, point clouds for lidar sensors, and video signals for radar sensors.

Changing weather conditions in already existing data saves test efforts due to the fact that less additional test kilometers are needed. Further, tests are independent of local weather condition, season, or time of the day and algorithms can be tested and optimized already in an early development stage.

Outline

This paper is organized as follows. Section II gives an overview on the effects of rain on each type of surround sensor data and basic work towards the modeling of such effects. Section III describes the general process for generating sensor specific noise filters, while Section IV focuses on their application. Section V presents initial test results by comparing modified data with the output of real surround sensors under rainy conditions. Finally, Section VI summarizes the contribution of this paper and discusses limitations and future work.

II. Related Work

Related work on simulating the effects of rain on camera sensors can be found in a variety of areas. However, the effects on active sensors such as radar and lidar are still relatively unexplored.

The authors are with the Center of Automotive Research on Integrated Safety Systems and Measurement Area (CARISSMA), Technische Hochschule Ingolstadt, Germany and Johannes Kepler University Linz, Austria `Email: sinan.hasirlioglu@thi.de, andreas.riener@thi.de`

978-1-7281-0323-5/18/$31.00 ©2018 IEEE

A. Camera Models

Garg and Nayar published several papers about the influence of rain on cameras. In [5], a geometric and photometric model for refraction and reflection from a single raindrop is presented. Unlike stationary raindrops, the intensities of rain streaks depend on the background scene and exposure time of the camera [6]. Further, they showed in [7], that rain strokes are only visible in the near field. When they are far away, they have the characteristics of fog. In [8], rain streaks are rendered by ray tracing images of falling drops. The approach is to render off-line and create a database of rain streaks. A complete overview can be found in [9]. Starik and Werman published a system to add artificial rain into video sequences by defining the raindrop size distribution and using a projective camera model [10]. They presented the additive adaptive model where raindrops depend on the background colors and lighting conditions. Blurring effects and darkening effects are also taken into consideration. Wang et al. presented a graphics processing unit based technique to render realistic rain in real-time [11]. Focusing on automotive systems, Gruyer et al. render images based on a two step filter mechanism, where the first one reproduces the falling raindrops and the second the drops on the camera lens or glass [12]. Hospach et al. presented in [13] a method to generate and apply synthetic falling rain to real recordings for evaluating the robustness of vision-based systems such as lane detection. They showed, that without knowing details on the scene illumination, the color of rain streaks can be set to semi-transparent white with opacity factor 0.5. Duthon et al. presented a digital rain image simulator, which is validated against data acquired in a real rain platform [14]. For specific camera settings, the digital rain simulator obtained lifelike images, which is assessed by the Harris Corner Detector.

In contrast to the before mentioned approaches, we use the ray tracing technique for generating rain images of static scenarios (snapshots where drops are spherical). The rain streaks are modeled based on axis-aligned anisotropic Gaussian filters.

B. Lidar Models

Environmental influences on lidar sensors are described in [15], where Mie's theory is used for the calculation of extinction and backscattering efficiency for single raindrops. It is shown that water drop reflections can result in false positive scan points, especially in the near field ($< 10\,\mathrm{m}$), and the sensor range decreases with increasing rain intensity. In [16], the impact of environmental water on lidar sensors was analyzed. Statistically, the reflectivity reduction due to surface wetness is approximately 10% for laser systems operating with a wavelength of 905 nm, but the effect is much less crucial than the impact of atmospheric extinction. According to noise modeling, previous papers consider Gaussian white noise, which is independently distributed at scan points [17].

In this work, we present a first model which can modify point clouds for simulating rain scenarios. The intensity values of false positive scan points are set to empirical values from real-world measurements.

C. Radar Models

Simulating rain for radar sensors has already been discussed back in the 1980's by Chandrasekar et al. [18]. The focus at that time was on meteorological radar sensors and the correlation between radar reflectivity and rain rate. Environmental influences on automotive radar sensors have been studied in [19]. In rainy scenarios, a reduction of the measured signal in the millimeter-wave range can be observed. An electromagnetic wave traveling through rain will be absorbed, depolarized, scattered, and delayed in time. Furthermore, the formation of water film on the surface of the radome causes a strong limitation of radar performance [20], [21]. Gourova et al. [22] presented experimental data and demonstrated the detectability of strong rain scenarios by a standard automotive radar sensor. Raindrops are especially visible in the near field.

We aim to modify a radar video signal by integrating raindrops into the free space area and simulate interactions such as backscattering and extinction.

III. Noise Filter Generation

In this section, we present a novel approach for generating sensor specific noise filters, which can be applied on surround sensor data (see Fig. 1).

