

Learning Sensor Models for Virtual Test and Development

Nils Hirsenkorn*, Hakim Kolsi*, Moez Selmi*, Alexander Schaermann†, Timo Hanke†, Andreas Rauch†, Ralph Rasshofer† and Erwin Biebl*

Summary: In order to keep up with the growing complexity of safety-critical driver assistance systems, testing methods need to advance alike. Virtual testing represents a controlled and deterministic environment, which is scalable with rising computing power. However, to obtain a valuable simulation, the realism of the simulated vehicular perception is crucial. This paper describes a method for modeling a sensor, which adapts to the behavior of sensor and reference measurements. The statistical framework enables generic use, as different environment and sensor-output types can be treated within the same model. To prove this flexibility, various quantities of different sensors are modeled. Presented models envelop the field of view, the estimated position, the estimation of other vehicles' dimensions and a point cloud model.

Keywords: Modeling, Sensor, Simulation, Testing

1 Introduction

As driver assistance systems become increasingly complex, the requirements for testing and development are rising alike. Established testing routines are reaching their limits and the development of other solutions is of interest [1, 2, 3].

Compared to road tests, virtual tests have various advantages: The controllable environment of the simulation enables direct testing of rare but critical scenarios without any risk. Moreover, virtual testing can be deeply integrated in the development process as instant feedback about critical function failure for developers. For major software releases it might be helpful to detect problems using test-drives in a virtual environment before starting long and expensive street tests.

The environment perception is the main input of ADAS. Therefore, the degree of realism acquired in the simulation is mainly dependent on the quality of the sensor-models. Current works on sensor-models can be divided into two categories: On one hand, „physical“ approaches exist [4, 5]. Objective of those models is to replicate electromagnetic propagation. The major advantage is the possibility to consider complex geometries and material properties. Disadvantages are the high computational cost and the dependence on low level interfaces of the simulator, which make portability to other simulators hard.

*Professorship of Microwave Engineering, Technical University of Munich (TUM), Arcisstraße 21, Munich, nils.hirsenkorn@tum.de, hakim.kolsi@tum.de, ga97maz@mytum.de (Moez Selmi), biebl@tum.de.

†BMW Group, Parking 19, Garching, alexander.schaermann@bmw.de, timo.hanke@bmw.de, andreas.rauch@bmw.de, ralph.rasshofer@bmw.de.

On the other hand, statistical sensor-models exist. The major advantage is the low computational cost. This enables extensive Monte-Carlo testing of various scenarios. Moreover their interfaces can be standardized [6], as statistical sensor-models rely on high-level information such as object-lists. Statistical models can be divided in two subclasses: Parametric models [7, 8, 9] assume a certain probability distribution, which is described by a fixed set of parameters. Those parameters are often interpretable, therefore model properties can be set before sensor-measurements are available, e.g. in early development stages. The second subclass are non-parametric methods. The density function is fitted to recorded data, without restriction to a certain class of distributions. A comprehensive comparison of the subclasses can be found in [10].

The non-parametric approach used in this contribution was first introduced in [11] and further developed in [10]. In this paper, a novel generalization to various types of sensor-outputs and environment descriptions is shown. This generalization empowers a generic use, which is demonstrated by modeling various quantities of different sensors with different types of reference measurements.

2 Modeling

In this section the problem is formalized, then the basic idea and the theoretical background of the model are described.

2.1 Problem description and basic approach

The task of a sensor is to measure a quantity in a certain environment. To formalize this, a feature vector \mathbf{X} describing the state of the environment, and a feature vector \mathbf{Z} describing the output of the sensor, are introduced. The transition from \mathbf{X} to \mathbf{Z} is assumed to be non-deterministic. Therefore a probabilistic connection is enabled. This can be described using the conditional probability density function (PDF) $p(\mathbf{z}|\mathbf{x})$.

