

# Uni-DAS

13. Workshop  
Fahrerassistenz  
und  
automatisiertes Fahren

FAS 2020



Copyright Uni-DAS e.V.

Alle Rechte, auch das des auszugsweisen Nachdrucks, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

ISBN: 978-3-941543-47-8

Druckerei  
Digitaldruck AixPress GmbH  
Eupener Straße 35  
52066 Aachen

Uni-DAS e.V.  
Otto-Verndt-Straße 2  
64287 Darmstadt

---

# **Uni-DAS**

---

**13. Workshop  
Fahrerassistenz  
und  
automatisiertes Fahren  
FAS 2020**

01. – 03.04.2020  
Hotel Gut Moierhof, Walting im Altmühlatal

---

# **Uni-DAS**

---

**13. Workshop  
Fahrerassistenz  
und  
automatisiertes Fahren  
FAS 2020**

01. – 03.04.2020  
Hotel Gut Moierhof, Walting im Altmühlatal

## Vorwort

Der Workshop „Fahrerassistenzsysteme und automatisiertes Fahren 2020“ knüpft als dreizehnte Veranstaltung dieser Art an die erfolgreichen Vorgänger an. Und zum elften Mal findet er in Walting statt, wodurch dieser Ort in der Community nahezu synonym für diese Veranstaltung steht. Auch wenn die Zahl der in diesem Bereich forschenden und entwickelnden Personen seit Beginn der Reihe um mindestens eine Größenordnung gewachsen ist, bleiben wir dem Konzept der Kombination von wissenschaftlicher Tagung und interaktivem Workshop mit beschränkter Teilnehmerzahl treu, um weiterhin den Tiefgang der Diskussion zu ermöglichen.

Fahrerassistenzsysteme haben sich mittlerweile etabliert und tragen nachweislich zur Steigerung der Verkehrssicherheit bei. Auch wenn sie noch deutliches Weiterentwicklungspotenzial besitzen, wird die aktuelle Diskussion durch das automatisierte Fahren ohne Fahrerüberwachung bestimmt. Den großen Versprechungen über die Zukunft der automatisch fahrenden Fahrzeuge stehen aber immer noch vielschichtige ungelöste Aufgaben gegenüber. Dazu gehören sowohl Aspekte der maschinellen Perzeption und Kognition, als auch methodische und juristisch relevante Fragen hinsichtlich der Absicherung, Freigabe und Zertifizierung. Die Notwendigkeit der Aktualisierung von Funktionen und das Potential von funktionalen Erweiterungen erfordern darüber hinaus innovative Architekturen und Prozesse, die deutlich über die Systemgrenze „Fahrzeug“ hinaus gedacht und gestaltet werden müssen. Im BMBF geförderten Projekt UNICARagil erforschen wir in einem Konsortium von 8 Universitäten und 8 Unternehmen an geeigneten Architekturen, die wir mit Ihnen diskutieren möchten. Unter dem Titel „Welche Rolle spielt die Zulassung?“ wollen wir im Workshop-Teil der Veranstaltung Herausforderungen und Lösungsansätze diskutieren, damit die Technik im Zuge der Markteinführung ihr volles Potential entfalten kann.

Die einzelnen Vorträge des Workshops und Beiträge dieses Tagungsbandes beleuchten ein breites Spektrum des aktuellen Wissensstandes im Bereich der Fahrerunterstützung und des automatischen Fahrens. In kaum einer anderen Reihe lässt sich so klar nachvollziehen, wie sich die Fahrerassistenzsysteme und das automatisierte Fahren in den zurückliegenden 20 Jahren entwickelt haben.

Mein Dank gilt allen Helfern bei der Organisation des Workshops, insbesondere Bastian Lampe, Isabell Müller und Martin Dietze sowie den Mitgliedern von Uni-DAS e.V. Ich wünsche allen Teilnehmern neue Einsichten, spannende Diskussionen und das Knüpfen neuer und Vertiefen bestehender Kontakte.

Aachen im April 2020

Lutz Eckstein

## Inhaltsverzeichnis

### Perception und Mapping

- |   |   |    |
|---|---|----|
| 1 | <b>Identification of Uncertainty in Artificial Neural Networks</b>  | 1  |
|   | N. Jourdan, E. Rehder, U. Franke,<br>Daimler AG   |    |
| 2 | <b>A machine learning approach for ultrasonic noise classification and suppression</b>                          | 11 |
|   | M.-E. Mohamed, H. Gotzig, R. Zoellner, P. Maeder, Valeo   |    |
| 3 | <b>Kontinuierliche, fahrstreifengenaue Kreuzungsschätzung aus den Fahrtverläufen anderer Verkehrsteilnehmer</b> | 21 |
|   | A. Meyer, J. Walter, M. Lauer, C. Stiller,<br>FZI Karlsruhe   |    |

### Lokalisierung

- |   |   |    |
|---|---|----|
| 4 | <b>Verfahren zur Korrektur von inkonsistenten Lokalisierungsdaten in modularen technischen Systemen</b> | 30 |
|   | T. Homolla, G. Gottschalg, H. Winner,<br>TU Darmstadt   |    |
| 5 | <b>Selbstlokalisierung für automatisierte Fahrzeuge mit tiefen neuronalen Netzen</b>                    | 42 |
|   | N. Engel, K. Dietmayer,<br>Uni Ulm  |    |
| 6 | <b>Zuverlässige und integre Landmarken-basierte Lokalisierungsmethoden zum automatisierten Fahren</b>   | 52 |
|   | I. Hofstetter, F. Ries, F. Schuster, M. Haueis, Daimler AG  |    |

## Absicherung I

- 7 **Validation of automated driving - a structured analysis and survey of approaches** 62  
J. E. Stelle, M. Woehrle, T. Brade, A. Poddey, W. Branz, Robert Bosch GmbH
- 8 **Saturation Effects in Recorded Vehicle and Pedestrian Maneuver Data for the Test of Automated Driving** 72  
L. Hartjen, R. Philipp, F. Schuldt, B. Friedrich, Volkswagen Group Innovation
- 9 **Verifikation von HD-Karten mittels räumlich und semantisch separierbarer 1D-Signale** 82  
J.-H. Pauls, T. Strauß, C. Hasberg, M. Lauer, C. Stiller, KIT

## Absicherung II

- 10 **Functional Decomposition of Automated Driving Systems for the Classification and Evaluation of Perceptual Errors** 91  
R. Philipp, F. Schuldt, F. Howar,  
Volkswagen Group Innovation
- 11 **How to evaluate synthetic radar data? Lessons learned from finding driveable space in virtual environments.** 104  
M. F. Holder, P. Rosenberger, C. Linnhoff, J. R. Thielmann, H. Winner, TU Darmstadt

## Mensch-Maschine Interaktion

- 12 **Bewertung der Sicherheit der Mensch-Maschine Interaktion teilautomatisierter Fahrfunktionen (Level 2)** 115  
A. Wiggerich, H. Hoffmann, N. Schömig, K. Wiedemann, K. Segler, BASF
- 13 **Manual Adaption of Steering Support in a Take-Over Scenario – A Technical Evaluation** 124  
L. Kalb, B. Karakaya, TU München

## Planung

- 14 **Learning Realistic High Level Decisions for Autonomous Driving at Complex Intersections** 132  
D. Kamran, M. Lauer, C. Stiller, KIT
- 15 **Systemarchitektur und Fahrmanöver zum sicheren Anhalten modularer automatisierter Fahrzeuge** 142  
S. Ackermann, H. Winner, TU Darmstadt

# Identification of Uncertainty in Artificial Neural Networks

Nicolas Jourdan\*    Eike Rehder\*    Uwe Franke\*

**Abstract:** Neural networks are the backbone of environment perception systems for autonomous driving. While they achieve state-of-the-art performance in most computer vision tasks, they typically do not provide self-evaluation with respect to their predictions. For autonomous vehicles, though, it is vital that the system actively reasons about its limitations. The aim of this work is to identify uncertainty in neural network decisions for semantic segmentation. To systematically evaluate this, we develop a methodology to compare neural networks' performance in out-of-distribution detection and uncertainty estimation. As the core contribution of our work, we propose a novel approach to learn uncertainty estimation for out-of-distribution detection from unlabeled parts of the training data. Our approach only extends the training strategy and therefore does not require any changes to network architecture or runtime. We show that resulting networks perform en par with state-of-the-art methods that require much greater computational efforts. Consequently, any given architecture for segmentation can be trained to also provide out-of-distribution detection.

**Keywords:** neural networks, out-of-distribution detection, semantic segmentation, uncertainty

## 1 Introduction

Neural networks have achieved state-of-the-art in most computer vision tasks and are the foundation of modern environment perception systems. While they have been deployed with great success, neural networks typically do not provide self-evaluation with respect to their predictions. If neural networks are used in environment perception systems of autonomous vehicles, though, it is mandatory that the system actively self-identifies its limitations as the human passenger does not provide a fallback option [13].

This work focuses on neural networks for road scene understanding from camera images. Thus, the aforementioned limitations include scene configurations the network was not trained to comprehend. In such situations the desired behavior of the classification system would be to express low confidence or high uncertainty for the detection. As neural networks can only be trained on a finite dataset, not every possible situation or even class configuration could be contained in a finite-sized and finite-class dataset. Thus, in deployment, a neural network will encounter situations that significantly differ from the data distribution it was trained with. For autonomous vehicles, it is crucially important

---

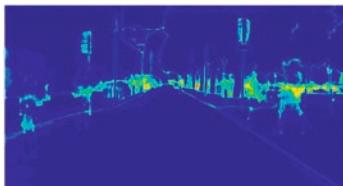
\*Daimler AG, Image Understanding Group (e-mail: {nicolas.jourdan, eike.rehder, uwe.franke}@daimler.com).

to detect these out-of-distribution (OOD) samples since failures may have catastrophic consequences.

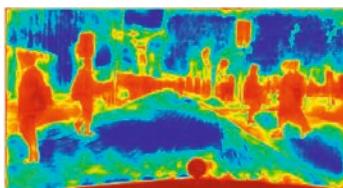
As an example, we provide semantic segmentation results of three different models, all trained on the Cityscapes dataset [1], in Figure 1. A standard network segments the image into relevant classes such as *road*, *infrastructure*, *vegetation* or *pedestrian* (Fig. 1 (a)). Now, suppose the same network was trained on data that did not include any *pedestrian* labels. Alarmingly, this network fills in the unknown pedestrians with highly confident predictions for the respective background, shown in Fig. 1 (b). An autonomous vehicle relying on this perception consequently would cause severe accidents. If we apply our proposed training method, the network achieves decent prediction in known regions while tagging the pedestrians as highly inconfident (*cf.* Fig. 1 (c)).



(a) Original image and prediction of the baseline network trained with all 19 classes.



(b) Uncertainty and prediction of the baseline network trained without *pedestrian* and *rider*.



(c) Uncertainty and prediction using our approach.

Figure 1: Example showcasing the behavior of the semantic segmentation network in the presence of unknown classes. Blue indicates low uncertainty while red indicates high uncertainty.

In this work, we present a method to extend any given segmentation architecture with uncertainty estimation. To put this into perspective, we initially review approaches from literature. We then present our simple yet effective training strategy. In order to evaluate and compare respective results, a methodology for out-of-distribution detection and network self-assessment is presented and used to benchmark the uncertainty estimation methods.

## 2 Uncertainty Estimation in Neural Networks

Any neural network that is used in the perception stage of an autonomous system should provide some means of self-evaluation. Unfortunately, the categorical decisions of neural networks do not provide this directly. Yet, the typical activation function for classification is the softmax  $s_i(\mathbf{a}) = \exp(a_i) / \sum_{j=1}^C \exp(a_j)$ , where  $a_i$  is the network's output logit for the  $i$ -th of  $C$  classes. The softmaxes can be interpreted as a categorical probability distribution. In the following, we will briefly explain how these outputs have been used to estimate uncertainty of network decisions in literature.

### 2.1 Reference Methods

**Naïve Baseline** The maximum softmax probability can be used as a confidence measure for out-of-distribution samples as well as misclassifications [6, 11]. This provides the baseline for the experiments of this work (*cf.* Fig. 1 (b)). In order to incorporate the prediction of all classes, the per-pixel entropy can be used instead. The entropy for this case with an input sample  $\mathbf{x}$  is defined as  $H(\mathbf{x}) = -\sum_{i=1}^C p_i(\mathbf{x}) \log p_i(\mathbf{x})$ . While the confidence score is easily acquired, it exhibits significant limitations. Since the softmax function normalizes the output distribution to  $\sum_{i=1}^C p_i(\mathbf{x}) = 1$ , the confidences by definition cannot be calibrated in open set conditions with unknown classes that are not accounted for during training. Consequently, the softmax probabilities only express relative confidences for the known classes (e.g. the image more likely shows a dog than a cat) but no overall confidence in the classification (e.g. the image shows neither a dog nor a cat).

**Temperature Scaling** Temperature scaling is an extension to the softmax function that is commonly used as the final layer in classification networks. For this, the inputs  $\mathbf{a}$  to the softmax are divided by a scalar constant *temperature*  $T \in \mathbb{R}^+$ ,

$$s_i(\mathbf{a}, T) = \frac{\exp(a_i/T)}{\sum_{j=1}^C \exp(a_j/T)}. \quad (1)$$

Since it only modifies the logits linearly, it can be applied to existing models without the need for architecture changes or retraining [11].

**Monte-Carlo Dropout** In Monte-Carlo dropout, multiple forward passes are performed for a test image. In each of the forward passes, random units are dropped by using dropout [8] layers in the network architecture, introducing variations in the predictions. Thus, they can be considered stochastic samples approximating a *Bayesian Neural Network* [4, 9]. The resulting  $N$  predictions are combined by class-wise averaging the predicted probabilities  $P_{final} = \frac{1}{N} \sum_{i=1}^N P_i$ . The combined per-pixel

probability distribution  $P_{final}$  is consequently used for uncertainty estimation. To enable the use of Monte-Carlo dropout for uncertainty estimation with the model used in this work, dropout layers are added to the central stages of the architecture, roughly following the setup of *Bayesian Segnet* but applied to an FCN [9]. We compute  $N = 10$  samples per image during testing. The major drawback of using Monte-Carlo dropout is the significantly increased computational cost since multiple forward passes are performed for each input image.

**Network Ensembles** Deep ensembles have been introduced for uncertainty estimation in image classification by Lakshminarayanan et al. [10]. In our implementation we employ an ensemble of  $N = 9$  network models for semantic segmentation. All ensemble members share the same architecture and training data and only differ in the random seed used for training, following the configuration for image classification in [10]. The random seed influences the initialization of the network weights as well as the shuffling of the training images. The predictions of the ensemble members are combined to  $P_{final}$  by class-wise averaging, equal to the combination of samples obtained by Monte-Carlo dropout. Consequently, the major drawback of this method is the significantly increased computational cost due to multiple forward passes being performed per image.

## 2.2 Margin-Entropy Loss

While sampling based uncertainty estimation methods like ensembles and Monte-Carlo dropout have shown decent results in recent publications, the need to process multiple network forward passes per image usually prohibits the deployment in autonomous vehicles due to the significantly increased runtime. In practice, an uncertainty estimation method is required that can be used to extend an existing network architecture without increasing the computational cost.

We employ the margin-entropy loss function that enforces a margin in predictive entropy between predictions on known classes and those on the unknown data. The margin-entropy loss function is defined as  $L_{me} = \max(m + \bar{H}_{id} - \bar{H}_{void}, 0)$ , where  $\bar{H}_{id}$  and  $\bar{H}_{void}$  are the average entropy of in-distribution and OOD samples, respectively. The hyperparameter  $m$  controls the margin between the two. In contrast to entropy maximization, the margin-entropy loss term  $L_{me}$  prevents overfitting on the OOD areas and limits negative effects on the performance on the original classification task. The margin-entropy loss is designed to punish high confidence values on out-of-distribution samples for image classification. The existing implementations of margin-entropy type loss functions require either a computationally expensive ensemble [15], or a separate dataset providing OOD samples for training [7]. In this work, we require neither of the two. Instead, we use Cityscapes labels as known classes while pixels that are unlabeled or ignored in Cityscapes images,  $\mathbf{X}_{void}$ , can be treated as OOD (shown in red in Fig. 2). The complete loss function used to train the network is the sum  $L = L_{ce} + \beta \cdot L_{me}$ , where  $L_{ce}$  is defined as the standard cross-entropy loss limited to the ID samples  $\mathbf{X}_{id}$  and  $\beta$  is a hyperparameter used to control the influence of the margin-entropy loss.

Note that the additional margin-entropy loss does not induce any changes to network architecture or required data. It can readily be applied to any given training pipeline and, thus, does not impact the network runtime in deployment.



Figure 2: Example for unused image areas in the Cityscapes dataset. Unused areas are highlighted in red.<sup>1</sup>

## 3 Experiments

In order to show the potential of our approach, we evaluate it experimentally in comparison to the presented baseline methods. For the first experiment, we construct a meaningful benchmark for Out-of-Distribution Detection. Secondly, we compare the networks' performance in Self-Assessment. All experiments in this work use the same FCN-based [12] architecture for semantic segmentation with a GoogLeNet [14] feature extractor trained with equal hyperparameter settings.

### 3.1 Out-of-Distribution Detection

Evaluation of out-of-distribution detection poses a severe challenge since well-defined unknowns are needed. In many works, it is customary to use out-of-distribution test data from a source other than the one the network was trained with. In segmentation, however, images are required that resemble the training data and comprise both, known and unknown classes. Unfortunately, Cityscapes void labels are not guaranteed to not contain known classes. This raises the need of *known unknowns* for evaluation. As a solution, we leave out well-defined classes in the dataset during training which will then be used as *known unknown* OOD samples during testing.

For evaluation, we employ the Receiver Operating Characteristic (ROC), which is commonly used to evaluate out-of-distribution detectors in recent literature [11]. The ROC describes the relative tradeoff between *true positive rate*  $TPR = TP/(TP + FN)$  and *false positive rate*  $FPR = FP/(FP + TN)$  of binary classifiers [2]. In the out-of-distribution experiment, in-distribution samples are labeled as positive, while out-of-distribution samples are labeled as negative. We use two metrics that summarize the ROC performance of a classifier:

- **AUROC:** The ROC curve of a continuous score classifier can be summarized in a single scalar by calculating the Area under the Receiver Operating Characteristic (AUROC). A perfect detector corresponds to an AUROC score of 1.0 [3].
- **FPR@0.95TPR:** The false positive rate at 0.95 true positive rate describes the probability that a negative sample is classified as positive when the TPR equals 0.95 [11].

<sup>1</sup>Note that this is an extreme example for unused regions.

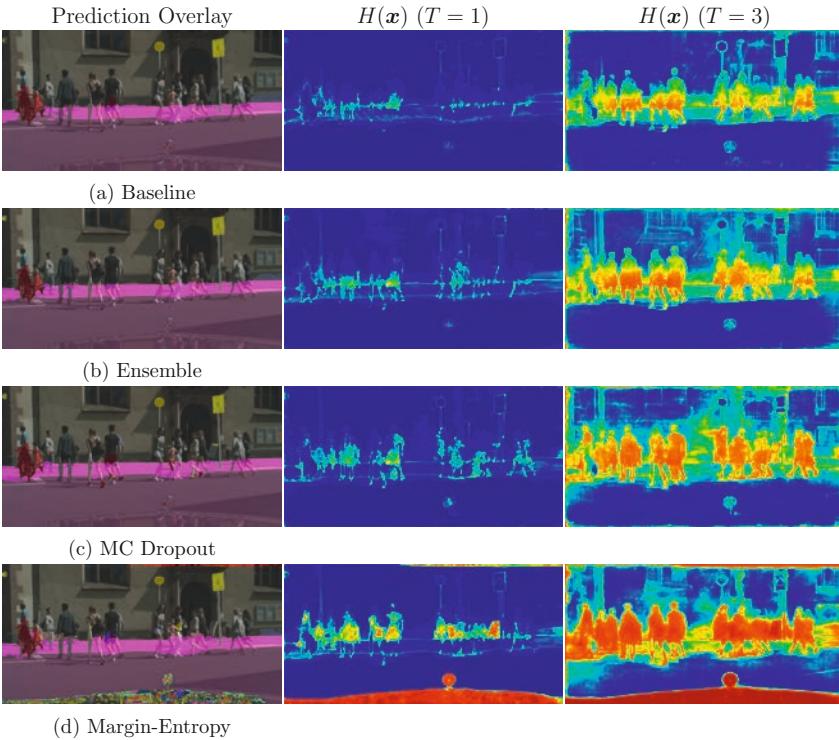


Figure 3: Uncertainty visualizations for OOD detection using the classes *pedestrian* and *rider* as out-of-distribution samples for evaluation. Blue indicates low uncertainty while red indicates high uncertainty.

The OOD detection performance is visualized for an example scene in Figure 3. In this experiment, the network is trained on the Cityscapes [1] training set, treating *pedestrian* and *rider* as *known unknowns* for evaluation. These classes are therefore not used in any of the loss calculations. Figure 3 shows the image with the model prediction as overlay (left), the predictive entropy as a heatmap (middle) and the same with temperature scaling applied (right).

The confidences of the baseline network on the unknown class *pedestrian* are nearly indistinguishable from the known classes, comparable to Figure 1. The OOD detection is slightly enhanced using ensembles and Monte-Carlo dropout. Using the proposed margin-entropy loss, the pedestrians show significantly lower confidence than the known classes, increasing OOD detection performance. The margin-entropy loss additionally causes low confidence levels on image areas such as the hood of the ego vehicle and the rectification border since these areas are ignored in Cityscapes. This is a desirable result since they cannot be classified correctly with the available set of classes. Temperature scaling

notably decreases confidence on pixels showing the OOD class *pedestrian* in all of the presented methods including the baseline model. Combined with the introduced methods for uncertainty estimation, the pedestrians show low confidence, clearly separating them from the known classes in the image. Using the margin-entropy loss, this effect is most pronounced.

Table 1 shows the quantitative results of this OOD detection experiment evaluated on the Cityscapes [1] validation set, again with persons treated as *known unknowns* for evaluation. The columns beneath each OOD detection metric show the results for using either the maximum predicted confidence or the negative predictive entropy as confidence scores. Without temperature scaling, the margin-entropy loss achieves the best OOD detection results in terms of FPR@0.95TPR. Combined with temperature scaling, the detection rates are significantly improved for all methods. While the sampling based methods combined with temperature scaling achieve higher detection rates than margin-entropy in this setting, the results of margin-entropy are comparable with only requiring a single forward pass. Using the predictive entropy as uncertainty score yields noticeably better results in this experiment when compared to the predicted confidence.

Configuration	FPR@0.95TPR in % ( $\downarrow$ )		AUROC in % ( $\uparrow$ )		$mIoU$ in % ( $\uparrow$ )
	Conf.	Entr.	Conf.	Entr.	
Baseline	57.5	49.7	89.7	90.8	74.0
Ensemble	53.3	39.7	91.3	92.6	76.4
MC Dropout	51.1	37.1	89.9	93.5	72.9
Margin-Entropy	46.6	34.0	92.0	93.2	71.7
Baseline + T	37.1	30.7	93.1	94.2	74.0
Ensemble + T	32.4	24.8	94.2	95.3	76.4
MC Dropout + T	26.7	18.7	95.3	96.4	72.9
Margin-Entropy + T	28.3	26.0	94.5	94.9	71.7

Table 1: Results of OOD detection experiment on the Cityscapes [1] validation set using *pedestrian* and *rider* as out-of-distribution classes. The subsequent +T denotes the use of temperature scaling.  $\uparrow$  indicates larger value is better and  $\downarrow$  indicates lower value is better.

The effects of temperature scaling are further analyzed in Figure 4. To measure the influence of temperature scaling on the calibration of the network confidences we additionally use the Expected Calibration Error (ECE) [5]. The ECE for  $n$  samples is defined as

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|. \quad (2)$$

The ECE expresses the weighted average difference between predicted confidence and achieved accuracy, discretized over  $M$  confidence intervals with their respective set of predictions  $B_m$ . Figure 4 (a) highlights the effects of temperature scaling on the calibration of the network predictions. When tuned for optimum OOD detection performance, the calibration of the network deteriorates significantly. At the optimum OOD detection temperature of  $T = 2.6$ , the network is strongly under-confident as visualized in the reliability curve in Figure 4 (b). Thus, the usage of temperature scaling depends on the calibration requirements of the specific usecase.

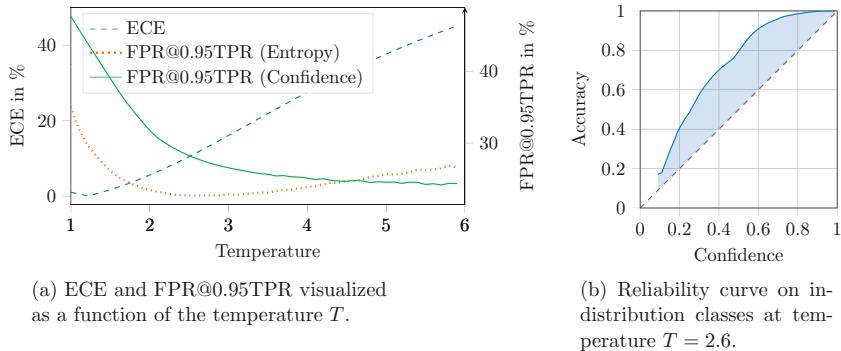


Figure 4: Temperature scaling effects on OOD detection and calibration using *pedestrian* and *rider* as out-of-distribution classes. Charts show the baseline model.

### 3.2 Network Self-Assessment

Out-of-distribution input data is not the only source of failure for classification systems. Similar to humans, classification systems based on neural networks sometimes fail to predict the correct class even for input samples that are close to the distribution of the training data. We evaluate the quality of the confidence score with regard to network self-assessment in two ways. First, the confidence score is used in a binary classification setting. In contrast to the OOD experiment, positive samples are now defined as correctly classified samples, while negative samples are defined as incorrectly classified. Second, we evaluate the calibration of the confidence scores. A network is perfectly calibrated if the accuracy of the predictions is equal to their respective confidences [5]. This property is crucial if the classification system is used for sensor fusion in conjunction with other perception systems. The confidence needs to be a meaningful, interpretable measure to be able to understand and compare predictions.

The results of the self-assessment experiments are summarized in Table 2. In addition to the binary classification metrics and the expected calibration error, we report the average predicted confidence for misclassifications  $\bar{p}_{\text{err}}$ . Contrary to the out-of-distribution detection experiments, temperature scaling showed no positive influence on network self-assessment and is not included in Table 2. The unchanged predicted softmax confidence of the baseline model is a competitive baseline for network self-assessment with an AUROC of 93.8 %. This score expresses a performance which significantly exceeds that of a random classifier. In general, capturing statistics about predicted confidences of correct and incorrect classifications is surprisingly effective for detecting whether an example is classified correctly, even though the prediction probability itself can be deceiving with  $\bar{p}_{\text{err}} = 0.747$ . This supports the results for image classification by Hendrycks & Gimpel [6].

The network ensemble outperforms the other approaches in terms of AUROC and FPR@0.95TPR. There is a correlation visible between these metrics and the ECE which can intuitively be explained by the similar objectives measured by both metrics. In contrast to the OOD detection experiments, there is no clear benefit observable for choosing either the predicted confidence or the predictive entropy as uncertainty quantification.

The margin-entropy loss introduces no deterioration in this task compared the baseline, indicating that the uncertainty related to out-of-distribution samples is fundamentally different when compared to misclassifications.

Configuration	FPR@0.95TPR ( $\downarrow$ ) in %	AUROC ( $\uparrow$ ) in %	$\bar{p}_{\text{err}}$	ECE ( $\downarrow$ ) in %	$mIoU$ ( $\uparrow$ ) in %
	Conf.	Entr.	Conf.	Entr.	
Baseline	32.8	32.9	93.8	94.1	0.747 1.47 74.3
Ensemble	28.7	29.2	95.2	95.1	0.692 0.44 77.4
MC Dropout	32.8	33.5	94.7	94.7	0.697 0.78 72.4
Margin-Entropy	33.9	34.7	94.2	94.3	0.716 1.19 72.3

Table 2: Results of network self-assessment experiments on the Cityscapes [1] validation set.  $\uparrow$  indicates larger value is better and  $\downarrow$  indicates lower value is better.

The performance of the network ensemble is visualized in comparison to the baseline model in Figure 5. A large area of the front of the truck is misclassified as *traffic sign* and *passenger car*. The ensemble (b) exhibits significantly lower confidence for the misclassified areas compared to the baseline model (a) while at the same time reducing the misclassified area.

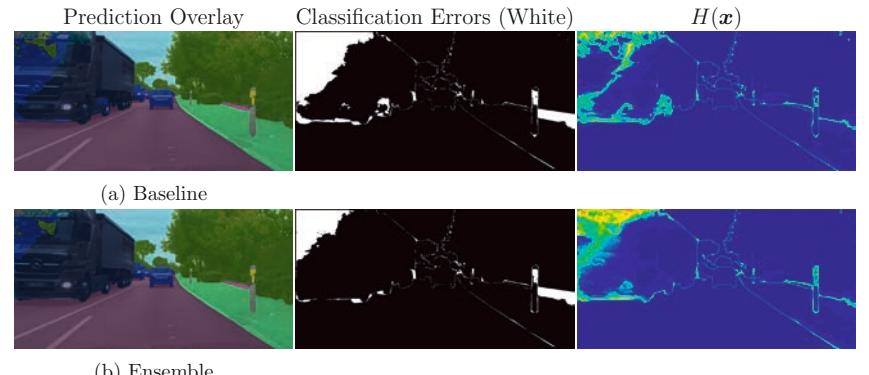


Figure 5: Example scene for misclassification detection

## 4 Conclusion

In this work, the uncertainty estimation of neural networks for semantic segmentation was analyzed. We provide an evaluation methodology for out-of-distribution detection and network self-assessment and compared multiple approaches experimentally. The results of the baseline out-of-distribution detection experiments in this work highlight the need for uncertainty estimation in black-box perception systems like neural networks. Recent research on uncertainty estimation in neural networks has shown the effectiveness of

sampling based methods like network ensembles [10] and Monte-Carlo dropout [9]. This, however, increases computational demands, potentially beyond real-time requirements. We, instead, propose a simple yet effective training strategy that can enable any given architecture to detect out-of-distribution data at similar detection accuracy. This enables existing systems to be extended by uncertainty estimation of which the downstream processing can benefit greatly.

### Acknowledgements

We would like to thank Maren Henzel and Prof. Dr. rer. nat. Hermann Winner from the Institute of Automotive Engineering (FZD) at the Technical University of Darmstadt for their support and supervision of this work.

## References

- [1] Marius Cordts et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [2] James P Egan. *Signal detection theory and ROC-analysis*. English. Includes bibliographies and index. New York : Academic Press, 1975.
- [3] Tom Fawcett. “An introduction to ROC analysis”. In: *Pattern recognition letters* 27.8 (2006).
- [4] Yarin Gal and Zoubin Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Proc. of the International Conference on Machine Learning (ICML)*. 2016.
- [5] Chuan Guo et al. “On Calibration of Modern Neural Networks”. In: *Proc. of the International Conference on Machine Learning, (ICML)*. 2017.
- [6] Dan Hendrycks and Kevin Gimpel. “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks”. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2017.
- [7] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. “Deep Anomaly Detection with Outlier Exposure”. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2019.
- [8] Geoffrey E Hinton et al. “Improving neural networks by preventing co-adaptation of feature detectors”. In: *arXiv preprint arXiv:1207.0580* (2012).
- [9] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. “Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding”. In: *Proc. of the British Machine Vision Conference (BMVC)*. 2017.
- [10] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles”. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*. 2017.

- [11] Shiyu Liang, Yixuan Li, and R. Srikanth. “Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks”. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2018.
- [12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.
- [13] Markus Maurer et al. *Autonomes Fahren: technische, rechtliche und gesellschaftliche Aspekte*. Springer, 2015.
- [14] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.
- [15] Apoorv Vyas et al. “Out-of-Distribution Detection Using an Ensemble of Self Supervised Leave-Out Classifiers”. In: *Proc. of the European Conference on Computer Vision (ECCV)*. 2018.

# A machine learning approach for ultrasonic noise classification and suppression

Mohamed-Elamir Mohamed<sup>\*</sup>, Heinrich Gotzig<sup>†</sup>, Raoul Zoellner<sup>‡</sup>  
and Patrick Maeder<sup>§</sup>

**Zusammenfassung:** In this paper we present a novel approach for using industrial grade ultrasonic sensors to perform echolocation by detecting ultrasonic echoes in a noisy environment using machine learning. A 2-step approach is presented starting by signal classification and followed by noise suppression. We show how this methodology is robust against the influence of ultrasonic noise sources in the environment as well as undesired reflections coming from the terrain. Several noise sources and noise powers are assessed as well as different terrain types. The results are bench-marked against the state of art energy thresholding, matched filters correlation algorithms and noise suppression methodologies, clearly showing the superiority of the machine learning based approach.

**Schlüsselwörter:** machine learning, artificial intelligence, ultrasonic and noise

## 1 Introduction

Ultrasonic sensors are commonly used in the automotive industry for obstacle detection and environment perception. Ultrasonic-based systems are usually comprised of several sensors distributed around the vehicle. These sensors fire ultrasonic waves and report the time of flight (TOF) between these firings and the respective echo reception to calculate the relative distance between the sensor and the obstacle. The TOF information from several sensors is then processed simultaneously using a form of triangulation to determine the exact position of the obstacle in a two dimensional map centered around the vehicle. This map is then used for several higher-end functionalities such as reporting these distances to the end user, automatic parking and braking on obstacles.

The quality of the ultrasonic signal plays an important role in the ability of the whole system to report correct information to the end user and to perform the higher-level driver assistance functionalities. In real life operation, there are several factors affecting the ability of the system to correctly identify the TOF of the ultrasonic reflected echo such as the presence of ultrasonic noise in the environment, the presence of other vehicles equipped with ultrasonic sensors in the vicinity of the vehicle and ground reflection from uneven terrain in the field of view of the sensor.

---

<sup>\*</sup>Valeo expert in artificial intelligence (email: mohamed-elamir.mohamed@valeo.com).

<sup>†</sup>Valeo master expert, Driving Assistance Product Group (email:heinrich.gotzig@valeo.com).

<sup>‡</sup>Prorektor Forschung, Transfer, Innovation, HS Heilbronn (e-mail: raoul.zoellner@hs-heilbronn.de).

<sup>§</sup>Prof. fuer Softwaretechnik fuer sicherheitskritische Systeme, TU Ilmenau (e-mail: patrick.maeder@tu-ilmenau.de).

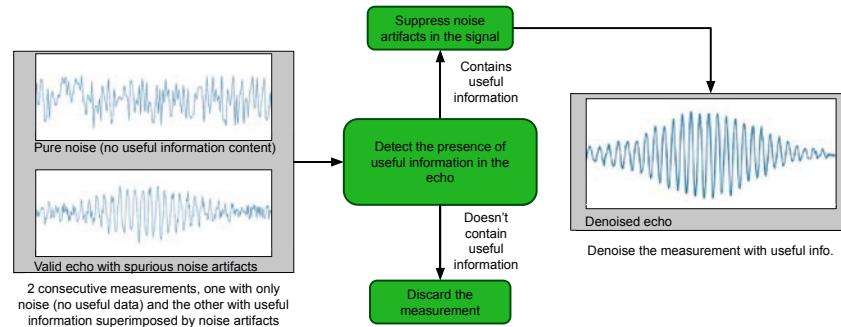


Abbildung 1: The 2-step machine learning based approach of filtering the noise signals with no actual content and further process the valid echoes to suppress spurious noise artifacts

In this work we present a 2-step approach using machine learning to process ultrasonic signals; figure 1. The first step identifies the presence of valid echoes as compared to signals with only noise artifacts [23]. In the second step, the signal with the valid echo is further processed to suppress any spurious disturbance artifacts originating from external noise sources.

## 2 State of the art

In this section we present a review of the existing methods for noise classification and noise suppression that fit the purpose at hand and also fit the nature of the ultrasonic signal generated by automotive-grade ultrasonic sensors.

### 2.1 Echo-Noise classification

Ultrasonic sensors are employed in many fields including biomedical imaging, structural integrity analysis and echolocation which will be the focus of this paper. Echolocation is used in different domains such as biomedical devices [1] where ultrasonic sensors are used for echolocation with acoustical feedback for aiding auditory challenged individuals. Robotics is another field where ultrasonic sensors are used for echolocation such as in [2] where ultrasonic sensors are used for robot localization and obstacles detection in the robot vicinity and [3] where ultrasonic sensors are used to detect walls and algorithms are used on top of the echolocation information to estimate the normal distance to the detected walls for more precise localization. Echolocation precision is investigated further in [4] by mitigating the effect of variating temperature on ultrasonic waves propagation speed. In this paper we focus further on the echolocation uses of ultrasonic sensors for automotive industry.

In automotive industry the echolocation functionality of ultrasonic sensors is used for perception of the surrounding environment and this functionality is used by other applications to build advanced driver assistance systems (ADAS). Ultrasonic sensors based

ADAS are used for detection of relevant obstacles [5] and signaling the driver through a human machine interfaces (HMI) with visual or acoustical signals or both. Another use for such systems is to avoid lateral collision [6] through the detection of obstacles in the lateral blind zone of a driver. More advanced ADAS based on ultrasonic sensors perform complete automatic parking maneuvers with the vehicle through the complete awareness of the surrounding environment [7] and [8] whose performance could be validated through methods as the ones presented in [9]

Further improvement of the performance of echolocation using ultrasonic sensors can be achieved by employing artificial intelligence (AI) methodologies and specifically, artificial neural network (ANN) based machine learning (ML) approaches. A general approach not specific to ultrasonic sensors is presented in [10] which discusses the use of ANN for processing sensor signals. The use of ANNs in ultrasonic signals allows for new functionalities and shows improvements in several fields such as in construction [11]. However the focus of this paper is on automotive industry and one of the first usage of ANN in vehicles is [12] which presents one of the early efforts in the direction of autonomous driving using ANNs for image processing based on cameras mounted on the vehicle. ANNs are used with ultrasonic sensors to achieve better distance measurements [13], extract attributes of the detected obstacle such as posture and shape [14], and the classification of obstacles [15] through the fusion between ultrasonic sensors and infrared sensors.

### 2.2 Noise suppression

In recent years several denoising methodologies are found in commercial products such as noise cancellation headsets. This family of denoising algorithms perform active noise cancellation [16] to counteract spurious signals coming from external sources. They rely on the fact that the SOI is known and coming from a definite source such as an MP3 player. This is not applicable for automotive ultrasonic sensors because the signal is already mixed at the source and it is not known apriori, which artifacts belong to the SOI and which are spurious and should be suppressed. Therefore we have to rely on other methodologies to identify the spurious noise sources, such as assuming the noise artifacts have only a certain range of frequency components or assuming that the SOI follows a specific model and all other signal components that do not fit this model should be suppressed.

Some trivial methods exist in literature where an assumption is made that the noise is present in certain frequency bands and that the useful signal is present in different frequency bands. In these methods simple filtering mechanisms satisfy the requirement of suppressing the unwanted signal components. This assumption doesn't hold in our case because the signal of interest and the noise components spread of intersecting frequency ranges.

Other methods such as adaptive multistage noise suppression filters [17] [18] are more adequate to the problem at hand. Such filters are by nature complex and include a large number of computations in subsequent steps which also lead to increased runtime and failure to satisfy the realtime constraints.

A family of algorithms that require relatively lower processing power and perform denoising over the whole spectrum are discrete wavelet transform (DWT) based methods. The DWT approach described in [19] and [20] achieves good denoising results in terms of suppressing unwanted spurious signals and extracting the SOI. It comes though with the cost that the extracted SOI is distorted, since it is the result of reconstructing the

superimposed wavelets passing the designed noise thresholds. This is not suitable for the ultrasonic sensor signal because the SOI is further processed to extract features pertaining to the reflecting obstacle such as height and class of the obstacle which is then fed into the higher layers of the ADAS for end-user functionalities such as braking on obstacles and automatic parking. Thus, the need arises to have a denoising algorithm that achieves similar noise suppressing results while maintaining the integrity of the SOI.

In literature there exist several publications discussing the use of machine learning and specifically deep neural networks for signal processing. Here we are interested in the family of algorithms that deal with the separation of signal components. Some approaches focus on the extraction of a specific SOI and discard the rest while other approaches consider all the signal components to be of interest and separates the different components such as [21] and [22]. There, the voice of the singer and the background music are both of interest and the algorithm tries to separate them without compromising signal integrity. The problem with denoising ultrasonic sensor signals is more in the domain of the former family of algorithms, where we only care about the echo from the obstacle and discard all other signal components.

In the following section we present the proposed 2-step machine learning based approach. An ANN structure is presented as a solution for the first phase of the approach. A deep CNN structure is also presented to perform noise suppression and extract the SOI from an ultrasonic signal mixture containing an echo from an actual obstacle and different types of environment noise in the same spectrum as the echo signal.

### 3 Proposed machine learning approach

In this section we present the concept behind the 2-step approach for noise classification and suppression. Each step is discussed separately as the connection between them is straight forward where the decision from the first step dictates whether the signal will be further processed or discarded.

#### 3.1 Echo-Noise classification

The ANN employed is a multi layer perceptron (MLP) composed of 4 layers. MLP architecture is chosen because of its higher execution speed compared to other ANN architectures such as convolution neural networks or recurrent neural networks which makes it suitable for realtime inference operation on microcontrollers used in automotive industry. The input layer is composed of 700 neurons which is equivalent to the number of samples in each snapshot of the digitized ultrasonic analog signal input. The output layer is composed of 2 neurons representing a one-hot approach for representing the 2 states of detected echo and noise respectively. The ANN also includes 2 hidden layers with 500 neurons each. All the layers in the ANN are structured in a dense fully connected manner as presented in Fig. The activation function used is the hyperbolic tangent (tanh) function and statistical gradient descent (SGD) is used as an optimization algorithm. The number of total samples is divided into 80% training, 10% validation and 10% testing.

#### 3.2 Noise suppression

The main concept of the approach is that the whole measurement is fed into the CNN and the denoised signal is completely regenerated at the output layer of the network. The network is trained on noisy signals with different types and levels of noise using supervised learning. The label at the output layer for each training sample is the same version of the input signal but with no noise component and only the echo features are present.

The structure of the network is the classical hour glass shape with the difference being that it is purely convolutional with no dense layers. The network is compressed using downsampling and decompressed using upsampling as described in fig. The number of layers is optimized to be 7 layers and the number of activation maps is set to a maximum of 128 kernels at the most compressed layer located in the middle of the neural network. The hyperbolic tangent (tanh) is used as activation function to limit the maximum values of the firing of the neurons, especially at the output layer where an extreme value could lead to the occurrence of false positives and unexpected behaviour. The output and the input layers are of the same size as the number of samples present in one measurement from the sensor.

### 4 Measurement campaign and data collection

The employed data acquisition setup is based on the Valeo ultrasonic sensor adapted with an analog signal interface. The analog signal output is sampled using an analog to digital converter (ADC) and preprocessed before being used. The carrier frequency of the ultrasonic sensor used is 51.2 KHz. The sampling rate at which the signal is sampled is 500 KSample/S which is higher than the nyquist rate of 102.4 KSample/S to avoid loss of information due to sampling. A resolution of 12 bits per sample is used to minimize the effect of the quantization error. The carrier frequency has a predetermined value and contains no information of interest therefore it is removed and the baseband signal is extracted which is also referred to as the envelop signal. In order to obtain the envelop signal, the in-phase and quadrature components are extracted by multiplying the input signal with the carrier frequency and a 90 degree phase shifted version of it respectively. The result is then passed through a low path filter (LPF) to eliminate the double frequency artifacts and isolate the baseband component and then the resulting signals are added to get the overall envelope; fig 2.

The signal samples are then decimated at a number of samples satisfying the condition of being above the nyquist rate of the baseband signal. The signal is further scaled to a range of [0,1] and the scaling factor is applied equally to all the samples at hand. The recorded samples are then divided into 3 groups; training, validation and testing. The extracted key performance indicators (KPIs) are based on the testing set that are never introduced to the ANN in the training phase.

A measurement campaign is carried out that includes several types of obstacles such

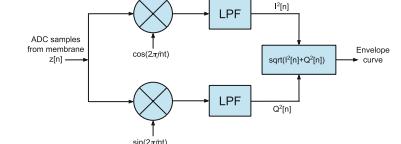


Abbildung 2: Pre-processing after performing data acquisition

as tubes, boxes, pedestrians, and objects of non uniform shapes. Measurements are also done under the influence of several noise sources such as truck brakes, clinging keys, air guns and rain.

## 5 Results and discussion

### 5.1 Echo-Noise classification

In order to assess the performance of the ML approach to detect echoes under noise conditions we use the F1 score for comparison purposes. The ANN is trained with samples from all the available SNR levels. In figure 3 the F1 score of the ML approach and the energy thresholding methodology with raw and correlated inputs at their respective optimum threshold levels are plotted against the same SNR range.

From figure 3 we see that the ML approach gives a superior performance as compared to the energy thresholding approaches especially under low SNR conditions.

The F1 score for the ML approach based on the same artificial neural network structure without the introduction of the different terrain samples in its training set is plotted in figure 4(a) against the optimum threshold levels for the thresholding methodology with both raw and correlated inputs for the gravel terrain.

From figure 4(a) it is evident that the ML approach shows some slight degradation in its capability of detecting the correct echo but remains superior to the thresholding methodology results at the same threshold and after re-tuning to their optimum thresholds under the modified terrain conditions.

From the results we see high dependence of the thresholding approach on the environmental conditions which states the need for re-tuning to the optimum threshold level under each set of conditions. Another approach is to use a universal threshold level but in this case we get a sub-optimum performance under all conditions. The ML approach shows an enhanced performance under different conditions without the need to re-tune the parameters. With prior knowledge of the 2 different terrain types and with the use of a universal optimal threshold for the thresholding method with raw and correlated input, the F1 score results against a dataset including the 2 terrain types are plotted together in figure 4(b).

Not only does figure 4(b) further highlight the superiority of the ML approach to the thresholding approaches but it also emphasizes the importance of augmenting the training set of the ML approach to further improve its capability of correctly detecting echoes in noisy conditions.

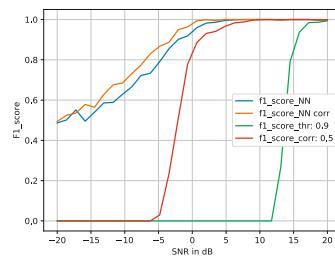


Abbildung 3: Results for echo/noise classification on smooth terrain

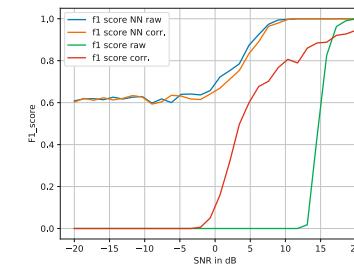


Abbildung 4: Results for echo/noise classification on gravel (ANN trained only for smooth terrain) and mixed terrain types

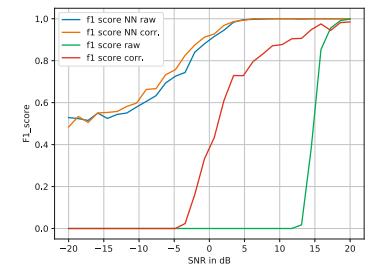
### 5.2 Noise suppression

To assess the performance of the ML-based noise suppression phase we bench mark it against the existing DWT noise suppression methodology.