A. Rain Model

Rain consists of water drops with varying diameters and fall speeds. Normally, a volume of $1\,\mathrm{m}^3$ of air contains about 10^3 raindrops with less large drops than smaller ones. Small raindrops are close to spherical, whereas larger drops become oblate spheroidal [23]. The diameter ranges between 0.1 and 9 mm and the fall speeds vary from 0.1 to greater than 9 m/s.

For the characterization of rain scenarios, different models have been developed in the past, based on exponential, gamma, and lognormal distributions. A detailed description is given in [24].

In this work, we focus on the three-parameter lognormal distribution presented by Feingold and Levin [25] given by

$$N(D) = \frac{N_T}{\sqrt{2\pi}\ln(\sigma)D}\exp\left(\frac{-\ln^2(D/D_g)}{2\ln^2(\sigma)}\right) \tag{1}$$

, where N_T is the total drop number per unit volume in m^3, D_g the geometric mean diameter in mm, and σ the standard geometrical deviation of D. All three parameters have physical meaning and all the moments of the distribution are themselves lognormally distributed and used to determine the parameters. Empirical relations, especially including physical parameters, simplify the parameter determination and are given by following approximations

$$\sigma = 1.43 - 3 \cdot 10^{-4} R \tag{2}$$

$$N_T = 172 R^{0.22} \tag{3}$$

$$D_g = 0.72 R^{0.23} \tag{4}$$

, where R is the rain intensity in mm/h. In a next step, the raindrops have to be distributed in the sensor's field of view (FOV) (see Fig. 2).

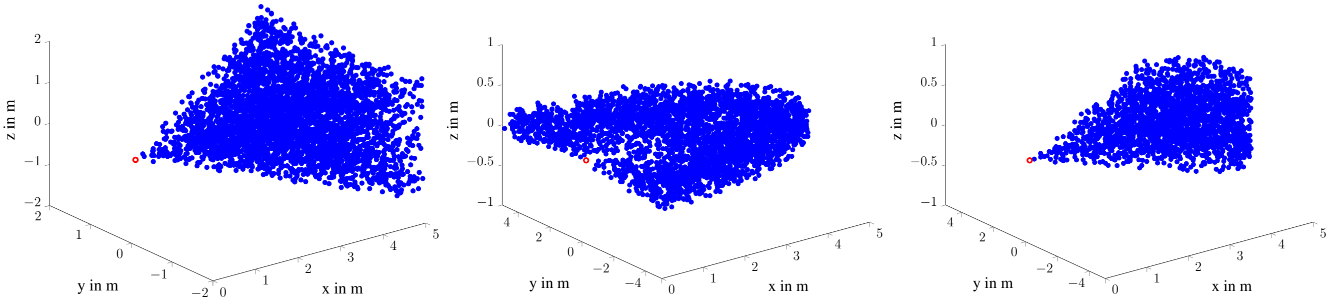

Fig. 2. FOV of each surround sensor including water drops. The left image represents the FOV of camera sensors, the center image the FOV of lidar sensors, and the image on the right the FOV of radar sensors. The number or drops is reduced by a factor of 10 for better clarity and the diameters of the drops are not considered.

B. Rain Volume Generation

A three-dimensional drop distribution is realized by assigning to each drop a normally distributed three-dimensional position, which ranges from 0 to the maximum extent of the rain volume in the respective direction. The higher the rain intensity and the larger the volume, the higher the number of drops. Note that the drop size distribution varies in nature from place to place. Therefore, single raindrops are normally distributed within the whole volume and not in unit volumes, that are repeatedly inserted. Assuming a rain intensity of $R = 100$ mm/h and inserting Eq. 2 - 4 into Eq. 1 results in a total number of 1,382 drops per cubic meter. We have used 24 diameter classes for diameters between 0 and 8.5 mm, where the size class width is varied from 0.125 to 0.5 mm.

C. Sensor Type

In this work, we present noise models for camera, lidar, and radar sensors due to the fact that each sensor outputs different data. Additionally, the sensors operate with varying wavelengths and have different FOVs. In this step, the rain volume will be limited to the sensor's FOV to reduce computation time of the filter generation. Information about the exact scene configuration can increase the level of realism due to the fact that raindrops are only present in the free space area. For camera sensors, the FOV is the area captured on the camera's imager. Lidar sensors have a wide FOV in horizontal direction and small FOV in vertical direction, which can be increased by using multiple scanning layers. For the sake of simplicity, we use only one scanning layer. In this work, the horizontal and vertical FOV of radar sensors is defined by the horizontal and vertical 3 dB beamwidth. An exemplary three-dimensional raindrop distribution can be seen in Fig. 2. The FOV is shown for all sensor types, where red circles indicate the sensor positions. Note that the diameters are not considered and the number of drops is reduced by a factor of 10 for better clarity.