When using a driving simulator, the simulated sensor output \mathbf{z}_{sim} is drawn from an estimated conditional PDF $\hat{p}(\mathbf{z}_{\text{sim}}|\mathbf{x}_{\text{sim}})$ in each time-step. Here the current state of the environment described by the simulator \mathbf{x}_{sim} is used. Throughout this paper, the index “sim” indicates a quantity in simulation, while “mea” is a quantity acquired by measurements on test-drives.

The basic idea of the approach is to return an output close to a recorded output, which was measured in a similar situation. Therefore, the current state of the simulation is compared to each state in which the measurements were received. The sensor-outputs recorded in the most similar environments are then considered to return an output close to one of them.

2.2 Non-parametric estimation

First, the conditional PDF is transformed to a joint PDF using Bayes’ theorem

$$\hat{p}(\mathbf{z}_{\text{sim}}|\mathbf{x}_{\text{sim}}) = \frac{\hat{p}(\mathbf{z}_{\text{sim}}, \mathbf{x}_{\text{sim}})}{\hat{p}(\mathbf{x}_{\text{sim}})} = \frac{\hat{p}(\mathbf{z}_{\text{sim}}, \mathbf{x}_{\text{sim}})}{c}. \quad (1)$$

The state of the sensor-output \mathbf{z}_{sim} and the state of the environment \mathbf{x}_{sim} are feature vectors describing for example object properties or weather conditions. As the state of the environment is already known, the denominator is regarded as a normalization constant.

The joint PDF $\hat{p}(\mathbf{z}_{\text{sim}}, \mathbf{x}_{\text{sim}})$ is estimated using a multivariate non-parametric approach

$$\hat{p}(\mathbf{z}_{\text{sim}}, \mathbf{x}_{\text{sim}}) = \frac{1}{N} \sum_{i=1}^N K \left(\begin{pmatrix} \mathbf{z}_{\text{sim}} \\ \mathbf{x}_{\text{sim}} \end{pmatrix} - \begin{pmatrix} \mathbf{z}_{\text{mea},i} \\ \mathbf{x}_{\text{mea},i} \end{pmatrix} \right) \quad (2)$$

$$= \frac{1}{N} \sum_{i=1}^N K \left(\begin{matrix} \mathbf{z}_{\text{sim}} - \mathbf{z}_{\text{mea},i} \\ \mathbf{x}_{\text{sim}} - \mathbf{x}_{\text{mea},i} \end{matrix} \right) = \frac{1}{N} \sum_{i=1}^N K \left(\begin{matrix} \Delta \mathbf{z}_i \\ \Delta \mathbf{x}_i \end{matrix} \right). \quad (3)$$

The function K is the kernel function, which can be interpreted as a measure of similarity [11]. The measurements were acquired in N time-steps. This approach requires the assumption that all environment descriptions \mathbf{x} and all sensor-outputs \mathbf{z} are respectively of the same type. Else K would have to treat multiple dimensionalities, or would compare quantities, which are not properly comparable. Section 2.4 further describes this issue and presents a solution for overcoming the limitations.

2.3 Kernel selection

This section briefly discusses the choice of kernel functions. Further description can be found in [12] and [13].

As a special kernel function, a “diagonal” Gaussian kernel, separating the input and the output, is particularly well interpretable

$$K \left(\begin{matrix} \Delta \mathbf{z}_i \\ \Delta \mathbf{x}_i \end{matrix} \right) = \frac{1}{c'} K_{\text{ref}}(\Delta \mathbf{x}_i) \cdot K_{\text{sen}}(\Delta \mathbf{z}_i) = \frac{1}{c''} \exp\left(-\frac{\Delta \mathbf{x}_i^T \boldsymbol{\Sigma}_{\text{ref}}^{-1} \Delta \mathbf{x}_i}{2}\right) \exp\left(-\frac{\Delta \mathbf{z}_i^T \boldsymbol{\Sigma}_{\text{sen}}^{-1} \Delta \mathbf{z}_i}{2}\right). \quad (4)$$