In figure 5(a) we see the performance of both algorithms in terms of suppressing noise presented over a range of SNR values [-5, 20] dB. The DWT performs slightly better in this regard. Nevertheless the main advantage obtained from using the machine learning based approach is that it introduces much less distortion compared to DWT as will be demonstrated.

To adequately judge the performance of the algorithm we take the changes in signal energy not originally present in the SOI as a measure of distortion. This will include added artifacts as well as removed components from the original SOI. This approach will also prevent the trivial solution where the output of the algorithm is an array of zeroes where the noise is completely suppressed but the SOI is also lost. In figure 5(b) we show a comparison between the DWT based method and the machine learning based method in terms of the defined distortion metric. We see that even though the 2 algorithms have comparable performance in terms of noise suppression, the machine learning based approach outperforms the DWT method in terms of conserving the SOI integrity and in terms of the introduced distortions to the original shape of the SOI. We also see that at low SNR levels, the DWT distortion saturates at a ratio of 1.0, indicating that the SOI is completely suppressed and the algorithm is not capable of differentiating between the SOI and the noise components of the signal. In contrast, the machine learning approach is still capable of extracting the SOI from the noisy input signal.

It is important to mention that with further optimization of the hyper parameters of the legacy denoising algorithms we could achieve better results over the available trace set. Consequently the performance could improve but only to a certain limited extent. This is manifested in the trade-off between the noise suppression capability and the distortion introduced to the signal.



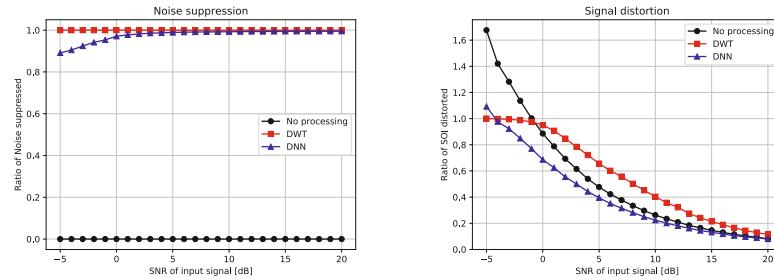


Abbildung 5: Results for noise suppression and distortion introduced by DWT and ML-based denoising algorithms

## 6 Conclusion

In this paper we presented some challenges facing the use of ultrasonic sensors in automotive industry such as the presence of external sources of noise and the changing terrain types affecting the functionality of the existing echolocation algorithms. To counteract these challenges we presented a 2-step approach for echo/noise classification and noise suppression. This includes MLP and CNN based ANN structures relying on ML techniques to overcome these challenges and provide a method to differentiate between echoes from real obstacles and external noise sources as well as suppressing noise artifacts. Furthermore, this methodology shows increased robustness against changing terrain types. The increase in performance is validated based on results collected from an extensive measurement campaign using recorded ultrasonic echoes from different obstacles as well employing different sources of noise and repeating the measurements of several terrains. The ML-based signal conditioning also proves to give comparable noise suppression compared to existing algorithms and is superior in terms of the levels of distortion introduced to the SOI.

The superior performance is explained by the fact that the ML based approach learns the typical shape of a SOI as well as the different noise patterns from the numerous measurements used to train the CNN, unlike the DWT algorithm which fits the SOI (the echo in this case) to the base wavelet shape.

By employing the proposed 2-step approach, the noisy measurements are eliminated directly without further processing which saves processing load and runtime. Furthermore, the measurements containing useful information are preserved and the integrity of the SOI is maintained while simultaneously suppressing the unwanted spurious noise artifacts. This leads to higher quality signal information and better functionality for the driver assistance systems relying upon these ultrasonic sensors.

## Literatur

- [1] J. Sohl-Dickstein, S. Teng, B. M. Gaub, C. C. Rodgers, C. Li, M. R. DeWeese and N. S. Harper, "A device for human ultrasonic echolocation," *IEEE Transactions on Biomedical Engineering*, vol. 62, pp. 1526–1534, 2015.
- [2] S. Kim and Y. Kim, "Robot localization using ultrasonic sensors," *Intelligent Robots and Systems, (IROS 2004)*, Proceedings of the 2004 IEEE/RSJ, vol. 4, pp. 3762–3766, 2004.
- [3] Nagashima, Yoshiaki and Yuta, Shinichi, 'Ultrasonic sensing for a mobile robot to recognize an environment-measuring the normal direction of walls,' *Intelligent Robots and Systems, Proceedings of the 1992 IEEE/RSJ*, vol. 2, pp. 805–812, 1992.
- [4] C. Canali, G. De Cicco, B. Morten, M. Prudenziati and A. Taroni, "A temperature compensated ultrasonic sensor operating in air for distance and proximity measurements," *IEEE Transactions on Industrial electronics*, num. 4, pp. 336–341, 1982.
- [5] M. Kodera, S. Akita and Y. Yoshino, "Rear obstacle detecting system for automotive vehicles," *The Journal of the Acoustical Society of America*, vol. 75, pp. 1312–1312, 1984.
- [6] K. T. Song, C. H. Chen and C. H. C. Huang, "Design and experimental study of an ultrasonic sensor system for lateral collision avoidance at low speeds," *Intelligent Vehicles Symposium*, pp. 647–652, 2004.
- [7] S. H. Jeong, C. G. Choi, J. N. Oh, P. J. Yoon, B. S. Kim, M. Kim and K. H. Lee, "Low cost design of parallel parking assist system based on an ultrasonic sensor," *International Journal of Automotive Technology*, vol. 11, pp. 409–416, 2010.
- [8] W. Wang, Y. Song, J. Zhang and H. Deng, "Automatic parking of vehicles: A review of literatures," *International Journal of Automotive Technology*, vol. 15, pp. 967–978, 2014.
- [9] Y. Lee and S. Chang, "Development of a verification method on ultrasonic-based perpendicular parking assist system," *The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014)*, pp. 1–3, 2014.
- [10] D. Masumoto, T. Kimoto and S. Nagata, "A sensory information processing system using neural networks," *IEEE International Conference on Neural Networks*, pp. 655–660, 1993.
- [11] S. Roy, F. K. Chang, S. J. Lee, P. Pollock and V. Janapati, "A novel machine-learning approach for structural state identification using ultrasonic guided waves," *Safety, Reliability, Risk and Life-Cycle Performance of Structures and Infrastructures*; CRC Press: Boca Raton, FL, USA, pp. 321–328, 2013.
- [12] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, pp. 305–313, 1989.

- [13] A. Carullo, F. Ferraris, S. Graziani, U. Grimaldi and M. Parvis, "Ultrasonic distance sensor improvement using a two-level neural-network," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, pp. 677–682, 1996.
- [14] K. Ohtani, M. Baba and T. Konishi, "Position and posture measurements and shape recognition of columnar objects using an ultrasonic sensor array and neural networks," *Systems and Computers in Japan*, vol. 33, pp. 27–38, 2002.
- [15] E. U. Warriach and C. Claudel, "A machine learning approach for vehicle classification using passive infrared and ultrasonic sensors," *Information Processing in Sensor Networks (IPSN)*, 2013 ACM/IEEE, pp. 333–334, 2013.
- [16] Elliott, Stephen J., and Philip A. Nelson. Äactive noise control. IEEE signal processing magazine 10, no. 4, 1993: 12-35.
- [17] Ikeda, Shigeji, and Akihiko Sugiyama. Än adaptive noise canceller with low signal distortion for speech codecs. IEEE Transactions on Signal Processing 47, no. 3, 1999: 665-674.
- [18] Qian, Wei, and Laurence P. Clarke. Äaptive multistage nonlinear filtering and wavelet for medical image enhancement. In *Proceedings of 1st International Conference on Image Processing*, vol. 3, pp. 711-715. IEEE, 1994.
- [19] Lang, Markus, Haitao Guo, Jan E. Odegard, C. Sidney Burrus, and Raymond O. Wells. NNoise reduction using an undecimated discrete wavelet transform. IEEE Signal Processing Letters 3, no. 1 (1996): 10-12.
- [20] Lang, Markus, Haitao Guo, Jan Erik Odegard, C. Sidney Burrus, and Raymond O. Wells Jr. NNonlinear processing of a shift-invariant discrete wavelet transform (DWT) for noise reduction. In *Wavelet Applications II*, vol. 2491, pp. 640-651. International Society for Optics and Photonics, 1995.
- [21] Chandna, Pritish, Marius Miron, Jordi Janer, and Emilia Gomez. MMonoaural audio source separation using deep convolutional neural networks. In *International conference on latent variable analysis and signal separation*, pp. 258-266. Springer, Cham, 2017.
- [22] Uhlich, Stefan, Marcello Porcu, Franck Giron, Michael Enenkl, Thomas Kemp, Naoya Takahashi, and Yuki Mitsuishi. Äproving music source separation based on deep neural networks through data augmentation and network blending. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 261-265. IEEE, 2017.
- [23] Mohamed, Mohamed-Elamir, Heinrich Gotzig, Raoul Zoellner, and Patrick Maeder. Ä Machine Learning Approach for Detecting Ultrasonic Echoes in Noisy Environments. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1-6. IEEE, 2019.

## Fahrstreifengenaue Kreuzungsschätzung anhand der Fahrtverläufe anderer Verkehrsteilnehmer

Annika Meyer\*, Jonas Walter\*, Martin Lauer<sup>†</sup> und Christoph Stiller<sup>†</sup>

**Zusammenfassung:** In diesem Beitrag wird ein Markov-Chain-Monte-Carlo-Ansatz vorgestellt, der ohne Kartenvorwissen sowohl das geometrische Layout einer Kreuzung als auch den genauen Verlauf der Fahrstreifen aus den Fahrtverläufen anderer Verkehrsteilnehmer schätzt. Das Modell ist flexibel gestaltet und nicht auf eine bestimmte Anzahl Arme oder Fahrstreifen beschränkt. Das Verfahren ist so konzipiert, dass laufend neue Messungen aufgenommen und genauso kontinuierlich Schätzungen abgegeben werden können. Der Ansatz wird auf 1 000 simulierten und 14 realen Kreuzungen evaluiert. Das Layout der Kreuzung kann dabei in unter 30 ms in mehr als 99 % der Fälle richtig geschätzt werden. Auf dieser Basis erreicht die Fahrstreifenschätzung in 100 ms einen Fehler von nur 14 cm. Nach den genannten Zeiten verbessert sich die Schätzung kontinuierlich durch Aufnahme neuer Messungen und Analyse weiterer Hypothesen.

**Schlüsselwörter:** Fahrstreifen, Kreuzungen, Kartenloses Fahren, Markov-Chain-Monte-Carlo

## 1 Einleitung

Heutige Systeme für automatisiertes Fahren verlassen sich größtenteils auf die Korrektheit einer hochgenauen Karte, um eine valide Verhaltensplanung zu ermöglichen. Hochgenaue Karten können dabei Wahrnehmungsaufgaben wie bspw. die Schätzung von Fahrstreifenrändern vereinfachen oder ersetzen. Durch Baustellen oder Verkehrsunfälle können sich diese Informationen jedoch ändern, sodass eine Karte veraltet. Der Aufwand, hochgenaue Karten aktuell zu halten, kann so schnell explodieren. Zusätzlich benötigen hochgenaue Karten eine hochpräzise Lokalisierung, um die gespeicherten Informationen adäquat nutzen zu können.

Bisherige Ansätze, die sich mit diesem Problem auseinandersetzen, konzentrieren sich primär auf die Fahrstreifenerkennung auf einfachen Straßen und Autobahnen [1, 2]. Wie viele andere betrachten Hou et al. [1] das Problem als eine pixelweise Segmentierung, wohingegen Homyounfar et al. [2] eine Polygon-Repräsentation wählen. Teilweise wird auch die Erkennung bzw. die Repräsentation eines Fahrstreifens übersprungen und direkt ein fahrbarer Pfad geschätzt [3, 4]. Dies hat den Vorteil, dass dynamische Objekte bereits beachtet werden, was bei einer reinen Fahrstreifenschätzung nicht der Fall ist. Eine Prädiktion andere Verkehrsteilnehmer und damit die genaue Verhaltensplanung auf Basis dieser generierten Pfade ist jedoch schwieriger, da das Straßenlayout nicht vollständig vorliegt oder hier wiederum auf eine Karte zurückgegriffen werden muss.

\*FZI Forschungszentrum Informatik, Karlsruhe (e-mail: ameyer@fzi.de)

<sup>†</sup>Institut für Mess- und Regelungstechnik, Karlsruhe Institut für Technologie (KIT), Karlsruhe

Kreuzungen sind im Vergleich zu einfachen Fahrstreifen komplexe Strukturen und wurden in bisherigen Arbeiten auf der Fahrstreifenebene kaum untersucht oder es wurden lediglich zeitaufwändige Ansätze vorgestellt.

Bestehende Ansätze befassen sich zum Beispiel mit dem Schätzen eines Verbindungsgraphen [5] oder der nicht-fahrstreifengenauen Geometrie. Geiger et al. [6] schränkt dabei das Modell der Kreuzung derartig ein, dass es kaum auf reale Kreuzungen generalisieren kann, da nur ein Fahrstreifen pro Fahrtrichtung modelliert wird. Ein geometrisch sehr flexibles Modell wird hingegen sowohl von Homayounfar et al. [2] als auch Liang et al. [7] vorgestellt, die die Ränder einer Kreuzung bzw. die Ränder von Autobahnfahrstreifen mit einem neuronalen Netz als Polygone präzidieren. Beide Ansätze können jedoch nicht die Fahrstreifen einer Kreuzung schätzen.

Andere Ansätze sind explizit für die Kartengenerierung entworfen und entsprechend zeitaufwändig. Roeth et al. [8] präsentieren dabei einen Ansatz, der aus Flottendaten Kreuzungs- und Fahrstreifengeometrien extrahiert. Die Modelle sind jedoch derartig konzipiert, dass eine Berechnung in Echtzeit nicht möglich ist. Die vorliegende Arbeit hingegen hat zum Ziel, einen echtzeitfähigen Ansatz vorzustellen, sodass einfache Modelle notwendig sind.

Wie einige andere Arbeiten in diesem Bereich [5, 6, 8] basiert die Schätzung hier auf den Fahrtverläufen anderer Verkehrsteilnehmer, die mit verschiedenen Sensoren gemessen werden können. Um mit der kombinatorischen Komplexität aller Kreuzungshypothesen umgehen zu können, wird das Markov-Chain-Monte-Carlo (MCMC) Verfahren verwendet, bei dem verschiedene Hypothesen untersucht und die beste Hypothese ermittelt werden kann. Ein Vorteil von MCMC ist die Möglichkeit, jederzeit Zwischenergebnisse bereitzustellen zu können. Die Besonderheiten dieser Arbeit sind dabei (i) eine fahrstreifengenaue Kreuzungsschätzung auf Basis des Fahrverhaltens anderer Fahrzeuge, (ii) progressive Aufnahme neuer Messungen und (iii) eine fortwährende und damit echtzeitfähige Schätzung.

## 2 Markov-Chain-Monte-Carlo-Ansatz

Um das optimale Kreuzungsmodell  $\hat{I}$  gegeben der gemessenen Fahrtverläufe  $Z$  zu schätzen, wenden wir ein Maximum-A-Posteriori-Verfahren an und erhalten

$$\begin{aligned} P(\hat{I}|Z) &= \arg \max_I P(I|Z) \\ &= \arg \max_I P(Z|I)P(I) \\ &= \arg \max_I \prod_i^{|Z|} P(z_i|I)P(I). \end{aligned} \quad (1)$$

Da das Modell hochdimensional ist und für die Verteilungen keine analytische Lösung existiert, kann nicht ohne Weiteres eine Wahrscheinlichkeit für jedes Modell berechnet werden. Gleichermaßen ist durch die Anzahl kombinatorischer Möglichkeiten der Parameter der Kreuzung eine Betrachtung aller Hypothesen sowie eine geschlossene Berechnung der Wahrscheinlichkeit nicht in begrenzter Zeit möglich. Daher eignet sich ein Monte-Carlo-Ansatz, der zufallsbasiert Hypothesen zur Untersuchung aus einer Markov-Kette zieht, das Markov-Chain-Monte-Carlo-Verfahren [9].

Die Hypothesen können dabei von einer Markov-Kette generiert werden, deren Verteilung  $q(x)$  proportional zu der tatsächlichen Verteilung  $p(x)$  ist. Für die generierten Hypothesen kann dann geprüft werden, ob sie der Zielverteilung entsprechen, indem die A-Posteriori-Wahrscheinlichkeit auf Basis der Messungen  $Z$  berechnet wird. Eine Hypothese  $I'$  wird somit als valides Sample aus der Verteilung  $p(x)$  akzeptiert, wenn sie im Bezug zum Vorgänger  $I$  die Akzeptanzbedingung

$$u \leq A(I', I) = \frac{P(I'|Z)q(I'|I)}{P(I|Z)q(I|I)} \quad (2)$$

erfüllt.  $u$  wird dabei aus einer Gleichverteilung  $u \leftarrow \mathcal{U}[0, 1]$  gezogen und neue Hypothesen durch Änderungen eines einzelnen Parameters generiert. Da wir in dieser Arbeit symmetrische Verteilungen annehmen und Abweichungen davon vernachlässigen (Metropolis-Algorithmus [10]), kann auf die Berechnung  $q(I'|I)$  verzichtet werden, sodass

$$\begin{aligned} u \leq A(I', I) &= \frac{P(I'|Z)}{P(I|Z)} \\ &= \frac{P(I') \prod_i^{|Z|} P(z_i|I')}{P(I) \prod_i^{|Z|} P(z_i|I)}. \end{aligned} \quad (3)$$

Durch den Einsatz eines Simulated Annealing Parameters  $T_s$  konvergieren die akzeptierten Hypothesen in ein optimales Kreuzungsmodell  $\hat{I}$ . Die Akzeptanzbedingung wird somit

$$u \leq A(I', I) = \left( \frac{P(I') \prod_i^{|Z|} P(z_i|I')}{P(I) \prod_i^{|Z|} P(z_i|I)} \right)^{\frac{1}{T_s}}. \quad (4)$$

## 3 Kreuzungsschätzung

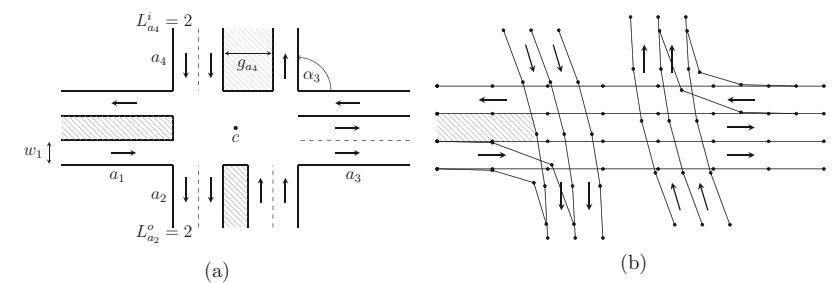


Abbildung 1: Zunächst werden die Parameter (a, Grobschätzung) und darauf aufbauend der genaue Fahrstreifenverlauf (b, Fahrstreifenschätzung) einer Kreuzung geschätzt.

Die Kreuzungsschätzung ist in zwei separate Schritte aufgeteilt. Im ersten Schritt werden nur die Parameter der Kreuzung geschätzt. Anschließend wird auf Basis dieser

Schätzung der eigentliche Fahrstreifenverlauf berechnet. In der Grobschätzung gehen wir zunächst davon aus, dass die Kreuzungsarme keine Krümmung aufweisen. In der Fahrstreifenschätzung wird diese Annahme dann verworfen und der tatsächliche, möglicherweise gekrümmte Verlauf bestimmt.

Für die Schätzung verwenden wir die Fahrtverläufe anderer Verkehrsteilnehmer  $Z$ , sodass jede Messung  $z$  aus der Position eines Objekts  $\mathbf{x}$  und dem Aufnahmzeitpunkt  $t$  besteht. Aus diesen Daten kann mit einer Objektverfolgung für jede Messung zusätzlich der Geschwindigkeitsvektor  $\mathbf{v}$  berechnet werden.

### 3.1 Grobschätzung

Für die Grobschätzung wird zunächst nur eine grobe Schätzung benötigt, sodass die Berechnung deutlich beschleunigt werden kann, indem nur zwei repräsentative Punkte  $\bar{z} = (\bar{\mathbf{x}}, \bar{\mathbf{v}})$  je Fahrtverlauf für die Schätzung verwendet werden. Dazu werden alle Trajektorien am Punkt der größten Krümmung in einen Teil vor und einen Teil nach der Kreuzung aufgeteilt und jeder dieser Teile auf einen Mittelpunkt  $\bar{\mathbf{x}}$  und einen mittleren Geschwindigkeitsvektor  $\bar{\mathbf{v}}$  reduziert.

Wie in Abbildung 1a veranschaulicht, besteht die Kreuzung  $I = (c, A)$  aus einem orientierten Zentrumspunkt  $c$ , um den wiederum eine Menge  $A$  aus Armen orientiert ist. Ein Arm wird dabei als  $a = (\alpha_a, g_a, L_a^i, L_a^o, w_a)$  modelliert und besteht dabei aus der Richtung  $\alpha_a$  eines Arms, der Breite der Lücke  $g_a$  zwischen den Fahrtrichtungen sowie der Anzahl Fahrstreifen, die in die Kreuzung hinein ( $L_a^i$ ) bzw. aus ihr heraus ( $L_a^o$ ) führen. Die einheitliche Breite der Fahrstreifen wird mit  $w_a$  bezeichnet.

In jedem Schritt des MCMC wird nur ein Parameter modifiziert (siehe Algorithmus 1). Über die Akzeptanzbedingung aus Gleichung 4 wird ermittelt, ob die neue Hypothese einem Sample aus der tatsächlichen Verteilung entspricht und somit sinnvoll für die weitere Hypothesengenerierung ist.

---

**Algorithmus 1** Neue Kreuzungshypothesen werden durch Änderung jeweils eines Parameters erzeugt.

```

 $\omega \leftarrow \mathcal{U}[0, 1]$ 
if  $\omega < 0.08$  then
    rotiere Arm um  $\Delta\alpha \leftarrow \mathcal{U}[-6^\circ; 6^\circ]$ 
else if  $\omega < 0.6$  then
    verschiebe Zentrum um  $\{\Delta c; \phi\} \leftarrow \mathcal{U}([0 \text{ m}; 6 \text{ m}] \times [0; 2\pi])$ 
else if  $\omega < 0.7$  then
    ändere den Abstand der Fahrstreifen um  $\Delta g \leftarrow \mathcal{U}[-1,8 \text{ m}; 1,8 \text{ m}]$ 
else if  $\omega < 0.85$  then
    füge Arm hinzu / entferne Arm  $a \leftarrow \mathcal{U}_D(A)$ 
else
    füge Fahrstreifen hinzu / entferne Fahrstreifen  $l \leftarrow \mathcal{U}_D(L)$ 
end if
```

---

Für den Vergleich zweier Kreuzungen auf Basis der Messungen  $\bar{Z}$  berechnen wir für jedes Modell  $I$  die Likelihood  $P(\bar{Z}|I)$  und die A-Priori-Wahrscheinlichkeit  $P(I)$ .

$P(I)$  hängt von der Anzahl der Fahrstreifen  $L_a^o$  bzw.  $L_a^i$  und Anzahl der Arme  $|A|$  ab und wurde empirisch bestimmt. Die Breite der Lücke  $g_a$ , der Winkel zwischen den Armen

$\alpha_a$  und der Zentrumspunkt  $c$  werden mit nicht-informativen A-Priori-Wahrscheinlichkeiten versehen. Somit ergibt sich

$$P(I) = P(|A|)P(c) \prod_{a \in A} P(L_a^o)P(L_a^i)P(\alpha_a)P(g_a). \quad (5)$$

Für die Likelihood  $P(\bar{z}|I)$  vergleichen wir jeden Trajektorienrepräsentanten  $\bar{z}$  mit der Zentrumslinie  $M_l$  der geschätzten Fahrstreifen, die zwischen dessen Rändern definiert ist. Dazu wird zunächst jede Trajektorie  $\bar{z}$  dem Fahrstreifen  $l$  zugeordnet, der die gleiche Fahrtrichtung wie  $\bar{\mathbf{v}}$  besitzt und die kürzeste orthogonale Distanz zu  $\bar{\mathbf{x}}$  aufweist. Die Likelihood selbst wird über die orthogonale Distanz  $d_\perp(\bar{\mathbf{x}}, M_l)$  sowie die Winkeldistanz  $d_\angle(\bar{\mathbf{v}}, M_l)$  berechnet, die jeweils als normalverteilt angenommen werden. Somit ergibt sich

$$P(\bar{Z}|I) = \prod_{\bar{z} \in \bar{Z}} \sum_{l \in L} P_\theta(\bar{z}, l)P_\perp(\bar{\mathbf{x}}|M_l)P_\angle(\bar{\mathbf{v}}|M_l), \quad (6)$$

wobei  $P_\theta(\bar{z}, l)$  die Wahrscheinlichkeit angibt, dass die Messung  $\bar{z}$  dem Fahrstreifen  $l$  zugeordnet wird. Da hier zur Reduktion der Rechenzeit jede Messung genau einem Fahrstreifen zugeordnet wird, nimmt diese nur 0 oder 1 an.

Aufgrund der Konstruktion des Algorithmus und der Reduktion auf einen einzelnen Repräsentanten pro Trajektorienstück lässt sich die Grobschätzung alternativ auch mit unverfolgten Objekten und reinen Detektionen, wie bspw. von einem Doppler-Radar, durchführen. Dabei fällt lediglich die Winkeldistanz  $d_\angle(\bar{\mathbf{v}}, M_l)$  als Teil der Likelihood weg. Der Vorteil dieser Alternative ist die Möglichkeit, auf sämtliche Vorverarbeitung im Sinne einer Objekterkennung und -verfolgung verzichten zu können, wenn lediglich das grobe Layout einer Kreuzung benötigt wird.

### 3.2 Fahrstreifenschätzung

Im zweiten Schritt wird die optimale, grob geschätzte Kreuzung  $\hat{I}$  verwendet, um genaue Fahrstreifen  $L$  zu schätzen. Jeder Fahrstreifen wird dazu mit einem rechten und linken Rand modelliert, der jeweils aus einem Polygonzug mit equidistanten Stützpunkten besteht (siehe Abbildung 1b). Hieraus ergibt sich zudem eine Zentrumslinie, die zwischen den Randpunkten verläuft.

---

**Algorithmus 2** Neue Fahrstreifenhypothesen werden durch Änderung jeweils eines Parameters erzeugt.

```

 $\omega \leftarrow \mathcal{U}[0, 1]$ 
if  $\omega < 0.67$  then
    verschiebe Zentrumslinie orthogonal zum Verlauf um  $\Delta m_l \leftarrow \mathcal{U}[-0,4 \text{ m}; 0,4 \text{ m}]$ 
else if  $\omega < 0.72$  then
    trenne Randpunkte; neuer Punkt  $\Delta b_l \leftarrow \mathcal{U}[-1,2 \text{ m}; 1,2 \text{ m}]$ 
else
    fusioniere zwei Randpunkte benachbarter Fahrstreifen zu einem Randpunkt
end if
```

---

Die Fahrstreifen werden mit Hilfe der Grobschätzung und der Trajektorien so initialisiert, dass sich alle Fahrstreifenpaare aus der Grobschätzung, die sich auf unterschiedlichen

Armen befinden und durch mindestens eine gemessene Trajektorie verbunden sind, auch durch einen Fahrstreifen innerhalb der Kreuzung verbunden werden. Anschließend wird, wie in der Grobschätzung, das Modell schrittweise mit den in Algorithmus 2 dargestellten Änderungsmöglichkeiten modifiziert.

Für die Bewertung der Fahrstreifen wird ebenfalls die A-Priori-Wahrscheinlichkeit  $P(L)$  und die Likelihood  $P(Z|L)$  für jede Trajektorie  $z$  berechnet. Hier werden jedoch im Gegensatz zur Grobschätzung alle Trajektorienpunkte  $z_i = (\mathbf{x}, \mathbf{v})$  verwendet.

Die A-Priori-Wahrscheinlichkeit  $P(L)$  wird aus der Anzahl der geteilten Randpunkte sowie der Glätte der Fahrstreifen als Summe der absoluten Winkelabweichungen zwischen aufeinanderfolgenden Zentrumspunkten berechnet.

Die Likelihood wird, wie bei der Grobschätzung, aus der orthogonalen Distanz  $d_{\perp}(\mathbf{x}_i, M_l)$  sowie der Winkeldistanz  $d_{\angle}(\mathbf{v}_i, M_l)$  zwischen Trajektorienpunkt  $z_i$  und dem nächsten Fahrstreifen  $l$  in gleicher Richtung bestimmt. Wir nehmen für beides eine Normalverteilung an und erhalten

$$P(Z|I) = \prod_{z_i \in Z} \sum_{l \in L} P_\theta(z, l) P_{\perp}(\mathbf{x}_i|M_l) P_{\angle}(\mathbf{v}_i|M_l). \quad (7)$$

Da hier ebenfalls mit  $P_\theta(z, l)$  jede Messung genau einem Fahrstreifen zugeordnet wird, nimmt diese nur 0 oder 1 an.

## 4 Evaluation

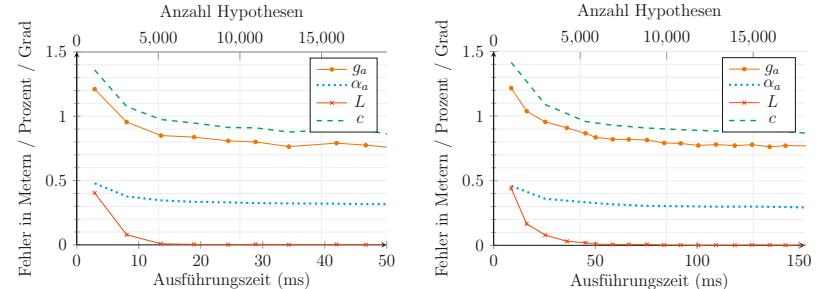
Für die Evaluation wurden 1000 zufällige Kreuzungen simuliert. Dabei wurden die Parameter des Modells verwendet und Kreuzungen generiert, die zwischen drei und fünf Armen mit bis zu vier eingehenden und ausgehenden Fahrstreifen besitzen. Die Arme wurden dabei mit einem Abstand von mindestens  $45^\circ$  positioniert und der Abstand der Fahrstreifen kann bis zu 3 m betragen.

Zudem haben wir für 14 urbane Kreuzungen im Raum Karlsruhe Kartenmaterial erstellt. Diese wurden in zwei Kategorien aufgeteilt. Kreuzungen mit einem Fahrstreifen je Richtung werden als *kleine* und alle Übrigen als *große* Kreuzungen bezeichnet. Für sowohl synthetische als auch reale Kreuzungen wurden zufällige Fahrzeugtrajektorien simuliert, die die Zentrumslinien der Fahrstreifen befahren. Die einzelnen Messpunkte der Trajektorien wurden normalverteilt mit  $\Delta z_i \leftarrow \mathcal{N}(0 \text{m}; 1 \text{m})$  verrauscht.

Für die Evaluation wurde zudem die kontinuierliche Aufnahme von Messungen vernachlässigt und stattdessen sämtliche Messungen auf einmal in die Schätzung aufgenommen. Dies hat den Vorteil, dass die Gesamtschätzung besser beurteilt werden kann.

**Genauigkeit der Schätzung** Um vergleichbare Experimente durchzuführen, wurden die Berechnungen auf eine bestimmte Stichprobenanzahl begrenzt und somit Ergebnisse nach 10 000 Stichproben in der Grobschätzung und nach 20 000 Stichproben in der Fahrstreifenschätzung analysiert. Die Ausführungszeiten für unterschiedlichen Anzahl Hypothesen und deren Genauigkeit in der Grobschätzung ist in Abbildung 2 dargestellt.

Für die Grobschätzung (siehe Tabelle 1) wurde jeder Parameter einzeln untersucht und es konnte festgestellt werden, dass die Anzahl der Arme in jeder Kreuzung richtig geschätzt wird. Die Anzahl der Fahrstreifen ist in allen Kreuzungen bis auf zwei bzw.



(a) Eine Trajektorie pro Fahrstreifen.  
(b) Drei bis fünf Trajektorien pro Fahrstreifen.

Abbildung 2: Entwicklung der Fehler über Ausführungszeit für 1000 künstlich erzeugte Kreuzungen.

drei korrekt. Hier wurden im Fehlerfall Fahrstreifen zu viel geschätzt. Bis zum Zeitpunkt der Schätzung mit jeweils einer Trajektorie pro Fahrstreifen lief das Verfahren 28 ms. Bei der Hinzunahme von mehr Trajektorien erhöht sich die Ausführungszeit für eine Ad-Hoc-Schätzung, das heißt eine Schätzung ohne Vorwissen mit sämtlichen Messungen, auf 84 ms, wobei sich die Qualität der Schätzung nur um  $0,02^\circ$  im Winkel und 1 cm in der Breite der Lücke zwischen den Fahrstreifen sowie dem Zentrum verbessert. Dies zeigt, dass bereits mit wenigen Daten eine sehr gute Schätzung durchgeführt werden kann.

Zur Evaluation der genauen Fahrstreifen (siehe Tabelle 1) wurden die geschätzten Zentrumslinien mit denen der Kartendaten verglichen und hierfür die durchschnittliche, orthogonale Distanz  $E$  verwendet. 20 000 Hypothesen sind hier nur nötig, wenn die Fusion der Fahrstreifenränder von Bedeutung ist, da die Positionierung der Fahrstreifen nach wenigen Schritten konvergiert und lediglich die Fusion mehr Schritte benötigt.

Auf den 14 realen Kreuzungen können für beide Schätzungen vergleichbare Ergebnisse schneller berechnet werden, da diese deutlich weniger Fahrstreifen aufweisen (siehe Tabelle 1). Die Ergebnisse für drei reale Kreuzungen aus Karlsruhe zeigt Abbildung 3.

Tabelle 1: Genauigkeit der Grobschätzung ① und Fahrstreifenschätzung ②. Die Tabellen zeigen zudem den Vergleich zwischen Experimenten mit einer Messung pro Fahrstreifen und drei bis fünf Messungen. Ergebnisse werden angegeben als Ausführungszeit in Millisekunden, Winkelabweichung in Grad, Anzahl Kreuzungen mit fälschlicher Anzahl Fahrstreifen und Abweichung der Lücke sowie des Zentrums in Zentimetern.

①	Synth					②	Synth		Real Klein		Real Groß	
	ms	$\alpha$	$L$	$g$	$c$		ms	$E$	ms	$E$	ms	$E$
1 T	28	$0,3^\circ$	3	80	91	1 T	104	14	78	18	89	27
3-5 T	84	$0,3^\circ$	2	79	90	3-5 T	173	9	110	11	137	19

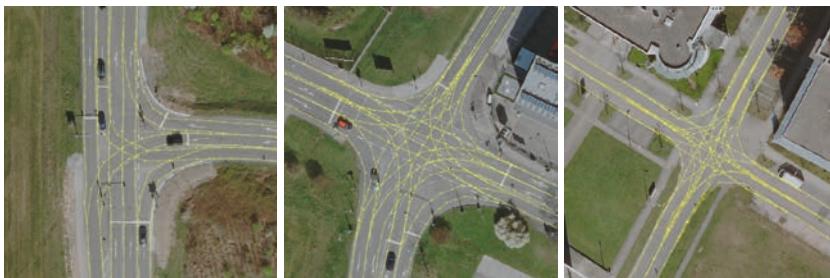


Abbildung 3: Ergebnisse der Schätzung von realen Kreuzungen in Karlsruhe<sup>1</sup>

**Abhängigkeit von der Anzahl an Messungen** Zur Evaluation der Robustheit des Ansatzes, haben wir untersucht, welche Änderungen sich zwischen vielbefahrenen Kreuzungen und weniger frequentierten Kreuzungen ergeben. Dazu wurde die Schätzung zunächst auf lediglich einer Trajektorie pro Fahrstreifen ausgeführt und zum Vergleich ein Experiment mit drei bis fünf Trajektorien pro Fahrstreifen als Messung in die Schätzung aufgenommen. Abbildung 2 veranschaulicht den unterschiedlichen Verlauf dieser Experimente für die Grobschätzung. Hier ist klar ersichtlich, dass zur Konvergenz zwar ähnlich viele Hypothesen generiert werden müssen, diese jedoch deutlich schneller berechnet werden können, wenn weniger Messungen vorhanden sind. Dabei erhöht sich die Verarbeitungszeit zwischen den beiden Experimenten von 3 µs pro Hypothese auf 8 µs. Die Qualität der Ergebnisse ändert sich jedoch nur marginal, solange nicht ein starkes Messrauschen oder sehr unterschiedliche Fahrweisen vorliegen, die sich mit mehr Messungen pro Fahrstreifen leichter ausgleichen lassen. In der anschließenden Fahrstreifenschätzung (siehe Tabelle 1) werden pro Hypothese 4 µs bzw. 9 µs benötigt, wobei sich die Ergebnisse durch mehr Trajektorien um wenige Zentimeter verbessern.

## 5 Diskussion

In dieser Arbeit wurde ein Ansatz vorgestellt, mit dem während der Fahrt das Kreuzungslayout sowie der Verlauf einzelner Fahrstreifen geschätzt werden können. In der Evaluation konnte gezeigt werden, dass selbst eine Ad-Hoc-Schätzung weniger als 180 ms benötigt.

Im Online-Einsatz nimmt die Kreuzungsschätzung, im Gegensatz zu dem in der Evaluation verwendeten Konzept der Einmal-Beobachtung, kontinuierlich Messungen auf. Dabei stehen zunächst nur wenige Trajektorien zur Verfügung, mit Hilfe derer jedoch bereits eine grobe Schätzung der Kreuzung erfolgen kann. Je mehr Fahrzeuge die Kreuzung befahren und somit je mehr Messungen in die Schätzung aufgenommen werden, desto stabiler kann eine valide Kreuzungshypothese gefunden werden. Dies bedeutet im Umkehrschluss, dass zwar für eine Schätzung einer Kreuzung mit bspw. einer Abweichung von lediglich 8 cm mehr als 100 ms benötigt werden, diese Zeit sich jedoch auf eine vollständige Neuschätzung bezieht. In der praktischen Anwendung des vorgestellten Ansatzes kom-

<sup>1</sup>Luftbilder: Stadt Karlsruhe, [www.karlsruhe.de](http://www.karlsruhe.de), dl-de/by-2-0

men kontinuierlich neue Messungen hinzu und eine neue Schätzung wird in regelmäßigen Abständen angestoßen, jedoch wird dabei die vorherige Modellschätzung beibehalten. Dadurch führen neue Messungen zwar zur Verlangsamung der Berechnung jedoch auch zur Verfeinerung der Hypothese. Eine initiale Schätzung ist bereits mit sehr wenigen Messungen möglich und reduziert somit bereits den nötigen zu untersuchenden Hypothesenraum für folgende Schätzungen.

Im Falle von ampelgesteuerten Kreuzungen kommt zudem hinzu, dass eine Rot-Phase Messungen in einem Zeitraum von mehreren Sekunden ermöglicht und somit eine äußerst präzise Schätzung erfolgen kann.

## Literatur

- [1] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, “Learning Lightweight Lane Detection CNNs by Self Attention Distillation,” in *The IEEE International Conference on Computer Vision (ICCV)*, pp. 1013–1021, 2019.
- [2] N. Homayounfar, W.-C. Ma, J. Liang, X. Wu, J. Fan, and R. Urtasun, “DAGMapper: Learning to Map by Discovering Lane Topology,” in *The IEEE International Conference on Computer Vision (ICCV)*, pp. 2911–2920, 2019.
- [3] D. Barnes, W. Maddern, and I. Posner, “Find Your Own Way: Weakly-Supervised Segmentation of Path Proposals for Urban Autonomy,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 203–210, 2017.
- [4] A. Amini, G. Rosman, S. Karaman, and D. Rus, “Variational End-to-End Navigation and Localization,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8958–8964, 2019.
- [5] O. Roeth, D. Zaum, and C. Brenner, “Road network reconstruction using reversible jump MCMC simulated annealing based on vehicle trajectories from fleet measurements,” in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 194–201, 2016.
- [6] A. Geiger *et al.*, “3D Traffic Scene Understanding from Movable Platforms,” *IEEE Trans. on Pattern Analysis and Mach. Intell.*, vol. 36, no. 5, pp. 1012–1025, 2014.
- [7] J. Liang, N. Homayounfar, W.-C. Ma, S. Wang, and R. Urtasun, “Convolutional Recurrent Network for Road Boundary Extraction,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9512–9521, 2019.
- [8] O. Roeth, D. Zaum, and C. Brenner, “Extracting Lane Geometry and Topology Information from Vehicle Fleet Trajectories in Complex Urban Scenarios Using a Reversible Jump MCMC Method,” in *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1-W1, pp. 51–58, 2017.
- [9] C. Andrieu, N. De Freitas, A. Doucet, *et al.*, “An Introduction to MCMC for Machine Learning,” *Machine learning*, vol. 50, no. 1-2, pp. 5–43, 2003.
- [10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, *et al.*, “Equation of State Calculations by Fast Computing Machines,” *The J. of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

# Verfahren zur Korrektur von inkonsistenten Lokalisierungsdaten in modularen technischen Systemen

Tobias Homolla<sup>\*†</sup>, Grischa Gottschalg<sup>\*‡</sup>, Hermann Winner<sup>§</sup>

**Zusammenfassung:** Automatisiert gesteuerte, mobile Systeme verfügen über eine planende und eine ausführende Instanz, die sowohl integriert als auch unabhängig voneinander vorliegen können. Beide benötigen Informationen über die aktuelle Pose des Systems. Um ein unerwünschtes Systemverhalten zu verhindern, ist es notwendig, dass beide Instanzen identische (konsistente) Informationen über die aktuelle Pose des Systems erhalten. Am Beispiel eines automatisiert gesteuerten Fahrzeugs wird ein Verfahren vorgeschlagen, das ermöglicht, für Planung und Regelung inkonsistente Lokalisierungsdaten zu verwenden. Dazu wird der Offset der ermittelten Posen bestimmt, überwacht und korrigiert. Außerdem werden die Effekte von Sprüngen der Posen im Rahmen der Sensordatenfusion auf die Bewegungsregelung unterdrückt und ein Beitrag zur Selbstwahrnehmung des Systems geleistet.

**Schlüsselwörter:** Automatisiertes Fahren, Lokalisierung, Sensordatenfusion, Regelung

## 1 Einführung

Automatisierte mobile Systeme (z.B. automatisierte Kraftfahrzeuge) sind von großem Forschungsinteresse und werden in zahlreichen Projekten in Wissenschaft und Industrie erforscht. Für die automatisierte Fahrzeugführung spielt unter anderem die Rückführungsstrategie der fahrdynamischen Zustandsgrößen im Rahmen der Bewegungsregelung eine wichtige Rolle [1].

### 1.1 Rückführungsstrategien

In vielen Projekten aus der Forschung und Industrie wird entsprechend der Klassifizierung aus [1] eine High-Level-Stabilisierung verwendet, wie bspw. beim Fahrzeug *BOSS*, Gewinner der *DARPA Urban Challenge 2007* [2]. High-Level-Stabilisierung bedeutet, dass die Trajektorie in jedem Berechnungsschritt der Planung neu berechnet wird, wohingegen bei

<sup>\*</sup>Die markierten Autoren sind Erstautoren mit gleichen Beiträgen.

<sup>†</sup>Tobias Homolla ist Wissenschaftlicher Mitarbeiter am Fachgebiet Fahrzeugtechnik der TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt, (e-mail: homolla@fzd.tu-darmstadt.de).

<sup>‡</sup>Grischa Gottschalg ist Wissenschaftlicher Mitarbeiter am Fachgebiet Physikalische Geodäsie und Satellitengeodäsie der TU Darmstadt, Franziska-Braun-Str. 7, 64287 Darmstadt (e-mail: gottschalg@psg.tu-darmstadt.de).

<sup>§</sup>Hermann Winner ist Professor und Leiter des Fachgebiets Fahrzeugtechnik der TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt (e-mail: winner@fzd.tu-darmstadt.de)

der Low-Level-Stabilisierung die Trajektorie fest im Raum liegt. Bi-Level-Stabilisierung basiert auf der Low-Level-Stabilisierung und berechnet nur bei Überschreitung von bestimmten Kriterien eine neue Trajektorie, so dass die Vorteile beider Rückführungsstrategien kombiniert werden können.[1]

### 1.2 Forschungsprojekt UNICARagil

Im Forschungsprojekt UNICARagil arbeitet ein Konsortium aus acht deutschen Universitäten und acht Industriepartnern an disruptiven, modularen Strukturen für agile, automatisierte Fahrzeugkonzepte [3]. Dabei sind einzelne Funktionen als abgeschlossene Software-Dienste mit definierten Schnittstellen konzipiert, die im Rahmen einer Systemarchitektur sinnvoll miteinander kombiniert werden müssen. Dies schließt den Entwurf einer Rückführungsstrategie der fahrdynamischen Zustandsgrößen mit ein.

### 1.3 Struktur des Beitrags

Bei der Verwendung von mehreren Lokalisierungsfunktionen in einem automatisierten Fahrzeug, wie es in UNICARagil der Fall ist, ergeben sich zahlreiche Möglichkeiten sowie Herausforderungen im Bezug auf die Gestaltung der Fahrzeugführung. In Kapitel 2 wird die in [3] für UNICARagil definierte Systemarchitektur erläutert, bevor in Kapitel 3 auf die mit dieser Architektur verbundenen Herausforderungen eingegangen wird. In Kapitel 4 werden überarbeitete Architekturen vorgestellt und mit Hilfe der definierten Anforderungen eine Lösung ausgewählt, die in Kapitel 5 erläutert wird. Abschließend folgt in Kapitel 6 ein Fazit und Ausblick.

## 2 Initiale Systemarchitektur

In [3] wird die zu Beginn des Projekts UNICARagil angedachte Systemarchitektur beschrieben, die als Basis für die weiteren Betrachtungen genutzt wird. Das Fahrzeug verfügt dabei über eine serviceorientierte Systemarchitektur mit drei makroskopischen Systemebenen: Großhirn, Stammhirn und Rückenmark. Im Großhirn findet die Umfeldwahrnehmung sowie die Verhaltens- und Trajektorienplanung statt. Im ausfallsicher gestalteten Stammhirn ist die Fahrdynamik- und Trajektorienregelung (FTR) und das sog. Sichere Anhalten verortet, während die Aktorik des Fahrzeugs dem Rückenmark zugeordnet ist. Durch die vorhandene Einzelradaktorik verfügt das Fahrzeug über den zusätzlichen, von der Bewegungsrichtung unabhängigen Freiheitsgrad des Gierwinkels.