D. Noise Filter Generation

This step uses information about rain volume and sensor type and generates the sensor specific noise filters, which can be applied to the ground truth surround sensor data.

1) Camera: The noise filter generation for camera sensors is based on the ray tracing technique which creates a two-dimensional representation of a three-dimensional world. The sensor position is used as the eye position and the field of view as the so-called viewing frustum. The orientation of the sensor is defined by the image plane, which is mostly 1 unit away from the sensor position. The image plane is subdivided into pixels, which define the resolution of the resulting image. Basically, rays are created for each pixel in the image plane, starting at the sensor position and passing through the middle of the pixel. If that ray intersects any object in the viewing frustum, the closest hit is calculated. Within this paper, the objects are the normally distributed spherical water drops. Intersections can be found by substituting the ray equation into the sphere equation and solving the resulting quadratic equation, where a ray is defined by its normalized origin and direction vector, and a drop by its center position and radius. If the discriminant is negative, the ray misses the raindrop, otherwise there is at least one intersection. The smaller, positive real root is the closest hit. To keep the model simple, we are focusing here on primary rays only. For an extensive introduction to ray tracing, we refer to [26].

The left image in Fig. 3 shows the result of the ray tracing, using 640 x 480 rays, as a matrix. Note that the resolution of the matrix must be equal to the resolution of the ground truth image. The bar on the right side shows the number of intersections. It can be observed, that drops closer to the sensor appear larger and a single ray can intersect multiple drops. Additionally, we store information such as closest hit and drop diameter, but neglect their effects in this paper.

2) Lidar: Noise models for lidar sensors are also based on ray tracing, but differ in the ray generation process. The direction vector of beams rotates depending on the angular resolution. Each beam in turn consists of multiple rays, which are offset to each other by a fixed angle for simulating the divergence. The offset angle $\delta_{ray,h/v}$ for rectangular beams in horizontal or vertical direction in degree is given as

$$\delta_{ray,h/v} = \frac{\Theta_{h/v}}{N_{ray,h/v} - 1} \quad (5)$$

, where $\Theta_{h/v}$ is the horizontal or vertical beam divergence in degree and $N_{ray,h/v}$ the number of horizontal or vertical rays per beam.

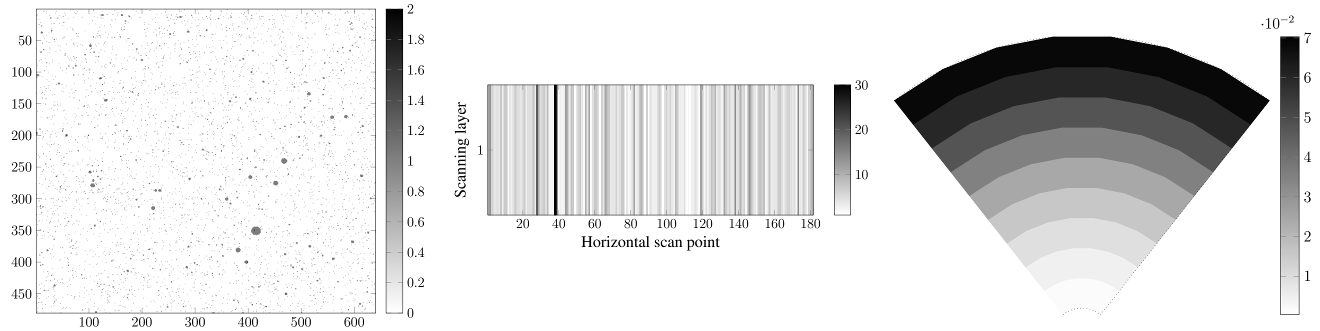

Fig. 3. Resulting noise filters for each sensor type. The left image represents the noise filter for camera sensors with a resolution of 640 x 480 pixels. Adding this mask to a camera output and applying additional adjustments leads to an image including the effects of rain. The center image shows the noise filter for lidar sensors. Each element includes information about ray-drop intersections, which models the effect of rain on every single beam of a scanning sensor. The image on the right represents the radar noise model and shows the radar cross sections of rain within range cells.