The relevance of each measurement is determined by K_{ref} [11]. In other words, it quantifies the similarity of each measurement to the current state of the simulation. As common in literature, we used a Gaussian kernel with a diagonal covariance matrix for all models. The variances, $\boldsymbol{\Sigma}_{\text{ref}}$ and $\boldsymbol{\Sigma}_{\text{sen}}$, should be chosen according to the “bias-variance trade-off” [13]: Too narrow variances lead to only few recordings to be “close” to the current situation \mathbf{x}_{sim} . Those few measurements usually do not expose all possible behaviors given this situation. Choosing the variances too high, leads to oversmoothing: Behavior of states, which are not representative for the current situation \mathbf{x}_{sim} , is biasing the true PDF. In general, the variances should decrease as the number N of measurements rises. Currently we set the values manually.

The shape of the contribution of each sensor-measurement to the resulting PDF is defined by K_{sen} . Again, a Gaussian kernel with a diagonal covariance matrix is chosen. In former works, we set the variances according to a leave one out cross-validation method [12]. However, in the authors’ opinion, the effect of smoothing in K_{sen} can be neglected, when a sufficient number of measurements was recorded. Therefore, its variances are set towards zero, resulting in a Dirac delta function. This reduces the computational cost at run-time, but the predominant effect is that it enables an important property: Under the assumption of sufficiently well described environment states and sufficiently small variances in K_{ref} , only measurements are simulated, which truly appeared. Therefore,

each behavior in the simulation is also possible in the vehicle. Nothing is made up, what has not been recorded. Moreover, each behavior can be traced back to exactly one true measurement. The latter is also possible when the variances are non-zero [10].

2.4 Generalization to various environment and sensor-output types

The method described in the previous section is feasible as long as all vectors \mathbf{x} and \mathbf{z} are respectively of the same type. However, for sensors of the environmental perception, neither the environment, nor the sensor-output, should be limited to a certain type: Imagine one of the measurements $\mathbf{z}_{1:N}$ would not contain a sensor-output, e.g. due to occlusion. This measurement would not be properly comparable to the other sensor-outputs. Necessary conditions for two vectors to be of the same class (i.e. kind, type), are for example an equal dimensionality, and equal units in the respective dimensions. However, a counterexample to the necessary conditions is the comparison of two positions: one describes a tree, the other describes a car. Even though the necessary conditions are met, it is not possible to compare them, as they describe fundamentally different situations. In other words: it is like trying to compare apples and pears.

In the following, we drop the assumption that all sensor-outputs \mathbf{z} are of equal type, whilst maintaining this assumption on all \mathbf{x} . We introduce the type of the simulated sensor-output $T_{\mathbf{z}}$ using the law of total probability

$$\hat{p}(\mathbf{z}_{\text{sim}}|\mathbf{x}_{\text{sim}}) = \sum_{d=1}^D \hat{p}(\mathbf{z}_{\text{sim}}|\mathbf{x}_{\text{sim}}, t_{d,\mathbf{z}}) \cdot \hat{P}(t_{d,\mathbf{z}}|\mathbf{x}_{\text{sim}}). \quad (5)$$

$T_{\mathbf{z}}$ is a random variable which is dependent, even determined by a sensor-output \mathbf{Z} :

$$T_{\mathbf{z}}(\mathbf{Z}) = \begin{cases} t_{1,\mathbf{z}} & \text{if } \mathbf{Z} \text{ is of output type 1} \\ \dots & \dots \\ t_{D,\mathbf{z}} & \text{if } \mathbf{Z} \text{ is of output type } D \end{cases} \quad (6)$$

The values represented by $t_{1:D,\mathbf{z}}$ have to be unique and constant, but are of no further importance. D represents the number of different output types.