Das Fahrzeug verfügt über zwei unabhängige Systeme zur Ermittlung der derzeitigen Pose (Position + Ausrichtung). Im Großhirn ist die kamera- und kartenbasierte Lokalisierung (KKLo) angesiedelt, die mittels eines Abgleichs von Landmarken mit einer digitalen Karte die Pose des Systems bestimmt. Im Stammhirn befindet sich die Fahrdynamikzustandsschätzung (FZS), die Informationen aus Navigationssatelliten, Odometriesensorik sowie einer inertialen Messeinheit nutzt, um ebenfalls eine Pose zu schätzen.

Durch die serviceorientierte Softwarearchitektur verfügt das System für die Fahrzeugführung über eine Wirkkette hintereinandergeschalteter Dienste. Aufbauend auf der Wahrnehmung des Umfelds durch entsprechende Sensorik wird das Verhalten des Fahrzeugs und anschließend eine Soll-Trajektorie geplant. Diese Soll-Trajektorie besteht aus

einer zeitgestempelten Abfolge von Posen und ihrer beiden zeitlichen Ableitungen und wird nur bei einer Änderung des dynamischen Umfelds, nicht jedoch bei einer Regelabweichung angepasst. Die nachgelagerte FTR führt einen Soll-Ist-Vergleich durch, für den die Soll-Trajektorie und der Ist-Zustand des Fahrzeugs aus der FZS verwendet werden, und generiert Stellbefehle für die Aktorik des Fahrzeugs. Bezogen auf die Bewegungsregelung des Fahrzeugs ist die Trajektorienplanung daher im Rahmen der initialen Architektur als Steuerung (Open-Loop) ausgelegt, die Aufgaben der Trajektorienplanung und -regelung sind entkoppelt. Jeder Schritt dieser Wirkkette ist latenzbehaftet. Daher besteht ein zeitlicher Verzug zwischen dem Messzeitpunkt der Wahrnehmung des Umfelds bis zur Umsetzung einer Reaktion durch die Aktorik.

Eine Besonderheit des UNICARagil-Konzepts stellt die Funktion Sicherer Anhalten dar. Dieser separate Betriebsmodus steht als Rückfallebene zur Verfügung, um bei einem Ausfall des Großhirns das Fahrzeug in einen sicheren Zustand zu überführen. Dazu wird von der Trajektorienplanung zusätzlich zur Soll-Trajektorie in jedem Zeitschritt eine Notbahn geplant, die aus einer Abfolge von Posen und Soll-Geschwindigkeiten besteht. Der Dienst Sicherer Anhalten überprüft diese Notbahn mithilfe separater Umfeldsensorik auf Belegung, reduziert wenn nötig die Geschwindigkeit und transformiert die Bahn damit in eine Trajektorie, die als Eingang für die FTR dient. Für die FTR ist daher keine Kenntnis des Betriebsmodus notwendig, da eine identische Schnittstelle vorliegt. Abbildung 1 stellt die relevante Systemarchitektur dar, wie sie in [3] vorgesehen ist.

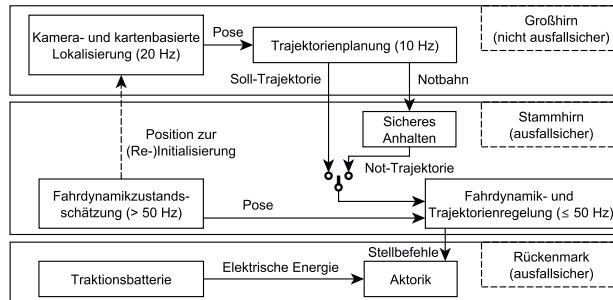


Abbildung 1: Wirkkette der Bewegungsregelung im Projekt UNICARagil (gemäß [3])

### 3 Problemstellung

Aus der in [3] vorgestellten Architektur ergeben sich mehrere Herausforderungen, die für eine Umsetzung zu bestehen sind. Zunächst wird die Abbildung 1 betrachtet. Die planerische und die ausführende Ebene greifen auf zwei unterschiedliche Lokalisierungsfunktionen zurück. Die Soll-Trajektorie wird auf Basis der Pose der KKLo in der Trajektorienplanung ermittelt, wohingegen die Ist-Werte des Fahrdynamikzustands für die FTR in der FZS berechnet werden. Innerhalb der FTR wird ein Soll-Ist-Vergleich durchgeführt. Inkonsistenzen in den Lokalisierungsinformationen der zwei Lokalisierungsfunktionen können daher zu scheinbaren Regelabweichungen führen, die nicht zu Regeleingriffen führen dürfen,

da sie nicht auf einer realen Abweichung zwischen den Soll- und Ist-Werten beruhen. Eine solche Inkonsistenz tritt bspw. durch eine Ungenauigkeit der Karte der KKLo [4] oder bei einem Drift in der Position der FZS aufgrund einer Abschattung des globalen Navigationssatellitensystems (GNSS) auf. Es ergibt sich daher die Notwendigkeit, die Systemarchitektur aus Abbildung 1 zu überarbeiten, um solche unerwünschten Regeleingriffe zu verhindern.

Außerdem ist in Abbildung 1 zu erkennen, dass die FTR mit einer Taktrate von bis zu 50 Hz arbeitet. Daraus ergibt sich als weitere Anforderung an eine neue Systemarchitektur, dass die für die Regelung benötigten Lokalisierungsinformationen mindestens mit der Taktrate der FTR zur Verfügung stehen. Des Weiteren folgt eine Anforderung aus den im vorherigen Abschnitt erläuterten Betriebsmodi in UNICARagil und der Annahme, dass nur Einzelfehler betrachtet werden. Die Fahrzeugführung muss sowohl im Regelbetrieb als auch bei Einzelfehlern im Modus Sicherer Anhalten sichergestellt werden. Bei der Überarbeitung der Systemarchitektur sind Schleifenbildungen in der Informationsverarbeitung zu vermeiden, da diese im Falle nicht berücksichtigter Korrelationen zu Fehlfunktionen bis hin zur Instabilität des Gesamtsystems führen können.<sup>1</sup> Im Projekt UNICARagil benötigt die KKLo für (Re-)Initialisierungen eine absolute Position, die durch die FZS zur Verfügung gestellt wird. Ein Beispiel für eine Schleifenbildung in der Informationsverarbeitung wäre eine Verarbeitung der Pose aus der KKLo im Filter der FZS, woraus eine Korrelation der beiden Posen folgt. Bei der anschließenden Verarbeitung der Posen aus der FZS in der KKLo zur (Re-)Initialisierung würde eine Schleife in der Informationsverarbeitung entstehen.

Zusammenfassend ergeben sich vier Anforderungen an die Systemarchitektur, die in Tabelle 1 dargestellt sind.

Tabelle 1: Anforderungen an die Systemarchitektur

Nr	Anforderung
A1	Regeleingriffe durch systematischen Offset der Ist-Posen der beiden Lokalisierungsfunktionen werden verhindert.
A2	Die benötigten Größen des Fahrdynamikzustands aus der Lokalisierung stehen mindestens mit der Datenrate der FTR zur Verfügung.
A3	Die automatisierte Fahrzeugführung ist sowohl im Regelbetrieb als auch im Betriebsmodus Sicherer Anhalten sichergestellt.
A4	Schleifenbildung in der Informationsverarbeitung wird vermieden.

### 4 Überarbeitung der Systemarchitektur

Im vorangegangenen Kapitel wurde aufgezeigt, weshalb die in [3] präsentierte Systemarchitektur nicht ohne Modifikation umgesetzt werden kann. Des Weiteren wurden Anforderungen abgeleitet, die bei der Wahl einer finalen Systemarchitektur zu erfüllen sind. Innerhalb dieses Kapitels werden zunächst verschiedene Lösungsvarianten (LV) für die Systemarchitektur des Fahrzeugs aufgezeigt. Dabei werden nur LV betrachtet, die sich

<sup>1</sup>Bei einer Schleifenbildung bzw. Rückkopplung ist eine positive Kopplung möglich. Diese führt zur Selbstverstärkung im System, welche die Instabilität des Regelkreises zur Folge haben kann [5, S. 56].

signifikant voneinander unterscheiden. Durch Abgleich mit den zuvor ermittelten Anforderungen findet anschließend eine Auswahl einer Systemarchitektur statt.

## 4.1 Mögliche Architekturen

### 4.1.1 High-Level-Stabilisierung

Die Bewegungsregelung kann als High-Level-Stabilisierung konzipiert werden (Abbildung 2A). Dabei werden die von der Trajektorienplanung bzw. dem Sicheren Anhalten berechneten Geschwindigkeits- und Beschleunigungsvorgaben in Abhängigkeit des Fahrzeugzustands ( $x(t)$ ) angepasst, was einer Posenregelung entspricht. Die unterlagerte FTR im Stammhirn ist als Geschwindigkeits- und Beschleunigungsregelung ausgelegt. Die Trajektorienplanung und die Posenregelung des Fahrzeugs finden in diesem Fall auf Basis der von der KKLo bestimmten Pose statt. Die Ausgangsgrößen der FZS werden ausschließlich zur Geschwindigkeits- und Beschleunigungsregelung innerhalb der FTR genutzt, um Stellgrößen für die Aktorik ( $u(t)$ ) zu generieren.

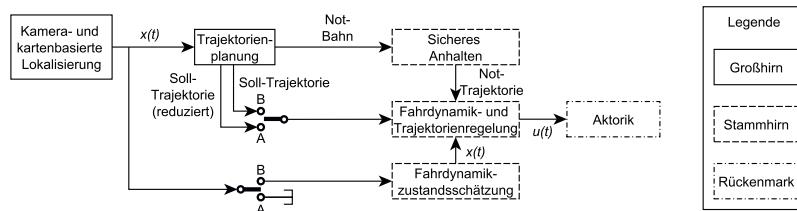


Abbildung 2: Systemarchitekturen für LV1 (A) und LV2 (B)

### 4.1.2 Low-Level-Stabilisierung mit Integration der KKLo-Pose in FZS-Filter

LV2 stellt die Einbeziehung der von der KKLo ermittelten Pose in das stochastische Filter der FZS dar. Die auf diese Weise zusätzlich verfügbare Pose wird als Stützinformation in der Sensordatenfusion der FZS verwendet. Durch eine entsprechende Gewichtung der Eingangsdaten können Inkonsistenzen zwischen den beiden Posen im System reduziert werden. Die resultierende Architektur entspricht Abbildung 2B. Im Unterschied zu LV1 liegt eine Low-Level-Stabilisierung vor, so dass die Posenregelung innerhalb der FTR stattfindet. Die Trajektorienplanung bzw. das Sichere Anhalten passt die Soll-Trajektorie in diesem Fall nicht in Abhängigkeit des Fahrzeugzustands an.

### 4.1.3 Low-Level-Stabilisierung mit Posen-Offsetkorrektur

LV3 ist eine Architektur mit Low-Level-Stabilisierung. Die Trajektorienplanung nutzt dabei die KKLo-Pose, die FTR die Pose der FZS (vgl. Abbildung 3). Um dabei die in Kapitel 3 dargelegten Probleme zu vermeiden, ist bei dieser Architektur ein zusätzlicher Berechnungsschritt im Form einer Offsetkorrektur notwendig. Dabei werden zunächst die Abweichungen zwischen den von den beiden Lokalisierungsfunktionen ermittelten Posen berechnet. Anschließend werden entweder die Ist-Posen der FZS (Variante A) oder die

in der Soll-Trajektorie bzw. Notbahn enthaltenen Posen (Variante B) um diesen Offset korrigiert.

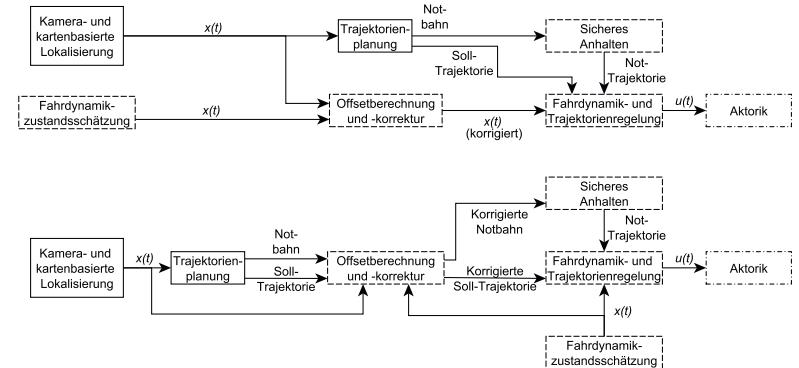


Abbildung 3: Systemarchitekturen für LV3 mit den beiden Varianten A (oben) und B (unten)

### 4.1.4 Einheitliche Lokalisierungs-Pipeline

LV4 umfasst eine Architektur mit Low-Level-Stabilisierung und den Aufbau einer gemeinsamen Lokalisierungs-Pipeline mit der gesamten vorhandenen Lokalisierungssensorik (KKLo und FZS). Dieser Ansatz ist in [6] beschrieben.<sup>2</sup> Die so ermittelte Pose wird sowohl für die Trajektorienplanung als auch für die FTR genutzt (vgl. Abbildung 4).

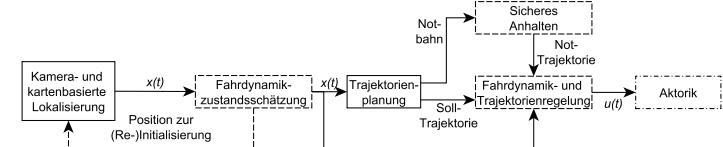


Abbildung 4: Systemarchitektur für LV4

### 4.1.5 Separater Posenregler im Großhirn

LV5 basiert auf einer Architektur mit Low-Level-Stabilisierung (vgl. Abbildung 5). Die Trajektorienplanung im Großhirn plant die Trajektorie auf Basis der KKLo-Pose. Die von der Trajektorienplanung unabhängige Posenregelung befindet sich dabei ebenfalls im Großhirn und nutzt für den Soll-Ist-Vergleich ebenfalls die KKLo-Pose. Die FTR im Stammhirn reduziert sich auf die Funktion einer Geschwindigkeits- und Beschleunigungsregelung.

<sup>2</sup>Bei dieser Lösung können die gegenseitigen Korrelationen berücksichtigt werden, da der Quellcode aller Teilfunktionen bei der Entwicklung der Pipeline zur Verfügung steht.

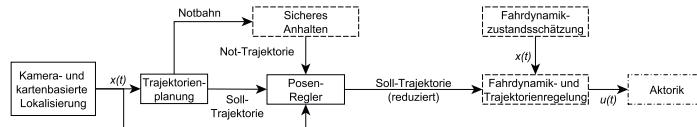


Abbildung 5: Systemarchitektur für LV5

#### 4.1.6 Low-Level-Stabilisierung mit Soll-Ist-Vergleich im Großhirn

LV6 nutzt ebenfalls eine Low-Level-Stabilisierung, verlagert jedoch den Soll-Ist-Vergleich für die Posenregelung in das Großhirn. Dabei wird ausschließlich die KKLo-Pose genutzt. Die Regelabweichung wird an das Stammhirn übertragen, in dem der eigentliche Posenregler angesiedelt ist. Diese Architektur bietet den Vorteil, dass keine Inkonsistenzen beim Soll-Ist-Vergleich der Posen entstehen kann, da ausschließlich die von der KKLo ermittelte Pose verwendet wird. Die Architektur entspricht LV5, wenn der Posenregler durch einen Dienst zur Bestimmung der Posenabweichung ersetzt wird und statt der Pose aus der FZS die KKLo Pose verwendet wird (siehe Abbildung 6).

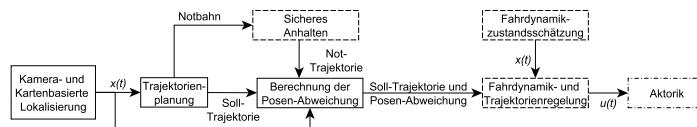


Abbildung 6: Systemarchitektur für LV6

## 4.2 Auswahl eines Verfahrens

Auf Basis der definierten Anforderungen und des aufgezeigten Lösungsraums wird nachfolgend die Entscheidung für eine Systemarchitektur getroffen.

LV1 steht im Widerspruch zu Anforderung A3, nach der die FTR in allen Betriebsmodi in der Lage sein muss, die Fahrzeugführung zu gewährleisten. Im Betriebsmodus Sicheres Anhalten muss als Worst-Case-Szenario von einem vollständigen Ausfall des Großhirns ausgegangen werden. Bei einer High-Level-Stabilisierung mit Posenregelung im Großhirn ist ein Abfahren der vorab berechneten Notbahn mit dem bestehenden Konzept für das Sichere Anhalten nicht möglich. Aus demselben Grund sind LV5 und LV6 zu verwerfen.

LV2 ist für die vorliegende Systemarchitektur nicht geeignet, da aufgrund der nicht bekannten Korrelationen die Gefahr von Schleifenbildungen (positive Kopplung) besteht<sup>3</sup> und eine Stabilität des Gesamtsystems daher u.U. nicht garantiert werden kann (vgl. Abschnitt 3). Dieselbe Gefahr besteht bei LV4, da die Position der FZS zur Initialisierung der KKLo benötigt wird. LV3 Variante B verhindert die rechtzeitige Unterdrückung

<sup>3</sup>Eine Berücksichtigung aller Korrelationen zwischen den Modulen ist nicht möglich, da die einzelnen Module von verschiedenen Partnern erarbeitet werden und der zugehörige Quellcode nicht vollständig zur Verfügung steht.

von Sprüngen der FZS im Betriebsmodus Sicheres Anhalten, da der Dienst Sicheres Anhalten mit einer niedrigeren Frequenz als die FTR ausgeführt wird und die korrigierte Trajektorie deshalb zeitverzögert an die FTR weiterleitet, was zu einer unerwünschten Regelabweichung führt. Dies entspricht einem Verstoß gegen Anforderung A3.

Aufgrund der nicht erfüllten Anforderungen der LV1-2, LV3B und LV4-6 stellt einzig die Low-Level-Stabilisierung mit Posen-Offsetkorrektur in Variante A (LV3A) eine geeignete Systemarchitektur dar und wird daher weiter verfolgt.

Tabelle 2: Übersicht der LV und der ermittelten Ausschlusskriterien

Nr.	Beschreibung	Verstoß gegen
LV1	High-Level-Stabilisierung	A3
LV2	Low-Level-Stabilisierung mit Einbezug der KKLo-Pose in FZS	A4
LV3A	Low-Level-Stabilisierung mit Posen-Offsetkorrektur	Keine
LV3B	Low-Level-Stabilisierung mit Posen-Offsetkorrektur	A3
LV4	Einheitliche Lokalisierungs-Pipeline	A4
LV5	Low-Level-Stabilisierung mit Posenregler im Großhirn	A3
LV6	Low-Level-Stabilisierung mit Soll-Ist-Vergleich im Großhirn	A3

## 5 Offsetkorrektur

Die im vorausgegangenen Kapitel ausgewählte Offsetkorrektur lässt sich in drei Unterfunktionen einteilen: die Grundfunktionalität zur Bestimmung des Posenoffsets und zur Umrechnung der Soll-Trajektorie aus der Trajektorienplanung und der Ist-Pose aus der FZS in ein gemeinsames Bezugssystem, eine Zusatzfunktion zur Vermeidung von unerwünschten Effekten von Sprüngen der Ist-Pose aus der FZS auf die FTR und eine Zusatzfunktion als Beitrag zur Selbstbeobachtung.

### 5.1 Funktionalität

#### 5.1.1 Aufbau

Wie in Abbildung 3 zu erkennen ist, wird die Offsetkorrektur zwischen der planerischen und der ausführenden Ebene verschaltet und besteht aus der Berechnung und der Korrektur des Offsets. Die Soll-Trajektorie wird auf Basis der Ist-Pose aus der KKLo geplant. Die  $j$ -te Soll-Trajektorie  $T_j$  enthält Soll-Posenabfolgen  $P_{j,i}$  mit Zeitstempeln  $t_{j,i}$ , deren zeitlichen Ableitungen  $\dot{P}_{j,i}$  und  $\ddot{P}_{j,i}$  sowie die Ist-Pose  $P_{j,U}$  aus der das Umfeld (U) betrachtenden KKLo, auf der die Trajektorie geplant wurde, mit dem Zeitstempel  $t_{j,U}$  an dem diese Pose gültig war:

$$T_j = \left\{ (t_{j,U}, P_{j,U}) ; (t_{j,i}, P_{j,i}, \dot{P}_{j,i}, \ddot{P}_{j,i})_{i=1,\dots,n} \right\} \quad (1)$$

Dabei steht  $P_j$  für eine kontinuierliche Posenabfolge und  $P_{j,i}$  für eine diskretisierte Posenabfolge (gilt ebenso für die Ableitungen). Die Ist-Pose  $P_{j,U}$  aus Gleichung (1) wird an die Offsetberechnung übermittelt.

Außerdem erhält die Offsetberechnung die Ist-Pose der FZS  $P_F(t_k)$  zum aktuellen Zeitpunkt  $t_k$ . Der bestimmte Offset wird zusammen mit den Größen  $P_F(t_k)$ ,  $\dot{P}_F(t_k)$ ,  $\ddot{P}_F(t_k)$  der FZS an die Offsetkorrektur gesendet. Die korrigierten Größen  $P_{F,korr}(t_k)$ ,  $\dot{P}_{F,korr}(t_k)$ ,  $\ddot{P}_{F,korr}(t_k)$  werden von der Offsetkorrektur an die FTR übermittelt. Diese berechnet daran zusammen mit der Soll-Trajektorie und den anderen Fahrdynamikgrößen aus der FZS die Stellgrößen für die Aktorik des Fahrzeugs.

### 5.1.2 Grundfunktionalität

Die Funktion der Offsetberechnung und -korrektur lässt sich folgendermaßen zusammenfassen: Die empfangenen Ist-Posen der FZS  $P_F(t_k)$  werden zusammen mit den zugehörigen Zeitstempeln  $t_k$  in einem Ringspeicher vorgehalten. Zunächst wird im Rahmen der Offsetberechnung ein korrespondierendes Posen-Paar aus Ist-Pose der FZS  $P_F(t_{j,U})$  und der Ist-Pose der KKLo  $P_{j,U}$ , auf deren Basis die aktuelle Soll-Trajektorie berechnet wurde, ermittelt. Im Regelfall existiert keine Ist-Pose der FZS  $P_F(t_k)$  mit  $t_k = t_{j,U}$  und es wird daher zwischen den zwei nächstgelegenen Zeitpunkten interpoliert. Anschließend wird die systematische Abweichung zwischen den beiden Posen („Offset“) bestimmt:

$$\Delta P_{j,U} = P_F(t_{j,U}) - P_{j,U} \quad (2)$$

Die Abweichung zwischen den beiden Posen  $\Delta P_{j,U}$  setzt sich aus der Abweichung in der Position und der Abweichung in der Orientierung zusammen, wobei die Matrix  $\Delta R_{j,U}$  die Rotation um die Abweichungen in der Orientierung beschreibt.

Nachfolgend werden im Rahmen der Offsetkorrektur die Ist-Posen  $P_F(t_k)$  um den ermittelten Offset der Posen sowie die Ist-Geschwindigkeiten  $\dot{P}_F(t_k)$  und Ist-Beschleunigungen  $\ddot{P}_F(t_k)$  um den Offset in der Orientierung korrigiert:

$$P_{F,korr}(t_k) = P_F(t_k) - \Delta P_{j,U} \quad (3)$$

$$\dot{P}_{F,korr}(t_k) = \Delta R_{j,U} \dot{P}_F(t_k) \quad (4)$$

$$\ddot{P}_{F,korr}(t_k) = \Delta R_{j,U} \ddot{P}_F(t_k) \quad (5)$$

Auf diese Art und Weise führen Abweichungen der beiden Lokalisierungsfunktionen nicht zu einer Regelabweichung und damit zu Stellbefehlen für die Aktorik. Des Weiteren wird die Konsistenz der Geschwindigkeits- und Beschleunigungsvorgaben sichergestellt.

### 5.1.3 Zusatzfunktionen

Wie zu Beginn des fünften Abschnitts aufgezeigt wurde, enthält die Offsetkorrektur zwei Zusatzfunktionalitäten.

Die erste Zusatzfunktion unterdrückt unerwünschte Effekte auf die FTR, die durch Sprünge in der Ist-Pose der FZS entstehen würden. Diese Sprünge in der Ist-Pose der FZS treten nur bei definierten Schritten im Verarbeitungssablauf, wie bspw. beim Messupdate im Kalman-Filter oder einer (Re-)Initialisierung des Filters auf. Bei diesen Verarbeitungsschritten gibt es neben dem Sprung in der Ist-Pose auch einen Sprung in der zugehörigen Standardabweichung. Daher können die Sprünge bei gleichzeitiger Betrachtung der Verläufe von Ist-Pose und dessen Standardabweichung erkannt werden. Um die rechtzeitige Erkennung der Sprünge vor einer Weitergabe an die FTR sicherzustellen, läuft die Offsetkorrektur mit der gleichen Datenrate wie die FTR. Wenn ein Sprung erkannt

wird, wird dessen Höhe in der Offsetkorrektur ermittelt und der Einfluss des Sprungs auf die FTR kompensiert. Dazu wird die Veränderung der Pose der vergangenen zwei Zeitschritte verwendet, um die erwartete Posenänderung aufgrund der Fahrzeugbewegung im aktuellen Zeitschritt zu prädizieren. Der verbleibende Anteil der Posenänderung wird auf den Sprung zurückgeführt und herausgerechnet.

Die zweite Zusatzfunktion leistet einen Beitrag zur Selbstwahrnehmung, indem zuvor definierte Kriterien überprüft werden. Diese können bspw. Grenzen für den Posen-Offset oder dessen zeitliche Änderung enthalten. Der aktuelle Posen-Offset sowie ggf. die Nichteinhaltung der Kriterien werden an die Selbstbeobachtung des Gesamtsystems zur Funktionsüberwachung gemeldet.

## 5.2 Programmablauf

Die Offsetberechnung und -korrektur läuft in mehreren Schritten ab, wobei die Offsetberechnung mit der Datenrate der Trajektorienplanung und die Offsetkorrektur mit der Datenrate der FTR läuft. Die Offsetberechnung wird demnach nur ausgeführt, wenn eine neue Soll-Trajektorie vorliegt, wohingegen die Offsetkorrektur immer vor einem Berechnungsschritt der FTR ausgeführt wird. Im ersten Schritt der Offsetkorrektur wird der berechnete Offset, wie in Abschnitt 5.1.2 dargelegt wird, mit den Größen aus der FZS verrechnet. Falls ein Sprung in der Ist-Pose der FZS erkannt wird, wird dieser im zweiten Schritt kompensiert. Im dritten Schritt wird die Überprüfung der definierten Kriterien durchgeführt (vgl. Abschnitt 5.1.3).

## 5.3 Implementierung und Testkonzept

Das vorgestellte Verfahren wird im weiteren Verlauf des Projekts UNICARagil im entwickelten Forschungsfahrzeug implementiert. Der erste Schritt der Implementierung ist eine Simulation als Proof-of-Concept. Als Testszenario wird die Durchfahrt eines Tunnels gewählt. Dieses Szenario ist für einen Proof-of-Concept geeignet, da im Tunnel ein signifikanter Unterschied zwischen den ermittelten Posen der beiden Lokalisierungsfunktionen erwartet wird. Während die Genauigkeit der FZS im Tunnel durch die Abschaltung des Satellitenempfangs stark beeinträchtigt ist, trifft dies auf die Genauigkeit der KKLo nicht zu, da die KKLo auf Landmarken basiert. Ohne Satellitenempfang sind in der FZS keine GNSS-Messupdates möglich. Da Inertialsensorik und Odometrie keine absolute Position messen, wächst der Positionsfehler mit der Zeit an (Drift).

Als Test werden die definierten Anforderungen an die Positionierungsgenauigkeit des Fahrzeugs herangezogen, wobei diese Genauigkeit den Unterschied zwischen Soll-Pose und in der Simulation bekannter Ist-Pose (ground truth) beschreibt. Durch den im Tunnel auftretenden Drift in der durch die FZS geschätzten Pose wird ein Einhalten der Anforderungen im Tunnel verhindert. Im Rahmen des Proof-of-Concepts wird gezeigt, dass die Anforderungen in mehreren Simulationen eingehalten werden können. Wird ein statistisch signifikantes Ergebnis erreicht, gilt der Test als bestanden.

Des Weiteren ist es möglich, anhand des Szenarios die erste Zusatzfunktion des Verfahrens zu testen, die den Einfluss von Sprüngen in der Position der FZS unterdrückt. Ein solcher Sprung tritt bspw. auf, wenn die im Tunnel angewachsenen Fehler in der geschätzten Pose der FZS nach Ausfahrt aus dem Tunnel durch ein GNSS-Messupdate korrigiert

werden. Durch die nach einem solchen Messupdate signifikant gesunkene geschätzte Standardabweichung der durch die FZS ermittelten Position ist das Erkennen des Sprungs möglich und der nächste Korrekturschritt wird frühzeitig durchgeführt. Auf diese Weise wird das Entstehen einer unerwünschten Regelabweichung auch hier verhindert.

## 6 Zusammenfassung und Ausblick

Im Rahmen dieses Beitrags wurden verschiedene Systemarchitekturen für die Bewegungsregelung serviceorientierter, mobiler technischer Systeme abgeleitet und diskutiert. Anhand generierter Anforderungen an das System wurde gezeigt, weshalb etablierte Architekturen nicht geeignet sind, um die mit der Nutzung unterschiedlicher Lokalisierungsfunktionen für Planung und Regelung verbundenen Herausforderungen zu bestehen. Anschließend konnte gezeigt werden, wie eine neuartige Architektur mit integrierter Posen-Offsetkorrektur eine geeignete Lösung der Problematik darstellt. Die Offsetkorrektur erlaubt es, einen Einfluss inkonsistenter Lokalisierungsdaten auf die Fahrzeugführung zu verhindern und eröffnet daher neue Gestaltungsspielräume bei der Entwicklung modularer Systeme. Des Weiteren ist es mit dem vorgeschlagenen Verfahren möglich, den Einfluss von Sprüngen in der Lokalisierung auf die Regelung zu reduzieren und die Selbstwahrnehmung des Systems zu unterstützen.

Die Anwendung des präsentierten Verfahrens ist nicht auf automatisierte Kraftfahrzeuge beschränkt, sondern kann immer dort erfolgen, wo ein mobiles System mit getrennter planerischer und ausführender Ebene sowie mit mehr als einer Lokalisierungsfunktion zum Einsatz kommt.

## Danksagung

Diese Forschungsarbeiten wurden im Rahmen des Projekts „UNICARagil“ durchgeführt (FKZ 16EMO0286). Wir bedanken uns für die finanzielle Unterstützung des Projekts durch das Bundesministerium für Bildung und Forschung (BMBF).

## Literaturverzeichnis

- [1] Moritz Werling. *Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien: Zugl.: Karlsruhe, KIT, Diss., 2010.* Print on demand. Vol. 34. Schriftenreihe des Instituts für Angewandte Informatik - Automatisierungstechnik, Universität Karlsruhe (TH). Hannover and Karlsruhe: Technische Informationsbibliothek u. Universitätsbibliothek and KIT Scientific Publishing, 2011. ISBN: 978-3-86644-631-1. DOI: 10.5445/KSP/1000021738. URL: <https://edocs.tib.eu/files/e01fn12/685609162.pdf>.
- [2] Chris Urmson, Joshua Anhalt, et al. “Autonomous driving in urban environments: Boss and the Urban Challenge”. In: *Journal of Field Robotics* 25.8 (2008), pp. 425–466. ISSN: 15564959. DOI: 10.1002/rob.20255.

- [3] Timo Woopen, Bastian Lampe, et al. “UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts”. In: *27th Aachen Colloquium*. Aachen, Germany, Oct. 2018.
- [4] Clément Fouque and Philippe Bonnifait. “Navigable map-aided differential odometry to enhance GNSS in adverse conditions”. In: *Accurate Localization for Land Transportation*. Paris, France, 2009, p. 5. URL: <https://hal.archives-ouvertes.fr/hal-00445307>.
- [5] Bernard P. Zeigler, Herbert Prähofer, and Tag Gon Kim. *Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems*. 2. ed., transferred to digital print. Amsterdam and Heidelberg: Academic Press, 2000. ISBN: 0-12-778455-1.
- [6] Chassis Systems Control, Robert Bosch GmbH. *Safe GNSS/Inertial Positioning by VMPS for Highly Automated Driving*. 2018. URL: <https://www.bosch.co.jp/aee2018/pdf/bosch-automotive-engineering-exposition-2018-automated-cc-vehicle-motion-and-position-sensor-vmps-02.pdf>.

# Selbstlokalisierung für automatisierte Fahrzeuge mit tiefen neuronalen Netzen

Nico Engel und Klaus Dietmayer\*

**Zusammenfassung:** Für das automatisierte Fahren ist die Kenntnis über die eigene Fahrzeugpose (bestehend aus Position und Orientierung) unerlässlich, um ein hohes Maß an Sicherheit und eine robuste Fahrweise im Straßenverkehr zu garantieren. Weiterhin erfordern viele Softwaremodule, wie zum Beispiel die Trajektorienplanung, eine hoch genaue und verlässliche Schätzung der eigenen Pose.

In diesem Beitrag wird ein Verfahren vorgestellt, welches auf Basis einer hochgenauen digitalen Karte mit Landmarken, also leicht wiedererkennbaren Objekten in der Umgebung, und aktuellen Sensormessungen des automatisierten Fahrzeugs die Fahrzeugpose ermittelt. Dabei wird ein neuartiges Verfahren mit tiefen neuronalen Netzen präsentiert, welches aktuelle Messungen und Landmarken aus der Karte als 2D Punktlisten verarbeitet. Die Ergebnisse werden auf einem neu erstellten Datensatz in verschiedenen Szenarien in urbanen und ländlichen Gebieten evaluiert und es wird gezeigt, dass die vorgestellte Methode state-of-the-art Genauigkeiten im Bereich von 20 cm bei einer Rechenzeit von nur 2 ms erreicht. Weiterhin erfolgt ein Vergleich mit einer Netzwerkarchitektur, welche anstatt der Punktlisten auf Rasterkarten basiert, um die Vorteile der direkten Verarbeitung der Punkte aufzuzeigen.

**Schlüsselwörter:** Deep Learning, Digitale Karte, Lokalisierung, Punktwellen

## 1 Einleitung

Unter der Selbstlokalisierung eines automatisierten Fahrzeuges versteht man die Aufgabe, die eigene Position und die Ausrichtung innerhalb eines gegebenen Koordinatensystems zu bestimmen [14]. Diese Informationen sind essentiell für den sicheren und zuverlässigen Betrieb von automatisierten Fahrzeugen, da mit Hilfe der Positionsdaten unter anderem Informationen aus einer digitalen Karte mit in die Umgebungswahrnehmung integriert werden können [4]. Eine weit verbreitete Methode zur Bestimmung der Position ist die Verwendung von globalen Satellitennavigationssystemen (GNSS), wie beispielsweise das Global Position System (GPS). Die Genauigkeiten dieser Systeme hängt stark von Umwelteinflüssen (z.B. atmosphärischer Störungen) ab und kann in manchen Fällen durch Mehrwegausbreitungen in Häuserschluchten bei nur 10 m liegen [7]. Besonders in hoch komplexen und unstrukturierten Gebieten, zum Beispiel in der Stadt, ist jedoch ein präzises Lokalisierungsergebnis im Bereich von 20 – 50 cm erforderlich. Ein Lösungsansatz stellt die Nutzung von Korrekturdaten für die empfangenen GPS-Information dar (DGPS), womit Genauigkeiten im Zentimeterbereich erreicht werden können. Jedoch ist

\*Die Autoren sind vom Institut für Mess-, Regel- und Mikrotechnik, Universität Ulm, Albert-Einstein-Allee 41, 89081 Ulm, Deutschland. Email: nico.engel@uni-ulm.de, klaus.dietmayer@uni-ulm.de

auch ein DGPS System von Abschattungen in Häuserschluchten betroffen. Im Bereich der Fahrerassistenzsysteme und des automatisierten Fahrens sind Verfahren für die Lokalisierung verbreitet, die verschiedenen Sensoren, wie Lidar, Radar, Kamera und GPS kombinieren und mit Hilfe einer hochgenauen digitalen Karte die Fahrzeugpose schätzen. Typischerweise werden Lokalisierungsverfahren in zwei Gruppen eingeteilt: rasterkartenbasierte [15, 8] und landmarkenbasierte Methoden [2, 11]. Landmarken sind statische und wiedererkennbare Objekte in der Umgebung, wie zum Beispiel Lichtsignalanlagen, Bäume oder Straßenschilder, die mit verschiedenen Merkmalen (hochgenaue 2D Position, Größe, etc) in einer digitalen Karte gespeichert werden. Während der automatisierten Fahrt werden (vorverarbeitete) Sensormessungen mit der Karte abgeglichen um die eigene Position innerhalb des Kartenkoordinatensystems zu schätzen.

In diesem Beitrag wird ein Verfahren für die Selbstlokalisierung basierend auf Landmarken und multimodalen Sensormessungen vorgestellt, das zur Posenschätzung trainierte tiefe neuronale Netze verwendet. Die landmarkenbasierte Karte wird a-priori aufgenommen und als gegeben angesehen. Während der Fahrt werden aus den Messungen Features extrahiert, die mithilfe eines tiefen neuronalen Netzwerkes mit der Karte verglichen werden und die Fahrzeugpose geschätzt. Eine große Herausforderung hierbei ist die hohe Dynamik der Umgebung: Landmarken, die zum Zeitpunkt der Kartenerstellung vorhanden sind, können während der Lokalisierungsfahrt entfernt worden sein, oder statische Objekte (wie zum Beispiel parkende Autos) können neu auftauchen. Weiterhin ist sowohl die Anzahl der Landmarken aus der Karte, als auch die Anzahl der aktuellen Messungen für jeden Zeitschritt unbekannt. Traditionelle lernbasierte Ansätze wie Convolutional Neural Networks (CNN) sind für dieses Problem nicht geeignet, da die Eingangsdaten in einer geordneten und strukturierten Form vorliegen müssen, was bei 2D Punktlisten (Messungen und Landmarken) nicht gegeben ist. Eine Möglichkeit, dieses Problem zu umgehen besteht darin, die Punkte aus Messungen und Landmarken in einer Rasterkarte (Graustufenbild) einzutragen.

Zur Lösung des Problems der Selbstlokalisierung wird eine neuartige Netzwerkarchitektur vorgestellt, die auf der kürzlich eingeführten PointNet-Struktur basiert [9] und dahingehend modifiziert, dass zwei unabhängige Punktlisten verarbeitet und aufeinander abgebildet werden können. Das Ziel ist es, aus den Eingangsdaten Features zu generieren, die für die Posenschätzung verwendet werden können. Im Gegensatz zu dieser auf Punktlisten basierenden Methode wird zum Vergleich ein weiterer Ansatz auf Basis von Rasterkarten gezeigt, der als Eingang Bilder verwendet. Außerdem lassen sich die vorgestellten Ansätze mit bekannten Filtermethoden, wie einem Extended Kalman Filter (EKF), kombinieren, so dass eine noch genauere Lokalisierung ermöglicht werden kann.

## 2 Selbstlokalisierung mit Neuronalen Netzen

### 2.1 Digitale Karte und Messungen

Für die spätere Evaluierung wurde ein neuer Datensatz mit realen Szenarien in urbanen und ländlichen Gebieten erstellt. Dabei wurde für die Aufnahme ein Fahrzeug verwendet, welches mit einem Stereokamerasetup, Lidar- und Radarsensoren, sowie einem DGPS-System ausgestattet ist. Das DGPS-System dient der Generierung einer Ground-Truth Trajektorie, damit die Genauigkeit des vorgestellten Lokalisierungsverfahrens evaluiert

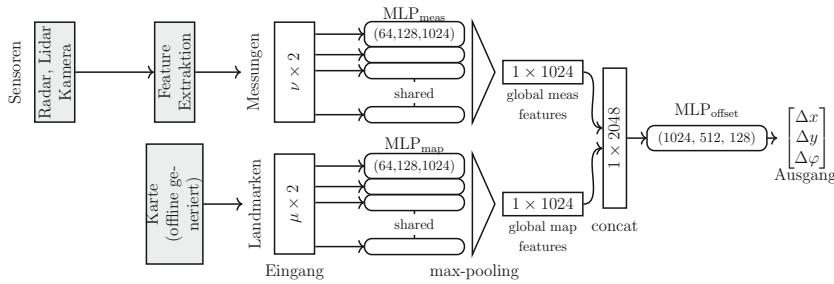


Abbildung 1: Übersicht der Netzwerkarchitektur auf Basis von Punktlisten [3]. Als Eingang dienen die vorverarbeitete Messungen und die Landmarken aus der digitalen Karten. Mittel Multilayerperzepron, welches die Gewichte für jeden Punkt der Messungen und der Landmarken teilt, und einer maxpool Operation werden für beide Eingangspunktlisten globale Featurevektoren erzeugt. Anschließend werden diese zusammengeführt und mittels eines dritten MLP der Ausgang (Korrekturvektor) geschätzt.

werden kann. Die landmarkenbasierte Karte wurde im Dezember 2017 aus insgesamt drei Rundfahrten auf der Teststrecke in Ulm-Lehr an verschiedenen Tagen erstellt und enthält 3860 Landmarken, wobei 1731 aus Lidarmessungen, 1411 aus Kamerabildern und 718 aus Radarmessungen kommen. Unabhängig davon wurden acht Messfahrten im November 2018 auf der Teststrecke durchgeführt, um Messungen zu generieren. Insgesamt enthält der Datensatz 145 000 Zeitschritte, die neben den Sensormessungen auch eine Ground-Truth Pose des Fahrzeugs beinhalten.

## 2.2 Methode auf Basis von Punktlisten

Das Ziel ist es, die Fahrzeugpose  $\mathbf{p} = [x, y, \varphi]^T$  in Bezug auf die Landmarken in der Karte  $\{z_1, \dots, z_\nu\}, z_{\{\cdot\}} \in \mathbb{R}^2$  und durch die Nutzung der Messungen  $\{m_1, \dots, m_\mu\}, m_{\{\cdot\}} \in \mathbb{R}^2$  zu bestimmen. Da eine direkte Bestimmung der Pose in einem globalen Koordinatensystem, wie zum Beispiel dem Universal Transverse Mercator (UTM), eine große Herausforderung darstellt, wird in diesem Beitrag vorgeschlagen, den Offset  $\Delta\mathbf{p}$ , also die Differenz, zu einer vorherigen, bekannten Pose  $\mathbf{p}_{t-1}$  zu schätzen. Die Idee besteht darin, eine Abbildung der Messungen und den Landmarken auf den Offset der Fahrzeugpose zu lernen [3]

$$f(\{z_1, \dots, z_\nu\}, \{m_1, \dots, m_\mu\}) \rightarrow [\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\varphi}]. \quad (1)$$

Ist die Pose des vorherigen Zeitschritts  $\mathbf{p}_{t-1}$  bekannt, so lässt sich mit Hilfe des Netzausgangs auf die aktuelle Pose schließen

$$\mathbf{p}_t = \mathbf{p}_{t-1} + [\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\varphi}]. \quad (2)$$

Das Lernziel ist die Minimierung der Differenz von dem Ground-Truth Pose Offset  $[\Delta x, \Delta y, \Delta \varphi]$  zum prädizierten Offset  $[\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\varphi}]$ . Da der translatorische Offset und

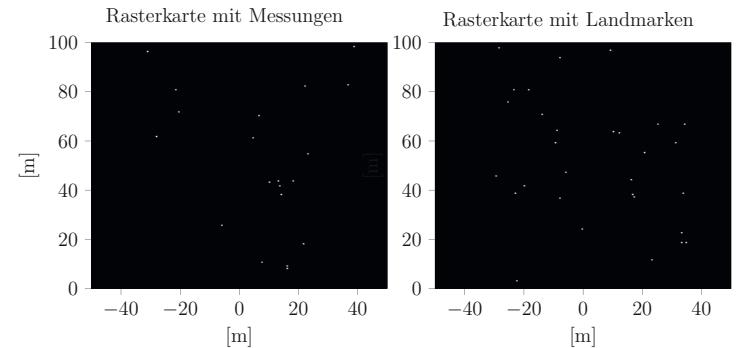


Abbildung 2: Darstellung der Landmarken und Messungen in einer Rasterkarte.

der rotatorische Offset unterschiedliche Einheiten besitzen, werden sie für die Minimierung unabhängig behandelt

$$\mathcal{L}_{\text{tran}} = \mathbb{E}[(\Delta\hat{x} - \Delta x)^2] + \mathbb{E}[(\Delta\hat{y} - \Delta y)^2], \quad \mathcal{L}_{\text{rot}} = \mathbb{E}[(\Delta\hat{\varphi} - \Delta\varphi)^2]. \quad (3)$$

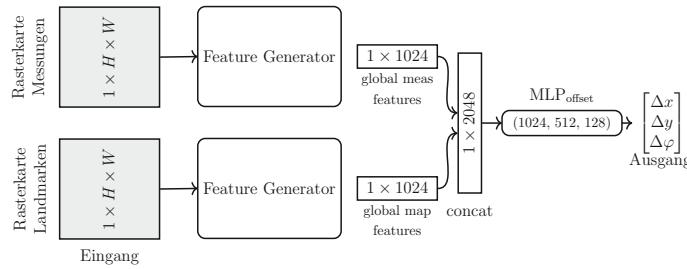
Zusätzlich wird die Gewichtung der beiden Loss-Terme mit gelernt [6], so dass für die Loss Funktion folgt

$$\mathcal{L} = \mathcal{L}_{\text{tran}} e^{-s_{\text{tran}}} + s_{\text{tran}} + \mathcal{L}_{\text{rot}} e^{-s_{\text{rot}}} + s_{\text{rot}}, \quad (4)$$

wobei  $s_{\text{tran}}$  und  $s_{\text{rot}}$  die jeweilige Unsicherheit der Aufgabe angibt und als Gewichtungsfaktor verstanden werden kann. Die verwendete Netzwerkarchitektur ist in Abb. 1 dargestellt. Die Messungen und die Landmarken aus der Karte liegen als 2D Punktlisten vor. Jeder Punkt aus der Liste der Messungen und jeder Punkt aus der Liste der Landmarken wird mit Hilfe mit jeweils einem unabhängigen Multilayerperzepron (MLP) von  $\mathbb{R}^2$  in einen höher dimensionalem Raum transformiert, hier  $\mathbb{R}^{1024}$ . Wichtig dabei ist, dass die Parameter der MLPs für die Messungen ( $\text{MLP}_{\text{meas}}$ ) geteilt werden. Das gleiche gilt für die MLPs der Landmarken ( $\text{MLP}_{\text{map}}$ ) (siehe Abb. 1). Es folgt eine *maxpool* Operation über die 1024 Dimensionen, so dass ein Vektor der Größe  $1 \times 1024$  folgt, der globaler Featurevektor genannt wird. Es ist ersichtlich, dass durch diese Operation unabhängig von der Anzahl der Eingangspunkte als auch unabhängig von der Reihenfolge der selbe Featurevektor entsteht. Somit sind die eingangs erwähnten Probleme der Permutationsinvarianz (Punktlisten sind unsortiert) und der unbekannten Anzahl der Datenpunkte zu jedem Zeitschritt gelöst. Die beiden globalen Featurevektoren werden anschließend zusammengeführt und mit Hilfe eines dritten MLPs ( $\text{MLP}_{\text{offset}}$ ) wird der Offset der Pose geschätzt.