As already described in the previous subsection, it is checked for each ray, if a ray-drop intersection exists or not. All ray-drop information of a single beam are stored in a matrix element. The center image in Fig. 3 shows an example for single-layer sensors with 10x10 rays per beam. The number of rows equals the number of scanning layers which is in our example equal to 1. The number of columns depends on the number of horizontal beams. Here, we generated 180 beams horizontally. The bar on the right side shows the number of intersections. It can be seen that the number of intersections per beam varies widely, where a high number can result from drops very close to the sensor. These values are used for calculating the hit ratio of each element by dividing the number of intersecting rays by the total number of rays per beam.

3) Radar: For generating radar noise models, we assume that the transmitted waves interact with all drops in the sensor's FOV. Further, we assume that all raindrops are spherical in order to make use of the Mie theory, which can be applied if the particle size is comparable equal to the wavelength of the incident wave. The theory uses the concept of cross sections, that indicate the area interacting with the particles [27], given by the ratio of the reflected, scattered, or absorbed power to the incident power density. Radar cross section (RCS) is a measure for the ability of a target to backscatter the signals in the direction of the radar sensor. All cross sections are calculated by MiePlot [28]. A detailed description about scattering theory is given in [24].

The smallest range increment of a radar sensor is defined by the range cell and depends on the bandwidth. Our approach is to modify the reflection and attenuation properties of each free space range cell due to the presence of raindrops. The RCS of a range cell σ_{cell} is the sum of all RCS values of raindrops $\sigma_{raindrop}$ within the range cell and given as

$$\sigma_{cell} = \sum_{i=1}^{n_{Cell}} \sigma_{raindrop,i} \tag{6}$$

, where n_{Cell} is the total number of raindrops per range cell. The result can be seen in the rightmost image in Fig. 3. The volume of range cells increases with distance and therefore the RCS values increase.

Apart from backscattering properties, each range cell leads to power loss due to scattering and absorption, called extinction [29]. Absorption is described by the imaginary part of the complex refractive index, given by Segelstein [30]. Hence, the specific attenuation α in dB over a length L can be described by the extinction coefficient γ_{ext} as [24]

$$\alpha = 4.434\, L\, \gamma_{ext}. \tag{7}$$

The extinction coefficient is expressed as $\gamma_{ext} = NC_{ext}$, where N is the number of drops per unit volume and C_{ext} the extinction cross section [29]. Due to the fact that rain consists of drops with different sizes, the extinction coefficient must be determined by integrating over all drop sizes. It results the specific attenuation per range cell as

$$\alpha_{cell} = 4.343\, L_{cell} \int_0^\infty N(D) C_{ext} dD. \tag{8}$$

, where L_{cell} is the length of a range cells.

IV. Data Modification

In this section, we apply the presented noise filters on already existing surround sensor data which can originate either from real surround sensors or surround sensor models.

1) Camera: Camera noise filters are matrices with the dimensions equal to the ground truth image. Therefore, each element of the noise filter includes information, whether or not the corresponding pixel is modified. If so, intersection information such as drop distance and diameter are available. The opacity factor can be set depending on scenario and sensor configuration. For the sake of simplicity, we set the raindrop color empirically to a semi-transparent white [13]. Rain streaks depend on parameters such as terminal velocity of raindrops and shutter speed of cameras. Here, we simulate streaks by applying axis-aligned anisotropic Gaussian filters on the noise model with an increased standard deviation in the vertical direction. As proposed by [10], darkening effects can be integrated to the modification process by moving the intensity value of each channel towards the mean. Finally, the resulting image can be blurred using a 2-D Gaussian smoothing kernel. This procedure modifies fundamental image properties and can be used for initial and generic tests of vision-based algorithms.

2) Lidar: The lidar noise models are matrices with the dimensions equal to the number of horizontal and vertical scan points of a single scan. Each element includes information about the interaction between the laser beam and raindrops, represented by the hit ratio. Hasirlioglu et al. [3] showed that raindrop reflections can result in scan points, but not every single raindrop leads to a false positive scan point. Therefore, we introduce a threshold TH_{lidar} in $\%$ depending on the hit ratio, which determines whether a scan point of the ground truth scan is modified or not. The higher the threshold value, the lower the number of false positive scan points. This parameter enables to configure the sensitivity of the model. As an example, a value of $TH_{lidar} = 10$ means that a hit ratio of minimum $10\,\%$ per beam is needed for modifications. When this threshold is exceeded, we use the closest hit point of the matrix element as the new distance of the scan point with same angle to the sensor. The closest hit causes the highest intensity, if we assume that all drops are of same size and irradiated across the whole area. However, different assumptions can be made at this position, including information about drop distance and diameter. Finally, the intensity values of the scan points need to be modified. Random numbers within the lowest $1\,\%$ of the maximum intensity value show, based on our measurements, a realistic behavior. According to [16], the intensity values of remaining scan points are reduced to $90\,\%$ of their initial values.