For example, when in reality one vehicle is present, the sensor will return an output of one of the following types:

$t_{1,\mathbf{z}}$: the sensor does not perceive the vehicle (object loss or out of the field of view)

$t_{2,\mathbf{z}}$: the sensor does perceive the vehicle

$t_{3,\mathbf{z}}$: the sensor does perceive the vehicle and one more object (track-split or clutter)

$t_{4:D,\mathbf{z}}$: ... increasing the number of track-splits

Summing up, the introduction of the type $T_{\mathbf{z}}$ is used to determine the type $t_{d,\mathbf{z}}$ of all measurements, which are used in the kernel estimation of the PDF $\hat{p}(\mathbf{z}_{\text{sim}}|\mathbf{x}_{\text{sim}}, t_{d,\mathbf{z}})$ using eq. (2). In other words, for each class $t_{1:D,\mathbf{z}}$, a distinct estimator is trained, just using the measurements of its belonging class. Regarding the kernel function, we currently use the same K_{ref} for each estimator.

The second term $\hat{P}(t_{d,z_{\text{sim}}}|x_{\text{sim}})$ indicates the probability of choosing an output of type $t_{d,z_{\text{sim}}}$, given the current state of the environment. It is also estimated using a non-parametric approach

$$\hat{P}(t_{d,z}|x_{\text{sim}}) = \frac{1}{N \cdot C_2} \sum_{i=1}^N K_2 \left(\begin{pmatrix} t_{d,z} \\ x_{\text{sim}} \end{pmatrix} - \begin{pmatrix} T_z(z_{\text{mea},i}) \\ x_{\text{mea},i} \end{pmatrix} \right). \quad (7)$$

The kernel function K_2 has to assure that its value is zero, when $t_{d,z_{\text{sim}}}$ is not equal to $T_z(z_{\text{mea},i})$. We currently use a kernel function of the type described in eq. (4) and choose $K_{2,\text{ref}}$ equal to K_{ref} .

The assumption of a fixed type in \mathbf{x} can be dropped using a similar manner. For each class of \mathbf{x} , a different non-parametric estimator is trained. However, for estimating the type of the reference vector $T_x(x_{\text{sim}})$, no estimator is required since x_{sim} , and therefore its type, is already known.

A general, vivid introduction to the method is shown in [11]. Regarding performance, the efficient implementation described in [10] allows real-time capabilities with a huge number of measurements.

3 Application

This part describes the modeling of various quantities. A comprehensive overview of existing sensor-errors is given in [14].

The described models can either run in a subsequent order. This would neglect correlations that are not induced by the environment description \mathbf{x} . The alternative of merging the models will drop all independence assumptions. However, as the dimensionality of the state vectors \mathbf{x} and \mathbf{z} increases, the curse of dimensionality will lower the model quality. Therefore, more measurements need to be available for a proper estimation of the PDF.

3.1 Field of view and position estimation

Probably the most important quantities for the environmental perception are the detection of objects and the estimation of their position. The „field of view“ denotes the positions where objects are perceived. The field of view and the position estimation can be modeled in one step, using the state description

$$\mathbf{x} = (o_{\text{ref},x,\text{corner}}, o_{\text{ref},y,\text{corner}})^T, \quad \mathbf{z} = (\Delta o_x, \Delta o_y, o_{\text{object reference point}})^T. \quad (8)$$

$o_{\text{ref},x,\text{corner}}$ and $o_{\text{ref},y,\text{corner}}$ describe the longitudinal and the lateral position of the closest corner of the target vehicle relative to the ego (i.e. host) vehicle. The reference vectors $\mathbf{x}_{\text{mea},1:N}$ are recorded using a high precision reference system in the host and the target vehicle, consisting of a GPS with carrier-phase and an inertial measurement unit. Δo_x and Δo_y denote the estimated position relative to the true position of the closest corner of the vehicle $o_{\text{ref},x,\text{corner}}, o_{\text{ref},y,\text{corner}}$. By inserting the sensed position relative to true position, the behavior of the transition from \mathbf{x} to \mathbf{z} stays similar, when varying \mathbf{x} by reasonable values. Therefore, more smoothing can be enabled by using high values in Σ_{ref} . Currently, the diagonal variances in Σ_{ref} are set to 0.5 [m] ($o_{\text{ref},x,\text{corner}}$) and 0.5 [m] ($o_{\text{ref},y,\text{corner}}$).