## 2.3 Methode auf Basis von Rasterkarten

Die Idee der Selbstlokalisierung von automatisierten Fahrzeugen auf Basis von Rasterkarten und neuronalen Netzen orientiert sich an der Struktur von RegNet [10]. Der Eingang besteht dabei aus zwei Rasterkarten (Graustufenbilder), die das Sichtfeld (engl.: field of



view - FOV) des Fahrzeugs und die Umgebung darstellen. Jedes Pixel der Rasterkarte modelliert einen Bereich des Sichtfeldes der Größe  $10 \text{ cm} \times 10 \text{ cm}$ . Befindet sich innerhalb dieses Bereichs entweder eine Landmarke oder eine Sensormessung zum aktuellen Zeitpunkt, so wird der Bereich der zugehörigen Rasterkarte weiß eingefärbt. Freiraum wird dabei durch ein schwarzes Pixel dargestellt. Beispielhaft sind in Abbildung 2 für einen Zeitschritt die generierten Rasterkarten für die Landmarken im FOV des Fahrzeugs sowie die aktuellen Sensormessungen dargestellt. Anhand der gezeigten Rasterkarte wird ersichtlich, dass in dem Graustufenbild, welches bei einer Rastergröße von  $10 \text{ cm} \times 10 \text{ cm}$  und einem FOV von 100 m eine Größe von  $1000 \times 1000$  Pixel besitzt, im Durchschnitt über 99.9% der Rasterzellen Freiraum sind und somit keine Informationen enthalten. Die grundlegende Struktur ist ähnlich zu Abbildung 1 mit dem Unterschied, dass die Messungen und Landmarken wie bereits erwähnt zu Rasterkarten transformiert werden und somit in einer geordneten Struktur vorliegen. Für die Generierung der globalen Features der Landmarken (*global map feat*) und der Messungen (*global meas feat*) werden bekannte Netzwerkarchitekturen verwendet. In diesem Beitrag werden jeweils die Featuregeneratoren von GoogleNet [13], InceptionV4 [12] und ResNet [5] verwendet und miteinander verglichen. Der Unterschied zum Netzwerk mit Punktlisten besteht darin, dass die Features anstatt mit MLPs und einer maxpool Operation mit Hilfe von Convolutional Neural Networks (CNN) erstellt werden. Diese Featurevektoren werden anschließend verknüpft um einen Zusammenhang zwischen den beiden Eingangs rasterkarten zu lernen und mittels des  $\text{MLP}_{\text{offset}}$  zu einem Korrekturvektor zusammengefasst. Die resultierende Architektur ist in Abbildung 3 dargestellt. Für das Training und die Inferenz wurde die Größe der Rasterkarten auf  $320 \times 320$  Pixel reduziert um auf Kosten der Genauigkeit Rechenzeit und Rechenleistung einzusparen.

## 2.4 Filterung des Netzausgangs

Da das Netzwerk ohne eine zeitliche Komponente trainiert und eingesetzt wird, kann es zum Beispiel vorkommen, dass eine Position prädiziert wird, die entgegen der eigentlich

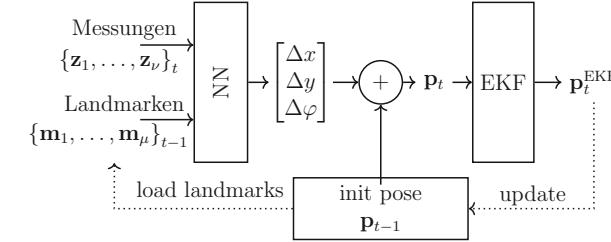


Abbildung 4: Implementierung des EKF [3]. Mit Hilfe der Messungen und Landmarken wird mittels des vorgestellten neuronalen Netzwerkes (NN) ein Offset-Vektor geschätzt, der eine initiale Pose korrigiert. Die korrigierte Pose  $p_t$  dient als Eingang eines Extended Kalman Filters mit einem CTRV Bewegungsmodells. Der Ausgang ist eine geglättete Posenschätzung  $p_t^{\text{EKF}}$ .

chen Fahrtrichtung liegt. Um diesen Effekt zu verhindern, wird zusätzlich ein Extended Kalman Filter (EKF) implementiert, welches als Messung die um den Netzausgang korrigierte Pose des Fahrzeugs erhält und eine zeitlich geglättete Trajektorie ausgibt. Die resultierende Pose wird dabei als neuer Startzustand für den nächsten Zeitschritt verwendet. Die Struktur ist in Abbildung 4 dargestellt. Als Bewegungsmodell wird ein Modell konstanter Geschwindigkeit und konstanter Drehrate (engl. constant turn rate and velocity - CTRV) eingesetzt. Die Rauschparameter wurde dabei experimentell bestimmt und die Updatefrequenz beträgt 20 ms.

## 3 Experimente und Evaluation

In diesem Kapitel werden die durchgeföhrten Experimente evaluiert. Insgesamt werden für den auf Punktlisten basierenden Ansatz zwei verschiedene Versuche mit vergleichbaren state-of-the-art Methoden verglichen. Zum Einem wird der Fall untersucht, dass eine ungenaue (z.B. durch atmosphärische Störungen verursachte) GPS-Position als initiale Lösung angenommen wird und die Aufgabe des neuronalen Netzes besteht darin, diese mit Hilfe der zum aktuellen Zeitpunkt registrierten Messungen und der digitalen Karte zu korrigieren (GPS-basierter Ansatz). Zusätzlich wird die in Kapitel 2.4 vorgestellte Filter-basierte Lokalisierung evaluiert. Die Besonderheit liegt darin, dass nur eine einzige initiale (und möglicherweise verursachte) GPS-Messung notwendig ist. Alle weiteren Zeitschritte nehmen als initiale Position die vorherige durch Netzausgang und EKF korrigierte Position an. Für die auf Rasterkarten basierte Methode werden, wie in Kapitel 2.3 beschrieben, verschiedene Featuregeneratoren (GoogleNet, InceptionV4 und ResNet) verwendet und auf Genauigkeit des Lokalisierungsergebnisses sowie Rechenzeit untersucht. Als Evaluationsmetrik wird der *Root Mean Square Error* (RMSE) verwendet. Außerdem werden ähnliche, auf Landmarken basierende Ansätze auf dem vorgestellten Datensatz zum Vergleich evaluiert. Die Ergebnisse sind in Tabelle 1 dargestellt. Für die auf Punktlisten basierenden Ansätze werden für die GPS-basierte Methode Positionsgenauigkeiten im Bereich von 23 – 27 cm erreicht bei einer Rechenzeit von nur 1.7 ms. Der Nachteil dieser Methode liegt darin, dass der Algorithmus in jedem Zeitschritt von einem GPS-Signal

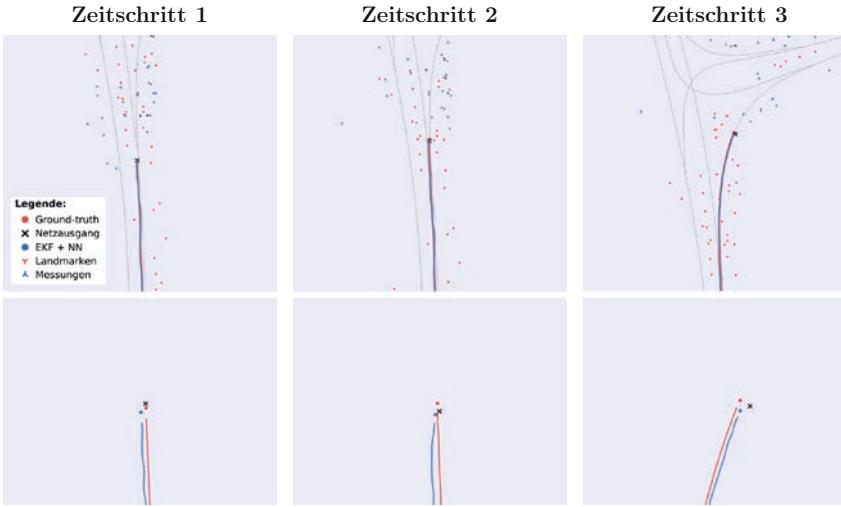


Abbildung 5: Ergebnisse der Testsequenz auf dem Ulm-Lehr Datensatz. Der Netzausgang kommt aus dem Netzwerk auf basis von Punktlisten. Ein roter Punkt ( $\bullet$ ) symbolisiert die Ground-Truth Position, ein blauer Punkt ( $\circ$ ) der vom EKF geglättete Netzausgang ( $\times$ ). Weiterhin sind die Landmarken in den oberen Grafiken als rote Kreuze und die aktuellen Messungen als blaue Kreuze dargestellt.

abhängt, welches beispielsweise in städtischen Szenarien durch Abschattungen nicht immer gegeben ist. Um dieses Problem zu umgehen wird der Netzausgang gefiltert und die geglättete Pose als neuer Startzustand gewählt (NN + EKF und NN + EKF + GPS in Tabelle 1). Somit ist der Algorithmus nur noch von einer initialen GPS-Messung abhängig und erreicht ohne weitere GPS-Messungen Genauigkeiten im Bereich von 30 – 50 cm. Werden zusätzlich vorhandene und verrauschte GPS-Messung im EKF fusioniert, erhöht sich die Genauigkeit auf  $\approx 20$  cm. Der Einsatz eines EKF hat auf die Rechenzeit einen vernachlässigbaren Einfluss.

Für die auf Rasterkarten basierenden Ansätze werden die drei vorgestellten Featuregeneratoren verglichen. GoogleNet, InceptionV4 und ResNet erreichen ähnliche Genauigkeiten im Bereich von 40–60cm. Der Nachteil dieser Methoden liegt in der hohen Rechenzeit von 20 – 80 ms. Dies ist darauf zurückzuführen, dass durch die Umwandlung der Landmarken und Messungen in Rasterkarten ein großer Überschuss an Daten generiert wird, der zur Lösung der Aufgabe keinen Vorteil bringt (unbelegte Zellen). Dieses Ergebnis legt nahe, dass die Verwendung von Punktlisten für die Selbstlokalisierung mit Landmarken besser geeignet ist.

Der Iterative Closest Point (ICP) Algorithmus [1] versucht mittels Rotation und Translation zwei Punktwolken aufeinander abzubilden, indem die Abstände zwischen den Punkten der beiden Punktwolken iterativ minimiert werden. Da die Messungen und die Landmarken in dem vorgestellten Datensatz sich in einigen Zeitschritten stark unterscheiden, erreicht der ICP Algorithmus eine durchschnittliche Genauigkeit von nur 1.17 m. Weiter-

Tabelle 1: Übersicht der durchgeföhrten Experimente mit den zugehörigen Genauigkeiten (RMSE) und Rechenzeiten.

Experiment	x [m]	y [m]	$\varphi$ [ $^{\circ}$ ]	Rechenzeit [ms]
<b>Punktlisten-basiert:</b>				
NN (GPS-basiert)	0.27	<b>0.23</b>	0.85	1.7
NN + EKF	0.56	0.33	1.20	2.1
NN + EKF + GPS	<b>0.27</b>	0.24	<b>0.82</b>	2.3
<b>Rasterkarten-basiert:</b>				
GoogleNet [13]	0.47	0.40	1.32	79.9
InceptionV4 [12]	0.61	0.52	1.22	62.6
ResNet [5]	0.52	0.45	1.7	20.9
<b>Vergleichbare Ansätze:</b>				
ICP [1]	1.17	1.4	4.96	7.6
EKF + GPS	0.59	0.54	6.58	<b>0.1</b>
RFS-MCL + GPS [11]	0.28	0.26	1.99	28

hin wurde die Pose mittels Extended Kalman Filter (EKF) geschätzt, wobei ungenaue GPS-Positionen als Messungen verwendet wurden ( $RMSE \approx 0.5$  m und  $\approx 6.5^{\circ}$ ). Als state-of-the art Methode wird die Random Finite Set Monte-Carlo Lokalisierung (RFS-MCL) von Stübler et al. [11] herangezogen, die vergleichbare Ergebnisse erreicht. Der Vorteil des vorgestellten Lokalisierungsverfahrens auf Basis tiefer neuronaler Netze liegt neben der hohen Genauigkeit in der sehr schnellen Rechenzeit von nur 2 ms im Vergleich zu den 28 ms bei der RFS-MCL.

Eine kurze Sequenz der Ergebnisse des Netzwerkes auf Basis von Punktlisten auf der Teststrecke ist exemplarisch in Abb. 5 dargestellt. Die obere Reihe zeigt einen größeren Ausschnitt mit dem Straßenverlauf, den in der Karte gespeicherten Landmarken, den Messungen sowie der Ground-Truth und der geschätzten Trajektorie (und Pose). In der unteren Reihe ist lediglich ein vergrößerter Ausschnitt gezeigt, um die Genauigkeit der Lokalisierung zu beurteilen. Es ist ersichtlich, dass die vorgestellte Methode ausreichende Genauigkeiten für das automatisierte Fahren liefert und mit Hilfe der Filterung (EKF) des Netzausgangs die Ground-Truth Trajektorie auf wenige Zentimeter genau geschätzt wird.

## 4 Zusammenfassung

In diesem Beitrag wurde ein Verfahren für die Selbstlokalisierung von autonomen Fahrzeugen auf Basis von tiefen neuronalen Netzen vorgestellt. Dabei wurden zwei Ansätze gewählt, die auf verschiedenen Eingangsformen basieren. Die Besonderheit liegt darin, dass durch die an PointNet angelegte Netzwerkarchitektur zwei unabhängige und ungeordnete 2D Punktlisten verarbeitet und aufeinander abgebildet werden können. Das

Netzwerk ist dabei ein Regressor, der den Offset der Fahrzeugpose zu einer bekannten, vorherigen Pose schätzt. Um das Lokalisierungsergebnis zu verbessern und zu glätten, wird ein Extended Kalman Filter mit einem Bewegungsmodell eingesetzt, welches als Messung den Ausgang des vorgestellten Netzwerkes erhält. Die Genauigkeit der vorgestellten Methode liegt im Bereich der Anforderungen für den sicheren Fahrbetrieb in städtischen sowie ländlichen Regionen und ist mit aktuellen bekannten Methoden vergleichbar. Zusätzlich kann die Methode mit nur einer einzigen initialen (und möglicherweise) ungenauen GPS Messungen zufriedenstellende Ergebnisse liefern. Weiterhin wurde gezeigt, dass die Rechenzeit des neuronalen Netzwerkes in Verbindung mit dem EKF bei etwa 2 ms liegt.

## Literatur

- [1] P. J. Besl and N. D. McKay. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.
- [2] H. Deusch, S. Reuter, and K. Dietmayer. The labeled multi-bernoulli SLAM filter. *IEEE Signal Processing Letters*, 22(10):1561–1565, 2015.
- [3] N. Engel, S. Hoermann, M. Horn, V. Belagiannis, and K. Dietmayer. Deeplocalization: Landmark-based self-localization with deep neural networks. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 926–933, Oct 2019.
- [4] F. Gies, A. Danzer, and K. Dietmayer. Environment perception framework fusing multi-object tracking, dynamic occupancy grid maps and digital maps. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3859–3865. IEEE, 2018.
- [5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [6] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.
- [7] T. Kos, I. Markezic, and J. Pokrajcic. Effects of multipath reception on GPS positioning performance. In *Proceedings ELMAR-2010*, pages 399–402. IEEE, 2010.
- [8] J. Levinson, M. Montemerlo, and S. Thrun. Map-based precision vehicle localization in urban environments. In *Robotics: Science and Systems*, volume 4, page 1. Citeseer, 2007.
- [9] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.
- [10] N. Schneider, F. Piewak, C. Stiller, and U. Franke. Regnet: Multimodal sensor registration using deep neural networks. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1803–1810. IEEE, 2017.

- [11] M. Stübler, S. Reuter, and K. Dietmayer. Consistency of feature-based random-set Monte-Carlo localization. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2017.
- [12] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [14] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.
- [15] J. Wiest, H. Deusch, D. Nuss, S. Reuter, M. Fritzsche, and K. Dietmayer. Localization based on region descriptors in grid maps. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 793–799. IEEE, 2014.

# Zuverlässige und integre Landmarken-basierte Lokalisierungsmethoden zum automatisierten Fahren

Isabell Hofstetter\*, Florian Ries†, Frank Schuster ‡, Martin Haueis §

**Zusammenfassung:** Nachweisbare Zuverlässigkeit Landmarken-basierter Lokalisierungsverfahren bleibt bis heute eine der größten Herausforderungen auf dem Gebiet des automatisierten Fahrens. Wesentliches Fehlerpotential steckt dabei vor allem in der Assoziation von detektierten Landmarken mit Kartenelementen. Periodische Muster und uneindeutige Lokalisierungsmerkmale stellen Fehlerquellen für eine korrekte Datenassoziation dar und gefährden damit eine integre Lokalisierung.

Diese Arbeit beschäftigt sich damit, Integrität von Lokalisierungsverfahren sicherzustellen. Durch eine Vorprozessierung des Kartenmaterials werden zunächst mehrdeutige Landmarkenmuster identifiziert, die mögliche Fehlerquellen darstellen. Zusätzlich wird eine neue Datenassoziationsmethode basierend auf Geometric Hashing eingeführt, welche keinerlei Informationen über die vorhergehende Position des Fahrzeugs benötigt und somit das Risiko der Fehlerfortpflanzung ausschließt. Abschließend wird die Identifikation eindeutiger Landmarkenkonstellationen zur integren Fahrzeuglokalisierung beschrieben.

**Schlüsselwörter:** Fahrzeuglokalisierung, Datenassoziation, Geometric Hashing, Integrität

## 1 Einleitung

Landmarken-basierte Lokalisierungsmethoden zum automatisierten Fahren wurden in der Vergangenheit ausgiebig erforscht und deren großes Potential verdeutlicht. Diese Verfahren nutzen eine zuvor generierte Karte, die Merkmale aus der Umgebung des Fahrzeugs enthält. Während der Lokalisierungsfahrt werden die detektierten Merkmale mit den Kartenelementen in Verbindung gebracht und daraus die Fahrzeugpose berechnet.

Die Herausforderung besteht somit vor allem darin, die korrekten Assoziationen zwischen kartierten Landmarken und Detektionen zu finden. Um Integrität sicherzustellen, sollte außerdem nachgewiesen werden können, dass die Wahrscheinlichkeit einer Falschassoziation hinreichend gering ist [1].

Lokalisierungsverfahren, die nur spärliche Landmarken mit schwachen Deskriptoren verwenden, bieten viele Vorteile, wie zum Beispiel eine reduzierte Kartengröße oder die Wiederverwendbarkeit der Merkmale für andere Aufgaben [4], [5], [6]. Allerdings geht mit

\*Isabell Hofstetter ist Doktorandin bei der Mercedes-Benz AG, isabell.hofstetter@daimler.com.

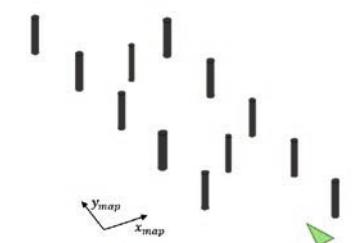
†Florian Ries ist Entwicklungsingenieur bei der Mercedes-Benz AG, florian.ries@daimler.com.

‡Frank Schuster ist Entwicklungsingenieur bei der Mercedes-Benz AG, frank.fs.schuster@daimler.com.

§Martin Haueis ist Leiter des Teams Fahrzeuglokalisierung bei der Mercedes-Benz AG, martin.haueis@daimler.com.



(a) Kamerabild einer städtischen Straße in Karlsruhe.



(b) Zugehöriger Kartenabschnitt mit zylindrischen Lokalisierungsmerkmalen.

Abb. 1: Beispiel einer städtischen Straße mit zugehörigen Lokalisierungsmerkmalen in der Karte.

der Extraktion solcher Lokalisierungsmerkmale auch ein großer Informationsverlust einher, und die Eindeutigkeit einer lokalen Umgebung wird in der Karte nicht immer wiedergespiegelt. Dies führt zu Mehrdeutigkeiten in der Assoziation von detektierten Merkmalen mit der Karte. Eine Garantie, dass Falschassoziationen ausgeschlossen werden, ist somit nicht mehr unbedingt gegeben.

Die meisten Ansätze verlassen sich bei der Datenassoziation auf die prädiizierte Pose des Fahrzeugs aus früheren Schätzwerten [2], was jedoch das Risiko der Fehlerfortpflanzung mit sich bringt. Andere Methoden greifen auf verschiedene Arten von Pattern Matching zurück [4], [7] um gewisse Mehrdeutigkeiten in der Zuordnung der Lokalisierungsmerkmale auszuschließen. Doch auch bei der Berechnung lokaler Deskriptoren aus Merkmalsmustern in [8], [9] fällt auf, dass solche Muster mehrdeutig sein können.

In dieser Arbeit wird auf ein Lokalisierungsverfahren aufgebaut, welches zylindrische Landmarken als Lokalisierungsmerkmale verwendet [4]. In Abbildung 1 ist die Art der Lokalisierungsmerkmale visualisiert: Zu einem Kamerabild in Abbildung 1a ist der passende Kartenabschnitt in Abbildung 1b zu sehen. Die zylindrischen Objekte, die hier in der Karte abgespeichert sind, repräsentieren in diesem Beispiel Bäume und Straßenlaternen. Im Folgenden werden diese Zylinder durch ihren Mittelpunkt auf der Bodenebene als zwei-dimensionale Punkte im Kartenkoordinatensystem dargestellt.

Um auch mittels solcher Lokalisierungsmerkmale eine integre Fahrzeuglokalisierung sicherstellen zu können, werden im Folgenden drei neue Ansätze beleuchtet: Kapitel 2 beschäftigt sich mit der Identifikation möglicher Fehlerquellen durch die a priori Detektion von Mehrdeutigkeiten im Kartenmaterial. Dazu wird eine Methode basierend auf Geometric Hashing genutzt, die in Kapitel 3 ebenfalls zur zuverlässigen Datenassoziation ohne Risiken durch Fehlerfortpflanzung vorgeschlagen wird. Abschließend wird in Kapitel 4 die Nutzung ausgewählter Landmarken zur integren und zuverlässigen Fahrzeuglokalisierung vorgeschlagen. Dazu wird die Identifikation eindeutiger Landmarkenmuster in der Karte erläutert.

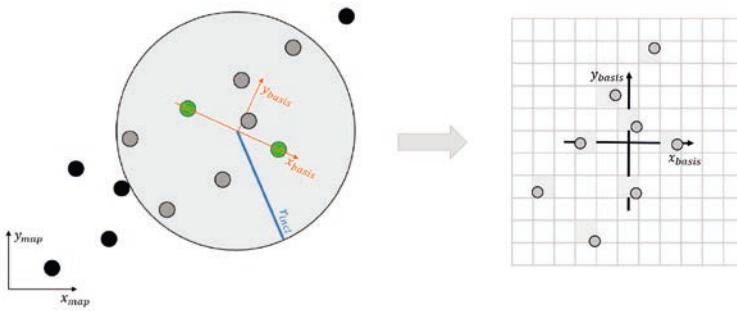


Abb. 2: Visualisierung der Hashing Parameter: Landmarken im Kartenkoordinatensystem (schwarz), ein Basispaar (grün), die zugehörige geometrische Basis (orange), Landmarken  $I_k$  (grau) im Umkreis  $r_{incl}$  (blau).

## 2 Vorprozessierung von Kartenmaterial zur Identifikation von Fehlerpotential

Mehrdeutigkeiten in den Kartendaten führen zu einem nicht eindeutig lösbar Problem der Datenassoziation. Solche Mehrdeutigkeiten entstehen durch uneindeutige Lokalisierungsmerkmale, ähnliche Landmarkenkonstellationen und periodische Muster von Lokalisierungsmerkmälern.

In diesem Kapitel wird eine Methode basierend auf Geometric Hashing vorgeschlagen, um strukturiert und effizient nach mehrdeutigen Mustern in Karten zu suchen. Das Wissen über die Existenz und räumliche Verteilung solcher Mehrdeutigkeiten erlaubt eine a priori Abschätzung von möglichen Assoziationsfehlern während der Lokalisierung und kann zu einer integren Lokalisierung beitragen.

Geometric Hashing wurde ursprünglich von Lamdan et al. im Gebiet des Bildverstehens zur Objekterkennung entwickelt [3], [10]. Inzwischen ist es eine bekannte Methode zur Assoziation von geometrischen Merkmalen, die einer Transformation unterzogen wurden.

Im Folgenden wird Geometric Hashing mit Hinblick auf die Verwendung der Methode zur Detektion von Mehrdeutigkeiten in Kartendaten hergeleitet. Das Vorgehen beinhaltet im ersten Schritt die Hash Table Generierung für einen Kartenabschnitt (siehe Abschnitt 2.1) und eine anschließende Suche nach Mehrdeutigkeiten mit Hilfe dieses Hash Tables (siehe Abschnitt 2.2).

### 2.1 Hash Table Generierung

In einem offline Schritt werden alle Merkmale in einer Karte  $M = \{f_1, f_2, \dots, f_n\}$  in einer Vielzahl von Koordinatensystemen dargestellt und in einem schnell durchsuchbaren, tabularen Format, dem *Hash Table*, gespeichert. Dafür werden Koordinatensysteme wie folgt definiert: Jede Kombination von zwei Landmarken  $k = (f_i, f_j)$  mit  $f_i, f_j \in M$  für  $i, j \in \{1, \dots, n\}, i \neq j$ , auch *Basispaar* genannt, definiert eine *geometrische Basis*. Den Ursprung dieser Basis stellt der Mittelpunkt zwischen  $f_i$  und  $f_j$  dar und die  $x$ -Achse ist

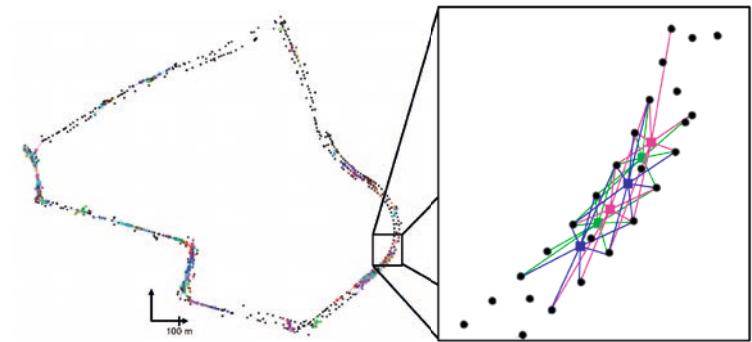


Abb. 3: Kartenausschnitt mit mehreren kongruenten Landmarkenkonstellationen: Kongruente Muster sind in der gleichen Farbe dargestellt und deren zugehörige Landmarken durch Kanten mit ihrem Schwerpunkt (als Quadrat dargestellt) verbunden.

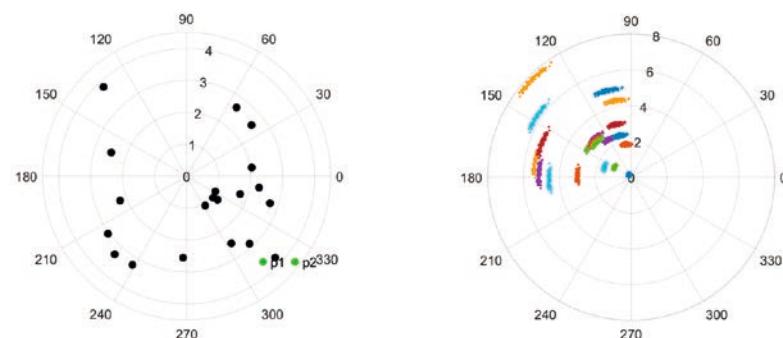
definiert als die Gerade, die durch die Punkte  $f_i$  und  $f_j$  geht. Um die Anzahl der dadurch entstehenden Koordinatensysteme einzuschränken, werden nur Basispaare verwendet für die gilt, dass  $\|f_i - f_j\| < d_{basis}$  für ein maximales Basispaar-Limit  $d_{basis}$ .

Anschließend werden alle Merkmale von Interesse  $I_k$ , d.h. diejenigen, die in einem gegebenen Umkreis  $r_{incl}$  um den Koordinatenursprung liegen, in das neue Koordinatensystem transformiert und diskretisiert. Schließlich werden für die Koordinaten der Lokalisierungsmerkmale  $I_k$  relativ zur geometrischen Basis Hashwerte berechnet, welche in einem Hash Table zusammen mit Informationen über das Basispaar abgespeichert werden. Abbildung 2 veranschaulicht den Hashing-Prozess und seine wichtigsten Parameter.

### 2.2 Detektion von Mehrdeutigkeiten

Der in Kapitel 2.1 generierte Hash Table enthält nun Informationen über lokale Umgebungen in der Karte, die in unterschiedlichsten Koordinatensystemen dargestellt und durch die Landmarken selbst definiert sind. Nun kann dieser Hash Table zur effizienten und strukturierten Suche nach kongruenten Landmarkenmuster in einer Karte genutzt werden. Dazu werden alle Koordinatensysteme oder *Layer* des Hash Tables nach ihren Basispaaren sortiert. Alle Layer mit übereinstimmenden Basispaaren werden anschließend nach weiteren Gemeinsamkeiten durchsucht. Diese Gemeinsamkeiten stellen ein kongruentes Landmarkenmuster dar, welches mindestens zweimal im Kartenkoordinatensystem vorkommt. Die Menge dieser Lokalisierungsmerkmale wird anschließend extrahiert und als mehrdeutige Konstellation abgespeichert. Ebenso wird jede mögliche Teilkonstellation in die Liste mit aufgenommen. Jede dieser Konstellationen wird mit Informationen über ihren Gegenpart und die Transformation zwischen den beiden abgespeichert. Dies ermöglicht die offline Abschätzung von möglichen Fehlern bei einer Falschassoziation von kongruenten Konstellationen.

Ein Kartenausschnitt mit detektierten kongruenten Konstellationen aus sieben Landmarken ist in Abbildung 3 dargestellt. Im hervorgehobenen Bereich ist deutlich sichtbar,



(a) Landmarken (schwarz) und zufällig gewähltes Basispaar  $p_1$  und  $p_2$  (grün) in Polarkoordinaten dargestellt.

(b) Simulierte Messungen der Landmarken, welche in das verrauschte Basis-Koordinatensystem transformiert wurden.

Abb. 4: Rauschverstärkung durch Geometric Hashing mit starren Transformationen.

dass kongruente Konstellationen im Kartenkoordinatensystem leicht verschoben zueinander auftreten können. Für weitere Details und Ergebnisse verweisen die Autoren auf [11].

### 3 Integre und effiziente Datenassoziation ohne Annahmen über vorhergehende Fahrzeugposen

Um den in Kapitel 2.1 generierten Hash Table nicht nur zur Detektion von Mehrdeutigkeiten sondern auch online zur effizienten und zuverlässigen Datenassoziation zu nutzen, muss er um ein Modell für das Messrauschen erweitert werden. Dies wird in 3.1 näher beschrieben. Anschließend wird die Datenassoziation mit Hilfe des Hash Tables in 3.2 hergeleitet. Das vorgeschlagene Verfahren ermöglicht die Datenassoziation ohne Prädiktion der Fahrzeugpose anhand von früheren Schätzwerten. Dies birgt viel Potential, da das Risiko der Fehlerfortpflanzung ausgeschlossen werden kann und somit eine sichere Lokalisierung ermöglicht wird.

#### 3.1 Rauschmodellierung für Geometric Hashing mittels starrer Transformationen

Während der Generierung des Hash Tables wird eine Monte Carlo Simulation durchgeführt, um das Messrauschen zu beschreiben, welches durch die Transformation in die verrauschte geometrische Basis verstärkt wird. Abhängig von der Wahl des Basispaars, sowie der Lage der restlichen Landmarken in der geometrischen Basis, muss eine Verteilung im Basis-Koordinatensystem geschätzt werden. Bei nah beieinander liegenden Basis-Landmarken wird das Messrauschen in der Winkelkomponente deutlich verstärkt,

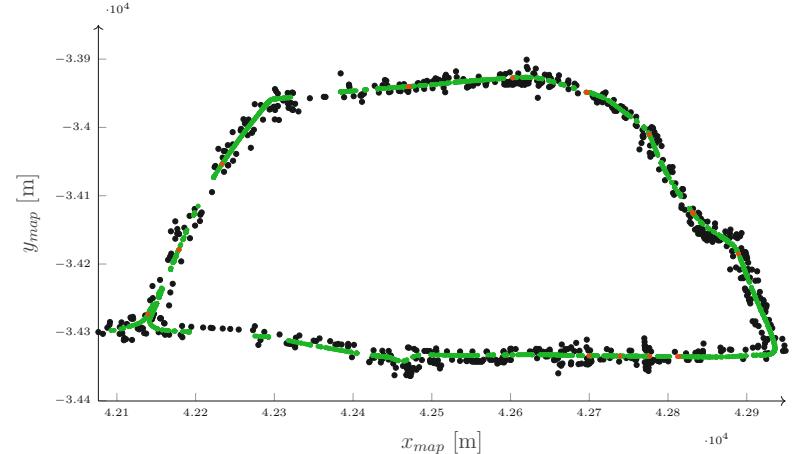


Abb. 5: Karte mit 738 zylindrischen Lokalisierungsmerkmalen (schwarz) und Positionen an denen eine sichere Assoziation möglich war (grün).

während das Rauschen in radialem Richtung konstant bleibt. Deswegen empfehlen die Autoren, die resultierende Verteilung in Polarkoordinaten zu betrachten (siehe Abb. 4).

Simulativ werden Messungen mit gegebenem Messrauschen  $\sigma$  erzeugt, welche dann in die Tranformation mit einfließen. Für jede geometrische Basis des Hash Tables und jede darin repräsentierte Landmarke werden aus diesen Messungen Standardabweichungen in radiale Richtung  $\sigma_r$  und in der Winkelkomponente  $\sigma_\theta$  berechnet. Diese Standardabweichungen werden dann im Hash Table gespeichert.

#### 3.2 Datenassoziation basierend auf Geometric Hashing

Gegeben sei eine Menge von detektierten Lokalisierungsmerkmalen  $D$  und ein Hash Table  $H$  zu einer zugehörigen Karte  $M$ .

Um Assoziationen zwischen den Detektionen  $D$  und Kartenelementen in  $M$  zu finden, müssen die detektierten Landmarken zunächst passend zum Hash Table verschlüsselt werden. Dazu wird ein zufälliges Paar an Merkmalen ausgewählt, welches eine geometrische Basis definiert. Alle verbleibenden Merkmale werden anschließend in dieses Koordinatensystem transformiert und entsprechend ihrer Koordinaten verschlüsselt. Für jeden berechneten Hashwert wird der zugehörige Eintrag im Hash Table gesucht und jede dort vorhandene geometrische Basis bekommt eine Stimme. Im folgenden Schritt werden die Basen, die eine gewisse Anzahl an Stimmen bekommen haben, extrahiert. Diese sind Kandidaten für mögliche Assoziationen. Abschließend wird eine Validierung durchgeführt, mit der entschieden wird, welche der Kandidaten die richtige Assoziation darstellt. Um weitere Sicherheit zu bekommen, können zusätzliche lokale geometrische Basen generiert und der Prozess wiederholt werden.

In Abbildung 5 ist eine Karte mit 738 Lokalisierungsmerkmalen zu sehen. Grüne Punkte stellen Positionen dar, an denen zuverlässige Assoziationen ohne Annahmen über ei-

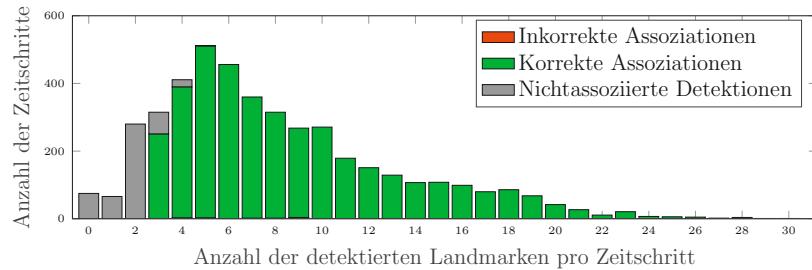


Abb. 6: Häufigkeit von verschiedenen Anzahlen von detektierten Landmarken und entsprechenden Assoziationen.

ne vorhergende Position des Fahrzeugs gefunden werden konnten. Rote Punkte stellen Fahrzeugposen dar, an denen mehrdeutige Konstellationen detektiert wurden und eine zuverlässige Assoziation somit nicht garantiert werden konnte. Die Häufigkeit von Landmarkendetektionen pro Zeitschritt und die zugehörigen korrekten bzw. inkorrekten Assoziationen sind mit Hilfe eines Balkendiagramms in Abbildung 6 dargestellt.

## 4 Ermittlung von Landmarkenkonstellationen zur integren Lokalisierung

Eine sichere Lokalisierung ist genau dann möglich, wenn keine Mehrdeutigkeiten in der Assoziation von Kartendaten und Detektionen vorhanden sind oder diese mit einer sehr hohen Wahrscheinlichkeit ausgeschlossen werden können.

In diesem Kapitel wird vorgeschlagen nur ausgewählte Landmarkenmuster zur Lokalisierung zu verwenden, die eine lokale Umgebung in der Karte eindeutig zuordnbar machen. Sobald ein solches Merkmalsmuster während der Fahrt detektiert wird, können Falschassoziationen dank der Eindeutigkeit der Konstellation unter gewissen Annahmen an Sensorfehler ausgeschlossen werden.

Im Folgenden wird zunächst definiert, was unter einer eindeutigen Landmarkenkonstellation zu verstehen ist. Anschließend wird beschrieben, wie solche Muster aus der Karte extrahiert werden können. Experimentelle Ergebnisse für einen Kartenabschnitt von ca. 4 km werden abschließend demonstriert.

### 4.1 Definition von eindeutigen Landmarkenmustern

Landmarkenmuster, die hier von Interesse sind, sollen in einem vorgegebenen Kartenabschnitt eindeutig sein. Dies gelte selbst dann, wenn die detektierten Landmarken mit einem Messrauschen  $\sigma$  belegt sind. Mit anderen Worten: Es gibt für ein eindeutiges Landmarkenmuster kein zweites approximatives Matching [12], [13] in der Karte. Ebenso sollen für eine eindeutige Konstellation auch alle Teilmengen der Merkmale wiederrum eindeutig sein um auch bei teilweisen Verdeckungen noch eine eindeutige Assoziation zu ermöglichen.

Um diese Voraussetzung mathematisch beschreiben zu können, sei der Abstand  $d$

zwischen einzelnen Landmarken  $x \in \mathbb{R}^2$  und Landmarkengruppen  $K$ , wobei  $K \subset \mathbb{R}^2$ , definiert als

$$d(x, K) := \min\{\|x - k\|_2 \mid k \in K\}.$$

Der minimale Abstand  $D$  zwischen zwei Landmarkengruppen  $K$  und  $L$  sei dann gegeben durch

$$D(K, L) := \min\{d(k, L) \mid k \in K\}.$$

Sei nun  $\mathcal{T}$  die Menge aller euklidischen Transformationen, d.h. alle Kombinationen aus Rotationen und Translationen. Dann ist ein Landmarkenmuster  $M_1 = \{f_1, \dots, f_N\}$  mit  $N > 2$  genau dann eindeutig in einem Kartenabschnitt  $M$ , wenn gilt

$$\forall T \in \mathcal{T} \exists F \subset M_1 \text{ mit } |F| = |M_1| - 2 : D(M_{sub,F}, T(F)) > \epsilon \quad (1)$$

für alle Teilmengen der Kartenmerkmale  $M_{sub,F} \subset M \setminus M_1$  mit  $|M_{sub,F}| = |F|$  und einem Mindestabstand  $\epsilon$ .

Das bedeutet, unter jeder beliebigen starren Transformation  $T$  gibt es höchstens zwei Merkmale  $f_i, f_j \in M_1$ ,  $i \neq j$ , deren Abstand zu den Kartenelementen in  $M \setminus M_1$  kleiner als  $\epsilon$  ist. Somit gibt es nur eine eindeutige Transformation, welche ein approximativeres Matching der vollständigen Konstellation ermöglicht. Gleichermaßen gilt auch für alle Teilkonstellationen  $F$  der Größen  $|F| = [3, |M_1| - 1]$ .

### 4.2 Identifikation eindeutiger Landmarkenmuster in der Karte

Um die Detektion vollständiger Landmarkenkonstellationen zu ermöglichen, ist zusätzlich zur Eindeutigkeit der Landmarkenmuster eine möglichst geringe Ausdehnung des Musters im Kartenkoordinatensystem wünschenswert, d.h.

$$d(f_i, f_j) < L \quad \forall f_i, f_j \in M_1, i \neq j, \quad (2)$$

mit gegebener maximaler Ausdehnung  $L$ .

Um Landmarkenkonstellationen mit Eigenschaften (1) und (2) aus den Kartendaten zu extrahieren, wird ein Clustering-Verfahren angewendet, welches die Informationen über Mehrdeutigkeiten gemäß der in Kapitel 2 beschriebenen Methoden nutzt.

Dazu werden zufällige Cluster von Lokalisierungsmerkmalen gebildet und mit der Liste der mehrdeutigen Konstellationen abgeglichen. Cluster, die die oben genannten Anforderungen (1) und (2) erfüllen, werden als eindeutige Landmarkenkonstellationen extrahiert. Die restlichen Cluster werden wiederholt zufällig aufgeteilt bis hier ebenfalls eindeutige Konstellationen gefunden werden.

In Abbildung 7 ist eine Karte zu sehen, in der Landmarkenmuster durch Farbe und verbindende Kanten gekennzeichnet sind. Diese Cluster stellen jeweils in der gesamten Karte eindeutige Konstellationen aus jeweils fünf Landmarken dar. Periodische Muster werden in verschiedene Konstellationen zerlegt, wie ebenfalls in Abbildung 7 zu sehen ist. Während der Lokalisierungsfahrt kann nun im Fall der Detektion einer solchen Konstellation unter gewissen Annahmen garantiert werden, dass keine Falschassoziation vorliegt.

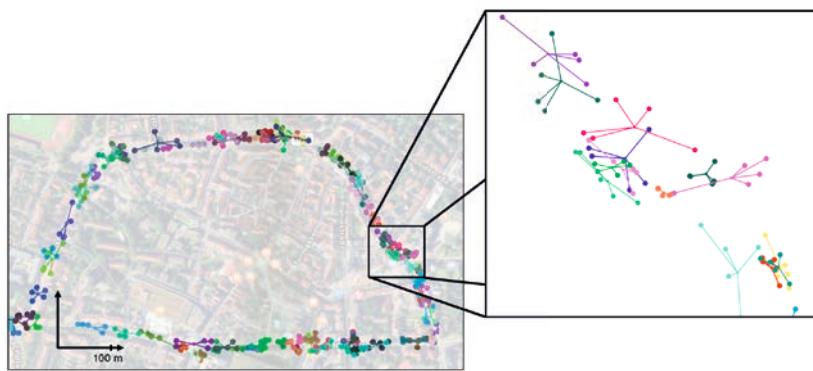


Abb. 7: Eindeutige Konstellationen von Lokalisierungsmerkmalen. Jede eindeutige Konstellation ist eindeutig farblich gekennzeichnet. Lokalisierungsmerkmale, die zu einem eindeutigen Merkmalsmuster gehören, sind durch Kanten mit dem Schwerpunkt der Konstellation verbunden.

## 5 Zusammenfassung und Ausblick

Die vorliegende Arbeit bietet einen Überblick zu drei neuen Ansätzen zur Sicherstellung integrier Landmarken-basierter Lokalisierung. Im ersten Teil wird die offline Vorprozessierung von Kartenmaterial vorgeschlagen, um mögliche Fehlerquellen schon a priori zu identifizieren. Dazu werden alle vorhandenen Mehrdeutigkeiten in der Geometrie der Landmarkenverteilung extrahiert und analysiert.

In einem zweiten Schritt wird eine neue Datenassoziationsmethode basierend auf Geometric Hashing vorgeschlagen. Dieses Verfahren benötigt keinerlei Information über vorige Fahrzeugposen und ermöglicht eine effiziente und sichere Datenassoziation, welche allein auf der Geometrie der Lokalisierungsmerkmale beruht. Da somit die Prädiktion der Fahrzeugpose anhand früherer Schätzwerte in der Datenassoziation vernachlässigt werden kann, ist eine Fehlerfortpflanzung ausgeschlossen und eine integre Lokalisierung möglich.

Abschließend wird die Ermittlung von eindeutigen Landmarkenkonstellationen in Karten beschrieben. Mit Hilfe der Informationen über Mehrdeutigkeiten wird bestimmt, welche Landmarkenmuster eine lokale Umgebung eindeutig zuordnenbar machen. Dies sind wertvolle Informationen um online Aussagen zur Wahrscheinlichkeit einer Falschassoziation zu treffen, welche in die Berechnung eines Integritätsmaßes einfließen können.

Zukünftige Arbeiten werden sich mit der online Nutzung der bereitgestellten Informationen und einer darauf basierenden Validierung von Landmarkenassoziationen beschäftigen.

## Literatur

- [1] PETOVELLO, MARK, S PULLEN, J SYRJARINNE und L WIROLA: *Quantifying the performance of navigation systems and standards for assisted-GNSS*. Inside GNSS, Seiten 20–24, 2008.