3) Radar: The radar noise models are matrices with the dimensions equal to the number of longitudinal and lateral range cells which include information about RCS and attenuation. Basic principle is increasing the near field reflections and decreasing the far field reflections. Therefore, the presence of raindrops limits the sensor range. First, the received power from the ground truth scenario is simulated with and without rain. The difference between these logarithmic signals are gain and attenuation factors in dB and can be directly added to the ground truth video signal. The reflected power P_R for each range cell can be calculated by the radar equation, which is given as

$$P_R = \frac{P_T \cdot G^2 \cdot \lambda^2}{(4\pi)^3 \cdot R^4} \cdot \sigma \tag{9}$$

, where P_T is the transmitted power, G the antenna gain, λ the wavelength, R the range, and σ the radar cross section [31]. For clear conditions, we enter the target object RCS, which is assumed to be 10 dBsm [32], into Eq. 9 and assume a constant noise for the free space which is 30 dB lower than the maximum target peak. This value is extracted from the ground truth signal. For rainy conditions, we enter the RCS of each range cell (see Eq. 6) into Eq. 9. Finally, the received power of each range cell is attenuated by twice the attenuation (see Eq. 8), due to the transmit and return path. As a consequence, the target object reflection decreases. The difference between the two signals in dB can be applied to the ground truth video signal. Given sensor parameters can be used to adapt the model to the real measurement data, such as range cell size. Otherwise, interpolation must be performed to enable the modification process.

V. Test Results

To evaluate the noise modeling, we defined a simple static scenario. A standardized Euro NCAP Vehicle Target (EVT) [33] is placed at a distance of 10 m from the sensors without a lateral offset. This imitates an urban car-following scenario. Measurements are performed in a test facility capable of simulating rainy conditions with various intensities and drop size distributions over a length of 50 m. The rain is produced by the use of different full cone nozzle combinations. A high rain intensity of 100 mm/h was selected for the real and virtual scenario in order to clearly show the effects.

The sensor setup is placed in front of the rain simulator due to the fact that some sensors are not waterproof. The left images in Fig. 4 - 6 are recorded by an uEye camera (U3-3250LE-C-HQ) with a resolution of 1.92 MP and a 6 mm focal length lens. We cropped the region of interest to the size of 640x480 pixels. The center images show the outputs of a Velodyne VLP-16, where the red circles represent scan points from one single layer. The images on the right show the video signals of a radar sensor (Inras RDL-77G-TX2RX16). We set the sweep bandwidth to 300 MHz and the ramp slope to 10 MHz/µs, which results in a range resolution of 0.5 m.

Fig. 4 shows the low-level sensor data under clear conditions, where Fig. 5 shows the sensor data under rainy conditions. As already mentioned, it is clearly visible that rain leads to contrast loss in camera images and false positive scan points in lidar point clouds. Additionally, the radar video signal changes, where reflections from the near field are increased and decrease with distance. The target peak at a distance of 10 m decreases by 1 dB, but is still clearly visible.

Fig. 6 shows the modified sensor data by using the noise models presented in this paper. Rain streaks are simulated by applying the axis-aligned Gaussian filter with the parameters $\sigma_x = 0.1$ and $\sigma_y = 4$ to the noise filter. The raindrop color in the camera image is set to a semi-transparent white with the opacity factor 0.15. Darkening effects are neglected due to indoor measurements. Finally, the resulting image is blurred with a 2-D Gaussian smoothing kernel with standard deviation of 0.8. For more realistic modeling, these parameters can be adjusted depending on the camera settings. The point cloud of the lidar sensor is modified using a threshold value of $TH_{lidar} = 10$. Depending on the number of object scan points, we extract only intersection information from respective matrix elements. The positions of false positive scan points are chosen as the closest hit with a random intensity value within the lowest $1\,\%$ of the possible intensity values. The Velodyne sensor outputs reflectivity data with a 256-bit resolution. According to [16], the intensity values of remaining scan points can be decreased by $10\,\%$, but are not analyzed further. The video signal of the radar sensor is modified by adding the noise model data to the measured radar signal. The specific attenuation leads to changes in the target reflection. It can be seen that the signal response is similar to the video signal under physical rain. However, the simulated effects are stronger than in reality.