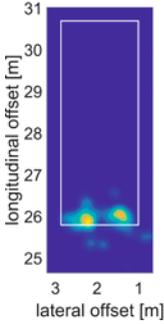
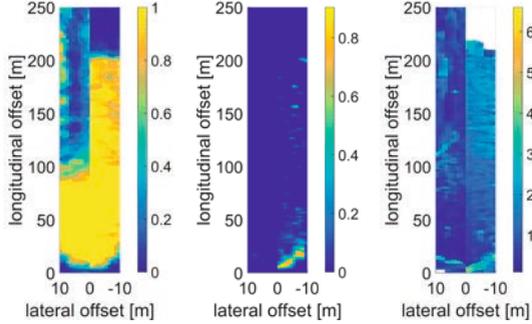


Figure 1: Exemplary radar model PDF

$$\hat{p}(\mathbf{z}_{\text{sim}} |$$

$$\mathbf{x}_{\text{sim}} = \begin{pmatrix} 25.8[\text{m}] \\ 1[\text{m}] \end{pmatrix}, t_{2,z}).$$



(a) Detection

$$P(t_{2:D,z} | \mathbf{x}_{\text{sim}}).$$

(b) Track-split

$$P(t_{3:D,z} | \mathbf{x}_{\text{sim}}).$$

(c) Mean

$$\text{absolute error}$$

in meters.

Figure 2: Various quantities of a lidar- (to the left, positive lateral values), and a radar-system (to the right, negative lateral values).

The variances of the output side Σ_{sen} are all set to zero as explained in section 2.3. $o_{\text{object reference point}}$ denotes the point on the target vehicle to which the position $\Delta o_x, \Delta o_y$ refers (e.g. bounding box middle, rear left corner, rear middle, undefined, ...). With the chosen state variables, possible classes or the sensor-output \mathbf{z} are indicated in the example of section 2.4.

It is assumed, that the sensor behaves in a symmetric manner in lateral direction (i.e. when changing the sign of the lateral position $o_{\text{ref},y,\text{corner}}$). This was introduced by adding “mirrored“ measurements $\mathbf{x}_{\text{mea},1:N}$ and sensor-outputs $\mathbf{z}_{\text{mea},1:N}$ to the pool of measurements.

To couple the reference and the sensor measurements, an offline track to track association is employed. It mainly compares the position and the velocities over the whole life cycle of the objects. This enables associations, even when deviations become high (e.g. when the object is far away).

The measurements were acquired on a test track, which resembles a motorway. The weather was often foggy. Note that the sensors were recorded at once, hence both encountered the same situations and weather conditions. In order to exclude influences of occlusions, there neither were objects, nor guard-rails between the host and the target vehicle. Furthermore, the deployed systems are not the latest versions. Therefore the most recent systems might exhibit even higher performance. In total, two and a quarter hours of test-drive, which results in approximately $N = 10^5$ time-steps, were recorded ($2 \cdot 10^5$ when the symmetry assumption in lateral direction is included).

Fig. 1 shows the PDF of the sensed position of a radar system. The white rectangle indicates the bounding box of the target vehicle, measured by the reference system. For visualization purposes, the first two diagonal variances of Σ_{sen} were set to 0.01 [m] opposed to the usual zero [m].