- [2] JOERGER, MATHIEU und BORIS PERVAN: *Quantifying Safety of Laser-Based Navigation*. IEEE Transactions on Aerospace and Electronic Systems, 2018.
- [3] LAMDAN, YEHEZKEL und HAIM J WOLFSON: *Geometric hashing: A general and efficient model-based recognition scheme*. 1988.
- [4] KÜMMERLE, J, M SONS, F POGGENHANS, M LAUER und C STILLER: *Accurate and Efficient Self-Localization on Roads using Basic Geometric Primitives*. In: 2019 International Conference on Robotics and Automation (ICRA), Seiten 5965–5971. IEEE, 2019.
- [5] SCHREIBER, MARKUS, CARSTEN KNÖPPEL und UWE FRANKE: *Laneloc: Lane marking based localization using highly accurate maps*. In: IEEE Intelligent Vehicles Symposium (IV), Seiten 449–454, 2013.
- [6] POGGENHANS, FABIAN, NIELS OLE SALSCHEIDER und CHRISTOPH STILLER: *Precise Localization in High-Definition Road Maps for Urban Regions*. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Seiten 2167–2174, 2018.
- [7] ASCANI, ANDREA, EMANUELE FRONTONI, ADRIANO MANCINI und PRIMO ZINGARETTI: *Feature group matching for appearance-based localization*. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Seiten 3933–3938, 2008.
- [8] SCHLICHTING, ALEXANDER und CLAUS BRENNER: *Localization using automotive laser scanners and local pattern matching*. In: 2014 IEEE Intelligent Vehicles Symposium Proceedings, Seiten 414–419. IEEE, 2014.
- [9] BRENNER, CLAUS: *Global localization of vehicles using local pole patterns*. In: Joint Pattern Recognition Symposium, Seiten 61–70. Springer, 2009.
- [10] RIGOUTSOS, ISIDORE: *Massively parallel Bayesian object recognition*. Technischer Bericht, New York University, Department of Computer Science, 1992.
- [11] HOFSTETTER, ISABELL, MICHAEL SPRUNK, FRANK SCHUSTER, FLORIAN RIES und MARTIN HAUEIS: *On Ambiguities in Feature-Based Vehicle Localization and their A Priori Detection in Maps*. In: 2019 IEEE Intelligent Vehicles Symposium (IV), Seiten 1192–1198. IEEE, 2019.
- [12] GOSHTASBY, A ARDESHIR: *Image registration: Principles, tools and methods*. Springer Science & Business Media, 2012.
- [13] GOODRICH, MICHAEL T, JOSEPH SB MITCHELL und MARK W ORLETSKY: *Approximate geometric pattern matching under rigid motions*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(4):371–379, 1999.

# Validation of automated driving – a structured analysis and survey of approaches

Jan Erik Stellet, Matthias Woehrle, Tino Brade,  
Alexander Poddey, Wolfgang Branz\*

**Zusammenfassung:** Validating the safety of automated driving (AD) is a problem of remarkable complexity and practical relevance. New approaches are needed since a statistical proof of safety based on field testing does not scale. Despite the attention paid to this topic in industry and academia, a consensus or unified framework has not yet been reached. This work describes and compares four distinct validation approaches. Our findings reveal that the current fragmentary landscape can be partly explained by differences in the AD use cases. On the one hand, there are different problem spaces characterised, e.g., by the operational domain and the driving tasks. On the other hand, the solution space differs for business models related to end-customer vehicles and mobility as a service.

**Schlüsselwörter:** Automated Driving, Validation, Safety, SOTIF

## 1 Introduction

Although modern driver assistance systems (SAE L2 [1]) can temporarily take over control of the vehicle, they do not actually take over the responsibility which remains with the human driver. Thus, advancing to automated driving (AD) of SAE L3 and beyond poses much higher requirements for the development and validation of safe systems. Over the last years, this topic has received remarkable attention in industry and academia. However, there seems to be no consensus yet.

The principal challenges of validation of complex systems operated in an open context have been discussed and formalised in [2]: Firstly, acceptance criteria are situation-dependent and currently informal. Secondly, a real-world operational design domain (ODD) is unstructured and bears infinitely many possible interactions [3]. Thirdly, the emergent behaviours are hard to predict since they are the result of a complex interplay of components (possibly including machine learning).

In general, none of these three interdependent aspects can be formally expressed in a sufficiently complete manner. Therefore, there does not exist a standard notion of coverage in this domain. Additionally, a naive approach based on the enumeration of combinations of equivalence classes on relevant dimensions, e.g., road topology, dynamic objects and environmental conditions, results in exponential blowup and renders an exhaustive set of verification tests practically infeasible. Note that the primary concern lies beyond the

\*The authors are with Robert Bosch GmbH, Corporate Research, Robert-Bosch-Campus 1, 71272 Renningen, Germany, [firstname.lastname@de.bosch.com](mailto:firstname.lastname@de.bosch.com)

scope of ISO 26262 [4] which does not define the necessary '*nominal performance*' of a safe system. The complementary standard ISO PAS 21448 [5] addresses '*functional insufficiencies*' but currently does not provide a detailed strategy on how to identify them.

In this research work, we compare prominent proposals for AD safety validation and highlight their similarities and differences. The goal is to help the reader understand the different approaches and inherent challenges. Thereby, we want to initiate a discussion in the academic community about research directions related to validation of AD.

This paper is organised as follows: First, related work will be discussed (Sec. 2). Thereafter, a formal problem description will be introduced in Sec. 3. Based on this, Sec. 4 will analyse four proposals. The paper concludes with Sec. 5.

## 2 Related work

Our previous publication [6] surveys the state of the art of testing of driver assistance systems (SAE L1–2) at that time. The present work provides an extended and updated view with a focus on automated driving systems (SAE L3–5).

Concerning the development of AD systems, [7] identifies and describes challenges from both sides of the V-model process, e.g. challenges related to fail-operational designs and complex requirements as well as testing of non-deterministic algorithms. The work [8] discusses stakeholder (e.g., legislative) perspectives on testing of AD systems. Three method clusters are identified, namely real world driving, formal verification and scenario-based tests. The test methods are categorised based on the representation of the object under test, the stimulus and assessment criteria.

Complementing the existing work, we identify practical AD validation challenges that allow us to differentiate among validation approaches.

## 3 Problem description and challenges

The question how to validate an AD system via testing procedures has become more and more important. Due to the more complex problem, more advanced methods than a representative field test (black-box test) as employed for validation of driver assistance systems are needed. Thereby, several practical challenges have to be addressed:

C1: **Representativeness challenge:** The goal of a test is to predict properties of the realisation in its operational domain. Therefore, a set of situations has to be sampled in order to accurately predict the operational context with respect to the purpose [11].

However, the openness of the context means that enumeration of situations does not scale. Instead, a global random sampling strategy can be used to obtain a representative set of situations, as it is done in field test drives. But, since such an undirected sampling will encounter (small) critical subspaces only with a certain (small) probability, the test duration increases with a decreasing frequency of critical events.

These considerations are illustrated by the empirical statistics on disengagements of autonomous vehicles [9]. Two examples, visualised in Fig. 1, show how the rate of

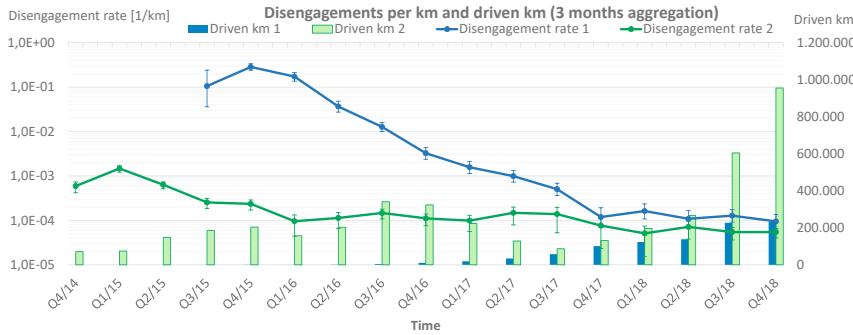


Figure 1: Two examples of human driver intervention (*disengagement*) rates and driven kilometres as published at [9]. Note that it is generally not known if all data is comparable because the driving profile is unknown and might change over time, e.g. towards more challenging situations. Further discussions can be found in [10].

improvement reduces over time although testing distances are significantly increased. This could be regarded as a first indication for a '*heavy tail safety ceiling*' [12].

C2: **Closed-loop challenge:** The AD function influences the vehicle's trajectory and thus the sensors' perspective of the world. Moreover, due to interactions, the future trajectories of other traffic participants are also influenced by the automated vehicle's behaviour. Thus, assessing an AD system requires to observe it in closed loop, at least partially to consider the interaction with the environment.<sup>1</sup>

C3: **Modelling challenge:** Testing in a virtual environment requires that the context (including e.g. behaviour of other traffic) and the realisation (including e.g. environment sensors such as cameras) have to be modelled. Additionally, the validity of the models has to be shown, see [11] for a detailed formalisation of this challenge.

C4: **Continuous improvements challenge:** Due to the complexity of the task the development of an AD function is necessarily iterative. However, each modification may change the behaviour of the automated vehicle and the interaction with others (cf. closed-loop challenge) compared to a recorded situation. Thus, conclusions from previous testing data as well as models may become invalid.

C5: **Insights challenge:** Testing on system level, i.e. in a black-box manner without exploiting structure in the realisation, prevents insights on sub-critical failures (cf. [13, 14]). Thus, this kind of testing becomes largely uninformative and inefficient.

C6: **Cross-domain, system-of-system challenge:** End-to-end performance relies on several, interacting systems from different domains in sensing, planning and actuation that rely on different design and validation approaches.

<sup>1</sup>Note that this is different for certain driver assistance systems, such as an automatic emergency brake, where solely the occurrence of a false intervention of an otherwise passive system shall be validated.

C7: **Supervision challenge:** Supervising an active automated driving function during a real-world driving test is a very demanding task for humans, especially if necessary interventions are very rare.

Note that there are additional challenges that are common to all approaches, such as legal aspects and the safety of data-based algorithms (e.g. machine learning) [7].

## 4 Survey of validation approaches

### 4.1 Passive AD (shadow mode)

The basic idea of the *shadow mode* or *passive AD* approach is to use a large fleet of human-driven vehicles to collect raw sensor data. After these open-loop recordings are retrieved, a simulation environment is used for a closed-loop replay of the data to the AD function [15, 16]. In another variant, the simulation environment runs on-line in the vehicle and only some data, determined by trigger conditions, is retrieved [17].

Since the AD function is passive during the data collection phase, it exhibits no safety risk. This enables a wide-spread roll-out to many end-user vehicles and would allow achieving large test distances in manageable time. Technical challenges for implementing the approach are:

- A sufficiently complete and correct transfer of the vehicle's environment to the simulation environment is necessary. Additional reference sensors, sensor data post-processing or annotations by humans can be used [17], but the correctness and completeness needs to be validated.
- For the simulation environment, the behaviour of other traffic participants has to be modelled, in order to close the loop.
- If trigger conditions are used to identify and transmit only relevant data from critical situations, these triggers have to be validated for sufficient completeness and correctness. Otherwise, a situation where the AD function would have planned a dangerous trajectory could remain unnoticed.
- The transferability of data collected with human-controlled vehicles to autonomously driven ones has to be argued. For example, it is sensible to assume that an AD vehicle drives defensively, e.g. leaving sufficient gaps, which might provoke more cut-in situations than a human driver would experience.

**Conclusion** Silently testing in end-customer vehicles improves scalability since there is no supervision challenge (C7) and the passive function can be continuously improved in the background (C4). However, there are several caveats that must be addressed before the open-loop recordings can be used as a validation argument. On the one hand, a sufficiently accurate simulation environment is needed to close the loop (C2). The dependence on models (e.g. reference environment model, sensor and behaviour models) reinforces the modelling challenge (C3). On the other hand, the representativeness of the data from human-controlled vehicles with respect to the AD function needs to be argued (C1). Additionally, if data can be recorded only selectively, there is limited insight into the system (C5).

## 4.2 Formally safe planning and statistically validated sensing

The authors of [18]<sup>2</sup> argue that scalability in the sense of mass production and ‘*everywhere*’ automated driving is not feasible with merely statistical data-driven validation. As a solution, they propose to combine a data-driven validation of the perception system and a formal model that guarantees for the safety of planned trajectories given a correct environment perception.

**Data-driven validation of the perception system** Firstly, the rate of situations being erroneously considered unsafe (*safety-critical ghosts*) or erroneously considered safe (*safety-critical misses*) shall be statistically validated by field tests.<sup>3</sup> Multiple parallel sub-systems, preferably based on different sensor technologies, with an assumed known probability of common errors are used in order to argue a lower failure rate of the combined perception system. Technical challenges with this approach are:

- Assumptions about the probability of common errors have to be validated. Note that small errors can cause a strong underestimation of the overall failure rate [19].
- Reference (ground truth) data of the environment model is needed in order to estimate the rates of safety-critical ghosts and misses. A straight-forward calculation based on the assumptions in [18] necessitates reference data on the order of magnitude of  $10^5$  h.
- Estimating the frequency of ghosts and misses is a not symmetric problem. While ghosts may occur at any time, a safety-critical miss can only occur if there is a dangerous situation. This imbalance may lead to much wider confidence intervals (or increased test durations) when estimating the frequency of safety-critical misses.
- Similar to the approach from Sec. 4.1, the profile of data collected for perception validation might differ from the statistics if the vehicles drive autonomously.

**Formal guarantees of the safety of the planned trajectories** The planning component is designed to be intrinsically safe by means of a safety envelope. To this end, the Responsibility Sensitive Safety (RSS) model is introduced. This model aims for a universally valid and explainable rule set that confines the actions of the actual trajectory planner. To achieve this, the ‘*elusive directive called duty of care*’ is made explicit in form of five rules, e.g. ‘*right-of-way is given, not taken*’ [18]. From these rules, quantitative constraints, e.g. in the sense of minimum safety distances, are derived.

Achieving generally accepted rules relies on two important premises. Firstly, the set of traffic scenarios used to derive the rules are exhaustive, especially when it comes to exceptional situations [3]. Secondly, the behaviour of other traffic participants can be

<sup>2</sup>Note that we directly refers to v6 of [18], as there have been substantial changes in the past.

<sup>3</sup>Note that there is an important distinction between *safety-critical ghosts* or *misses* and *false positive* or *false negative detections* in the perception system. Although detection errors are the usual causes for the former, a ghost or miss can also be caused by measurement errors, e.g. a noisy distance measurement. Furthermore, not every false positive or negative detection of the sensors will change the judgement of a situation as safe or dangerous, i.e. produce a safety-critical ghost or miss, respectively. Therefore, these error definitions are tied to the formulation of safety envelope which decides whether a situation is safe or dangerous.

modelled with a realistic set of parameters, e.g. the maximum reasonable deceleration that a lead vehicle might apply. It is currently unclear how realistic parameter values can be obtained or if it is even possible to assign single values that are adequate in all situations. Instead, it might be necessary to let the parameter values depend on the current situation to represent the societally accepted boundary between agile and dangerous behaviour. Implementing a rule set requires modelling of behaviour and motion of the ego vehicle and other traffic participants. Shalev-Shwartz *et al.* [18] use simple kinematic constraints, however more general techniques such as reachability analysis provide stronger guarantees and allow integrating further sources of uncertainty [20].

**Conclusion** Overall, the idea of formalising acceptable behaviour of an AD vehicle has its strengths in formally resolving the closed-loop (C2) and supervision challenges (C7). Moreover, explainable rules can help in increasing insight and transparency of the behaviour planning (C5). The cross-domain challenge (C6) is addressed by separation of concerns between perception and planning. However, to be applicable, error probabilities of the perception system have to be estimated and a generally valid set of parameters in the RSS model has to be argued. Since the model and its parametrisation take a central role, the corresponding modelling challenge (C3) is a key challenge for this approach.

## 4.3 PEGASUS: Scenario-based (top-down) approach

PEGASUS<sup>4</sup>, a German publicly funded project, aimed at developing methods for ensuring the safety of automated driving on the example of a SAE L3 (conditional automation) ‘*highway chauffeur*’ function.

The core of PEGASUS approach are scenarios. Their description is based on a six-layer model to compositionally model static and dynamics aspects in a joint description [21]. Scenarios can be identified from system knowledge [22], domain modelling [21,23] and field observation [17]. Scenarios feature parameters that can be varied in order to increase their coverage.

Scenarios are used as test cases that are executed and evaluated in simulation or on test tracks. Corresponding criticality metrics are used to determine the automated driving capabilities. Virtual testing enables reproducible tests and large scale parameter variations. One goal of test track testing is a point-wise validation of the simulation model. Additionally, real-world drive tests are conducted.

The scenario-based approach is developed together with an assessment of its validity and two major limitations have been identified:

1. Risk of generating the wrong scenarios: On the one hand, if scenarios are derived deductively the completeness and relevance is difficult to achieve. On the other hand, if scenarios are identified inductively from field data, the completeness depends on the metrics and models used. Additionally, the data might be incompletely mapped to a test case.
2. Risk of a wrong selection (reduction) of scenarios: Test cases might be defined based on equivalence classes although the underlying scenarios are in fact not equivalent.

<sup>4</sup>Project for the Establishment of Generally Accepted quality criteria, tools and methods as well as Scenarios and Situations

Concerning scenario parameters, the challenge is to find representative value ranges, a suitable discretisation and to cope with the exponentially increasing number of combinations.

**Conclusion** The PEGASUS approach attempts to standardise and make AD safety requirements transparent with a scenario-based approach. The most challenging aspect of this is to find the right set of scenarios (C1 representativeness). The closed-loop (C2) and insight (C5) challenges are alleviated by also using simulations for test case execution. However, before this can be achieved, the simulation environment has to be validated (C5 modelling). The PEGASUS concept addresses this with a cross-check between test cases executed on test tracks and in simulation. Nevertheless, representativeness and modelling are core challenges for the PEGASUS approach.

#### 4.4 Continuous validation

The previously described validation approaches mostly break down the overall problem into several successive steps, e.g. sensor data is collected and the safety of the AD system argued by simulation (Sec. 4.1) or statistical analysis and a formal model (Sec. 4.2). Thus, there are expectations that some elements, e.g. a simulation environment, will have been developed and validated because other elements rely on them.

However, this may be problematic in practice due to dependencies and a necessarily iterative development of an AD system. An alternative is to consider the development of the AD system and the validation infrastructure as a joint iterative process. An applied example with the scope of validating SAE L4 robotaxis can be found in [24].

**Iterative data- and simulation-driven development** The core of this approach is an iterative development and testing cycle with strong links between the different test strategies as illustrated in Fig. 2. This includes domain and system analyses, test track testing, virtual testing and field tests. The complementary nature is key to efficiency and effectiveness of the iterative cycle.

Exemplarily, field testing can identify edge case scenarios that are hard to identify otherwise. However, to cope with the continuous improvement challenge, recorded field test data must be reusable in a closed-loop simulation environment for reproducibly testing and generalising such scenarios. In addition to the random sampling-based field tests, test cases should be derived from systematic analysis of both the AD system and the simulation environment. Improving the realism of the simulation environment is equally important in order to rely on the efficiency of virtual testing. Otherwise, critical scenarios might be overlooked in virtual testing and would turn up later as surprises in field tests.

**Incremental deployment and supervised operation** A mobility as a service application benefits an iterative approach by means of the operation and deployment strategy. Firstly, the service can be launched under supervision of a safety driver. Secondly, a fine-grained and incrementally growing ODD is possible. The ODD for the fully automated vehicles (i.e. without safety driver) within a larger fleet can be defined on the level of individual customer trips. Third, frequent maintenance stops and remote monitoring of all vehicles are possible.

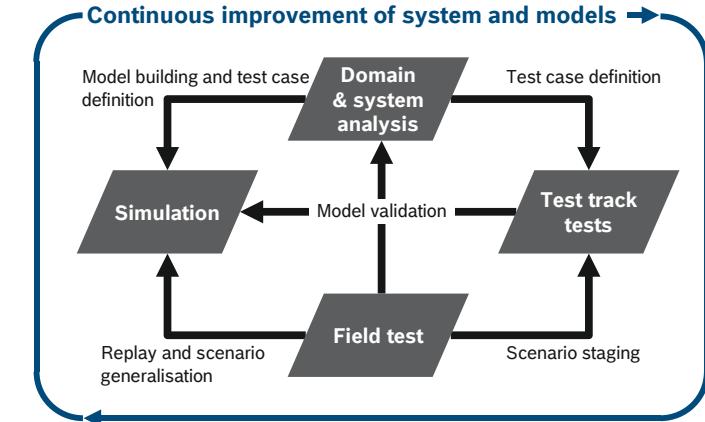


Figure 2: The iterative approach builds on continuous improvements of the system and models in a **feedback loop** with multiple ways to generate test cases.

**Conclusion** The sketched approach relies on similar elements as the ones discussed before, but integrates these complementary elements tightly. An iterative approach benefits from an incremental deployment that is only possible in mobility as a service applications. However, it is hampered with respect to scalability as new service locations are iteratively added and may exacerbate the continuous improvement challenge (C4), especially in the long tail, as new location may invalidate previously collected data, knowledge and models. This depends heavily on the balance between implicit knowledge such as field-data and explicit knowledge such as perception models including common errors. The supervision challenge (C7) must be specifically considered as it is difficult for humans to supervise an ever more capable automated driving function. Furthermore, the iterative nature must not conceal the fact that eventually, acceptance depends on explainable criteria. Thus, the approach could benefit from elements found in other, e.g. transparent rules for safe behaviour or coverage criteria.

## 5 Conclusion

As we have seen, different AD use cases, e.g. highway chauffeur or robotaxi, can induce quite different safety validation concepts. One reason is that the associated business case has a large impact on validation and its challenges, e.g., by carefully restricting the ODD or alleviating the supervision challenge. This concerns the initial release scope and its corresponding ODD but also scalability with respect to vehicle variants, geographic distribution and resulting diversity in ODDs. On the one hand, OEMs and suppliers of end-customer vehicles have to consider many vehicle variants in different price ranges that are used worldwide. On the other hand, driver-less mobility services are typically characterised by a homogeneous vehicle fleet and a restricted operational domain. The vehicle fleet might initially consist of vehicles with and without a supervisor, enabling a

fine-grained and growing ODD.

This work provides some first insights by identifying and discussing several validation challenges. While overall metrics and information for a public audience are available for individual approaches, detailed information to further analyse the progress with respect to the presented validation challenges is currently lacking. In future work, we try to reduce this information gap by further detailing on validation challenges and how current and future approaches could address them.

## References

- [1] SAE International, “Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles,” 2018.
- [2] J. E. Stellet, T. Brade, A. Poddey, and W. Branz, “Formalisation and algorithmic approach to the automated driving validation problem,” in *Intelligent Vehicles Symposium (IV), IEEE*, pp. 45–51, 2019.
- [3] P. Koopman and F. Fratrik, “How many operational design domains, objects, and events?,” 2019.
- [4] International Organization for Standardization, “Road vehicles – functional safety,” 2011.
- [5] International Organization for Standardization, “Road vehicles – safety of the intended functionality,” 2019.
- [6] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, and J. M. Zöllner, “Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions,” in *Intelligent Transportation Systems (ITSC), 18<sup>th</sup> IEEE International Conference on*, pp. 1455–1462, 2015.
- [7] P. Koopman and M. Wagner, “Challenges in autonomous vehicle testing and validation,” *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.
- [8] P. Junietz, W. Wachenfeld, K. Klonecki, and H. Winner, “Evaluation of different approaches to address safety validation of automated driving,” in *21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 491–496, 2018.
- [9] “Autonomous vehicle disengagement reports,” tech. rep., State of California: Department of Motor vehicles.
- [10] C. Lv, D. Cao, Y. Zhao, D. J. Auger, M. Sullman, H. Wang, L. M. Dutka, L. Skrypczuk, and A. Mouzakitis, “Analysis of autopilot disengagements occurring during autonomous vehicle testing,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 58–68, 2018.
- [11] E. Böde, M. Büker, U. Eberle, M. Fränzle, S. Gerwinn, and B. Kramer, “Efficient splitting of test and simulation cases for the verification of highly automated driving

functions,” in *Computer Safety, Reliability, and Security* (B. Gallina, A. Skavhaug, and F. Bitsch, eds.), (Cham), pp. 139–153, Springer International Publishing, 2018.

- [12] P. Koopman, “The heavy tail safety ceiling,” in *Automated and Connected Vehicle Systems Testing Symposium*, 2018.
- [13] S. Shalev-Shwartz and A. Shashua, “On the sample complexity of end-to-end training vs. semantic abstraction training,” *arXiv preprint arXiv:1604.06915*, 2016.
- [14] H. Winner, W. Wachenfeld, and P. Junietz, *Validation and Introduction of Automated Driving*, pp. 177–196. Cham: Springer International Publishing, 2018.
- [15] W. Wachenfeld and H. Winner, “Virtual assessment of automation in field operation a new runtime validation method,” in *Workshop Fahrerassistenzsysteme, Walting, Germany*, pp. 161–170, 2015.
- [16] A. Koenig, K. Witzlsperger, F. Leutwiler, and S. Hohmann, “Overview of had validation and passive had as a concept for validating highly automated cars,” *at-Automatisierungstechnik*, vol. 66, no. 2, pp. 132–145, 2018.
- [17] P. Junietz, W. Wachenfeld, V. Schönenmann, K. Domhardt, W. Tribelhorn, and H. Winner, “Gaining knowledge on Automated Driving’s safety – the risk-free VAAFO tool,” in *Control Strategies for Advanced Driver Assistance Systems and Autonomous Driving Functions*, pp. 47–65, Springer, 2019.
- [18] S. Shalev-Shwartz, S. Shamir, and A. Shashua, “On a formal model of safe and scalable self-driving cars,” *arXiv preprint arXiv:1708.06374v6*, 2017.
- [19] R. W. Butler and G. B. Finelli, “The infeasibility of quantifying the reliability of life-critical real-time software,” *IEEE Transactions on Software Engineering*, vol. 19, no. 1, pp. 3–12, 1993.
- [20] M. Althoff and J. M. Dolan, “Online verification of automated road vehicles using reachability analysis,” *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.
- [21] F. Schuldt, *Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen*. PhD thesis, Technische Universität Braunschweig, 2017.
- [22] M. Büker, B. Kramer, E. Böde, S. Vander Maele, and M. Fränzle, “Identifikation von automationsrisiken hochautomatisierter fahrfunktionen in pegasus,” in *AAET 2019 – Automatisiertes und vernetztes Fahren*, 2019.
- [23] G. Bagschik, T. Menzel, and M. Maurer, “Ontology based scene creation for the development of automated vehicles,” in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1813–1820, 2018.
- [24] “Waymo safety report: On the road to fully self-driving,” tech. rep., Waymo LLC, 2018.

# Saturation Effects in Recorded Maneuver Data for the Test of Automated Driving

Lukas Hartjen\*, Robin Philipp\*, Fabian Schulte\* and Bernhard Friedrich †

**Abstract:** One key challenge for a scenario-based verification and validation approach of automated vehicles is the completeness of a scenario set for a specific operational design domain of the system. A possible solution is the use of real-world datasets to argue for the representativity of a given set of scenarios. In order to make these arguments, it is necessary to estimate the amount of data that has to be collected. This contribution investigates the occurrence of saturation effects in the data collection for the test of automated vehicles to estimate the representativity of a set of scenarios. To this end, we analyze a dataset collected in the city of Hamburg with respect to saturation effects in the observed maneuvers.

**Keywords:** Data Collection, Saturation, Test Driving, Validation

## 1 Introduction

In order to introduce highly automated driving (HAD) to the market, the safety verification and validation of these systems is a key requirement. Since distance-based validation approaches will likely not be feasible for economic reasons alone [1], scenario-based verification and validation of HAD-Systems are a focus of current research activities [2]. However, it is currently an unsolved question how a finite set of scenarios for the test of an automated vehicle should be constructed. Especially it is a challenge to argue for the completeness and representativity of a scenario set for a given operational design domain.

A possible approach to this argumentation for completeness and representativity is the exposure of the scenario set during real-world traffic recordings. The recognition of semantic traffic behavior based on vehicle and pedestrian maneuvers can facilitate the estimation of scenario exposure [3]. Due to the description through maneuvers, the traffic behavior of objects is abstracted from, becomes differentiable and thereby countable and collectable. However, it is necessary to know the amount of traffic data that has to be collected through real-world driving to make a representative statement about the exposure of different traffic scenarios. Therefore, this publication analyzes saturation effects during the data collection of vehicle maneuvers in urban traffic to estimate the data quantities that are needed for this task.

\*Lukas Hartjen, Robin Philipp and Fabian Schulte are with Volkswagen Group Innovation, Wolfsburg, Germany, lukas.hartjen@volkswagen.de

†Bernhard Friedrich is a Professor at the Institute of Transportation and Urban Engineering, Technische Universität Braunschweig, Brunswick, Germany

## 2 Related Work

This work is following a suggestion by Wachenfeld et al. [4], that test drives with automated vehicles are not only useful to test the system functionality directly. Rather, the data collected on these drives can also be applied to model traffic behavior and assemble a set of scenarios to test the automated vehicle in simulation or proving ground tests.

The assembly of a set of scenarios is also addressed by Hauer et al. [5]. They propose to model the search for new scenario types as a coupon collector's problem to estimate the likelihood of detecting previously unseen scenario types in new collected measurement data. These abstract scenario types correspond to the *Functional Scenarios* introduced by Menzel et al. [6]. In this paper, we investigate new maneuver combinations as a part of these *Functional Scenarios* to facilitate this and other approaches to evaluate the completeness of scenario catalogues.

Amersbach and Winner [7] propose a method to calculate the number of concrete scenarios required for the validation of automated vehicles. Based on the work of Langner et al. [8], they assume the share of unique scenarios to be one fifth of the overall amount of observed scenarios. Accordingly, they estimate the amount of scenarios required for the validation to be  $n_{req} \approx 1.2 \cdot 10^{10}$ . In our work, we investigate the occurrence of new unique scenarios in measurement data by looking at the occurrence of abstract vehicle maneuvers. The findings could be incorporated into calculations such as the ones proposed by Amersbach and Winner [7].

Mauritz et al. [9] introduce a testing strategy for a lane change assistant based on an abstract semantic domain description. They demonstrate how it is possible to estimate the achieved test coverage over time using this semantic description. A transfer of this methodology to pedestrian traffic and the test of automated driving systems is described by Hartjen et al. [3], but did not include the semantic analysis of vehicle movements. While the approach proved helpful in identifying semantically redundant elements in recorded test drives, saturation behavior could not be observed due to the limited size of the analyzed data.

A framework to model driving maneuvers for the generation of test scenarios is presented by Krajewski et al. [10]. Their approach uses *Generative Adversarial Networks (GANs)* to synthesize new trajectories from a set of previously recorded maneuver instances. While this is an important addition to our work in this paper, the question of how much data should be collected is not addressed to the best of our knowledge.

An estimation of the necessary amount of Naturalistic Driving Data (NDD) for traffic modeling is presented by Wang et al. [11]. In their work, they apply the Kullback-Leibler (KL) Divergence [12] to measure the change in estimated probability density functions when using more data to compute them. While the focus of their work is the analysis of longitudinal driving behavior, the general methodology seems to be transferable to other applications as well. In this work, their presented methodology is transferred to analyze saturation effects in the collection of semantic traffic data for the test of automated vehicles. The usage of the KL-divergence measure is discussed in detail in the later part of this publication.

Despite an emerging consensus in the scientific community that scenario-based verification and validation approaches for automated driving will play an important role, it remains an unsolved challenge how to assemble a representative set of scenarios. In order to utilize real-world data collected during test drives, it is of interest to estimate the

amount of data that has to be collected to make informed statements about the surrounding traffic of the automated vehicle. To the best of our knowledge, this area of research has not been widely investigated for the purpose of verifying automated vehicles. In this work, we therefore address two research questions that will be discussed in the following section.

### 3 Research Questions and Hypotheses

Based on the analyzed related work, the following research questions were identified to quantify the amount of measurement data that should be collected to create a representative set of scenarios for scenario-based testing of urban automated driving functions.

**Research Question 1** *How much data is needed to achieve saturation with respect to the behavior of objects?*

In this contribution, the behavior of movable objects is defined as sequences of maneuvers [13]. To answer this research question, we will look at the parametric description of individual maneuvers on one side and at their occurrence in the form of maneuver sequences on the other side. A dataset is considered to be saturated in this respect when no previously unseen maneuver sequences are detected through additional data collection efforts. The amount of data necessary to observe these saturation effects is analyzed in this work.

**Research Question 2** *Is it possible to identify commonly occurring Functional Scenarios by means of semantic classification?*

Since identifying all possible *Functional Scenarios* (cf. [6]) in recorded measurement data is likely going to be very time and cost expensive, it could be of great practical interest to determine common scenarios. These scenarios would be likely to occur during the field operation of an automated vehicle. Thereby, a first goal of the verification and validation process of the system could then be to establish the safe behavior of the system in these common *Functional Scenarios*, for example in regression tests of new software revisions. In this research question, it will be investigated if and how the classification of abstract maneuvers in recorded measurement data can aid the process of identifying common *Functional Scenarios*.

## 4 Methodology

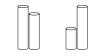
To investigate the aforementioned research questions, data was collected on test drives with automated vehicles in the city of Hamburg. The driven route is shown in Figure 1. Our dataset consists of approximately 3 hours of urban driving, in which the vehicles travelled roughly 50.6 km. In this database, we classified 179 488 maneuvers that were executed by surrounding objects.

In an earlier publication [13], we introduced a catalog of vehicle maneuvers, as well as the different layers of semantic analysis for urban traffic that will be the foundation of this work. The layers are shown in Figure 2. For the purpose of this work, classifiers

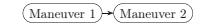


Figure 1: Routes driven for the collection of maneuver data in the city of Hamburg, Germany. Map data by OpenStreetMap [14]

**Scenario-Layer**  
Number of traffic participants



**Behavior-Layer**  
Sequences of maneuvers



**Maneuver-Layer**  
Parametric Description of individual maneuvers

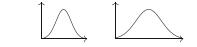


Figure 2: Layers of semantic analysis for traffic participants in urban traffic scenarios, extended from [13]

were implemented to detect all maneuvers of the catalog [13] in measurement data with the exception of *Follow Lane* and *Park* maneuvers. The following subsections describe the applied methodologies, as well as the preliminary results.

### 4.1 Saturation Analysis

With respect to *Research Question 1*, the collected data is analyzed on the *Maneuver Layer* and the *Behavior Layer* with respect to the occurrence of saturation effects.

#### 4.1.1 Maneuver Layer

On the lowest layer of analysis, the execution of individual maneuvers is examined. An example is the velocity profile or the trajectory curvature during *Turn Left* maneuvers at intersections. For the purpose of this work, the analysis on the *Maneuver Layer* will be limited to investigate one exemplary maneuver parameter, the initial velocity  $v_0$  observed in *Turn Left* maneuvers. Apart from being an illustrative example, this parameter could also be of practical interest to specify the initial state of a vehicle in a simulation scenario, even if the remaining movement of the object is determined by a traffic model.

In order to describe the observed execution of the maneuvers in the database in a uniform manner, the object movements are modeled by Beziér curves of third order [13]. Three curves describe the spatial movement as well as the observed velocity profile. Figure 3 depicts the first parameter of the Beziér curve modeling the velocity  $v_0$  during classified *Turn Left* maneuvers in the analyzed database. After the trajectory of each recorded maneuver of the chosen type has been converted to this local coordinate system, the Bézier points are calculated. By combining the Bézier representations of all observed maneuver instances, the underlying probability density function (*pdf*) for  $v_0$  is estimated using kernel density estimation (*kde*) with Scott's rule [15] for bandwidth selection.

Following Wang et al. [11], saturation effects regarding the amount of collected maneuver samples are investigated by repeating the process of *kde* for different sample sizes. To this end, 125 instances of *Turn Left* maneuvers are sampled to 30 different sets of

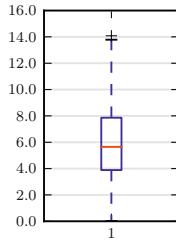


Figure 3: Recorded initial velocities during 125 *Turn Left* maneuvers

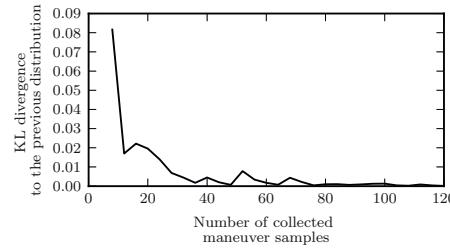


Figure 4: Kullback-Leibler divergence to the previous distribution of initial velocities during *Turn Left* maneuvers over the amount of maneuver samples used for kernel density estimation

equidistant size. The obtained *pdfs* are then compared using the Kullback-Leibler [12] divergence to the next smaller sample set. The results are shown in Figure 4.

#### 4.1.2 Behavior Layer Analysis

To analyse saturation effects on the *Behavior Layer*, the amount of observed unique maneuver sequences is plotted over the cumulated number of recorded objects. Since multiple maneuvers can be executed at each point in time, we introduce the concept of a *Parallel Maneuver Combination (PMC)*. A *PMC* captures all the maneuvers that an object is executing at a defined point in time (cf. Figure 5). A *Parallel Maneuver*

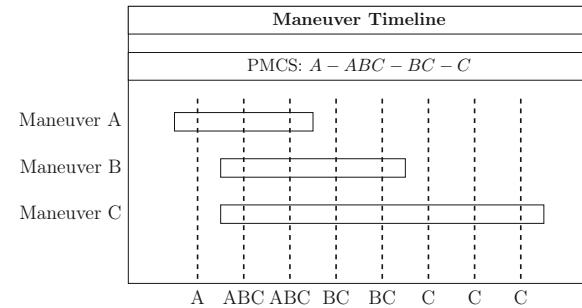


Figure 5: Abstract example for the notation of an object's *Parallel Maneuver Combination Sequence (PMCS)* from its maneuver timeline

*Combination Sequence (PMCS)* is subsequently defined as the chronological development of an object's *PMC* state. A *PMCS* is constructed by identifying the *PMC* at each discrete point in time and then appending it to the sequence if it differs from the previous *PMC*. An abstract example for the construction of a *PMCS* can be found in Figure 5. In the beginning, the object executes only the abstract maneuver *A*. Next, it simultaneously

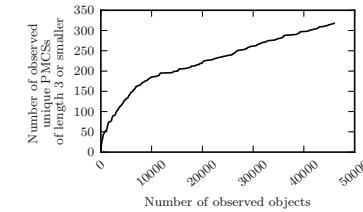


Figure 6: Amount of observed unique *Parallel Maneuver Combination Sequences (PMCSs)* of length 3 or smaller over the total number of observed objects

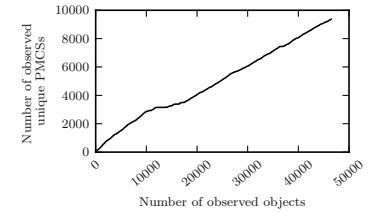


Figure 7: Amount of observed unique *Parallel Maneuver Combination Sequences (PMCSs)* over the total number of observed objects

starts to execute the maneuvers *B* and *C*, changing its current *PMC* from *A* to *ABC* and adding a new entry to its *PMCS*. After the execution of maneuver *A* terminates, the current *PMC* is now *BC*, adding another entry to the *PMCS* which is followed by *C* after the execution of maneuver *B* terminates as well. Note that repeated *PMC* samples are not duplicated in the *PMCS* since it reflects only changes in the object's current maneuver constellation. This constitutes an abstraction by neglecting the temporal extension of individual *PMC* states in the sequence.

Once the *PMCS* has been constructed for every object in the recorded database, the occurrence of new unique sequences is analyzed regarding saturation effects. For every *PMCS*, it is checked whether or not an identical *PMCS* was already observed for another object in the database. If not, it is added to the set of known sequences and the respective count is incremented by one. Figure 6 shows the development of the number of observed *PMCS* of length three or shorter over the time driven, while Figure 7 shows the development for all sequences.

## 4.2 Common Scenarios

To answer *Research Question 2*, common patterns of maneuver sequences are identified and their frequencies of occurrence are compared. This way, commonly occurring elements of scenario layer 4 [16] can be found and subsequently be turned into *Functional Scenarios*[6] for simulation or proving ground tests. To this end, Figure 8 shows the 15 *Parallel Maneuver Combination Sequences* with the highest number of occurrences in the database.

## 5 Discussion

In the first research question of this paper, the goal is formulated to estimate the amount of recorded data after which repetitions in the observed maneuver data make the collection of more data inefficient. When looking at the analysis of the *Maneuver Layer*, a qualitative evaluation of the results in Figure 4 shows the occurrence of saturation behavior with

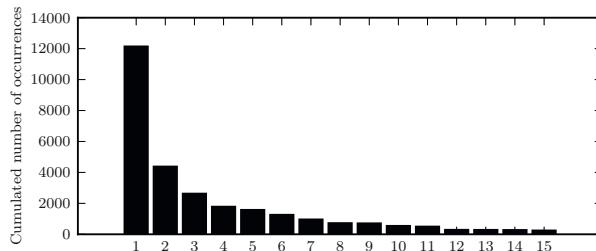


Figure 8: 15 Parallel Maneuver Combination Sequences (PMCS) with the highest number of occurrences in the database

respect to the distribution of initial velocities  $v_0$  during *Turn Left* maneuvers. As shown by Wang et al. [11], the detection of saturation can be quantified by introducing a threshold parameter  $\epsilon$ . The dataset is then considered saturated if none of the remaining KL-divergence values exceeds  $\epsilon$ . Therefore a discussion should be held if and for what value of  $\epsilon$  this kind of saturation effect should be incorporated into the overall validation process.

While saturation behavior was observable for the investigated example of initial velocities during *Turn Left* maneuvers after around 80 samples, this does not necessarily apply to other parameters or other maneuvers for that matter. The introduced methodology could be applied to a wider range of parameters to research similar effects for these maneuver properties.

Analysis on the *Behavior Level* showed no significant saturation effects in the data quantities that were investigated in this paper. In fact, a linear regression for the number of unique PMCS  $n_{PMCS}$  over the amount of observed objects  $n_{objects}$  yields  $R^2 \approx 0.99$ , indicating a nearly linear relationship between the two variables. This absence of saturation effects in the observed behavior could have multiple causes. On the one hand, it is possible that the quantity of analyzed data was simply not sufficient. Since we are not looking at entire scenarios, but rather at individual object behavior as an element of scenario layer 4 [17], it is difficult to compare the obtained results to estimations of the amount of required scenarios such as by Amersbach and Winner [7] to assess plausibility. Another possibility would be that the concept of PMCS is not well suited to compare object behavior. This cannot be ruled out but is viewed to be unlikely since it introduces a significant abstraction from the actual concrete behavior. Therefore an overestimation of the actual traffic complexity should not take place in our current opinion. Nevertheless, our approach manages to quantify the variability of the underlying dataset by differentiating the sequences of abstract maneuvers from each other. Thereby, we hope to further contribute to the understanding of large traffic datasets with this approach, which could not only be applicable for the verification and validation of automated vehicles, but also in training data selection for machine learning algorithms.

Another important factor influencing the saturation results are the classification algo-

rithms for the individual maneuvers. Since recognizing them reliably in noisy measurement data is not a trivial task, the output can generally include false positive as well as false negative detections. The classification algorithms written in the context of this publication are considered to be experimental software that could be subject to bugs. Therefore, the results should be treated with care, as future improvements could also influence the statistical findings discussed in this paper. After manual inspection, we do however believe the general trend of our results to remain, even if the implementation of the classification algorithms could change in the future.

Looking at the second research question concerning the identification of commonly occurring scenarios, only a partial answer can be given. By looking at the occurrence of different PMCS in Figure 8, common elements of scenario layer 4, the movable objects, can be identified. For example, a common sequence that was identified this way was the combination of a *Cross Junction* maneuver with a simultaneous *Keep Velocity* maneuver. The most common sequence was a single *Standstill* maneuver. This could be an indication to increase testing efforts with respect to this behavior, since there seems to be a high exposure to it, at least in the dataset that was analyzed in this work.

These analyzed sequences of maneuvers only represent one element of scenario layer 4, the movable objects. They can however be used as building blocks in the construction of test scenarios. For example, a strategy could be to construct scenarios that include one of the most occurring PMCS to test the automated vehicle's reaction to this common behavior.

## 6 Conclusion and Future Work

In this paper, we have investigated the occurrence of saturation effects in collected measurement data for the verification and validation of automated driving. While our analysis of individual maneuver parameters indicated saturation behavior for the chosen exemplary maneuver parameter, more research is necessary to determine the applicability of the obtained quantitative statements for other parameters and a wider range of maneuvers. The investigation should also be extended to include parameter correlations to more accurately reflect the complex nature of urban traffic. On the *Behavior Layer*, saturation of the observed maneuver sequences could not be determined. Future work will likely focus on investigating larger quantities of data. It was, however, possible to determine commonly occurring elements of *Functional Scenarios* for the analyzed test drives in the form of *Parallel Maneuver Combination Sequences (PMCS)*. These could be a valid starting point to further refine and construct a data-driven set of scenarios for the test of urban automated vehicles.

Finally, the effect of route choice on the qualitative and quantitative structure of the classified maneuver data will be analyzed in the future to further facilitate the practical application of the proposed methodology in the verification and validation process of automated vehicles.

## Acknowledgment

We would like to thank Sven Chlosta, Stefan Diddens, Ting Fu, Hansang Lee, Asis Das, Heden Qian and Zhijing Zhu for their help in implementing the proposed concepts.

## References

- [1] H. Winner, W. Wachenfeld, and P. Junietz, “(How) can safety of automated driving be validated?” 2016, presentation at Virtual Vehicle Symposium Graz. [Online]. Available: [https://www.fzd.tu-darmstadt.de/media/fachgebiet\\_fzd/publikationen\\_3/2016\\_5/2016\\_Wi\\_Wf\\_Ju\\_ViV-Symposium\\_Graz.pdf](https://www.fzd.tu-darmstadt.de/media/fachgebiet_fzd/publikationen_3/2016_5/2016_Wi_Wf_Ju_ViV-Symposium_Graz.pdf)
- [2] “Pegasus Research Project,” <https://www.pegasusprojekt.de/en/home>, accessed: 2019-04-05.
- [3] L. Hartjen, F. Schuldt, and B. Friedrich, “Semantic classification of pedestrian traffic scenarios for the validation of automated driving,” in *2019 22nd IEEE International Conference on Intelligent Transportation Systems (ITSC)*. Auckland, New Zealand: IEEE, 2019, pp. 3696–3701.
- [4] W. Wachenfeld and H. Winner, “Die Freigabe des autonomen Fahrens,” in *Autonomes Fahren*, M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, Eds. Springer Vieweg, 2015, pp. 439–464.
- [5] F. Hauer, T. Schmidt, B. Holzmüller, and A. Pretschner, “Did we test all scenarios for automated and autonomous driving systems?” in *2019 22nd IEEE International Conference on Intelligent Transportation Systems (ITSC)*. Auckland, New Zealand: IEEE, 2019, pp. 2950–2955.
- [6] T. Menzel, G. Bagschik, and M. Maurer, “Scenarios for development, test and validation of automated vehicles,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*. Changshu, Suzhou, China: IEEE, 2018, pp. 1821–1827.
- [7] C. Amersbach and H. Winner, “Defining required and feasible test coverage for scenario-based validation of highly automated vehicles,” in *2019 22nd IEEE International Conference on Intelligent Transportation Systems (ITSC)*. Auckland, New Zealand: IEEE, 2019, pp. 425–430.
- [8] J. Langner, J. Bach, L. Ries, S. Otten, M. Holzapfel, and E. Sax, “Estimating the uniqueness of test scenarios derived from recorded real-world-driving-data using autoencoders,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, Suzhou, China, 2018, pp. 1860–1866.
- [9] M. Mauritz, F. Howar, and A. Rausch, “Assuring the safety of advanced driver assistance systems through a combination of simulation and runtime monitoring,” in *Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications*. Corfu, Greece: Springer International Publishing, 2016, pp. 672–687.