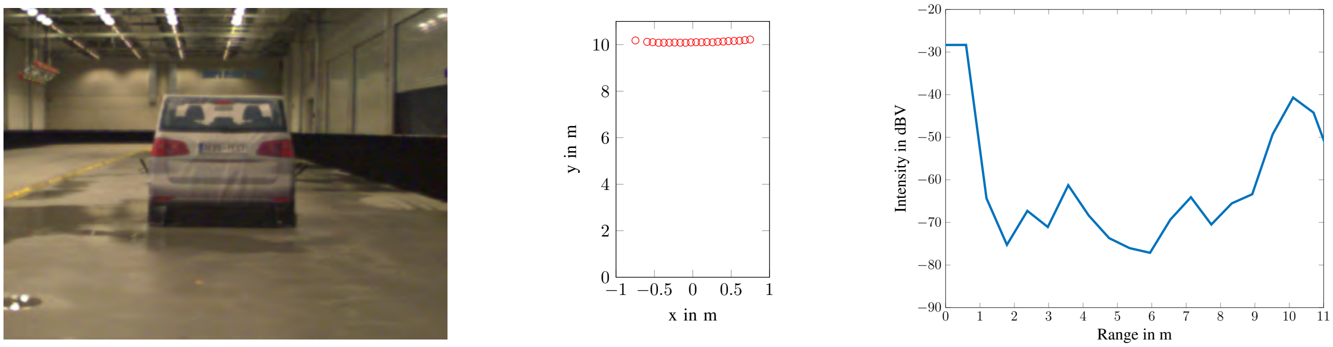

Fig. 4. Low-level sensor data under clear weather conditions. A vehicle target is placed at a distance of 10 m. The left image represents the output of a camera sensor, the center image the scan points of a lidar sensor, and the image on the right the video signal of a radar sensor.

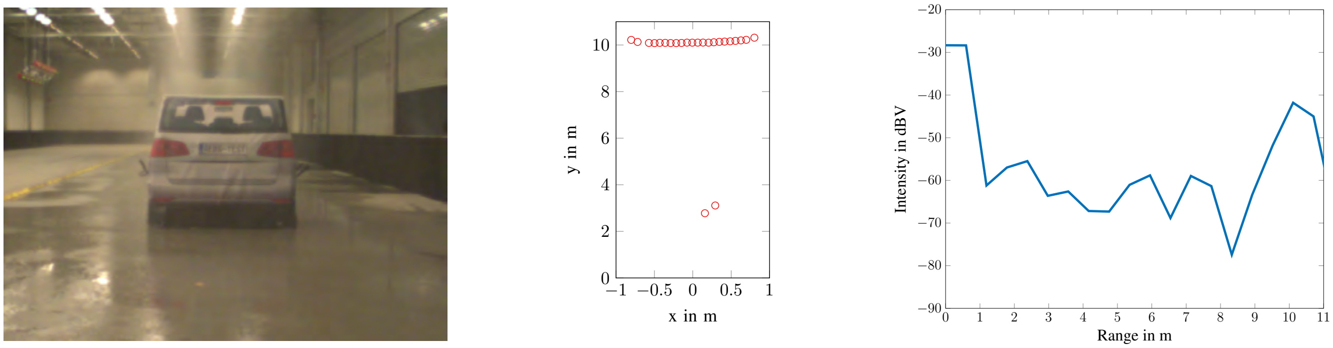

Fig. 5. Low-level sensor data under rainy weather conditions with an intensity of 100 mm/h, replicated in a test facility. A vehicle target is placed at a distance of 10 m. The left image represents the output of a camera sensor, the center image the scan points of a lidar sensor, and the image on the right the video signal of a radar sensor.

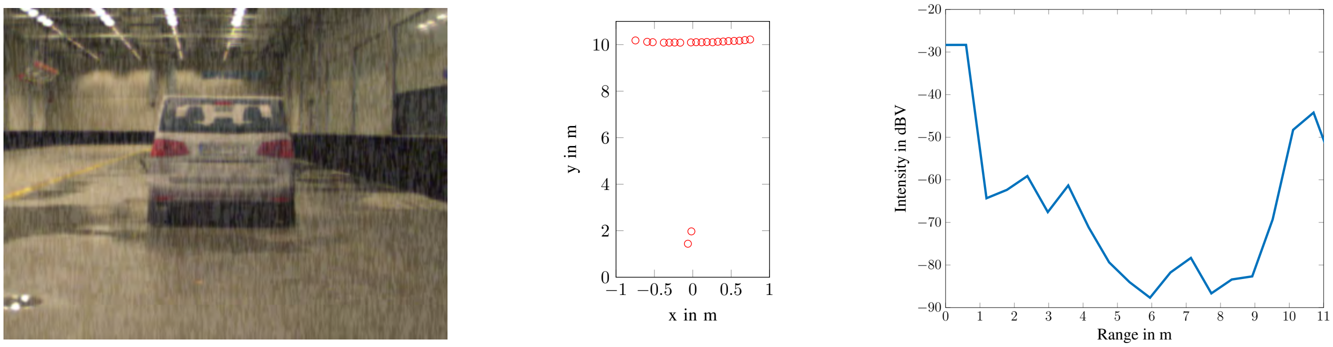

Fig. 6. Modified low-level sensor data using the noise models presented in this work. Rain effects are added to already existing sensor data in a post-processing step. The left image represents the modified output of a camera sensor, the center image the scan points of a lidar sensor, and the image on the right the video signal of a radar sensor.