Using calculated PDFs at each location, fig. 2 compares the detection rate (field of

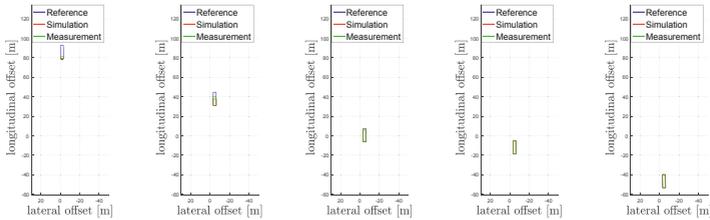


Figure 3: Comparison of reference, measurement, and simulation.

view), the rate of track-splits and the accuracy of the position estimation of a commercial automotive radar and a lidar system. The laserscanner offers a robust detection of objects in a range up to 80 m, almost no track-splits (i.e. multiple detections of the same vehicle) and a high accuracy, even at poor weather conditions. The latter was quantified by the mean of the absolute position difference between the sensed position and the position measured by the reference system. Note that the position of the reference system was transformed to the reference point indicated by the sensor (e.g. the rear right corner). For the accuracy calculation only sensor-outputs containing one sensed vehicle were regarded (i.e. type $t_{2,z}$). The radar shows a high detection rate up to 200 m, beyond which it appears to be filtered by the signal processing. A drawback is the existence of track-splits at the close border of the field of view. As our radar does not indicate a reference point, we assume the middle of the rear of the vehicle for the accuracy calculation. This is the best guess, when observing PDFs such as fig. 1. At good weather conditions, all quantities of the lidar improved remarkably. This especially holds for the range of detection, which was then robust up to approximately 200-230 m.

3.2 Length and width estimation

The aim of this model is to simulate the length and width of an object, estimated by a sensor. An exemplary behavior can be seen in figure 3. The figure shows an overtaking maneuver, where the host vehicle passes a truck. To simulate the behavior, the following feature vectors are chosen

$$\mathbf{x} = (o_x, o_y, l_{\text{ref}}, w_{\text{ref}}, l_{T-1}, w_{T-1})^T, \quad \mathbf{z} = (l, w)^T. \quad (9)$$

o_x and o_y represent the position of the target vehicle measured by the laserscanner. These measurements are a sufficient reference, since the sensor behavior changes relatively slow in o_x and o_y . Therefore, the error in the estimated position is negligible. Due to their good angular resolution, laserscanners can determine the dimensions of a vehicle well, when the vehicle is orientated sidelong to the sensor. However, when the object has not been visible from the side, a length which is too low is assumed. The estimation is particularly good when the vehicle is close, and viewed from the side. Therefore, the reference length l_{ref} and width w_{ref} are estimated using the lidar system. The reference is represented in figure 3 by the blue box, whose estimation was automatically derived approximately from the fourth plot of the series. The object is visible and tracked in

all directions as our sensor-setup consists of multiple fused laserscanners. To acquire the best estimation, a holistic view of time is used. We assume the sensor returns its best estimation of the dimensions when the vehicle was already seen from the side. We can use this measurement as reference for all other measurements of the vehicle, as the dimensions are a constant property of an object. The obtained reference is considerably more accurate than the sensor-measurement, especially regarding those of the beginning. The estimation of the dimensions in step T is dependent on the former estimations. By including the previous sensor-estimations l_{T-1} and w_{T-1} , we assume that a Markov-chain of first order is sufficient (e.g. if a Kalman filter is used).

The sensor-output \mathbf{z} is characterized using the estimated length l and width w estimated by the sensor. To obtain the sets of measurements $(\mathbf{x}_{\text{mea},i}, \mathbf{z}_{\text{mea},i})_{i=1:N}$ using the described reference generation, 8000 km of test drives were used. 424 overtaking maneuvers were extracted for whom we expect to know a proper reference. The diagonal entries of the variances Σ_{ref} used in the kernel are chosen to 7[m] (σ_x), 7[m] (σ_y), 1[m] (l_{ref}), 0.2[m] (w_{ref}), 0.1[m] (l_{T-1}) and 0.1[m] (w_{T-1}). In order to assure the model is not solely overfitting, the track shown in figure 3 was excluded from the training of the model.