- [10] R. Krajewski, T. Moers, D. Nerger, and L. Eckstein, “Data-driven maneuver modeling using generative adversarial networks and variational autoencoders for safety validation of highly automated vehicles,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, Hawaii, USA, 2018, pp. 2383–2390.
- [11] W. Wang, C. Liu, and D. Zhao, “How much data are enough? a statistical approach with case study on longitudinal driving behavior,” *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 2, pp. 85–98, June 2017.
- [12] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [13] L. Hartjen, R. Philipp, F. Schuldt, F. Howar, and B. Friedrich, “Classification of driving maneuvers in urban traffic for parametrization of test scenarios,” accepted for publication in 2019 Tagung Automatisiertes Fahren, Munich, Germany, 2019.
- [14] OpenStreetMap contributors, “Planet dump retrieved from <https://planet.osm.org/>,” <https://www.openstreetmap.org>, 2019.
- [15] D. W. Scott, “On optimal and data-based histograms,” *Biometrika*, vol. 66, no. 3, pp. 605–610, 12 1979. [Online]. Available: <https://doi.org/10.1093/biomet/66.3.605>
- [16] F. Schuldt, “Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen - english title: Towards testing of automated driving functions in virtual driving environments,” Ph.D. dissertation, Technische Universität Braunschweig, 2017.
- [17] G. Bagschik, T. Menzel, and M. Maurer, “Ontology based scene creation for the development of automated vehicles,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, Suzhou, China, 2018, pp. 1813–1820.

## Verifikation von HD-Karten mittels räumlich und semantisch separierbarer 1D-Signale

Jan-Hendrik Pauls\*, Tobias Strauß†, Carsten Hasberg†,  
Martin Lauer\* und Christoph Stiller\*

**Zusammenfassung:** Für automatisierte Fahrfunktionen haben sich Karten sowohl als perfekte Ergänzung als auch als Rückfallebene für Sensordaten erwiesen. Gerade hochauflöste Karten können jedoch sehr schnell veralten.

In teilweise geänderten Umgebungen ist eine Lösung des Lokalisierungsproblems schwierig oder uneindeutig – insbesondere wenn diese strukturiert oder periodisch sind. Um dennoch die Aktualität einer Karte sicherstellen zu können, werden in diesem Beitrag 1D-Signale vorgestellt, die sich nur selten oder geringfügig entlang einer Straße ändern, aber signifikante Änderungen aufweisen, sobald sich die kartierte Umgebung gegenüber der Karte geändert hat.

Drei beispielhafte Signale werden durch die Kombination zweier Verfahren – einer räumlich-semantischen Gruppierung und einer ordnungsinduzierenden Projektion – sowohl aus punkt- als auch aus linienförmigen Landmarken hergeleitet.

**Schlüsselwörter:** Aktualitätsverifikation, Automatisiertes Fahren, HD-Karten, Kartenverifikation

## 1 Einleitung

In den vergangenen Jahren gab es sowohl im Bereich der Sensorik als auch bei den Algorithmen, die Sensordaten verarbeiten, große Fortschritte. Dennoch scheinen hochauflöste Karten (HD-Karten) bislang unerlässlich für (hoch-)automatisierte Fahrfunktionen [1]. Gegenüber Sensorik zeichnen sich Karten durch eine unbegrenzte Reichweite, eine quasi beliebige Auflösung, eine völlige Unabhängigkeit von möglichen Verdeckungen sowie durch eine vorab verifizierbare Korrektheit der Repräsentation [2] aus. Im Gegensatz zu Sensoren, die ein geringfügig verzögertes Bild der Umgebung liefern, enthalten Karten mitunter Jahre altes Kartenmaterial. Hierdurch ergibt sich der Fehlerfall von Karten: Sie veralten.

Insbesondere HD-Karten, also Karten, die nicht nur den Straßenverlauf, sondern auch detailliertere Merkmale wie Fahrbahnmarkierungen, Ampeln, Verkehrsschilder oder gar spezifische Verkehrsregeln enthalten, veralten sehr schnell. Je mehr Details eine Karte enthält, desto größer die Wahrscheinlichkeit, dass sich auch nur ein Merkmal ändert.

Während die oben beschriebenen, komplementären Eigenschaften HD-Karten zur perfekten Ergänzung sowie einer möglichen Rückfallebene für Sensordaten machen, muss

\*Institut für Mess- und Regelungstechnik am Karlsruher Institut für Technologie (KIT), 76131 Karlsruhe (e-mail: {pauls,martin.lauer,stiller}@kit.edu).

†Robert Bosch GmbH, 74232 Abstatt. (e-mail: {tobias.strauss, carsten.hasberg}@de.bosch.com)

damit auch die Korrektheit, d. h. die Aktualität, der verwendeten Karte sichergestellt werden. In vorhergehenden Veröffentlichungen [3] haben wir zwischen Kartenvorläufigkeit, also der anwendungsspezifischen Absicherung einer Karte, und Kartenverifikation, also einem merkmalsbasierten Abgleich von Karte und kartierter Umgebung, unterschieden. Dieser Beitrag befasst sich mit der Verifikation einer HD-Karte auf der Detailebene von semantischen Merkmalen bzw. Merkmalsgruppen.

Voraussetzung für einen Abgleich zwischen Kartendaten und der durch Sensorik wahrgenommenen Umgebung ist eine korrekte Lokalisierung. Für den Fall einer eindeutigen und hochgenauen Lokalisierung wurde bereits ein Verfahren zur Kartenverifikation vorgestellt [4]. Schnellstraßen und Autobahnen sind sehr strukturierte Umgebungen, in denen Merkmale oft periodisch oder in sich wiederholenden Mustern auftreten. Dadurch ist eine solch eindeutige Lokalisierung vor allem entlang der Fahrtrichtung nicht immer gegeben – insbesondere, wenn Teile der Karte veraltet sind. In lateraler Richtung kann hingegen eine Lokalisierung bis hin zum korrekten Fahrstreifen angenommen werden [5]. Dies wirft die Frage auf, wie man unter diesen speziellen Voraussetzungen dennoch die Aktualität einer HD-Karte sicherstellen kann.

In diesem Beitrag werden zur Lösung dieses Problems 1D-Signale vorgeschlagen, die sowohl aus stückweise quasi-konstanten Umgebungsmerkmalen, wie der Asphaltähnlichkeit, aber insbesondere – wie in diesem Beitrag – auch aus stückweise quasi-konstant periodischen Landmarken wie (gestrichelten) Fahrbahnmarkierungen oder den Aufhängepfosten von Schutzplanken abgeleitet werden können. Allgemein können all solche Signale gewählt werden, die sich entlang der Fahrtrichtung nur langsam oder selten ändern, bei Änderungen der kartierten Umgebung gegenüber der Karte jedoch signifikante Änderungen aufweisen.

## 2 Stand der Technik

Der Stand der Technik unterscheidet sich nach der Art der zu verifizierenden Karte. Dudek u. a. [6] beschäftigten sich erstmals mit der Verifikation einer Karte, die durch einen Graphen repräsentiert wurde. Wenngleich die Arbeit nicht ansatzweise den Umfang von HD-Karten abdeckt, so kann sie doch auf den oft in HD-Karten hinterlegten Routing-Graphen angewandt werden.

Spätere Arbeiten befassten sich mit Karten, wie sie für Navigations-, Komfort- und Fahrerassistenzanwendungen üblich sind. Zinoune u. a. [7, 8, 9] verifizierten eine Navigationskarte mit Befahrungen durch ein Fahrzeug, um beispielsweise neu gebaute Kreisverkehre einzufügen. Diese Anwendung ist sehr ähnlich zur Arbeit von Raaijmakers und Bouzouraa [10], die mittels Lidar und einer Karte neu gebaute Kreisverkehre detektierten und schätzten. In seiner Dissertation stellte Raaijmakers darüber hinaus das erste uns bekannte probabilistische Modell zur Verifikation von Karten vor [11]. Hartmann u. a. [12] fusionierten verschiedene Sensordaten mit abgetasteten Geometrieinformationen einer Karte in einem neuronalen Netz, um die Aktualität der Karte zu verifizieren.

Karten in der mobilen Robotik enthalten oft nur Landmarken, die eine globale Lokalisierung erlauben. Für unstrukturiert verteilte Landmarken beschäftigten sich Stübner u. a. mit einem Konsistenzmaß [13] sowie der Anwendung dessen für langzeitstabile Karten [14]. Ebenfalls im Bereich der Robotik ist außerdem das Forschungsfeld langzeitstabiler Karten und zugehöriger Lokalisierungsverfahren zu verorten. Hier wird das Problem der Karten-

verifikation allerdings nicht explizit gelöst.

Die einzigen bemerkenswerten Ausnahmen sind die Ansätze von Jo u. a. [4] sowie von Pannen u. a. [15]. Pannen u. a. schlugen einen bzw. zwei Partikelfilter zur Lokalisierung vor, aufgrund deren Partikel ein Klassifizierer die Aktualität der Karte beurteilt. Hiervon unterscheidet sich der in diesem Beitrag vorgestellte Ansatz, da verifizierte Teile der Karte weiterhin verfügbar sind.

Jo u. a. erweiterten das klassische Problem der gleichzeitigen Lokalisierung und Kartierung (SLAM) um eine auf der Dempster-Shafer-Theorie basierende Update-Komponente. Dies setzt aber immer noch eine eindeutige, hochgenaue Lokalisierung voraus, da eine punktweise Assoziation zwischen Karten- und detektierten Elementen notwendig ist. Diese ist unter der in strukturierten Umgebungen nicht unbegründeten Annahme einer Fehllokalisierung nicht mehr gangbar, da die Assoziationskomplexität mit zunehmender Unsicherheit explodiert.

### 3 Räumlich-semantische Merkmalsgruppierung

Wie anfangs erwähnt können aus bereits stückweise quasi-konstanten Umgebungsmerkmalen direkt passende Signale abgeleitet werden. Typischerweise sind solche Umgebungsmerkmale aber nicht Teil aktueller Karten oder Datensätze. Daher wird in diesem und den folgenden Kapiteln gezeigt, wie aus beispielhaften Landmarken dennoch passende Signale generiert werden können.

Die Eingangsdaten für das vorgestellte Verfahren sind somit semantisch klassifizierte punkt- oder linienförmige Landmarken, beispielsweise Schutzplankenpfosten (SPF) oder Fahrbahnmarkierungen (FBM) sowie eine lateral fahrstreifengenaue, longitudinal aber unsichere Lokalisierung. Weiterhin wird die Fahrstreifentopologie als unverändert oder rekonstruiert angenommen. Dies kann mit dem Stand der Technik detektiert und sogar kompensiert werden [16]. Eine Einschränkung ist bisher noch, dass die eingehenden Landmarken als perfekt kartiert bzw. detektiert angenommen wurden, d. h. es gibt weder Messfehler noch Fehldetections.

Basierend auf diesen Landmarkendaten und der geschätzten Fahrzeugposition können nun Trenngrenzen (sog. Separatoren) eingeführt werden, die die semantisch bereits klassifizierten Landmarken auch räumlich gruppieren. Hierbei können durchgezogene FBM sowie SPF auf den meisten Autobahnabschnitten durch nur einen Separator in der Fahrbahnmitte gruppiert werden. Gestrichelte FBM sind jeweils zwischen zwei Fahrstreifen zu finden, sodass hier je Fahrstreifen ein Separator in der Fahrstreifenmitte eingebracht werden kann. Das Verfahren ist in Abb. 1 dargestellt und hat semantisch greifbare Merkmalsgruppen wie „rechte gestrichelte Fahrbahnmarkierungen“ als Ergebnis.

### 4 Ordnungsinduzierende Dimensionalitätsreduktion

Wenngleich aufgearbeitete Kartendaten bereits eine Ordnung besitzen können, sind insbesondere Merkmalspunkte aus Sensordaten im Allgemeinen noch ungeordnet. Um wohldefinierte 1D-Signale auf den Merkmalen definieren zu können, ist aber eine Ordnungsrelation notwendig. Diese lässt sich durch Projektion auf eine Referenzlinie, besser noch auf einen durch Referenzpunkte gelegten Referenzspline, induzieren. Als Referenzpunkte eignen sich

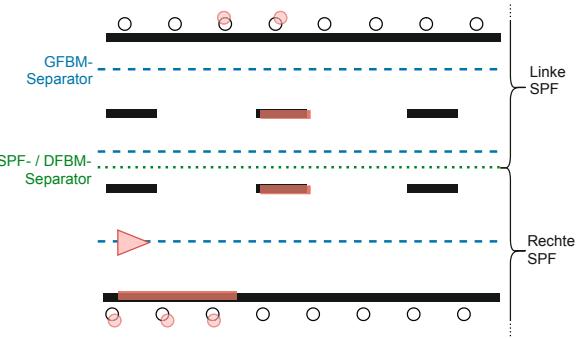


Abbildung 1: Illustration der räumlichen Gruppierung semantischer Merkmalsgruppen. Dargestellt ist die Kartenrepräsentation (schwarz), die aktuelle Pose mit zugehörigen Sensordaten (rot) sowie Separatoren für gestrichelte FBM (GFBM; blau) und SPF bzw. durchgezogene FBM (DFBM; grün). Rechts sind zwei der sich dadurch ergebenden Merkmalsgruppen angedeutet.

sowohl in der Karte verortete Geometrien wie Punkte auf der linken durchgezogenen FBM als auch dynamisch eingeführte Punkte, die beispielsweise durch die Egopositionen definiert sein können.

Nun können die Merkmale  $\tilde{f}_k$  jeder Merkmalsgruppe  $\tilde{F}_k$  durch Projektion auf den Referenzspline geordnet werden. Hierbei ist zu beachten, dass sich ein generisches Optimierungsverfahren [17] numerisch stabiler zeigte als die vermeintlich optimale Lösung für dieses Problem [18]. Die Projektion erfolgt somit durch Minimierung des Gütfunktionalen  $J(t_i, i) = \|\tilde{f}_i - \mathcal{S}(t_i)\|$ , das den Abstand zwischen dem kartesischen Merkmalspunkt  $\tilde{f}_i = (x_i, y_i) \in \tilde{F}_k$  und dem entsprechenden Projektionsaufpunkt  $\mathcal{S}(t_i)$  minimiert. Dies überführt jeden Merkmalspunkt in ein Frenet-Koordinatensystem definiert durch die Splinekoordinate  $t_i$  sowie den orientierten Abstand  $u_i$ :  $f_i = (t_i, u_i)$ .

Dies wiederum induziert nun die Ordnungsrelation  $<_t$ , sodass man die Merkmalspunkte in  $\tilde{f}_i \in \tilde{F}_k$  nun als wohlgeordnete Menge  $\mathcal{F}_k$  auffassen kann.

### 5 1D-Signale

Nachdem aus ungruppierten und ungeordneten Merkmalspunkten nun geordnete Merkmalsmengen hergeleitet wurden, können in diesem Kapitel 1D-Signale definiert werden, die die Eigenschaft haben, dass sie sich bei Fehlern in der longitudinalen Lokalisierung kaum, bei Änderungen der Umgebung jedoch signifikant ändern. Hierbei sind die vorgestellten Signale nur beispielhaft. Weitere Beispiele werden in Kapitel 5.4 eingeführt.

Die bisher angedachten Signale lassen sich in drei Kategorien ordnen: *Existenz* ist ein notwendiges Meta-Signal, ohne das alle weiteren Signale nicht mehr wohldefiniert sind. Sogenannte *intrinsische* Signale sind innerhalb nur einer Merkmalsgruppe definiert. Beispiele sind der Abstand zweier gestrichelter FBM oder deren Länge. *Extrinsische* Signale hingegen setzen zwei verschiedene Merkmalsgruppen zueinander in Beziehung.

## 5.1 Existenz

Grundlage aller Signale ist die sogenannte *Existenz*  $\exists$ . In unserer Implementierung ist die Existenz als binäres Signal über ein Fenster  $w$  in Spline-Koordinaten für eine geordnete Merkmalsmenge  $\mathcal{F}_k$  und eine Spline-Koordinate  $t$  definiert:

$$\exists(\mathcal{F}_k, t, w) := \begin{cases} 1 & \left( \sum_{\{f_i \in \mathcal{F}_k : t_i \in [t-w, t+w]\}} 1 \right) \geq T_\exists \\ 0 & \text{sonst} \end{cases} \quad (1)$$

Hierbei gilt für perfekte Eingangsdaten  $T_\exists = 1$ , während für mit Fehldetections behaftete Daten ein größerer Schwellwert zu wählen ist.

## 5.2 Intrinsische Signale

In diesem Beitrag werden zwei intrinsische Signale vorgestellt. Das erste ist der longitudinale Abstand  $\mathcal{D}_{\text{long}}$  zweier aufeinanderfolgender Merkmalspunkte:

$$\begin{aligned} \mathcal{D}_{\text{long}}(\mathcal{F}_k, t) &:= t_j - t_i \\ \text{u.d.N. }^1 i &= \arg \max_{l \in \mathcal{F}_k : t_l \leq t} (t_l) \\ j &= \arg \min_{l \in \mathcal{F}_k : t_l > t} (t_l). \end{aligned} \quad (2)$$

Für punktförmige Merkmale wie SPF ist diese Definition trivial. Für linienförmige oder anderweitig ausgedehnte Merkmale muss ein konsistenter Referenzpunkt gefunden werden. Bei gestrichelten FBM kann dies beispielsweise der Anfangspunkt oder der Endpunkt der jeweiligen Markierung sein.

Für ausgedehnte Merkmale ergeben sich dafür weitere mögliche Signale. Eines davon ist die kontinuierliche Länge  $\mathcal{L}$  einer gestrichelten FBM-Gruppe, trivial definiert durch die Längen  $\ell_i$  der einzelnen Markierungen. Diese kann entweder entlang dem Referenzspline linear interpoliert werden oder stückweise konstant sein:

$$\begin{aligned} \mathcal{L}(\mathcal{F}_k, t) &:= \|\ell_i\| \\ \text{u.d.N. } i &= \arg \max_{l \in \mathcal{F}_k : t_l \leq t} (t_l). \end{aligned} \quad (3)$$

## 5.3 Extrinsische Signale

Extrinsische Signale setzen zwei Merkmalsgruppen zueinander in Beziehung. Ein Beispiel hierfür ist die laterale Distanz  $\mathcal{D}_{\text{lat}}$ , die für zwei geordnete Merkmalsmengen  $\mathcal{F}_k$  und  $\mathcal{F}_l$  an jeder Koordinate  $t$  definiert ist:

$$\begin{aligned} \mathcal{D}_{\text{lat}}(\mathcal{F}_k, \mathcal{F}_l, t) &:= \|u_j - u_i\| \\ \text{u.d.N. } i &= \arg \min_{m \in \mathcal{F}_k} (\|t_m - t\|) \\ j &= \arg \min_{m \in \mathcal{F}_l} (\|t_m - t\|). \end{aligned} \quad (4)$$

Hierbei ist  $u_i$  bzw.  $u_j$  die zum Referenzspline orthogonale Koordinate.

<sup>1</sup>unter den Nebenbedingungen

## 5.4 Generalisierung

Neben den bereits vorgestellten 1D-Signalen sind zahlreiche weitere denkbar. Neben den vorgestellten Signalen für periodische Umgebungen sind hierbei insbesondere Signale denkbar, die bereits stückweise quasi-konstant bzw. nur sehr langsam veränderlich als kontinuierliche Größen vorkommen. Beispiele hierfür sind Asphalthelligkeiten, Breiten von Markierungen und Bordsteinen, aber auch Höhen von Bordsteinen oder Schutzplanken.

## 6 Kartenverifikation und Änderungsdetektion

Schließlich können nun durch einen direkten Vergleich der Karten- mit den Messdatensignalen Änderungen in der Umgebung detektiert werden. Hierfür wird, wie in Abb. 2 dargestellt, ein dreistufiger Ansatz verfolgt. In einem ersten Schritt wird die Existenz derselben Merkmalsgruppe in Karte und Sensordaten geprüft. Ist diese für keine der beiden gegeben, so ist die Karte trivialerweise verifiziert. Stimmt die Existenz nicht überein, so kann eine Änderung detektiert werden. Existiert eine Merkmalsgruppe sowohl in der Karte als auch in den Sensordaten, so wird zur zweiten Stufe übergegangen. Hier werden die intrinsischen Signale auf Übereinstimmung geprüft. Dies geschieht mit einem Schwellwertverfahren auf der Differenz der Signale. Wurde weiterhin keine Änderung detektiert, so kommt die Merkmalsgruppe in die dritte Stufe. Hier werden nun potentiell noch valide Merkmalsgruppen auf ihre extrinsischen Signale geprüft. Dies geschieht ebenfalls durch ein Schwellwertverfahren.

Nach drei Schritten können so einige Merkmalsgruppen als bereits geändert angesehen werden. Für andere ist dies noch nicht klar, da eine konsistente Verschiebung in Fahrtrichtung prinzipiell nicht festgestellt werden kann, ohne sich mit einer punktweisen Assoziation longitudinal zu lokalisieren. Es kann aber für einen anschließenden punktweisen Abgleich die Menge der auszuwertenden Merkmale teils erheblich eingeschränkt werden, da alle bereits als geändert bekannten Merkmalsgruppen ausgeschlossen werden können.

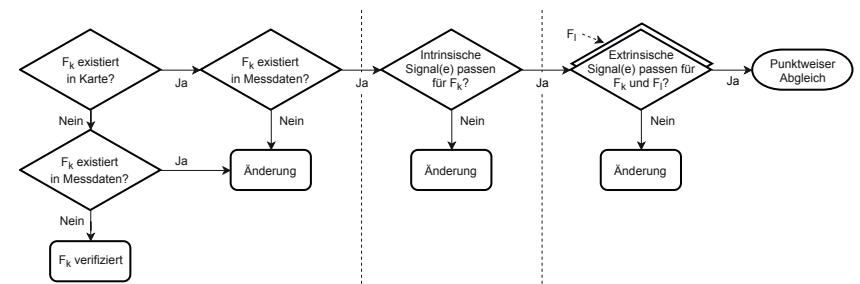


Abbildung 2: Schema zur Änderungsdetektion. Gestrichelte Linien markieren die Abtrennung der drei Detektionsstufen Existenz, intrinsische Signale und extrinsische Signale.

## 7 Evaluierung

Da in diesem Beitrag stets perfekte Karten- und Sensordaten angenommen werden, konnte die Evaluierung direkt auf einem bereits publizierten Datensatz [3] erfolgen. Da das Label-Rauschen im Datensatz in lateraler Richtung in derselben Größenordnung wie die zu beobachteten Änderungen liegt, wurden jedoch nur die ersten beiden Stufen der in Kapitel 6 beschriebenen Änderungsdetektion ausgewertet. Dies wäre bei automatisierter Erfassung der Merkmale nicht zu erwarten.

Als Metriken dienen die Richtig-Positiv-Rate und die Falsch-Positiv-Rate auf den vermeintlich<sup>2</sup> perfekten Daten. Abb. 3 zeigt beide über einen simulierten, zeitlich konstanten Lokalisierungsfehler. Wie anfangs beschrieben ist dieser Lokalisierungsfehler rein longitudinal, d. h. entlang der Fahrtrichtung.

Hierbei konnten stets 49 % bis 98 % aller Änderungen erkannt werden. Gleichzeitig blieb die Falsch-Positiv-Rate meist bei unter 15 %. Bei Lokalisierungsfehlern von 20 m wären die allermeisten punktweise arbeitenden Verfahren längst funktionsunfähig.

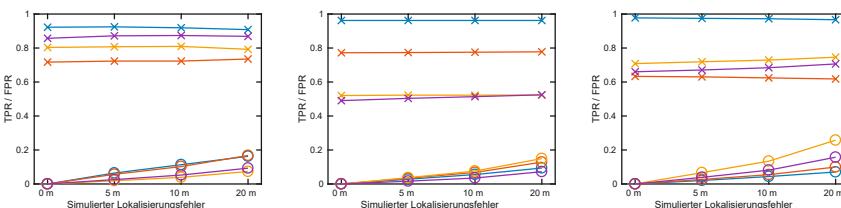


Abbildung 3: Evaluierung auf drei Autobahnabschnitten. Abgebildet sind Richtig- ( $\times$ ) und Falsch-Positiv-Rate ( $\circ$ ) der linken/rechten SPF (blau/rot) sowie der linken/rechten gestrichelten FBM (gelb/lila) über ansteigendem simuliertem Lokalisierungsfehler.

Gerade bei den SPF kann bereits aufgrund dem Meta-Signal *Existenz* ein Großteil der Änderungen erkannt werden, da diese zwischen den Fahrbahnen mehr und mehr durch Betonrückhaltesysteme ersetzt werden.

Falsch-positive Änderungsdetektionen treten typischerweise an den Übergängen auf, in denen die lokale Konstanz der Signale nicht mehr gegeben ist. Dies ist dem Ansatz geschuldet und wurde entsprechend erwartet.

Die fehlenden richtig-positiven Änderungsdetektionen stammen von longitudinalen Verschiebungen der Merkmale. Diese sind mit dem vorgestellten Ansatz nicht zu beobachten, sondern erfordern – ob nun direkt oder indirekt – eine punktweise Assoziation, die stets einer Lokalisierung gleichkommt. Diese sollte im Anschluss an das vorgestellte Verfahren auf den potentiell noch korrekten Signalen/Merkmalen erfolgen.

## 8 Fazit und Ausblick

In diesem Beitrag wurde ein Ansatz vorgestellt, der eine Aktualitätsverifikation von HD-Karten erlaubt, obwohl teils erhebliche Lokalisierungsfehler – wie beim aktuellen Stand

<sup>2</sup>Falsch-Positiv-Detektionen aufgrund von Label-Fehlern wurden ignoriert und die Falsch-Positiv-Raten entsprechend kompensiert.

der Technik auf Autobahnen und Schnellstraßen zu erwarten – vorliegen. Der Ansatz nutzt stückweise quasi-konstante 1D-Signale, die sich longitudinal kaum, im Falle von Änderungen der kartierten Umwelt aber signifikant ändern. Neben bereits in dieser Form vorliegenden Signalen wurde in diesem Beitrag darüber hinaus vorgestellt, wie strukturiert oder periodisch auftretende Landmarken wie Schutzplankenpfosten oder Fahrbahnmarkierungen in eine solche Signalform gebracht werden können.

An einem Verfahren zur Rekonstruktion periodischer Signale auch aus verrauschten bzw. mit Fehldetections behafteten Sensordaten wird aktuell geforscht. Darüber hinaus ergibt sich fast schon natürlich die Frage, ob direkt im Signalraum das SLAM-Problem gelöst werden kann. Dies geht auch damit einher, dass die unveränderten Teile der Karte zur Lokalisierung genutzt werden, um die geänderten Teile zu aktualisieren. Damit stellt sich schließlich die Frage, wie man geänderte Teile der Karte – ob nun als Landmarken oder als Signale – probabilistisch aktualisieren kann.

## Literatur

- [1] F. Poggenhans u. a. „Lanelet2: A high-definition map framework for the future of automated driving“. In: *21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, S. 1672–1679.
- [2] F. Poggenhans *Generierung hochdetaillierter Karten für das automatisierte Fahren*, Dissertation, Karlsruher Institut für Technologie (KIT), 2019.
- [3] J. Pauls u. a. „Can We Trust Our Maps? An Evaluation of Road Changes and a Dataset for Map Validation“. In: *21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, S. 2639–2644.
- [4] K. Jo u. a. „Simultaneous Localization and Map Change Update for the High Definition Map-Based Autonomous Driving Car“. In: *Sensors*, Vol. 18, No. 9, S. 3145, Sep. 2018.
- [5] M. Harr u. a. „Robust Localization on Highways Using Low-Cost GNSS, Front/Rear Mono Camera and Digital Maps“. In: *AME 2018 - Automotive meets Electronics; 9th GMM-Symposium*, Mär. 2018, S. 1–7.
- [6] G. Dudek u. a. „Map validation and self-location in a graph-like world“. In: *Int. Joint Conf. Artificial Intell.*, Aug. 1993, S. 1648–1653.
- [7] C. Zinoune u. a. „Detection of missing roundabouts in maps for Driving Assistance Systems“. In: *2012 IEEE Intelligent Vehicles Symposium*, Jun. 2012, S. 123–128.
- [8] C. Zinoune u. a. „A sequential test for autonomous localisation of map errors for driving assistance systems“. In: *15th Int. Conf. Intell. Transp. Syst. (ITSC)*, Sept. 2012, S. 1377–1382.
- [9] C. Zinoune u. a. „Sequential FDIA for Autonomous Integrity Monitoring of Navigation Maps on Board Vehicles“. In: *IEEE Trans. Intell. Transp. Syst.*, Jan. 2016, S. 143–155.
- [10] M. Raaijmakers und M. E. Bouzouraa „In-vehicle Roundabout Perception Supported by A Priori Map Data“. In: *18th Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2015, S. 437–443.

- [11] M. Raaijmakers *Towards environment perception for highly automated driving: with a case study on roundabouts*, Dissertation/Prüfschrift, TU Eindhoven, 2017.
- [12] O. Hartmann u. a. „Robust and real-time multi-cue map verification for the road ahead”. In: *17th Int. Conf. Intell. Transp. Syst. (ITSC)*, Okt. 2014, S. 894–899.
- [13] M. Stübner u. a. „Consistency of feature-based random-set Monte-Carlo localization”. In: *2017 Eur. Conf. on Mobile Robots*, Sep. 2017, S. 1–6.
- [14] M. Stübner u. a. „A continuously learning feature-based map using a bernoulli filtering approach”. In: *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Okt. 2017, S. 1–6.
- [15] D. Pannen u. a. „HD Map Change Detection with a Boosted Particle Filter”. In: *2019 Int. Conf. Robot. Autom. (ICRA)*, Mai 2019, S. 2561–2567.
- [16] F. Dierkes u. a. „Towards a Multi-Hypothesis Road Representation for Automated Driving”. In: *18th Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2015, S. 2497–2504.
- [17] J. Lagarias u. a. „Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions”. In: *SIAM Journal on Optimization*, Band 9, Jan. 1998, S. 112 – 147.
- [18] H. Wang u. a. „Robust and efficient computation of the closest point on a spline curve”. In: *5th Int. Conf. Curves and Surfaces*, 2002, S. 397–406.

## Functional Decomposition of Automated Driving Systems for the Classification and Evaluation of Perceptual Threats

Robin Philipp\*, Fabian Schuldt\* and Falk Howar†

**Abstract:** This contribution investigates dependability threats to automated driving systems pertaining to the environment perception. The identification of factors that can lead to safety-relevant system failures is essential for assuring safety of automated driving systems. We establish a comprehensive taxonomy for the classification of perceptual threats based on a functional decomposition of automated driving systems. Moreover, we use an exemplary lane keeping assistance system to describe different types of threats by using the taxonomy. The proposed taxonomy enables the opportunity for future work on a safety validation concept for perception components.

**Keywords:** Automated Driving Systems, Dependability, Functional Decomposition, Perception

## 1 Introduction

Safety validation of automated driving systems is a challenge that needs to be met for the introduction of self driving vehicles on public roads. However, safety validation concepts for higher<sup>1</sup> automation of vehicles are yet to be developed. One approach that is already present in other domains and currently researched for automated driving systems is the method of functional decomposition [2]. Instead of verifying the complex system as a whole, the verification of less complex single components is examined. Shifting from vehicle level verification to component level verification offers the advantage to apply more specific verification methods for different components. The verification process therefore gains more manageability and flexibility. However, a downside to a decomposition-based verification is that threats have to be accounted for separately, which are not safety-relevant for a single component, but can become safety-critical when propagating along the following components.

Regarding an automated driving system the verification of the perception component is challenging and therefore of special interest. The perception component must guarantee detection of all relevant objects with a certain quality in a fixed time interval to ensure

---

\*Robin Philipp and Fabian Schuldt are with Volkswagen Group Innovation, Wolfsburg, Germany (robin.philipp@volkswagen.de).

†Falk Howar is a Professor at the Chair for Software Engineering, Technische Universität Dortmund, Germany.

<sup>1</sup>An automated driving system with level 3 or higher as defined by the SAE [1]

safe behavior of the self driving vehicle in every possible scenario [3]. This requirement, however, is still too vague to be tested considering that there is an infinite amount of possible scenarios that can happen. In order to formulate meaningful requirements for environmental perception of an automated driving system it is essential to identify possible threats for a perception component, where they can potentially originate from and what influence they can have on the whole system performance. Moreover the identification of possible threats can enable a better understanding of dependability threats a perception component should handle itself, which threats should be handled by subsequent processing and which threats should not occur.

In this work, we establish a taxonomy for perceptual threats to automated driving systems. We characterize perceptual threats by functionally decomposing environmental perception components into its constituent processing parts. The resulting interfaces of the decomposed parts can then be used to derive potential dependability threats.

## 2 Related Work

In the following contributions related to functional decomposition and regarding dependability threats in general and specifically caused by the perception component are introduced.

Amersbach et al. [4] functionally decompose automated driving systems into six layers based on the human driving task for the definition of particular test cases. The decomposition layers are information access, information reception, information processing, situational understanding, behavioral decision, and action. The proposed decomposition is not further distinguished into more layers to be applicable for various automated driving systems. However, to define requirements for the perception component there is a need for the definition of dependability threats based on a more specific decomposition of the environmental perception and the subsequent processing into an environmental model. Therefore, in this work we focus on the information processing layer by decomposing it and identifying corresponding dependability threats. While the contribution by Amersbach et al. [4] lacks a more detailed decomposition of the task of perceiving the environment, Rosenberger et al. [5] take a closer look into the information processing layer and functionally decompose a lidar sensor system. They define differently abstract interfaces along the lidar data processing chain: the raw scan of the lidar sensor, the resulting point cloud and an object list which contains geometric and physical attributes. These interfaces are then used for a more detailed comparison of real and synthetically generated lidar measurement data using different metrics for different interfaces. A similar approach for the differently abstract representations of sensor data is also considered in this work.

A contribution that deals with the identification of perceptual uncertainty is proposed by Hanke et al. [6]. They examine the construction of a statistical sensor model for the virtual test of automated driving systems. To provide more realistic testing conditions they investigate the integration of lossy perception process characteristics into sensor models. To do so, they define the output interface of the model to consist of several model units where each of these units deals with one specific perception error. However, their work primarily focuses on objects and does not distinguish between different processing steps of sensor data. Another contribution for the classification of perceptual uncertainty is made by Dietmayer [3]. He describes the task of machine perception for automated driv-

ing and distinguishes its uncertainty into three uncertainty domains: state uncertainty, existence uncertainty and class uncertainty. State uncertainty deals with uncertainty regarding state variables such as position, kinematic or size of detected objects. Existence uncertainty refers to the uncertainty whether an object that was perceived actually exists. Class uncertainty describes the uncertainty concerning the semantic classification of detected objects. In this work we combine the classification of perception threats and where they can occur along the processing chain by considering differently abstract representations of sensor data. However, due to the different components processing the sensor data and therefore several potential causes for dependability threats arising, there is a need to differentiate these threats. A general approach to classify dependability threats is conducted by Avižienis et al. [7]. They establish basic concepts for the dependability of computing and communicating systems and distinguish threats to dependability into faults, errors and failures and define them subsequently. While faults are causes to errors, errors can propagate and eventually lead to a failure of a subsystem. Moreover, the characteristics of faults, errors and failures are discussed and different measurements to handle dependability threats are addressed. We adapt the definitions of Avižienis et al. to the perception component of automated driving systems.

While there are various contributions towards functional decomposition and categorization of perceptual threats and uncertainties, there is no known comprehensive taxonomy for the classification of threats to and from the perception component while also considering differently abstract levels of perception data in regards to the functional system architecture of automated driving systems.

## 3 Research Questions

**Research Question 1** *How can dependability threats to automated driving systems pertaining to perception components be characterized?*

For the establishment of a comprehensive taxonomy regarding dependability threats to and from perception components, we functionally decompose an automated driving system into components with well-defined tasks to receive precise interfaces. For that matter we extend existing approaches to functional decomposition (cf. [4], [5]) by respecting the individual steps of the perceptual processing chain of automated driving systems (cf. [8, p.47]). Moreover, we adapt the taxonomy for dependability threats by Avižienis et al. [7] (fault, error, failure) for the perception component of automated driving systems.

**Research Question 2** *What types of perception errors do exist and how can they be classified?*

When considering the task of perceiving the environment and processing different sensor data, there are several possibilities for the occurrence of errors from the raw scan of the environment up to the generated environmental model. Moreover, not every error or uncertainty has to be relevant for the automated driving system safely performing its driving task. Based on the functional decomposition, which is part of the first research question, we derive possible errors of perception components in this contribution. Furthermore, our proposed taxonomy for the classification of dependability threats to the perception component is evaluated by an exemplary lane keeping assistance system based on a camera.

## 4 Methodology

Robot systems are often distinguished into *Sense*, *Plan* and *Act* components. Adapted to a self driving vehicle, *Sense* includes the task of perceiving the surroundings and generating a model of the environment. *Plan* subsumes interpreting and predicting of future behavior of surrounding traffic participants based on the environmental model and then choosing a trajectory to be driven. *Act* stands for executing the planned trajectory by steering and accelerating or braking while also performing actions like indicating lane changes. This cycle is repeated for every scene<sup>2</sup>. A more detailed decomposition of automated driving systems is conducted by Amersbach et al. [4]. Figure 1 shows the decomposition layers of Amersbach et al. [4] mapped onto *Sense*, *Plan* and *Act* components.

Due to automated driving functions being highly complex systems consisting of various components, it is essential to identify the factors that can lead to safety-relevant system failures. In this section we propose a taxonomy for the classification of dependability threats to automated driving systems while focusing on the perception component. For that, we stick closely to the concept of faults, errors and failures introduced by Avižienis et al. [7] while also considering the differently abstract levels of sensor data representing the environment.

Avižienis et al. [7] define a fault as cause of an error. They distinguish between internal and external faults of a system. When a fault causes an error, it is active, otherwise it is dormant. An error is part of the total state of the system. When one or multiple errors cause the delivered service of the system to deviate from correct service, a failure occurs.

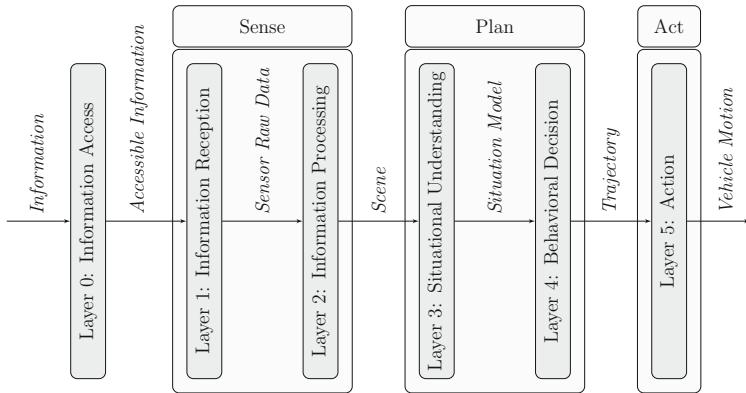


Figure 1: Functional decomposition by Amersbach et al. [4] mapped onto the *Sense-Plan-Act*-Paradigm

We assume that errors can occur in every step of processing environmental sensor data. Therefore we have to look at the data each component provides to the following

<sup>2</sup>We adopt the definitions of scene and scenario by Ulbrich et al. [9]

component. The raw scan of the surrounding environment is processed into a model of the surrounding environment and therefore exists in differently abstract levels during the processing. Considering the functional system architecture of automated driving functions [8, p.47] on the lowest level, there is a raw scan of the environment consisting of the data generated by the different sensors. Based on that different features like objects, traffic signs or road markings are detected. On the highest level all features are merged into a scene - a representation model of the environment. Figure 2 illustrates the processing of environmental sensor data and summarizes where the dependability threats, which are introduced in the following, can occur.

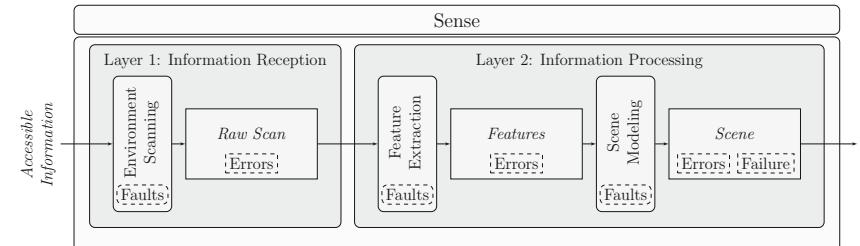


Figure 2: Processing chain of the perception component and potential occurrences of dependability threats relating to the *Sense* component as a system

### 4.1 Fault

Referring to our taxonomy, a fault is the cause of a perception error. Considering that there are different types of perception errors, there are also different types of faults to the perception component subsequently. On the one hand, errors that are propagating along the processing chain can be seen as faults to the resulting errors. On the other hand, each processing step of sensor data can contain its own faults (cf. Figure 2). When creating a raw scan of the environment, there are two types of faults: external faults and internal faults. External faults are disturbance variables like environmental conditions which can obscure the accessible information. Internal faults are either linked to the hardware, e.g. a systematic measurement error of a sensor, or are anchored in the software, e.g. a flawed point cloud generation out of received lidar beams. Faults to the processing of the raw scan into features are e.g. bugs in the object segmentation based on point clouds or images. When generating a scene, faults are either errors on feature level or present because of flaws in the scene modeling. An exemplary fault on this level is e.g. an incorrect lane matching algorithm for perceived vehicles.

### 4.2 Error

Each of the different representations of the environment can be inaccurate and therefore be subject to errors (cf. Figure 2). Examples for errors in these differently abstract representations are e.g. a blurred camera image on raw scan level, an object that is seen which is not existent on feature level and a correctly perceived traffic light that is,

however, linked to an incorrect lane on scene level. According to Avižienis et al. [7], many errors do not affect the system's external state.

### 4.3 Failure

According to Avižienis et al. [7] a system failure occurs when the delivered service deviates from correct service. In terms of the environmental perception, the question arises what correct service of the perception component of an automated driving system comprises. According to Dietmayer [3], correct service is delivered by a perception component when all relevant objects are detected with a certain quality within a fixed time interval. Moreover, the objects have to be correctly assigned to the traffic infrastructure. Hence, the delivered service deviates from correct service when either not all relevant objects are seen or when there is a mismatch in the modeled scene. In this case, the automated driving system would not be able to evaluate the situation appropriately anymore and therefore not be capable of performing its driving task safely enough.

## 5 Classification of Perception Error Types

In the following, both errors on raw scan level and on feature level are examined. To that end, raw data errors for the sensor technologies camera, Lidar and Radar are briefly discussed. Consecutively, we will derive errors on feature level by individually considering the single parts that make up the environment. While doing so, we are also referring to commonly used approaches on how this accessible information is included into the scene modeling.

### 5.1 Raw Scan

Errors on raw scan level are anchored in the raw data<sup>3</sup> generated by the deployed sensors. Due to the fact that different types of sensors generate different kinds of raw data, it is not possible to define common errors on this level of environmental representation which are applicable for every type of sensor. Instead the raw data of the different sensor types has to be looked at separately. Raw data generated by a camera are in general images consisting of pixels. Image noise due to the level of illumination or image distortions caused by effects like rolling shutter are therefore examples for camera raw data errors, as well as whole missing image sections (e.g. missing traffic signs due to flickering when capturing a variable-message sign over time). A Lidar sensor emits laser beams into the environment and measures their echoes. For each laser beam, a measured distance is recorded and, depending on the sensor implementation, other values like intensity or echo-puls-width are also obtained. Therefore, the raw data of a lidar consists of tuples of measured values. [5] Uncertainties in these measurement tuples due to noise, non-measured echoes or broken down channels can be considered as lidar raw data errors. According to Holder et al. [10] raw data of a radar is defined as the range-doppler-beam spectrum at the interface after the spectral analysis of the sensor readings and before the subsequent post-processing, which typically starts with a thresholding. Common distortions that occur in these raw data are defined as artifacts by Holder et al. [10]. While these artifacts obscure the

<sup>3</sup>We adopt the definition of raw data by Holder et al. [10].

accessible information, they can be seen as errors. Causes of such artifacts are e.g. mirror reflections, aliasing or electronic noise in the sensor [10].

### 5.2 Features

Errors on feature level are dependent from the different features that are considered for the scene modeling. For the definition of errors on this level it does not matter based on which kind of raw data the feature was extracted. Errors regarding features can be derived by looking at the elements which the environment consists of. Subsequently, we first decompose the environment into its parts. According to Ulbrich et al. [9] the environment consists of movable objects and the scenery. The scenery is then split up into the lane network, vertical elevation, stationary elements and environment conditions. Lanes and conflict areas belong to the lane network. Stationary elements are among other things obstacles, curbs, traffic signs and traffic lights. Figure 3 illustrates the decomposed elements of the environment.

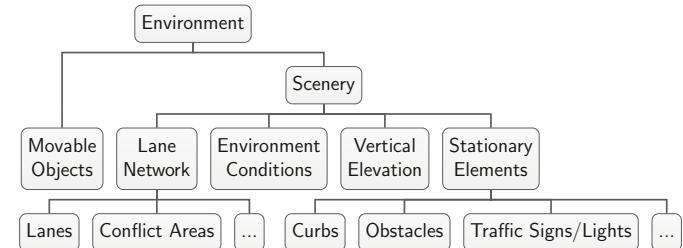


Figure 3: Elements of the environment according to Ulbrich et al. [9]

One part of the environmental perception is to detect existing movable objects. Whenever an object is not detected an object is missed by the environmental perception. A non-existing movable object, that is detected, is called a phantom object. Both of these cases can increase the risk during automated driving. But even when an existing object is perceived, there is an uncertainty that comes with every measurement. Ideally a movable object is represented by one bounding box instead of multiple ones. Regarding static non-continuous attributes of movable objects, like the classification, it is trivial to define that any deviation from the real classification is an error. However, concerning attributes that are continuous (e.g. dimensions) and attributes that are additionally dynamic and therefore can change over time (e.g. position and kinematics), it is not obvious when an uncertainty could propagate into a safety relevant error. This depends on the relevance of the perceived objects to the driving task as well as the robustness of the automated driving system. Possible errors regarding movable objects are summarized in Figure 4.