A comparison of Fig. 5 and Fig. 6 shows initial results of adding rain effects into surround sensor data. The structural similarity (SSIM) index [34] is a method for measuring the similarity between two images represented by values between -1 and 1. The simulated and real rain image result in a score of 0.83. Note that the real-world data strongly depends on sensor and scene configuration, for example, the rain simulator is limited in lateral direction and simulates foggy rain at the current development stage. Further, the modified lidar scan points show a lifelike behavior regarding the number of false positives. It can be seen that the scan points modified by the model are closer to the sensor than in real world, due to the selection of the closest hit point as the new scan point distance. Raindrop reflections in the modified radar video signal cause increased signal peaks in near field which can also be seen in the data under real rain. The correlation between simulated and real rain data is stronger (Spearman correlation = 0.58) than between clear and rainy data (Spearman correlation = 0.44). Therefore, the presented models contribute towards the simulation of adverse weather conditions in surround sensor data. Initial tests can be performed for camera-, lidar-, and radar-based object detection algorithms in an early development stage. Our results should help car manufacturers to transfer large portions of test effort into virtual environment and thus save time and costs.

VI. Conclusion

Active and integral safety functions use surround sensors to perceive their local environment. It is known that the sensor signals suffer from attenuation while propagating through the atmosphere, especially under adverse weather conditions. To enable virtual testing and development, the effects of weather, illustrated here using the example of rain, have to be modeled for surround sensors. This paper presents a model-based approach for integrating the effects of rain into low-level sensor data by applying novel noise models. Initial tests show that effects such as noisy images, false positives in point clouds, and changed reflections in video signals can be simulated.

The major disadvantage of this methodology is the computation effort required to generate noise models. Real time behavior, at least on standard PCs, is currently not achievable. Nevertheless, pre-calculated noise models stored in a database can shift the effort into the preparation phase.

Future work includes the acceleration of the generation process of noise filters using different methods, such as Kd-tree, Bounding Volume Hierarchy, or a machine learning approach. Furthermore, empirical values, especially for optical sensors, will be described by sensor specific parameters to increase the adaptability and the level of realism of the noise models. Additionally, the method for validating the sensor data modification in low- and object-level must be developed further. The indoor rain simulator will be adapted regarding its drop size distribution to consider also large drops. Rainy weather conditions also lead to secondary effects such as water covered sensor, target, and road, which cause additional attenuation and impact.

Acknowledgment

We applied the SDC approach for the sequence of authors. This work is supported under the FH-Impuls program of the German Federal Ministry of Education and Research (BMBF) under Grant No. 13FH7I01IA.

References

[1] H. L. Ross, *Functional Safety for Road Vehicles: New Challenges and Solutions for E-mobility and Automated Driving*. Springer International Publishing, 2016.

[2] D. Watzenig and M. Horn, Eds., *Automated Driving: Safer and More Efficient Future Driving*. Springer International Publishing, 2017.

[3] S. Hasirlioglu, A. Kamann, I. Doric, and T. Brandmeier, "Test methodology for rain influence on automotive surround sensors," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 2242–2247.

[4] S. Hasirlioglu, I. Doric, A. Kamann, and A. Riener, "Reproducible fog simulation for testing automotive surround sensors," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*. IEEE, 2017, pp. 1–7.

[5] K. Garg and S. K. Nayar, "Photometric model of a rain drop," *CMU Technical Report*, 2003.

[6] ——, "Detection and removal of rain from videos," *Computer Vision and Pattern Recognition*, 2004.

[7] ——, "When does a camera see rain?" in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. IEEE, 2005, pp. 1067–1074 Vol. 2.

[8] ——, "Photorealistic rendering of rain streaks," *ACM Transactions on Graphics*, vol. 25, no. 3, p. 996, 2006.

[9] ——, "Vision and rain," *International Journal of Computer Vision*, vol. 75, no. 1, pp. 3–27, 2007.