3.3 Point cloud of a laserscanner

In this section, the simulation of a point cloud generated by a laserscanner is demonstrated. The sensor-output \mathbf{z} in this model is a set of three dimensional points. Throughout this model, only dynamic objects are regarded. The state of a target vehicle is characterized by its position o_{ref} and orientation γ_{ref}

$$\mathbf{x} = (o_{\text{ref},x,\text{corner}}, o_{\text{ref},y,\text{corner}}, \gamma_{\text{ref}})^T. \quad (10)$$

$o_{\text{ref},x,\text{corner}}$ and $o_{\text{ref},y,\text{corner}}$ are defined equal to section 3.1. γ_{ref} denotes the orientation of the target vehicle relative to the host vehicle. \mathbf{x} was recorded using the same high-precision reference system as described in section 3.1.

The sensor-output \mathbf{z} is a point cloud consisting of M points

$$\mathbf{z} = (\Delta o_{x,\text{point},1:M}, \Delta o_{y,\text{point},1:M}, \Delta o_{z,\text{point},1:M})^T. \quad (11)$$

$\Delta o_{\text{point},i}$ denotes the position of a point in the point cloud relative to the nearest corner of the target vehicle. When recording the sensor-measurements $\mathbf{z}_{\text{mea},1:N}$, all points caused by static objects are neglected. The distinction of static and dynamic objects has already been performed by the sensor. Moreover, only points which were caused by the target vehicle measured by the reference system are regarded. Therefore the points have to be within a radius of three meters around the position of the target vehicle estimated by the reference system. The diagonal entries of the variances Σ_{ref} used in the kernel are chosen to 2[m] ($o_{\text{ref},x,\text{corner}}$), 2[m] ($o_{\text{ref},y,\text{corner}}$), and 0.1[rad] (γ_{ref}). Figure 4 compares recorded measurements to ones generated by the sensor-model in one run. In order to assure the model is not solely overfitting, the track shown in the figure was excluded from the training of the model. Summing up, figure 4 demonstrates that the sensor-model is able to generate an output similar to the recording.

To allow deeper insight into the model and the sensor, figure 5 shows the median number of points generated by the model, when the target vehicle is at different locations. The adapted behavior exposes an approximately quadratically decreasing number of points in distance, which was to be expected.

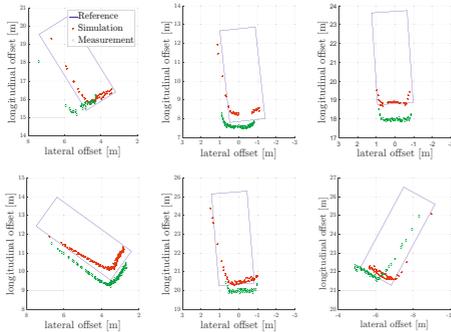


Figure 4: Comparison of reference, measurement, and simulation.

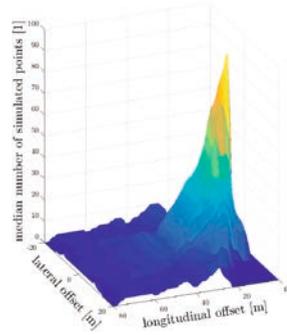


Figure 5: Median number of simulated points at various positions of the target vehicle.

4 Conclusion and outlook

This paper described a statistical approach for modeling sensor behavior. The main idea is to behave identically to the sensor, when measuring in a similar environment. Special attention was centered around sensor-outputs or environments exhibiting various, non-comparable types. The result is a statistical framework, which is capable of adapting to diverse behaviors, based on sensor and reference measurements. This was demonstrated by modeling different quantities of multiple kinds of sensors.

Future work should focus on the quantification of model quality. This is a necessity since the realism of the sensor-model is crucial for the realism of the whole virtual testing. Furthermore, this would allow the optimization of the model's degrees of freedom such as the choice of the state variables or the kernel functions.