Traffic signs and lights are mandatory for managing traffic flow. For an automated driving system to abide by the road traffic regulations, traffic signs and lights need to be correctly captured, matched to their corresponding lanes and considered in the path planning. Regarding the definition of perceptual errors related to traffic signs, we differentiate between missed traffic signs, phantom traffic signs and correctly perceived traffic signs, which are, however, afflicted with uncertainties. Because traffic signs are static

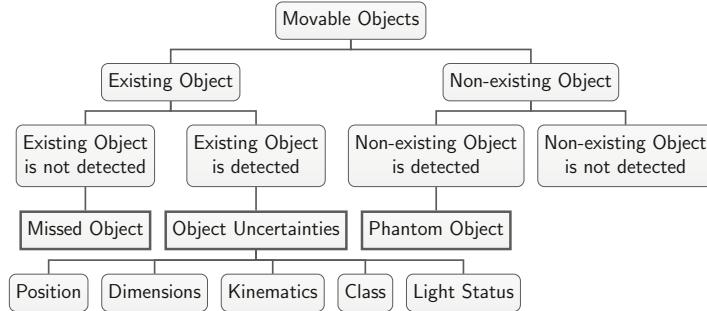


Figure 4: Errors regarding movable objects

(unlike movable objects), it is easier to define when an uncertainty might propagate into a safety relevant error. The position of the traffic sign needs to be captured accurately enough to be correctly matched to its corresponding lane. For the interpretation of the traffic sign both the class (e.g. a speed limit) and the value (e.g.  $80 \text{ km h}^{-1}$ ) have to be recorded correctly. While the value of most traffic signs does not change over time, traffic lights and variable-message signs are dynamic elements and therefore do not exclude changes regarding their value (e.g. a traffic light changing from green to yellow). Figure 5 summarizes the introduced errors.

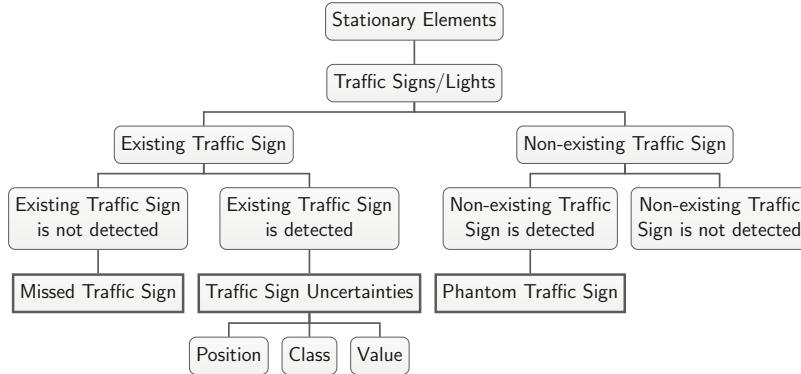


Figure 5: Errors regarding traffic signs

Lanes are defined by lane markings which imply the lane boundaries. In urban areas lane boundaries are additionally represented by curbs. Multiple lane marking segments form a continuous lane marking. For the automated driving system to construct these continuous lane boundaries, the lane marking segments need to be captured by the environmental perception. Moreover, overlapping lanes form conflict areas. We define overlooked lane marking segments as missed lane marking segments and detections of non-existing lane marking segments as phantom lane marking segments. Detected lane

marking segments can be uncertain in regards to their exact position and characteristics (e.g. curvature) and their class (e.g. solid, dashed, curbs), which also includes the color for lane markings (usually white or yellow). The class attribute is mandatory to know whether a lane boundary can legally be crossed and hence needs to be considered by the path planning. Any deviation from the real class can subsequently be considered as an error. Position and characteristics of lane marking segments are continuous values and need to be accurate enough to create a precise lane network. As soon as the lane network cannot be clearly derived by the detected segments, the uncertainty can be interpreted as safety-critical. Errors regarding lane marking segments are summarized in Figure 6.

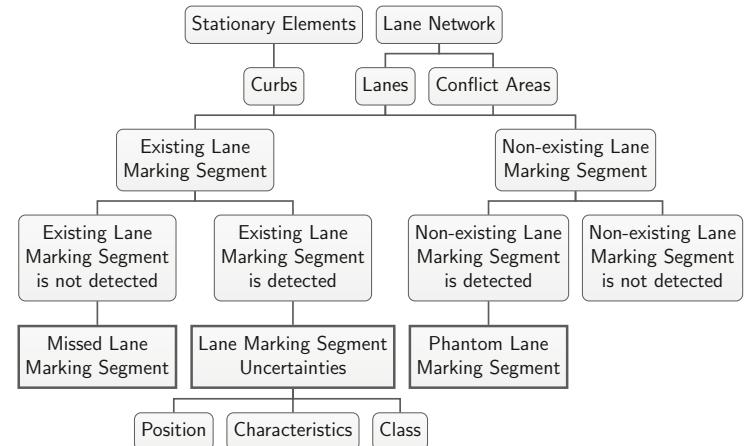


Figure 6: Errors regarding the lane network

One approach to capture vertical elevation is by estimating the ground plane. This information is not only important for path planning, but can also be used to improve quality of object detection [11]. Regarding a point in the environment, it either belongs to the ground plane or not. Subsequently, errors regarding ground mark classification are either overlooked ground marks or misleadingly classified ground marks (cf. Figure 7).

The integration of surrounding obstacles and not accessible areas into the path planning of a robot system is often implemented by creating an occupancy grid. For the creation of an occupancy grid, the environment is divided into grid cells. Afterwards, for each grid cell it is determined whether the cell is occupied or not. Hence, possible errors regarding the occupancy grid are either occupied cells which are classified as not occupied (overlooked obstacle) or not-occupied cells which are misleadingly classified as occupied (non-existing obstacle) (cf. Figure 8).

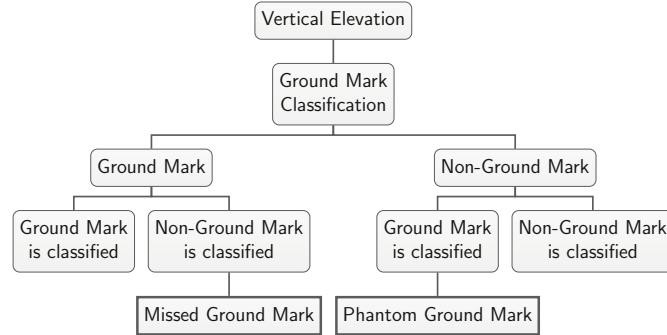


Figure 7: Errors regarding ground mark classification

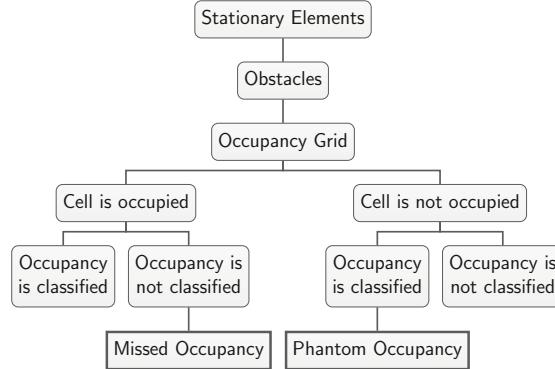


Figure 8: Errors regarding occupancy

## 6 Case Example: Lane Keeping Assistance System

To show the applicability of the presented taxonomy, we consider a lane keeping assistance system as case example and its handling of exemplary dependability threats in a hypothetical scenario. Task of the considered assistance system is to detect lane marking segments in a camera image, model them to lanes and subsequently assist the driver with lateral control of the vehicle to keep the lane. Figure 9 shows the functional architecture of the *Sense* component of the exemplary system and one possible hazard, which is analyzed in the following.

We now consider for the system to run into a scenario where the correct service cannot be maintained without making adjustments. While the camera captures lane marking segments, we assume a low hanging sun to blind the camera for a short time and therefore cause overexposed images. That results in *Errors* in the raw data because the image misses parts of the environment and therefore does not represent all of the accessible information. Extraction of lane marking segments based on these images leads to *Missed Lane Marking*

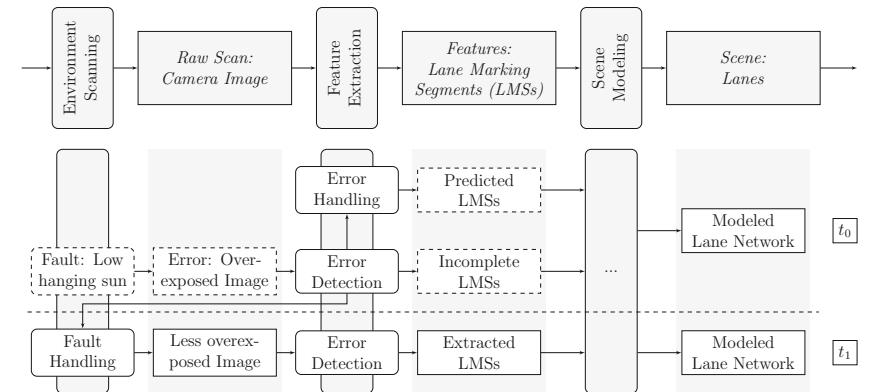


Figure 9: Case Example: Handling of exemplary dependability threats for a Lane Keeping Assistance System

*Segments* and therefore an incomplete set of lane marking segments. To deal with these false negative errors, the hypothetical system contains a component for *Error Detection*, which can trigger *Fault Handling* of the environment scanning to avoid *Errors* in the images of upcoming iterations and trigger *Error Handling* to cope with defective images for the current iteration. According to Avižienis et al. [7], the combination of *Fault Handling* and *Error Handling* form *System Recovery*.

In this case example, *Error Handling* is implemented by *Compensation* (cf. [7]). The compensation comprises relying on predicted lane marking segments that were generated during feature extraction of earlier iterations (e.g. by using a Kalman-Filter). Both predicted and the set of incomplete lane marking segments are then provided to the subsequent lane modeling. Simultaneously to *Error Handling*, *Fault Handling* in the environment scanning is triggered. To cope with the low hanging sun and to avoid *Errors* in the camera images, camera settings are reconfigured (e.g. light shade and exposure time). Therefore, according to Avižienis et al. [7] *Fault Handling* in this case means *Reconfiguration*. This results in less overexposed camera images for upcoming iterations.

Based on the executed *System Recovery*, lane marking segment extraction and subsequent lane modeling can then be sufficiently precise again for the *Sense* component to deliver correct service without considering predicted lane marking segments of an earlier iteration.

## 7 Conclusion and Future Work

In this contribution a taxonomy for the characterization of dependability threats to perception components is established. For that, the task of environment perception is functionally decomposed and thus precise interfaces are created. Subsequently, the concept of faults, errors and failures is implemented to include causalities between the threats along the processing chain of environmental perception. For the classification of perceptual error

types, both the raw scan of the environment and extracted features are closer examined. Within the scope of this work, raw data of camera, Lidar and Radar is briefly discussed. For the definition of perceptual errors on feature level, possible errors are derived by splitting up the environment into its subsequent parts and considering in which aspects extracted features can be flawed. Since for the definition of these errors the components of the environment established by Ulbrich et al. [9] have been considered, we do not claim for our error classification to be exhaustive. The proposed taxonomy is supported by an exemplary case example.

Since our focus was deducting perceptual errors on feature level, future work should also address dependability threats on raw scan and scene level in more detail, especially regarding possible faults. Moreover, future work should deal with the influence of different kind of threats on the robustness of automated driving systems focusing on the *Sense* component. Acquiring information about how precise environmental perception must be is a key step for safe system design. Therefore, not only false negative and false positive errors, but also uncertainties of true positives should be investigated in more detail. Additionally the importance of surrounding objects has to be considered, since not every object is of relevance for the automated driving system. Subsequently, not every error is safety-relevant and thereby results in higher safety risks of automated driving system. Defining and also refining of safety requirements revolving around the environmental perception will be a key challenge to solve for the safety validation of automated driving systems.

## References

- [1] SJ3016: *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, Jun 2018.
- [2] B. Lightsey, “Systems Engineering Fundamentals,” Department of Defense - Systems Management College, Fort Belvoir, VA, USA, Tech. Rep., 2001.
- [3] K. Dietmayer, *Predicting of Machine Perception for Automated Driving*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 407–424.
- [4] C. Amersbach and H. Winner, “Functional decomposition: An approach to reduce the approval effort for highly automated driving,” in *8. Tagung Fahrerassistenz*. München: Lehrstuhl für Fahrzeugtechnik mit TÜV SÜD Akademie, 2017.
- [5] P. Rosenberger, M. Holder, S. Huch, H. Winner, T. Fleck, M. R. Zofka, J. M. Zöllner, T. D'hondt, and B. Wassermann, “Benchmarking and Functional Decomposition of Automotive Lidar Sensor Models,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, Paris, France, 2019.
- [6] 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 *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 643–648.

- [7] A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr, “Basic Concepts and Taxonomy of Dependable and Secure Computing,” *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, Jan 2004.
- [8] F. Schultdt, “Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen,” Ph.D. dissertation, Technische Universität Braunschweig, 2017.
- [9] S. Ulbrich, T. Menzel, A. Reschka, F. Schultdt, and M. Maurer, “Defining and substantiating the terms scene, situation, and scenario for automated driving,” in *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*. Las Palmas, Spain: IEEE, 2015, pp. 982–988.
- [10] M. F. Holder, C. Linnhoff, P. Rosenberger, C. Popp, and H. Winner, “Modeling and simulation of radar sensor artifacts for virtual testing of autonomous driving,” in *Automatisiertes Fahren*. München: Lehrstuhl für Fahrzeugtechnik mit TÜV SÜD Akademie, 2019.
- [11] J. Rieken, R. Matthei, and M. Maurer, “Benefits of using explicit ground-plane information for grid-based urban environment modeling,” in *2015 18th International Conference on Information Fusion (Fusion)*. IEEE, 2015, pp. 2049–2056.

# How to evaluate synthetic radar data? Lessons learned from finding driveable space in virtual environments

Martin F. Holder\*, Jan R. Thielmann†, Philipp Rosenberger\*,  
Clemens Linnhoff\*, and Hermann Winner‡

**Abstract:** Generating synthetic sensor readings at scale by means of virtual sensors is expected to facilitate safety validation of autonomous driving functions. An absolute equality of real and synthetic data is not to be expected. Instead, it has to be proven that synthetic sensor data exhibits a comparable level of uncertainties as data from the real sensor, so that subsequent algorithms draw the same conclusions from the respective input data. This paper addresses this problem by comparing free space information inferred from real and synthetic radar data. It is shown that comparable free space can be calculated from the sensor simulation, although deviations between synthetic and real sensor data exist. The presented method to compare the calculated free space from synthetic and real data serves as evaluation of the simulation model.

**Keywords:** Automotive Radar, Sensor Modeling, Environment Perception, Virtual Validation

## 1 Introduction

During development of sensor simulation models for virtual validation of autonomous driving, it is of interest to identify the current performance level of the model. When generating synthetic sensor data as available from an implementation of a sensor model from a simulation environment such as IPG Carmaker, Vires Virtual Test Drive, CARLA etc., differences to measured data are naturally expected as models are always only an approximation to reality. Especially for the simulation of environmental sensors, the achievable accuracy of the virtual representation is limited: Detailed modeling of reflection properties of materials is an open research topic and modeling radar sensors is known to be particular challenging [1]. Besides, once a sensor simulation model is implemented, uncertainties in its parameterization remain. For example, uncertainty regarding main radar performance parameters, such as measurement ranges, window functions, antenna gains but also mounting position etc. is present and this uncertainty in the model is part of the overall *epistemic* uncertainty. Next, the parameter space of the virtual environment, in which the sensor model is executed, revolves around reflection properties of materials,

\*Martin Holder M. Sc., Philipp Rosenberger M. Sc., and Clemens Linnhoff M. Sc. are Research Associates at Institute of Automotive Engineering (FZD) at TU Darmstadt, E-Mail (holder,rosenberger,linnhoff@fzd.tu-darmstadt.de).

†Jan Thielmann M. Sc. conducted his Master's Thesis at the Institute of Automotive Engineering (FZD) at TU Darmstadt, E-Mail (janthielmann@gmx.de).

‡Prof. rer. nat. Hermann Winner is Head of Institute of Automotive Engineering (FZD) at TU Darmstadt, E-Mail (winner@fzd.tu-darmstadt.de).

such as objects, vegetation, pavement, etc., but also includes the virtual scenery itself with arbitrary level of complexity of the 3D models with respect to resolution and details. Since absolute equality of simulated and real values is never to be expected, methods are researched to evaluate a simulation model. An ad-hoc solution for model validation is to stimulate subsequent algorithms in the sensor data processing pipeline with both real and synthetic data and observe the amount of consensus, similar to the concept of runtime verification [2], which is subject of this work.

## 2 Comparing Synthetic and Real Radar Sensor Data

Already during model development, tools are needed to check the correct implementation (verification) and to make a statement about the degree of functional fulfillment of the requirements of the model (validation). On the one hand, it has to be proven that basic radar sensor parameters such as the measuring ranges in distance, azimuth and relative speed are correctly modeled. On the other hand, a model is considered falsified if the conclusions drawn by data processing algorithm from synthetic and real data are contradictory. For example, synthetic radar data could be fed into an object tracking algorithm and the results evaluated using specific metrics. In the sense of validating a simulation model, it would be considered (sample-)valid if the score achieved for a metric was within predefined limits [3]. However, this is difficult for many reasons: The data processing algorithms usually comprise a large number of parameters whose optimal choice must first be found. Second, it must be proven that these parameters can be transferred into the simulation to guarantee the equivalence of the algorithm. Ultimately, the comparison by means of (validated) metrics assumes that their calibration has already been performed. To the best of the authors' knowledge, no validity criteria are available for radar sensor models using metric-based key figures. This makes an objective evaluation of the quality of a sensor simulation more complicated and demands for new methods.

### 2.1 Grid Maps as Tool for Sensor Model Evaluation

The idea laid out in this paper is to initially allow for derivations between synthetic and measured radar sensor readings and shift the task of judging about the validity and meaningfulness of the synthetic data to subsequent data processing algorithms. The methodological approach is to recreate a test drive that was conducted in real world, in a virtual environment and then calculating the free space that is computed from the occupancy grid map and is available for planning in the scenery. Free space determination with Occupancy Grid Maps (OGM) based on Radar sensors has been extensively studied [4–7] and promises to avoid the challenges mentioned above: In OGM the Inverse Sensor Model (ISM) translates sensor readings into probabilities for occupancy of a specific grid cell. By evaluating the cumulative distribution of the sensor readings, the ISM adjusts its parameters accordingly. This facilitates comparison of synthetic to real data for evaluating environment occupancy. Comparison of OGM can be done either with heuristics [8], but this would again require calibrated metrics. In addition, a baseline offset is expected, since a real scene can only be reproduced in a simulation environment with finite and difficult to quantify accuracy. Instead, it should be checked whether essential information about free space is also contained in the synthetic data and whether a comparable path

for a self-driving vehicle is available in the free space. In order to compare synthetic and real data, this paper's working hypothesis is stated as: *Despite inaccuracies in the modeling, the free space that can be detected by a radar sensor is similarly reproduced in simulation.* It can be falsified if no similar path through free space could be derived from the OGM calculated from real or synthetic data. The hypothesis can also be understood as a qualitative requirement for a sensor simulation that a model must meet in order to be considered acceptable for simulation-based testing. Since the focus of this work is on the comparison of real and synthetic radar data, the pipeline for calculating free space is designed with fairly simple algorithmic components, see Figure 1.

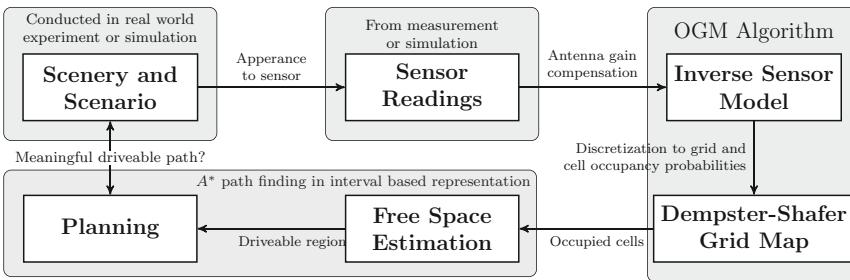


Figure 1: Road map for evaluation of synthetic sensor data: In both measurement and simulation, a scenery is observed by a radar sensor and its measurements are fed into an OGM algorithm. Free space and a driveable path are searched and their feasibility is evaluated.

## 2.2 Radar Measurements and Simulation

Automotive radar often uses chirp sequence waveforms with a patch antenna array. Subsequent Fourier transformations over each chirp, the sequence of chirps, and the antennas deliver a three dimensional spectrum, where each bin contains the spectral power for a certain range, range-rate, and azimuth bin. This data type is also called *radar cube* and can be seen as *raw* radar data where no thresholding has been applied. Basically, it is available for all chirp sequence radars with digital beamforming and a modeling approach that renders synthetic data on this interface has been proposed in earlier work by the authors [9], which motivates its usage in this work.

## 2.3 From Occupancy Grid Mapping to Driveable Path from a Free Space Estimate

The ISM represents grid-based environment occupancy in the posterior probability for occupancy of a grid cell  $O_j$  for a given power reading at cell  $j$  from the sensor  $P_{Tx,j}$  and vehicle pose  $\mathbf{z}_j$ . Raw radar data is notoriously difficult to interpret because of several pertinent artifacts, such as noise, ambiguities, and limited resolution. *Aleatoric* uncertainties are inherently present in this kind of data, simply due to the physical

measurement principle of the radar: Due to multipath propagation and noise artifacts in the radar, the area between the sensor and the obstacle cannot necessarily be considered free based on the sensor readings, although there is no real obstacle. In the sensor measurement, this is initially only reflected in areas with increased received power, for which the occupancy is justified via the ISM. Although the radar shows a better penetration of optical occlusions, it can be argued that a radar measurement can be better used to determine the occupied areas than to infer free space. The ISM proposed by [10] is used in this paper: It is parameterized from the empirical distribution of received power values. For capturing pure measurement noise, the lower threshold,  $P_{Tx,0.1}$ , is found as the upper limit of the 10% of the lowest observed values in the cumulative distribution power readings. Actual obstacles like cars are expected to show high power readings and therefore the upper threshold, while  $P_{Tx,0.9}$  is the lower limit of 90% of the largest observed values in the current measurement frame. The intermediate region is assumed to consist of large clutter targets as well as weak reflections from the terrain. In order to compensate for the directional dependence of the received power, it is equalized with the antenna gain characteristics. Consequently, the ISM requires a certain dynamic range between minimum and maximum power values in order to determine occupancy probabilities. The ISM and the cumulative distribution of the received power is illustrated in Figure 2.

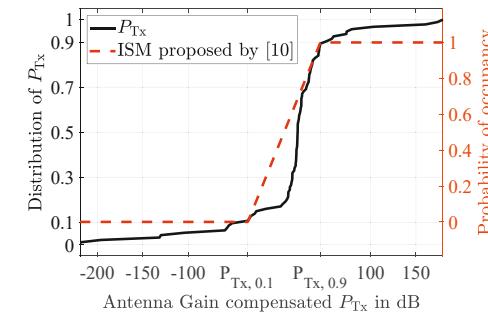


Figure 2: Schematic illustration of the proposed ISM based on the distribution of  $P_{Tx}$ .

The Dempster-Shafer Theory (DST) is a decision-making strategy that differentiates lack of information from conflicting or uncertain information, which makes it attractive for map building [11, 12]. In the OGM that is calculated from DST, the interval-based method [13] is used for obtaining free space from an occupancy grid map. This method is characterized by being computationally inexpensive and non-parametric: While it does not guarantee an optimal solution for obtaining free space it is easier to implement compared to parametric methods like B-splines [14]. The  $A^*$  path-finding algorithm is used to search a (driveable) path within the estimated free space that would be available for trajectory planning. For a passable path a minimum width of 2 cells and a minimum clearance of one cell from an obstacle is required.

## 2.4 Evaluation Aspects of OGM for Sensor Modeling

Two aspects of sensor modeling are evaluated via the OGM and the calculation of the free space: On the one hand, the dynamic range of the simulated sensor can be compared to the measurement results in order to evaluate whether the ISM is able to correctly calculate gradations in the probability of occupancy. On the other hand, the detailing of the environment is evaluated: In the proposed scenarios the modeling of the road edge is of interest, since the irregularity of the vegetation, as it exists in reality, can only be represented with an unreasonable effort in the virtual image of the environment. For this reason, a simple substitute modeling is used in the simulation model, which consists of a homogeneous vegetation strip along the line. Due to this (intentional) modeling inaccuracy, deviations in the free space are to be expected. An examination of the required faithfulness of the vegetation rendering would be possible in order to make a statement about the required degree of detailing.

## 3 Experiments

A far range radar with a measurement range of 200 m and approx.  $25^\circ$  in azimuth is used in real-world experiments. The first 15 m in front of the radar are excluded since they are assumed to contain only clutter reflections from the road. A respective simulation model has been parametrized according to this sensor. The size of a grid cell balances between the measurement accuracy of the sensor and the resulting total size of the grid. In order to take into account the range of the long-range radar with the limited lateral measurement range, a grid cell of 1 m was selected. Grid sizes of approx. 10 cm are common in lidar applications [15] and a similar grid size could be achieved with higher resolution radar sensors. With respect to the planning algorithm, it is expected to find a feasible trajectory for a regular passenger car within the required width of two cells.

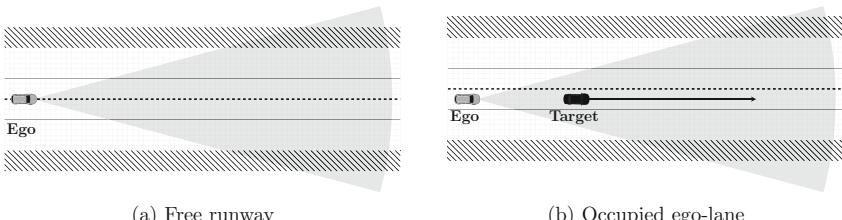


Figure 3: Illustration of the evaluation scenarios

Two experiments have been performed at the August Euler airfield in Griesheim, with its runway having a width of approx. 20 m, which corresponds to a width of 20 grid cells, see Figure 3. The first scenario involves a free runway and is used for evaluating the general capability of the pipeline to infer free space and learn about the accuracy of the environment simulation. Here, the ego vehicle drives straight ahead on the runway and the free space is therefore only limited by the vegetation at the roadside. In the second scenario, a target vehicle is located on the same lane as the ego vehicle. The target vehicle moves (slowly) away from the ego vehicle but overtaking through the left lane would be

possible at any time. The purpose of the experiment is to determine at which point both the simulation and the measurement, based on the free space recorded, would consider an overtaking attempt as possible.

## 4 Results

### 4.1 Sensitivity of $P_{Tx}$

The core element of free space estimation is the inverse sensor model. As mentioned at the beginning, this requires a certain dynamic range between minimum and maximum power which is examined for both scenarios. Figure 4 shows the cumulative distribution of all received power values,  $P_{Tx}$ , during both scenarios.

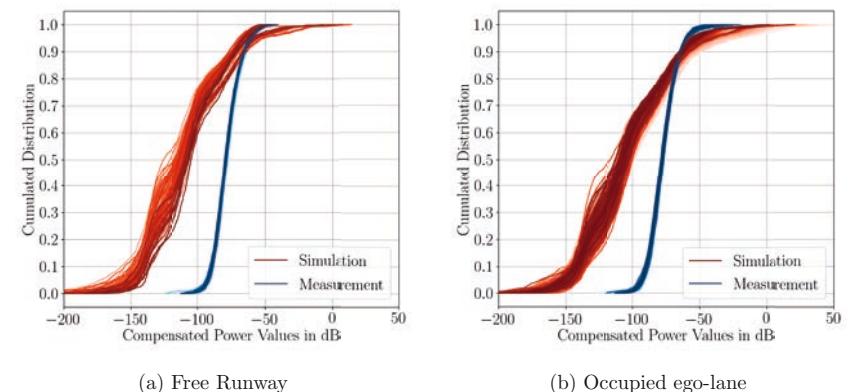


Figure 4: Cumulative Distribution of received power values for both scenarios. Individual time-steps are marked by darker color. The temporal process is coded in the color intensity. The darker, the more advanced the time.

As expected, the dynamic range in the measurement is increased due to the presence of the target vehicle in the second scenario. Basically, the measurement shows a similar course for both scenarios. Like the measurement, higher received power is sensed in the second scenario due to the presence of the target vehicle. With the empirical distribution the following aspects of the simulation can be falsified: The spread shows that the noise behavior is much more significant in the simulation than in the measurement. A higher dynamic range in the simulation shows that the simulated received power covers a higher bandwidth. The deviation of the median value indicates that the received power of *weak* targets such as vegetation is incorrectly simulated. Despite the differences, the synthetic data shows sufficient dynamics so that the ISM can parameterize itself and assign occupancy probabilities to the received power readings.

## 4.2 Free Runway Scenario

Figure 5 shows a qualitative comparison of the received power  $P_{Tx}$ , the probability of cell occupancy  $\mathcal{P}(\mathcal{O}_j)$  and the free space derived from it along with the proposed path. Simulated quantities are denoted with  $\tilde{\cdot}$ .

Obviously, deviations between simulation and measurement can already be detected in the comparison of the received power. Particularly noticeable in the synthetic data is the increased visibility of the edge of the road even at larger distances. A narrower width of the roadway is visible in the free space map because the lateral expansion of the reflections from the edge of the roadway is more significant in the synthetic data. Despite the deviations, in both cases it is possible to plan the (trivial) path, the straight route.

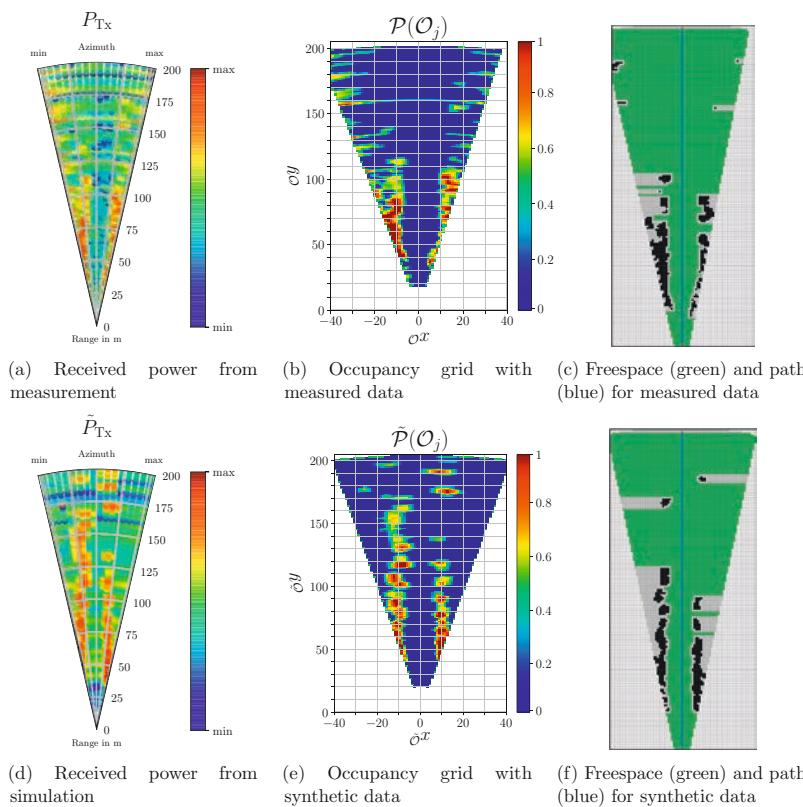
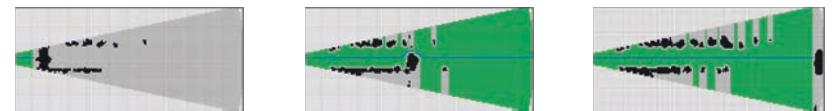


Figure 5: Qualitative comparison of measurement (top) and simulation (bottom) results for the free runway scenario.

## 4.3 Occupied Ego-lane Scenario

The task in this scenario is to find a path around the vehicle in front, which can be done by switching to the free left lane on the left. Through spectral windowing the vehicle smears in azimuthal direction, so that some azimuth cells are marked as occupied. Therefore, the sensor will only be able to capture the free space and plan a path around the vehicle after a certain distance. By comparing the required minimum distance from simulation and measurement, it is possible to evaluate the extent to which the window effects mentioned are correctly implemented in the simulation model. The definition of passability is shown in Figure 6. It should be noted that only the passability of the free space estimated by the sensor is considered and the target's backscattering is sufficiently high, see Figure 6c, where overtaking is possible, but is intentionally prevented for illustrative purposes.



(a) No path for overtaking found  
(b) Reasonable path for overtaking found  
(c) Visibility of the target at distance

Figure 6: Illustrative definition of a passable path

Figure 7 illustrates the distances at which sufficient space for overtaking is estimated on the left lane. Both measurement and simulation show sporadic time steps up to a distance of about 130 m, at which an overtaking path can be planned. At very close range, overtaking is not considered possible due to the limited visibility of the radar. Although the measurement still shows a few isolated time steps from 130 m upwards to which no path can be found, these can be traced back to noise effects. Deviating from the simulation, the measurement in the intermediate area shows a clearly more unsteady pattern with regard to the overtaking possibility. Because of the slow speed of the target vehicle, this is caused by interference patterns that cannot be found in the simulation in this form.

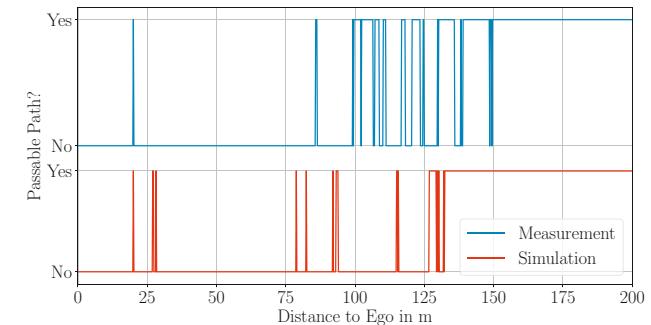


Figure 7: Comparison of the distance of the target to the ego vehicle at which a path for overtaking exists according to radar measurements.

## 5 Implications for Radar Sensor Modeling

Based on the results obtained, there are a number of implications for sensor modeling: First, the sensitivity analysis of the received power shows differences in dynamic range for synthetic data and reveals differences in noise behavior. The dynamics of real radar data is limited at the lower end by the noise level and at the upper end by the maximum level at the AD converter. The simulation model used in this paper shows no such dynamics limitation. In fact, this does not exist naturally for virtual sensors, since for example in raytracing based approaches, the signal strength of a ray can initially become arbitrarily small during propagation and reflection. For this application, this is of secondary importance due to the self-parameterization of the ISM to the cumulative power distribution. However, other applications, such as CFAR algorithms, make more restrictive assumptions about the signal-to-noise ratio, so that this must be taken into account in a simulation model. With regard to model parameters, the investigation shows the importance of correct modeling of azimuthal blurring as a consequence of spectral window effects in the azimuth direction. As the first experiment has shown, the free space in lateral direction at larger distances is incorrectly reproduced with the sensor model used in this work. Free space that can be detected by a sensor simulation model is closely linked to the accuracy of the environmental simulation. The results show that the design of the boundary area, i.e. the transition between the passable area and the vegetation, is of great importance. The simple vegetation model in the simulation already exhibits a well comparable shape of the OGM at the boundary area. This may serve as a first evidence for required detailing of such areas for radar sensor modeling.

## 6 Conclusion

This paper presents a method for evaluating a radar sensor simulation model by comparing the driveable space: By feeding synthetic data into an algorithm developed for real data, the strengths and weaknesses of a sensor simulation model can be investigated. The initial situation was that deviations between measurement and simulation were initially allowed, but the consequences related to the planning of a passable path were examined in two scenarios. Despite differences in the absolute value of simulated data, it is possible to evaluate a sensor model using the method presented: The advantage of the presented method is that initially deviations between synthetic and real data are allowed, because only the dynamic range is relevant for the parameterization of the ISM which results in a robustness against statistical deviations in the input data. By comparing the cumulative distributions of received power, model assumptions can also be falsified. The developed method is an extension to the research for the validation of sensor models and serves as a tool for testing the sensor model and its implementation in virtual environments.

## 7 Acknowledgement

This research is funded by the *SET Level 4to5* research initiative, promoted by the Federal Ministry for Economic Affairs and Energy (BMWi). The authors would like to thank Albert Schotschneider for his support in the development and implementation of the algorithms.

## References

- [1] M. Holder, P. Rosenberger, H. Winner, T. Dhondt, V. P. Makkapati, M. Maier, H. Schreiber, Z. Magosi, Z. Slavik, O. Bringmann, and W. Rosenstiel, "Measurements revealing Challenges in Radar Sensor Modeling for Virtual Validation of Autonomous Driving", in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018.
- [2] E. Bartocci and Y. Falcone, Eds., *Lectures on Runtime Verification*. Springer International Publishing, 2018.
- [3] P. Rosenberger, J. T. Wendler, M. F. Holder, C. Linnhoff, M. Berghöfer, H. Winner, and M. Maurer, "Towards a Generally Accepted Validation Methodology for Sensor Models - Challenges, Metrics, and First Results", in *Grazer Symposium Virtuelles Fahrzeug*, May 2019.
- [4] K. Werber, M. Rapp, J. Klappstein, M. Hahn, J. Dickmann, K. Dietmayer, and C. Waldschmidt, "Automotive radar gridmap representations", in *2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, Apr. 2015.
- [5] J. Lombacher, K. Laudt, M. Hahn, J. Dickmann, and C. Wohler, "Semantic radar grids", in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017.
- [6] R. Weston, S. Cen, P. Newman, and I. Posner, "Probably Unknown: Deep Inverse Sensor Modelling Radar", in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019.
- [7] M. Holder, S. Hellwig, and H. Winner, "Real-Time Pose Graph SLAM based on Radar", in *2019 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2019.
- [8] T. Colleens, J. Colleens, and D. Ryan, "Occupancy grid mapping: An empirical evaluation", in *2007 Mediterranean Conference on Control & Automation*, Jun. 2007.
- [9] M. Holder, C. Linnhoff, P. Rosenberger, and H. Winner, "The Fourier Tracing Approach for Modeling Automotive Radar Sensors", in *2019 20th International Radar Symposium (IRS)*, Jun. 2019.
- [10] M. Li, Z. Feng, M. Stolz, M. Kunert, R. Henze, and F. Küçükay, "High Resolution Radar-based Occupancy Grid Mapping and Free Space Detection", in *VEHITS*, 2018.
- [11] D. Pagac, E. Nebot, and H. Durrant-Whyte, "An evidential approach to map-building for autonomous vehicles", *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 623–629, 1998.
- [12] R. Murphy, "Dempster-Shafer theory for sensor fusion in autonomous mobile robots", *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 197–206, Apr. 1998.
- [13] T. Weiherer, S. Bouzouara, and U. Hofmann, "An interval based representation of occupancy information for driver assistance systems", in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct. 2013.

- [14] M. Schreier, V. Willert, and J. Adamy, “Compact Representation of Dynamic Driving Environments for ADAS by Parametric Free Space and Dynamic Object Maps”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 367–384, Feb. 2016.
- [15] J. Porebski, K. Kogut, P. Markiewicz, and P. Skruch, “Occupancy grid for static environment perception in series automotive applications”, *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 148–153, 2019.

## Bewertung der Sicherheit der Mensch-Maschine-Interaktion teilautomatisierter Fahrfunktionen (Level 2)

André Wiggerich\*, Heike Hoffmann\*, Nadja Schömig†,  
Katharina Wiedemann† und Katja Segler‡

**Zusammenfassung:** Kontinuierlich assistierende Systeme des Level 2 können per Definition die Quer- und Längsführung ausführen, wobei dem Fahrer primär die Rolle der Object and Event Detection and Response (OEDR) zukommt. Gleichzeitig übernehmen die Systeme Teile dieser Aufgabe auch selbst, wodurch es für den Fahrer schwierig sein kann, seine Rolle und Verantwortung korrekt zu erkennen und sich entsprechend zu verhalten. Im Rahmen einer Fahrsimulatorstudie wurde ein szenario-basierter Bewertungsansatz der Sicherheit der Mensch-Maschine-Interaktion (MMI) erprobt. Hierfür wurde ein neu entwickeltes kombiniertes Beobachtungs- & Befragungsinstrument für Probandenstudien eingesetzt. Die Reliabilität eines L2 Systems wurde anhand der Anzahl der nötigen Fahrereingriffe sowie der Güte der Spurführung manipuliert. Eine hohe L2 Systemreliabilität führte zu einer kritischen MMI. Dies konnte zuverlässig mit dem Instrument bewertet werden.

**Schlüsselwörter:** Sicherheit, Mensch-Maschine-Interaktion, Bewertungsverfahren, SAE Level 2.

## 1 Einleitung

Teilautomatisierung (so genannte Level-2- oder kurz L2-Systeme, gemäß SAE-Definition [1]) hält mit immer weitreichenderen Systemen in immer kürzer werdenden Entwicklungsintervallen Einzug in aktuelle Serienfahrzeuge. Diese Funktionen können im Gegensatz zu klassischen Fahrerassistenzsystemen sowohl die kontinuierliche Längs- als auch die Querführung des Fahrzeugs übernehmen, wobei der Fahrer weiterhin in der Verantwortung für die permanente Überwachung der Funktion und der Verkehrsumgebung ist. Diese Rolle des Fahrers wird gemäß SAE-Definition unter dem Begriff der Object and Event Detection and Response (OEDR) zusammengefasst. Ein Aspekt der diesbezüglich nicht unbeachtet bleiben sollte ist, dass die Fahraufgabe sowie die Interaktion des Fahrers mit dem Fahrzeug durch kontinuierliche

\* Bundesanstalt für Straßenwesen (BAST), Brüderstraße 53, 51427 Bergisch Gladbach ([wiggerich@bast.de](mailto:wiggerich@bast.de), [hoffmann@bast.de](mailto:hoffmann@bast.de)).

† Würzburger Institut für Verkehrswissenschaften (WIVW) GmbH, Robert-Bosch-Str. 4, 97209 Veitshöchheim ([schoemig@wivw.de](mailto:schoemig@wivw.de), [wiedemann@wivw.de](mailto:wiedemann@wivw.de)).

‡ Rheinische Fachhochschule (RFH) Köln, Schaevenstraße 1a-b, 50676 Köln ([katja.segler@rfh-campus.de](mailto:katja.segler@rfh-campus.de)).

Automatisierung verändert sind. Einerseits wird ein Teil der Fahraufgabe sowie auch ein Teil der OEDR durch die Funktionen übernommen, andererseits bleibt die letztendliche Verantwortung für die Überwachung und den unmittelbaren Eingriff in die Fahrzeugsteuerung im Falle von Systemgrenzen immer beim menschlichen Fahrer. In den bekannten „Ironies of Automation“ schrieb Bainbridge [2] schon in den 1980er Jahren, dass der Mensch kein guter Überwacher ist und nicht gut darin, in Situationen, in denen die Automation ausfällt, unmittelbar einzutreten. Eine dauerhafte Überwachung ohne dabei jedoch über weite Strecken aktiv eingreifen zu müssen, kann zu Missverständnissen und im Falle eines plötzlichen Ausfalls bzw. Fehlers der Funktion zu nicht adäquatem Verhalten beitragen [3]. Ein potentieller Sicherheitsgewinn kann so durch eine Veränderung des Verhaltens des Fahrers hin zu einer risikanteren Fahrweise oder mangelnder Überwachung verringert werden. Für eine sichere Interaktion ist es daher von höchster Relevanz, dass der Fahrer ein exaktes Verständnis von seiner Rolle im Umgang mit der Funktion hat und sein konkretes Verhalten entsprechend dieser Rolle anpasst. Gleichzeitig sollte die Funktion bestmöglich an die veränderten Anforderungen an den Fahrer angepasst sein und die korrekte Rolle des Fahrers explizit, bspw. über die entsprechende Gestaltung der Anzeige- und Bedienkonzepte, sowie implizit, bspw. über wahrnehmbare Fahrdynamik oder Systemreliabilität, einfach und unmissverständlich kommunizieren.

Diese Aspekte sind aufgrund der Heterogenität der am Markt verfügbaren Funktionen und Systeme aktuell nicht unmittelbar bewertbar. Hierzu fehlen valide und reliable Methoden, die eine ganzheitliche Betrachtung der Mensch-Maschine-Interaktion (MMI) von L2-Systemen zulassen. Die Studie hat somit insgesamt zwei Ziele: Zum einen soll ein neues Instrument zur Bewertung der Effizienz und Sicherheit der MMI für L2 Systeme validiert werden und zum anderen der Einfluss der spezifischen Gestaltung der Querführungsfunction des L2 Systems auf die Sicherheit der MMI hin untersucht werden.

## 2 Methoden

### 2.1 Versuchsaufbau

Die Studie fand im statischen Fahrsimulator der Bundesanstalt für Straßenwesen (BASt) statt. Für das Design und die Umsetzung der Szenarien wurde die Software SILAB® der WIVW GmbH verwendet. Das Mock-Up des Fahrerplatzes ist einem Kompakt- bis Mittelklassefahrzeug nachempfunden. Das Bedienelement für das L2 System ist einem BMW Serienfahrzeug entnommen und am Lenkrad des Simulators installiert. Die Anzeige des Systemzustands erfolgt über Icons im Kombidisplay. Der Versuchsleiter kann die Simulation in einem Operatorraum steuern sowie anhand von Kamera- und Simulationsansichten auf Monitoren überwachen (siehe Abbildung 1).

Das verwendete L2 System kombiniert eine assistierende Abstands- und Geschwindigkeitsregelung mit einer assistierenden Querführung, wobei die Funktionen der Quer- und Längsführung separat voneinander aktiviert werden können. Die Aktivierung der Querführung erfolgt in zwei Schritten: Erst wird die Querführung vom Fahrer in den Standby-Modus geschaltet und aktiviert sich sodann selbstständig, sobald das

Fahrzeug in einem definierten Korridor innerhalb des Fahrstreifens fährt und das System Fahrstreifen erkennt. Das System kann jederzeit durch den Fahrer über Lenk- und/oder Bremseingriffe übersteuert werden.



Abbildung 1: Darstellung des Fahrerplatzes des Probanden im Simulator (links) sowie der Versuchsleiterposition im Operatorraum zur Beobachtung des Versuchs.

### 2.2 Testszenarien

Im Rahmen des Versuchs wurden zwei Systemgrenzen als Testszenarien für die Probanden dargestellt. Aufgabe des Probanden war bei beiden Systemgrenzen ein aktives Eingreifen in die Fahrzeugführung, um ein Abkommen vom Fahrstreifen oder eine Kollision zu vermeiden. Die Systemgrenze 1 (enge Rechtskurve) wird in Abbildung 2 (rechts) graphisch verdeutlicht und stellt einen für die Querführung zu engen Kurvenradius dar, in welcher das notwendige Lenkmoment nicht vollständig durch das System aufgebracht werden kann. Das Szenario ist durch einen daraufhin erfolgenden Ausfall der Querführung in der Kurve gekennzeichnet. Der Fahrer erhält keinerlei visuelle oder akustische Übernahmeaufforderung in Form einer dedizierten Warnung. Die unmittelbare Notwendigkeit zum Eingriff durch den Fahrer wird lediglich visuell im Display durch einen Wechsel der Statusanzeige von aktiv zu inaktiv erkennbar. Ein Nichteingreifen des Fahrers führt in dieser Situation vom Verlassen des Ego-Fahrstreifens auf den benachbarten linken Fahrstreifen bis hin zur Kollision mit der linken Rückhalteinrichtung („Leitplanke“).