[10] S. Starik and M. Werman, "Simulation of rain in videos," in *Texture Workshop, ICCV*, vol. 2, 2003, pp. 406–409.

[11] L. Wang, Z. Lin, T. Fang, X. Yang, X. Yu, and S. B. Kang, "Real-time rendering of realistic rain," in *ACM SIGGRAPH 2006 Sketches on - SIGGRAPH '06*, J. Finnegan and H. Pfister, Eds. New York, New York, USA: ACM Press, 2006, p. 156.

[12] D. Gruyer, M. Grapinet, and P. de Souza, "Modeling and validation of a new generic virtual optical sensor for adas prototyping," in *2012 IEEE Intelligent Vehicles Symposium (IV)*, 2012, pp. 969–974.

[13] 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)*, 2016, pp. 233–236.

[14] P. Duthon, F. Bernardin, F. Chausse, and M. Colomb, "Methodology used to evaluate computer vision algorithms in adverse weather conditions," *Transportation Research Procedia*, vol. 14, pp. 2178–2187, 2016.

[15] R. H. Rasshofer, M. Spies, and H. Spies, "Influences of weather phenomena on automotive laser radar systems," *Advances in Radio Science*, vol. 9, pp. 49–60, 2011.

[16] J. Wojtanowski, M. Zygmunt, M. Kaszczuk, Z. Mierczyk, and M. Muzal, "Comparison of 905 nm and 1550 nm semiconductor laser ragefinders' performance deterioration due to adverse environmental conditions," 2014.

[17] Xianfang Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, "Noise in 3d laser range scanner data," in *2008 IEEE International Conference on Shape Modeling and Applications*, 2008, pp. 37–45.

[18] V. Chandrasekar and V. N. Bringi, "Simulation of radar reflectivity and surface measurements of rainfall," *Journal of Atmospheric and Oceanic Technology*, vol. 4, no. 3, pp. 464–478, 1987.

[19] A. A. Hassen, *Indicators for the signal degradation and optimization of automotive radar sensors under adverse weather conditions: Zugl.: Darmstadt, Techn. Univ., Diss., 2006*, ser. Berichte aus der Hochfrequenztechnik. Aachen: Shaker, 2007.

[20] A. Arage, W. M. Steffens, G. Kuehnle, and R. Jakoby, "Effects of water and ice layer on automotive radar," in *Proc. of the German Microwave Conf*, 2006.

[21] N. Chen, R. Gourova, O. Krasnov, and A. Yarovoy, "The influence of the water-covered dielectric radome on 77ghz automotive radar signals," in *2017 European Radar Conference EURAD*, 2017, pp. 139–142.

[22] R. Gourova, O. Krasnov, and A. Yarovoy, "Analysis of rain clutter detections in commercial 77 ghz automotive radar," in *2017 European Radar Conference EURAD*, 2017, pp. 25–28.

[23] L. W. Li, P. S. Kooi, M. S. Leong, and T. S. Yeo, "On the simplified expression of realistic raindrop shapes," *Microwave and Optical Technology Letters*, vol. 7, no. 4, pp. 201–205, 1994.

[24] S. Hasirlioglu and A. Riener, "Introduction to rain and fog attenuation on automotive surround sensors," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–7.

[25] G. Feingold and Z. Levin, "The lognormal fit to raindrop spectra from frontal convective clouds in israel," *Journal of Climate and Applied Meteorology*, vol. 25, no. 10, pp. 1346–1363, 1986.

[26] A. S. Glassner, *An introduction to ray tracing*. Elsevier, 1989.

[27] G. Mie, "Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen," *Annalen der Physik*, vol. 330, no. 3, pp. 377–445, 1908.

[28] P. Laven, "Simulation of rainbows, coronas, and glories by use of mie theory," *Applied Optics*, vol. 42, no. 3, p. 436, 2003.

[29] H. C. Hulst and van de Hulst, Hendrik C, *Light scattering by small particles*. Courier Corporation, 1957.

[30] D. J. Segelstein, "The complex refractive index of water," Ph.D. dissertation, University of Missouri, 1981.

[31] M. I. Skolnik, Ed., *Radar handbook*, 2nd ed., ser. Electronics electrical engineering. New York: McGraw Hill, 1990.

[32] N. Yamada, Y. Tanaka, and K. Nishikawa, "Radar cross section for pedestrian in 76ghz band," in *2005 European Microwave Conference*, vol. 2, 2005, pp. 4 p–.

[33] V. Sandner, "Development of a test target for aeb systems," *23rd Int. Technical Conference on the Enhanced Safety of Vehicles (ESV): Research Collaboration to Benefit Safety of All Road Users*, 2013.

[34] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.