Most fusion methods are based on forward or inverse sensor-models. The described approach is based on the estimation of the joint probability density function, which connects the state of the environment with the state of the sensor-output. This intermediate result can be used to replace the common, simple, and inaccurate sensor-models and therefore improve the accuracy of sensor fusion.

Our future work includes sensor-models derived from long test-drives without reference, and physical sensor-models based on ray-tracing. Furthermore, we plan to compare physical and statistical models.

Literature

- [1] Bundesministerium für Wirtschaft und Energie, “BMW startet Forschungsprojekt für Testverfahren für hochautomatisierte Fahrzeuge,” www.bmwi.de/DE/Presse/pressemitteilungen,did=749340.html, Accessed: 2016-11-03.

- [2] W. Wachenfeld and H. Winner, "Virtual Assessment of Automation in Field Operation A New Runtime Validation Method," in *10. Workshop Fahrerassistenzsysteme*, pp. 161–170, 2015.
- [3] S. Olivares, N. Rebernik, A. Eichberger, and E. Stadlober, "Virtual Stochastic Testing of Advanced Driver Assistance Systems," in *Advanced Microsystems for Automotive Applications*, Lecture Notes in Mobility, pp. 25–35, Springer International Publishing, 2015.
- [4] J. Maurer, *Strahlenoptisches Kanalmodell für die Fahrzeug-Fahrzeug-Funkkommunikation*. dissertation, Universität Karlsruhe, IHE, 2005.
- [5] M. Weiskopf, C. Wohlfahrt, and A. Schmidt, "Integrationslösung zur Absicherung eines realen Radarsensors im Systemverbund mit der Hardware-in-the-Loop Testtechnologie," *Automotive-Safety & Security*, 2014.
- [6] T. Hanke, N. Hirsenkorn, P. Garcia-Ramos, M. Schiementz, S. Schneider, and E. Biebl, "Open Simulation Interface." www.hot.ei.tum.de/forschung/automotive-veroeffentlichungen/, 2017. Accessed: 2017-01-10.
- [7] R. H. Rasshofer, J. Rank, and G. Zhang, "Generalized Modeling of Radar Sensors for Next-Generation Virtual Driver Assistance Function Prototyping," in *12th World Congress on Intelligent Transport Systems*, 2005.
- [8] T. Hanke, N. Hirsenkorn, B. Dehlink, A. Rauch, R. Rasshofer, and E. Biebl, "Generic architecture for simulation of ADAS sensors," in *International Radar Symposium*, pp. 125–130, IEEE, 2015.
- [9] S. Bernsteiner, Z. Magosi, D. Lindvai-Soos, and A. Eichberger, "Phänomenologisches Radarsensormodell zur Simulation längsdynamisch regelnder Fahrerassistenzsysteme," *VDI-Bericht 2169, Elektronik im Fahrzeug*, 2013.
- [10] N. Hirsenkorn, T. Hanke, A. Rauch, B. Dehlink, R. Rasshofer, and E. Biebl, "Virtual sensor models for real-time applications," *Advances in Radio Science*, vol. 14, pp. 31–37, 2016.
- [11] N. Hirsenkorn, T. Hanke, A. Rauch, B. Dehlink, R. Rasshofer, and E. Biebl, "A non-parametric approach for modeling sensor behavior," in *International Radar Symposium*, pp. 131–136, DGON, 2015.
- [12] J. S. Simonoff, *Smoothing methods in statistics*. Springer Science & Business Media, 2012.
- [13] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.
- [14] T. Hanke, N. Hirsenkorn, B. Dehlink, A. Rauch, R. Rasshofer, and E. Biebl, "Classification of Sensor Errors for the Statistical Simulation of Environmental Perception in Automated Driving Systems," in *International Conference on Intelligent Transportation Systems*, IEEE, 2016.