Die Systemgrenze 2 (Pannenfahrzeug als Hindernis auf der Fahrbahn), in der Abbildung 2 (links) dargestellt, zeigt die Einschränkung des Systems in der Längsführung bei stehenden Objekten auf. Vor einer Kuppe erscheint zunächst ein Warndreieck, kurz darauf, hinter der Kuppe, ein Pannenfahrzeug, welches sich auf dem rechten Fahrstreifen befindet. Auch in diesem Szenario erhält der Fahrer keine explizite Warnung oder Übernahmeaufforderung, sondern muss selbstständig durch Bremsen und/oder Fahrstreifenwechsel reagieren. Ein Nichteingreifen des Fahrers führt zur direkten Kollision mit dem Fahrzeug, da die simulierte Funktion konzipiert wurde, nicht auf stehende Objekte zu reagieren.



Abbildung 2: Darstellung der Simulation der Systemgrenze 1 enge Rechtskurve (rechts) sowie der Systemgrenze 2 Pannenfahrzeug als Hindernis auf der Fahrbahn (links) aus der Vogelperspektive.

### 2.3 Bewertungsinstrument

Da bislang keine standardisierte und anwendungsnahe Möglichkeit besteht, um die Sicherheit der MMI kontinuierlicher Automatisierungsfunktionen umfassend und ganzheitlich bewerten zu können, wurde ein tablet-basiertes Beobachtungs- und Befragungsinstrument, die sog. Standardized Application for Automated Driving Evaluations (S.A.D.E.)-App, entwickelt. Das Instrument dient dazu, die Effizienz und Sicherheit der MMI nicht nur über Checklist-Items anhand vorgefertigter Standards, sondern direkt und unmittelbar in der erlebbaren Interaktion anhand definierter Prüfszenarien messen und bewerten zu können. Das Instrument wurde auf Basis der S.A.F.E.-Anwendung [4] in SILAB programmiert.

Die Beobachtung des Interaktionsverhaltens findet anhand von drei Hauptkategorien, nämlich Bedienprobleme, fahrzeugführungsbezogene Probleme sowie Probleme bzgl. des Monitorings des Systems, statt. Diese Hauptkategorien werden anhand vonzählbaren Verhaltensankern, bspw. ob eine Gefährdung oder Kollision aufgetreten ist (siehe Abbildung 3), bewertet. In Abhängigkeit des identifizierten Verhaltens wird anschließend pro Szenario eine Gesamtbewertung der Interaktionsleistung auf einer Skala von 0 bis 10 in Anlehnung an die Fitness-to-Drive-Skala (FtD-Skala; nach [4] sowie [5]; in Anlehnung an [6]) vorgenommen. Die Zuordnung des Verhaltens zu den jeweiligen Ausprägungen der Skala wurde vorab definiert und allen Beobachtern intensiv im Rahmen einer Schulung inklusive Übungs- und Kontrollfahrten vermittelt. Zusätzlich stehen dem Anwender verschiedene Befragungsitems, bspw. hinsichtlich der Verständlichkeit des Systemverhaltens, zur Verfügung, welche während der Versuchsfahrt zur Bewertung der Usability des Systems durch die Probanden eingesetzt werden können.

Szenario					
301	nichts bemerkt	unsicher	verzögert	unangemessen	
	0	0	0	0	1. Wie klar war Ihnen, was zu tun ist?
Bedienprobleme	Fehlbedienung	nicht erfolgreich	Hilfestellung nötig		2. Wie nachvollziehbar war das Systemverhalten?
	0	0	0	X	-3 -2 -1 0 1 2 3
Fahrzeugführung	keine Reaktion	verzögert	Spurabkommen		3. Wie verständlich waren die Systemausgaben?
	0	1	5	0	? -3 -2 -1 0 X 2 3
Monitoring	unzur. gesichert	Gefährdung	Kollision		4. Wie schwierig war die Systembedienung / die Reaktion auf das Systemverhalten?
	0	1	0	X	? -3 -2 -1 0 1 X 3
VL-Rating Bewältigung Szenario					5. Optionale Zusatzfrage
?	0	1	2	3	X -3 -2 -1 0 1 2 3
?	0	1	2	3	4 5 6 7 X 9 10
?	0	1	2	3	4 5 6 7 X 9 10

Abbildung 3: Standardized Application for Automated Driving Evaluations (S.A.D.E.) zur Durchführung von Beobachtungs- und Befragungsstudien zur MMI bei SAE Level 2 Systemen per Tablet-PC.

### 2.4 Versuchsablauf

Vor der eigentlichen Versuchsfahrt bekam jeder Proband die Aufgabe, eine Bedienungsanleitung zum L2 System zu lesen. In dieser Anleitung wurden beide im Versuch dargestellten Systemgrenzen exemplarisch genannt und die Aufgabe des Fahrers in Reaktion auf das Auftreten von Systemgrenzen eindeutig kommuniziert. Anschließend wurde die Müdigkeit sowie das Vertrauen in das System auf Basis der gelesenen Anleitung mittels eines Fragebogens erfasst. Zusätzlich wurde eine circa fünfminütige Testfahrt mit dem L2 System unter Anleitung des Versuchsleiters durchgeführt, bei der das Ein- und Ausschalten, Übersteuern sowie die Displayanzeigen der Systeme erläutert wurden. Anschließend startete die Versuchsfahrt. Nach dem Auffahren auf die Autobahn wurden beide Systeme (Quer- und Längsführung) vom Probanden durch Tastendruck aktiviert. Dabei sollte eine Geschwindigkeit von 120 km/h eingestellt werden. Während der dreißigminütigen Autobahnfahrt sollte so lange und häufig wie möglich mit aktiven Systemen gefahren werden. Die Autobahn wies durchgängig zwei Fahrstreifen auf und verlief überwiegend durch eine ländliche Umgebung mit wenigen Reizen in der Umwelt oder aufgrund anderer Verkehrsteilnehmer. Während der Fahrt erlebt der Proband im Abstand von jeweils 12 Minuten zwei Systemgrenzen, welche bereits in Abschnitt 2.2 beschrieben wurden. Nach der Fahrt erfolgten die Beantwortung eines zweiten Fragebogens sowie ein teilstrukturiertes Interview.

### 2.5 Stichprobe

An der Studie nahmen  $N=56$  Probanden (24 weiblich, 32 männlich) im Alter zwischen 18 und 62 Jahren ( $MW=37.12$ ,  $SD=13.44$ ) teil. Ausgewählt wurden ausschließlich Personen, die eine Fahrleistung von mehr als 3000km/Jahr aufweisen und nicht regelmäßig mit

Längs- und/oder Querführungsassistenten fahren oder gefahren sind, um einen herstellerbezogenen Gewöhnungseffekt auszuschließen.

## 2.6 Versuchsdesign & Datenauswertung

Zur Prüfung der Fragestellung, welchen Einfluss die Gestaltung der Querführungsfunction des L2 Systems auf die Sicherheit der MMI hat, wurde die Qualität der Querführung anhand von zwei Faktoren manipuliert. Der erste Faktor umfasste die Variabilität bzw. Schwankungsbreite der Spurführung innerhalb des Fahrstreifens (zentriert vs. variabel/schwankend). Für die Umsetzung der Prüfbedingung wurden streckengebundene Trigger implementiert, die bewirkten, dass das System in den Bedingungen „hohe Variabilität“ regelmäßige Spurversäte von der Fahrstreifenmitte aus nach links und rechts ausführt, die gerade so stark waren, dass der Fahrer sie zwar als unangenehm wahrgenimmt und das Gefühl bekommt, „mitlenken“ zu müssen, aber noch nicht die Lenkung übersteuern muss, um im Fahrstreifen zu bleiben.

Der zweite Faktor wurde durch die Reliabilität des Systems (hoch vs. niedrig) dargestellt. Hierzu wurden zusätzlich in regelmäßigen Abständen streckengebundene Trigger gesetzt, die unter der Bedingung „niedrige Reliabilität“ einen Ausfall der Querführung für ca. 3-4 Sekunden bewirkten. Lenkte der Fahrer hier nicht aktiv mit, kam er in Kurven vom Fahrstreifen ab. Als Häufigkeit dieser Systemausfälle wurde eine Frequenz von ca. einem Ausfall pro 1000m Fahrtstrecke, d.h. etwa alle 30s bei einer Geschwindigkeit von 120km/h definiert. Die Zuordnung der Probanden zu den Faktorstufenkombinationen erfolgte randomisiert. Die Studie wurde als Doppelblind-Versuch durchgeführt, d.h. weder dem Probanden noch dem bewertende Versuchsleiter war die jeweilige Versuchsbedingung bekannt.

Als abhängige Variablen wurden u.a. die Versuchsleiter-Rating-Skalen der S.A.D.E.-App, die Standardabweichung der lateralen Position während der Kurve in Szenario 1 sowie das Minimum der Time-To-Collision (TTC) und die Anzahl an Kollisionen mit dem Hindernis im Szenario 2 ausgewählt. Die Untersuchung wurde als 2x2x2 fullfactorial mixed within-between-subjects-design gestaltet. Zusätzlich zur Betrachtung der beiden Zwischensubjektfaktoren Reliabilität und Variabilität bzw. Schwankungsbreite, erfolgte eine Betrachtung des Effekts der Reihenfolge der beiden Testszenarien innerhalb eines Probanden. Die Daten wurden mit inferenzstatistischen Verfahren, bspw. unterschiedlichen Varianzanalysen (ANOVA), ausgewertet.

## 3 Ergebnisse

### 3.1 Testszenario Kurve

Die erste ANOVA bezieht sich auf das Szenario 1, in welchem die Querführung in der Kurve ausfällt. Der Faktor Reliabilität beeinflusst die Stabilität der manuellen Spurführung nach Ausfall der Querführungsassistenz tendenziell signifikant ( $F(1,48)=3.58, p=.065, \eta^2=.069$ ). Probanden mit Ausfällen (geringe Reliabilität) weisen im Durchschnitt ( $MW=0.35, SD=0.30$ ) eine geringere mittlere Standardabweichung der

lateralen Position auf als Probanden ohne Systemausfälle ( $MW=0.50, SD=0.29$ ). Die Ergebnisse dazu sind in Abbildung 4 (links) dargestellt. Keinen signifikanten Unterschied gibt es zwischen den experimentellen Bedingungen der Variabilität ebenso wie der Reihenfolge ( $p>.05$ ). Auch eine signifikante Interaktion der Faktoren konnte nicht festgestellt werden ( $p>.05$ ).

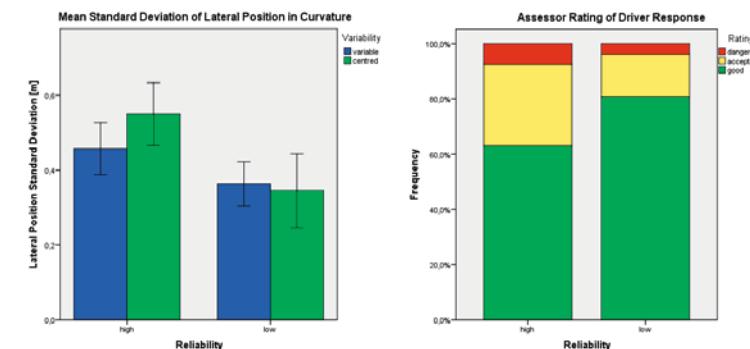


Abbildung 4: Mittelwerte der Standardabweichung der lateralen Position während der Kurvenfahrt in Abhängigkeit der Faktoren Reliabilität und Variabilität (links) sowie Zusammenfassung der Skala zur Bewertung der Interaktion der S.A.D.E.-App im Szenario 1 in Abhängigkeit des Faktors Reliabilität (rechts).

Die Skala zur Bewertung der Interaktionsleistung der S.A.D.E.-App zeigt, dass dieses Szenario über alle Bedingungen hinweg überwiegend gut gelöst wurde. Die Ergebnisse dazu sind in Abbildung 4 (rechts) dargestellt. Rein deskriptiv ist ein Unterschied zwischen einem System mit hoher und niedriger Reliabilität zu erkennen. Probanden mit einem hoch reliablen System reagieren tendenziell schlechter in dieser Situation. Die Ergebnisse des  $\chi^2$ -Tests dazu sind jedoch nicht signifikant ( $p>.05$ ). Die Korrelation zwischen der Standardabweichung der lateralen Position und Skala zur Bewertung der Interaktionsleistung liegt bei  $r=.80$  ( $p<.01$ ).

### 3.2 Testszenario Hindernis

Die zweite ANOVA bezieht sich auf das Szenario 2 (Pannenfahrzeug). Die Ergebnisse (siehe Abbildung 5 links) zeigen, dass der Faktor Reliabilität einen signifikanten Einfluss auf die  $TTC_{min}$  hat ( $F(1,48)=5.265, p=.026, \eta^2=.1$ ). Bei einer hohen Reliabilität ( $MW=0.88, SD=0.73$ ) kann eine signifikant niedrigere  $TTC_{min}$  festgestellt werden als bei einer niedrigen Reliabilität ( $MW=1.36, SD=0.86$ ). Keinen signifikanten Unterschied gibt es zwischen den experimentellen Bedingungen der Variabilität ebenso wie der Reihenfolge ( $p>.05$ ). Auch eine signifikante Interaktion der Faktoren konnte nicht festgestellt werden ( $p>.05$ ). Der Einfluss des Faktors Reliabilität zeigt sich rein deskriptiv betrachtet auch in der Anzahl der Kollisionen mit dem Hindernis. Bei einer hohen Reliabilität kollidierten  $n=5$  Probanden mit dem Hindernis, bei einer geringen Reliabilität

hingegen nur  $n=2$ . Dieser Unterschied ist statistisch gesehen jedoch nicht signifikant ( $p>0.05$ ). Die Skala zur Bewertung der Interaktionsleistung der S.A.D.E.-App zeigt, dass das Szenario mit einem hoch reliablen System signifikant kritischer gelöst wurde als mit einem System mit niedriger Reliabilität ( $\chi^2=5.00$ ,  $df=2$ ,  $p=.041$ ). Die Ergebnisse dazu sind in Abbildung 5 (rechts) dargestellt. Beim System mit niedriger Reliabilität haben 50.0% der Probanden das Szenario gut, 38.5% akzeptabel und 11.5% gefährlich gelöst. Beim System mit einer hohen Reliabilität hingegen haben 29.6% der Probanden das Szenario gut, 33.8% akzeptabel und 37.0% gefährlich gelöst. Die Korrelation zwischen der TTC<sub>min</sub> und der Skala zur Bewertung der Interaktionsleistung liegt bei  $r=-.56$  ( $p<.01$ ).

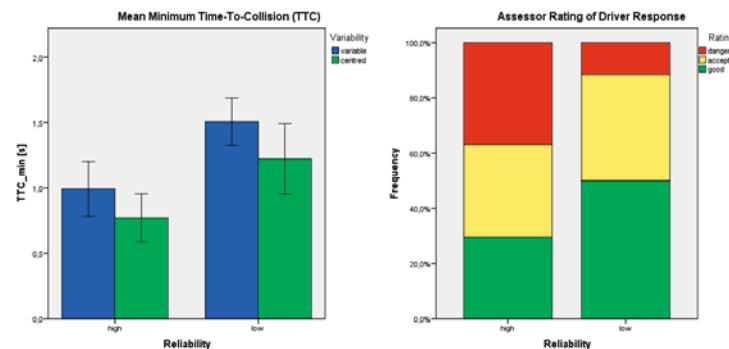


Abbildung 5: Mittelwert des Minimums der Time-To-Collision (TTC) vor dem Pannenfahrzeug in Abhängigkeit der Faktoren Reliabilität und Variabilität (links) sowie Zusammenfassung der Skala zur Bewertung der Interaktion der S.A.D.E.-App im Szenario 2 in Abhängigkeit des Faktors Reliabilität (rechts).

## 4 Diskussion und Zusammenfassung

Die Fahrimulatorstudie verfolgte insgesamt zwei Ziele: Zum einen sollte ein neu entwickeltes kombiniertes Beobachtungs- und Befragungsinstrument validiert und zum anderen die Auswirkungen bestimmter querführungsbezogener Systemdesignvarianten, in diesem Fall die Reliabilität und Variabilität der Querführung, auf die Güte der MMI untersucht werden. Die Ergebnisse zeigen, dass eine hohe Reliabilität in der Querführung des L2 Systems die Güte der MMI verschlechtert. Sowohl im Kurvenszenario als auch vor dem stehenden Fahrzeug als Hindernis führte eine hohe Reliabilität zu einer höheren Standardabweichung bzw. geringeren TTC. Obwohl alle Probanden vorab über die Systemgrenzen und die notwendige Reaktion durch den Fahrer aufgeklärt wurden, gelang dieser Eingriff Probanden mit einem hoch reliablen System nur schlecht. Dies deutet darauf hin, dass es Schwierigkeiten in der Wahrnehmung und Ausführung der korrekten Fahrerrolle gibt, die allein auf die Wahrnehmung des objektiven Verhaltens des Systems zurückzuführen sind [7]. Diese Effekte konnten auch mittels des Tablet-Tools valide gemessen und bewertet werden. Die Zusammenhänge zwischen den objektiven Maßen

und den beobachtungsbasierten Skalen des Bewertungsinstruments sind insgesamt hoch. Im Rahmen einer zukünftigen Studie mit Serienfahrzeugen auf einer Teststrecke soll daher überprüft werden, inwiefern das Tool in der praktischen Anwendung einsetzbar ist und sich die Ergebnisse auch auf reale Systeme und Prüfszenarien übertragen lassen.

Zukünftig können so Tests und Bewertungen zur Sicherheit der MMI von L2 Systemen verschiedener Hersteller, bspw. im Rahmen des Verbraucherschutzes, standardisiert und für Testlabore in einem angemessenen Rahmen durchgeführt werden. Die bisherigen Tests sehen hierfür eine getrennte Auswertung von technischen Anforderungen an die Systemgestaltung sowie Usability-Standards vor [8]. Dies führte in der Vergangenheit dazu, dass die eigentliche Interaktion zwischen Fahrer und System gar nicht bewertet werden konnte. Mit Hilfe der S.A.D.E.-App ist es nun erstmals möglich, dass die reelle Interaktionsleistung unmittelbar Teil der Bewertung sein kann.

## Literatur

- [1] SAE On-Road Automated Vehicle Standards Committee. (2018). Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems (J3016\_201806). SAE International, Warrendale, USA.
- [2] Bainbridge, L. (1983). Ironies of Automation. *Automatica*, 19(6), 775-779.
- [3] National Transportation Safety Board. (2017). Collision Between a Car Operating With Automated Vehicle Control Systems and a Tractor-Semitrailer Truck Near Williston, Florida May 7, 2016. NTSB/HAR-17/02 PB2017-102600 Notation 56955.
- [4] Kaussner, Y., Hoffmann, S., Fischer, F. & Kenntner-Mabiala, R. (2013). SPDE\_DFA: S.A.F.E: Anleitung zur Durchführung von Fahrverhaltensbeobachtungen mit der Standardized Application for Fitness to Drive Evaluations (S.A.F.E.) am Tablet-PC. Würzburg.
- [5] Kaussner, Y. (2013). *Assessment of driver fitness: An alcohol calibration study in a high-fidelity simulation*. Paper presented at the Fit to Drive 7th International Traffic Expert Congress, Berlin, 25.04.-26.04.2013.
- [6] Neukum, A., & Krüger, H.-P. (2003). Fahrerreaktionen bei Lenksystemstörungen - Untersuchungsmethodik und Bewertungskriterien. In VDI-Gesellschaft Fahrzeug- und Verkehrstechnik (Hrsg.), *Reifen-Fahrwerk-Fahrbahn*, VDI-Berichte Nr. 1791. Düsseldorf: VDI-Verlag.
- [7] Victor, T. W., Tivesten, E., Gustavsson, P., Johansson, J., Sangberg, F., & Ljung Aust, M. (2018). Automation expectation mismatch: Incorrect prediction despite eyes on threat and hands on wheel. *Human Factors*, 60(8), 1095-1116.
- [8] EuroNCAP (2018). Automatisiertes Fahren. <https://www.euroncap.com/de/fahrzeugsicherheit/sicherheitskampagnen/2018-automatisiertes-fahren/>, abgerufen am 11.12.2019.

## Manual Adaption of Steering Support in a Take-Over Scenario – A Technical Evaluation

Luis Kalb\* and Burak Karakaya\*

**Abstract:** This paper presents the technical development and evaluation of three systems to manually adapt steering support in a take-over scenario during conditional automated driving. Speech command, pressure sensors and clutch paddles were individually examined in a prototype state. Results indicate that speech commands exceed time limits during processing and pressure sensors are too sensitive for human fingers. Clutch paddles are most promising, both in signal processing and mounting in a vehicle mockup.

**Key words:** automated driving, take-over scenario, steering support, adaption

## 1 Introduction

Transitions of driving control during conditional automated driving, as defined in [1], have been researched intensely in the last decade ([2, 3]). [4] more recently introduced a concept for a transition of control with added steering support for a human driver. An automated system actively steers the vehicle from the point of the Request to Intervene (RtI) until the system boundary is reached in cooperation with the human driver. The automation's applied torque at the steering wheel decreases over time because it won't be able to control the vehicle at the system boundary at the latest. Human drivers might want to speed up this process or disengage steering support altogether sooner to regain full manual control earlier.

In this paper we present our technical development and evaluation process for systems or devices that will enable the driver to adapt this steering support. First we define the requirements and the scenario in which the systems will be used. We exclude certain established use concepts based on these requirements and introduce ideas from similar applications. Three concepts are chosen for implementation as prototypes. We describe our testing procedures and conclude with the recommendation of one system.

## 2 Requirements and Scenario

[4] described the scenario that is usually researched for transitions in conditional automated driving: The human driver is not paying attention to traffic or the environment and busy with a non-driving related task (NDRT). The feet are off the pedals and the hands are off the steering wheel.

The driver is now primed with the RtI for the upcoming steering support by a multimodal warning (see [5, 6]). The driver ideally already chose a preferred driving trajectory as he puts his hands on the steering wheel. At this moment the steering support sets in and the driver has to evaluate whether the performed assistance matches his chosen trajectory. If not, the driver can reevaluate the possible trajectories and give way to the steering support or explicitly adapt the influence of the system on the steering task to make way for his own steering choice.

If the driver chooses to adapt the steering assist, he or she now has an additional task, that a) is not yet represented with an interface in a standard car interior and b) has a major influence on the task of lateral vehicle control. One option would be to use this opportunity to rethink the steering interface altogether, for which [7] gives some examples. Another, more feasible approach, is to add another interface that is interlocked with the existing steering task and doesn't obstruct the conventional driving control.

Another factor was time efficiency. Available time for transitions in conditional automated driving ranges between five and seven seconds in the literature [2, 3]. Drivers already require between 1.45 seconds and 1.79 seconds ([2]) to place their hands on the steering wheel which is the point at which steering support would start ([4]). Drivers would lose more valuable steering time if they had to move their hands away from the steering wheel and towards a control element. Their situation awareness would also deteriorate if they had to move their gaze away from the road scene to localize a control element and coordinate their hand movements. Consequently, the planned concept should include control elements at the steering wheel or a control method through other modalities than the hands. Time efficiency is also required for any software processing following the usage of the concept control element.

Two more requirements are not yet specified but should be considered after an evaluation: A continuous or a discrete scale on which the driver can adapt the driving support and the possibility to correct given input by the driver.

## 3 Concept Development

### 3.1 State of Technology

We first took a look at existing control elements at the steering wheel and their suitability for the present use case. A standard multi-function steering wheel can offer (1) several buttons with dichotomous states on both wheel spokes (2) several levers behind and on both sides of the steering wheel with dichotomous states in two directions, possibly

\* Chair of Ergonomics at the Technical University of Munich, Boltzmannstr. 15, 85748 Garching (e-mail addresses: luis.kalb@tum.de, burak.karakaya@tum.de)

equipped with buttons with dichotomous states on the levers (3) two gear shift paddles with dichotomous states.

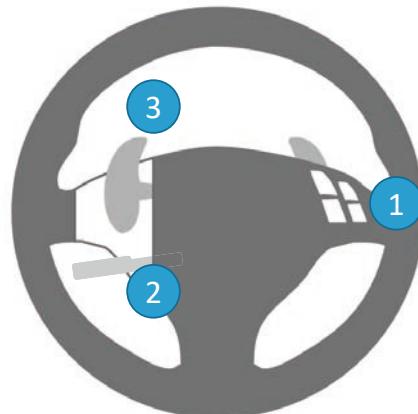


Figure 1: Example of a steering wheel with (1) buttons (2) levers (3) paddles.

- (1) The existing buttons offer the possibility to control steering support in general. The buttons have to incorporate the factor time for a continuous scale, i.e. holding a button for a certain time decreases steering support by a certain amount. Buttons like +/- or up/down would allow changes on a discrete scale.  
Yet these buttons are usually already equipped with a specific task or function. This would require multi-labeling of a button for it's at least two functions which might confusion in a time critical scenario. More buttons could offer a solution, but aren't feasible if the mock-up shouldn't be reconstructed. It is highly unlikely, that a driver would count the seconds he pushed a button in a critical take-over scenario, so the use of buttons also requires an additional HMI to signal the driver what effect his input had.
- (2) A minimum of two levers, turn signal and windshield wipers, is standard equipment in a current car. More advanced driver assisted systems may be controlled with an additional lever. The driver can shift this lever in at least two perpendicular directions and use several smaller buttons and other control elements on the lever.  
A possible fourth lever for the adaption of steering support might overcrowd the bottom space behind the wheel and induce extra workload for the driver in finding the correct lever in a time critical scenario.
- (3) Gear shift paddles are found in vehicles with "half-automatic" transmissions. They can be pushed or pulled to fixed positions and spring back to a neutral state when

the hand is released. Each side represents a shift up or down, regardless whether the paddle is pushed or pulled.

Gear shift paddles share disadvantages of buttons, i.e. pre quipped with functions and required additional HMI.

### 3.2 Prototyping

We set out to develop three concepts based on the initial requirements as well as the constraints and advantages of an existing steering wheel.

#### (1) Pressure sensors activated by finger

The first approach was to implement pressure sensitive sensors that allow measurements on a continuous scale. Figure 2 shows the size of a sensor which required little to almost no space for implementation on the steering wheel. The sensors were mounted on the back of existing gear shift paddles which in turn were fixated. The idea was for the driver to push on the sensor with a finger of his choice. The harder or stronger the push, i.e. the more pressure measured, the less support would be provided.



Figure 2: Implemented pressure sensor on the back of a gear shift paddle [8].

The used pressure sensors weren't originally designed to be used as an interface for a human operator. Pretests showed that the sensor was too sensitive to be actuated on a continuous scale by a human finger. Measurements almost immediately jumped to full saturation with subjectively different amounts of pressures applied by a finger. This led to the early exclusion of this concept.

#### (2) Voice commands

The concept of voice control is established in many other fields in research as well as in daily life. Most contexts push for a good understanding of content and accurate, natural responses. In our scenario we focused on processing times first. We conducted a preliminary study with 15 participants in a desktop setup consisting of a laptop with a

USB plugged microphone. Speech recognition was based on Googles Cloud Speech API. The setup is described in detail by [9]. Results during a simulated take-over scenario reveal processing times around six seconds for several verbal expressions intended to adapt the steering assistance (s. Figure 3).

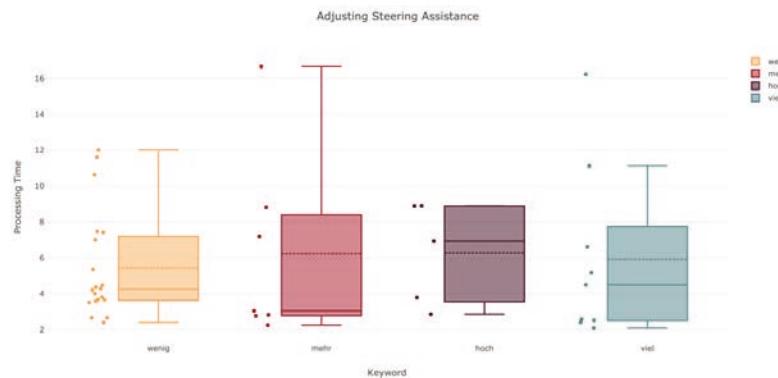


Figure 3: Distribution of processing times while adjusting the cooperation during the takeover using "less", "more", "high", and "much" [9]. The box plots show the quarter percentiles and the mean as dotted line.

We were not able to identify the cause of these comparatively long processing times although we used state-of-the-art software and hardware. Speech command was therefore not chosen for implementation in a vehicle mock-up

### (3) Clutch paddles

The third concept consists of two clutch paddles that were originally developed for racing simulation by Ascher Racing ([10]) (s. Figure 4a). The paddles were designed to control analog inputs, such as the clutch, throttle and brake, based on the Hall effect. Since our simulation software is different than a commercial racing simulation a direct communication is not possible. Therefore, a micro controller (Arduino Mega 2560) was used to translate the voltage differences into values between 0 and 100, i.e. starting position and fully pulled position. The standard steering wheel does not provide an appropriate position for the clutch paddles and since the changes should be reversible an additional aluminum component was manufactured. This in turn is placed on the backside of the steering wheel and has two flaps for mounting the paddles (s. Figure 4b). It was assured that the additional plate does not cause too much friction and not influences the steering behavior. This concept meets all requirements: (1) easy to grasp because of direct attachment on the steering wheel and near to usual gripping position, (2) a continuous scale and (3) time efficiency due to a direct transmitted signal to the controller.

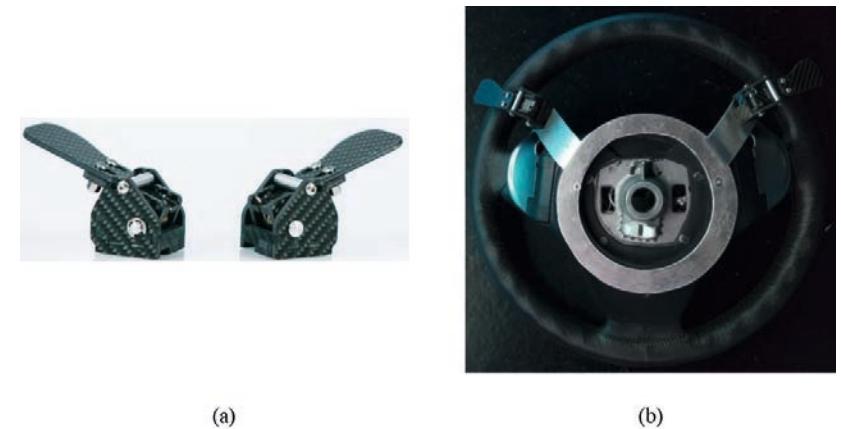


Figure 4: Clutch paddles: (a) picture from Ascher Racing ([10]); (b) installed on the steering wheel of the static driving simulator ([8])

The presented concept refers to three levels of cooperation according to [4]:

1. Authority *or* Allocation of tasks & responsibilities: the clutch paddles directly influence the allocation of the lateral driving task between driver and assistance system. The more the driver pulls the paddles, the more authority he gains.
2. Interface: the presented concept allows an usage at any time during the transition process and unlimited interruptibility.
3. Contact: the concept is mounted on the control element that is crucial for the lateral action, i.e. the steering wheel, and at a position that does not interfere with the normal grip position.

After the implementation of the final concept we conducted another user study with 34 participants. The mean age was 25 years ( $SD = 7.2$ ) among which 27 male participants took part. Drivers drove on a highway for approximately 10 km and then encountered a take-over scenario with a time budget of 6 seconds. The ego vehicle was on the middle lane at the time of the RTI with a blocked right lane. The cause for the RTI was a stranded vehicle on the middle lane. Therefore, the only possibility for an evasive maneuver was to the left lane which the automated system intended to support. All participants were instructed that they could, how they could and when they could use the paddles before starting the experimental drive. No participant reported any confusion or uncertainties about the use of the system.

We tested whether participants used the clutch paddles and if so, what effect it had on the steering support. The results indicate that only 6 out of 34 drivers used the paddles at all and only one driver pulled early enough to have an influence on the steering support. For the other five drivers the steering support was already reduced by the automated system

below the threshold that they have set by the paddles. No participant indicated a reason why they didn't use the paddles in the final questionnaire.

## 4 Conclusion

The goal of this research was to find a suitable interface to enable a driver to adapt steering support in a take-over scenario. Based on technical evaluations we were able to implement analog clutch paddles on the back of a steering wheel. The main reason of excluding the concept of the pressure sensors was that they were too sensitive for our use case. The second concept of voice commands was not further considered because of the long processing times of the system in relation to the time budget during transition phases.

The results of the user study show that only 6 out of 34 participants used the concept of clutch paddles during the transition phase in order to reduce the steering support by the system. We conclude that this was not due to the properties of our developed system but rather due to the general setting of a time critical take-over scenario. Drivers seem to have a very high workload in this situation and don't possess the necessary capabilities to deal with the possibility of reducing the steering support.

## References

- [1] *Standard J3016\_201806 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, SAE J3016, 2018.
- [2] C. Gold, D. Damböck, L. Lorenz, and K. Bengler, ““Take over!” How long does it take to get the driver back into the loop?,” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1, pp. 1938–1942, 2013.
- [3] R. Happee, C. Gold, J. Radlmayr, S. Hergeth, and K. Bengler, “Take-over performance in evasive manoeuvres,” *Accident Analysis & Prevention*, vol. 106, pp. 211–222, 2017.
- [4] B. Karakaya, L. Kalb, and K. Bengler, “Cooperative Approach to Overcome Automation Effects During the Transition Phase of Conditional Automated Vehicles,” in *Uni-DAS 12. Workshop Fahrerassistenzsysteme und automatisiertes Fahren*, pp. 79–86.
- [5] L. Kalb, L. Streit, and K. Bengler, Eds., *Multimodal Priming of Drivers for a Cooperative Take-Over*. 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018.
- [6] L. Kalb, L. Streit, and K. Bengler, Eds., *What Drivers Make of Directional Maneuver Information in a Take-Over Scenario*. 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018.
- [7] M. M. Kienle, “Kooperative Interaktionskonzepte zum Fahren eines teilautomatisierten Fahrzeugs,” Doctoral dissertation, Technical University of Munich, Munich, 2011.
- [8] T. C. Gümüşluk, “Entwicklung von haptischen Konzepten zur kooperativen Fahrzeugführung in der Transitionsphase,” Semesterarbeit, Lehrstuhl für Ergonomie, Technical University of Munich, Garching, 2018.
- [9] Kübra Ipekcioglu, “Interaction Design for Cooperative Automation: Developing and Evaluating a Speech Dialog System for Takeover Situations,” Master's Thesis, Lehrstuhl für Ergonomie, Technical University of Munich, Garching, 2018.
- [10] Ascher Racing GmbH. [Online] Available: <https://www.ascher-racing.com/>.

---

This research was conducted as part of the project PAKoS (Personalisierte, adaptive, kooperative Systeme für automatisierte Fahrzeuge) and funded by the Federal Ministry of Education and Research of Germany. The authors would like to thank Turan Cihad Gümüşluk and Kübra Ipekcioglu for their contributions in carrying out and documenting this research.

# Learning Realistic High Level Decisions for Autonomous Driving at Complex Intersections

Danial Kamran, Martin Lauer and Christoph Stiller\*

**Abstract:** In this work we use deep Q network (DQN) approach to learn high level actions for automated driving at un-signaled intersections. Using lanelet map, history of positions and velocities for vehicles close to the intersection are calculated that provide a generic state representation for the Reinforcement Learning (RL) agent to handle different types of intersections. Moreover, we define safety and utility reward functions and use weighted average of them as total reward in order to evaluate the situation more precisely. The goal is to learn optimal policy which is safe but also not overcautious. According to our experiments, using such reward function, the agent can successfully learn an optimal policy which only stops for the vehicles close to the intersection and will drive fast when vehicles are far from the intersection.

**Keywords:** Decision Making for Automated Driving, Reinforcement Learning, Un-signaled Intersections, Deep Q Networks

## 1 Introduction

One of the most important challenges for automated driving in urban areas is navigating through un-signaled intersections. In such situations, there is no traffic light to give way and take way to vehicles and therefore, the vehicles should themselves realize when to pass the intersection without having collisions. Besides safety, utility and comfort are also two important factors during crossing intersection which means the vehicles should prevent being too much cautious or having jerky maneuvers.

Several rule-based solutions have been presented for handling un-signaled intersections. In [1], hierarchical state machines were designed in order to handle different situations at intersections. In most of cases, time-to-collision [2] between ego vehicle and other vehicles is used as the main reasoning feature for handling intersection crossing. Although TTC can provide a safe and reliable solution, it assumes constant velocity for other vehicles and therefore it has similar reaction to vehicles with different intentions. In other words, TTC as the only reasoning factor can not provide information about future intention of vehicles which will result overcautious maneuvers.

Recently reinforcement learning (RL) techniques have been utilized for high level decision making for automated driving [3]–[6]. Generally RL based decision making approaches try to increase utility as much as possible without forgetting about safety of maneuvers. By utilizing plenty of interaction experiences between agent and environment

\*Institute of Measurement and Control Systems, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany (e-mail: {danial.kamran, martin.lauer, stiller}@kit.edu).

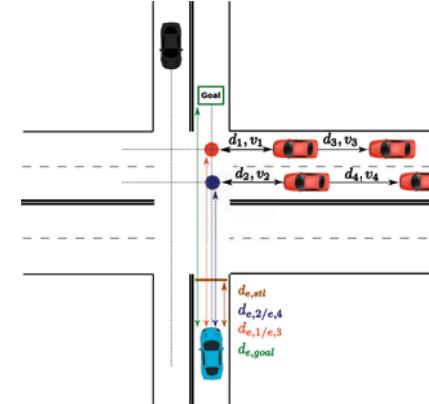


Figure 1: Overview of the yielding scenario and the features we used for situation representation. Blue car is ego vehicle and red cars are vehicles that have collision zone with ego lane. Gray car which has no collision zone with the ego lane is discarded.

ment (simulator), these approaches try to find long term optimal decisions which increase cumulative future reward.

One of the main challenges that still needs to be addressed for RL based automated driving solutions is realistic and generalized problem formulation for better portability in different simulation scenarios and also in real world. Therefore, the features that are being used for learning should be generalized and less dependent on the scenario. Moreover, complexity of learning structure and the amount of required training time for converging to an optimal policy has to be reduced as much as possible in order to prevent damages that the agent is likely to suffer specially due to the so called catastrophic forgetting [7], [8] and also overfitting [9] issues.

The main contribution of this paper is presenting a generic Reinforcement Learning (RL) algorithm for learning optimal behavior to drive through different types of un-signaled intersections safe and also without being conservative. Utilizing Lanelet2 maps [10] with enriched information about geometry of intersections, we propose a parameterized state representation based on the distance and velocities of the vehicles along the curvature (figure 1) which makes it less sensitive to the geometry and structure of intersection.

Another contribution of the proposed approach is learning high level actions as the behavior instead of low level control of the ego vehicle, which makes the RL agent suitable to be used as a conventional decision making module for generating velocity constraints for trajectory planning and control module in an automated driving pipeline (figure 2).

### 1.1 Related Works

There are some approaches that use reinforcement learning for the yielding scenario similar to this paper [3], [4]. In [3], authors try to learn optimal time or sequence of accelerations for passing occluded intersections using deep Q Networks (DQN). They use top view image

of the intersection marked with location and velocity of vehicles as state representation of their RL algorithm. In the output, they compared three different actions: time to start driving, sequential accelerations and creep-and-go for passing the intersection. The results show small amount of collisions and faster drives comparing to the TTC approach. However, the main disadvantage for this work is using image based state representation which makes the algorithm not generalizable and more sensitive to intersection and road geometries as part of state representation. To overcome this problem, parameterized state representation similar to [4], [5] can be more efficient and make the algorithm more generic. In [4] parameterized state has been used for the same yielding scenario but generating high level decisions in the output instead of time-to-go or acceleration values. Although they provide different techniques to improve network convergence, there is no evaluation on complex intersections where we can see how good the algorithm is generalized. Since there is only safety evaluations of the algorithm by measuring percentage of accidents, we also do not know if the presented algorithm is overcautious or not and how fast can drive through intersections.

## 2 Problem Statement

As explained in 1, we assume that the automated vehicle is supposed to cross an un-signaled intersection without any traffic light. We assume that there is one stop line behind intersection and the ego vehicle can stop and wait behind that line until the situation become safe for crossing. The goal is to cross the intersection without blocking oncoming vehicles on the other lanes. For that purpose, the RL agent decides about the action given to trajectory planning and control module during crossing the intersection (figure 2). The action can be stop, drive fast or drive slow (crawl) and is selected using DQN network (2.2). The input for the DQN is other vehicles velocities and distances to the collision point and also the ego vehicle velocity and distance to collision points and the stop line. The main challenge is to provide a generalized approach that can handle a variety of intersections with different number of lanes and structures.

### 2.1 Decision Making and Planning Pipeline

Figure 2 depicts the structure of decision making and planning pipeline that we present in this paper. Decision making module gets the information about upcoming intersections from the Lanelet map [10]. This information includes distance to the stop line and crossing lanes at the intersection. A Lanelet matching module compares the position of detected vehicles from perception with the crossing lanes and keeps those that have collision zone with our vehicle. Distance and velocity along curve for these vehicles are then sent to the RL module in order to choose the best optimal action as velocity constrain for the ego vehicle. Since the path in our scenario is fixed, trajectory planning only performs speed control in order to follow velocity constraints generated by the DQN module.

### 2.2 Reinforcement Learning with Deep Q Networks

Reinforcement Learning (RL) is used to solve different problems having interaction with the environment specifically for robotics. It models the problem as Markov Decision

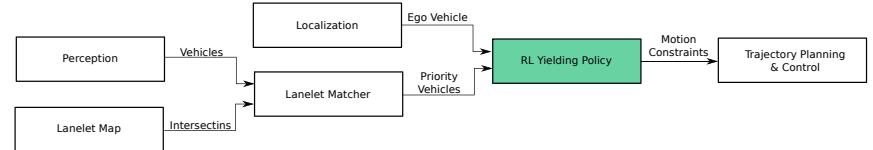


Figure 2: Overview of decision making and planning pipeline used in this paper for driving through intersections in urban area.

Process (MDP). RL agent selects the best action ( $a_t$ ) for the current state ( $s_t$ ) which provides the highest expected cumulative reward. The action is executed and the environment changes to a new state ( $s_{t+1}$ ). According to the reward function defined for the RL agent, the action that was selected is evaluated and its reward value will be provided as  $r(s_t, a_t)$  according to the new situation in the whole environment. After recording several experiences through interactions with the environment, the agent should learn the best actions for each state which will have highest discounted future reward:

$$R_t = \sum_{i=t}^T \gamma^{(i-t)} r(s_i, a_i) \quad (1)$$

In order to find the best policy, action-value function helps to find out the cumulative discounted reward of each action for each state:

$$Q^\pi(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}[r_t + \gamma Q^\pi(s_{t+1}, \pi(s_{t+1}))] \quad (2)$$

If we assume that we have optimal Q function, a greedy policy selects actions with maximum Q value as the best action for each state [11]:

$$\mu(s_i) = \arg \max_a Q(s_i, a | \theta^Q) \quad (3)$$

The Q function can be approximated using deep neural networks known as Deep Q Networks (DQN)[12]. Assuming  $\theta^Q$  as the parameters of the Q function estimator, the Q function is learned through minimizing the loss function using  $B$  random samples from reply buffer:

$$L(\theta^Q) = \sum_{i=1}^B (y_i - Q(s_i, \mu(s_i) | \theta^Q)) \quad (4)$$

where  $y_i$  is the network training target:

$$y_i = r(s_i, a_i) + \gamma Q(s_{i+1}, \mu(s_{i+1}) | \theta^Q) \quad (5)$$

## 3 Learning Yielding Policy at Intersections

### 3.1 Parameterized State Representation

In this section, the main structure of reinforcement learning algorithm for finding best optimal actions during driving at the intersection will be presented. As explained in section 2.1, using lanelet map, the distance and velocity for ego vehicle and all other vehicles at intersection will be calculated along their path as one dimensional values.

Also the ego vehicle distance to the stop line will be required for full representation of the situation for the RL agent. All of these values are normalized assuming maximum distance and maximum velocity of objects in our experiments. Finally, the situation for the RL agent at time step  $t$  is represented as below:

$$\text{situation}_t = \begin{bmatrix} d_{e,stl} & d_{1,e} & d_{2,e} & \dots & d_{n,e} \\ v_e & v_1 & v_2 & \dots & v_n \\ d_{e,goal} & d_{e,1} & d_{e,2} & \dots & d_{e,n} \end{bmatrix}^T$$

The first row in the situation represents distance of ego vehicle to the stop line, velocity of ego vehicle and its distance to the goal as  $d_{e,stl}$ ,  $v_e$  and  $d_{e,goal}$  respectively. In the remaining rows, the distance of each vehicle to the collision zone with ego vehicle, its velocity and the distance of ego vehicle to this vehicle are represented as  $d_{i,e}$ ,  $v_i$  and  $d_{e,i}$  respectively where  $i$  is the id of vehicle. It should be noted that using this representation, only  $N$  vehicles can be feed into DQN network. Therefore, when there are more than  $n$  vehicles at intersection which have collision zone with ego lane, the ones with highest criticality  $c_i$  will be selected where  $c_i$  for each vehicle is calculated as below:

$$c_i = 1 - \frac{\sqrt{d_{i,e}^2 + d_{e,i}^2}}{\sqrt{2}} \quad (6)$$

Using this equation 3.1, the vehicles which have smaller distance to the collision zones and closer to ego vehicle will be more critical and are selected among the others with lower criticality. When the number of vehicles is lower than  $N$ , we fill the rows of representation with 1, 0, 1 values meaning virtual cars at maximum distance with zero velocity that can help the DQN network to neglect them.

In order to give more information about previous distances and velocities for ego and other vehicles for better reasoning, we provide history of situations for the RL algorithm as our final state representation:

$$s_t = [\text{situation}_t \text{ } \text{situation}_{t-1} \text{ } \text{situation}_{t-2} \text{ } \text{situation}_{t-3} \text{ } \text{situation}_{t-4}]$$

## 3.2 Action Representation

In the output, the reinforcement learning policy will choose about the vehicle behavior at intersection. Every decision can be interpreted as a high level action which will be sent for the trajectory planner as a velocity constraint:

- Stop: Full stop with maximum deceleration ( $a_{min}$ )
- Drive-fast: Reach  $v_{fast}$
- Drive-slow: Reach  $v_{slow}$

## 3.3 Reward Function

The reward function that is used for the proposed DQN algorithm will qualify the status of ego vehicle and other vehicles at the intersection in terms of safety and utility. For

that, two factors  $R_{safety}(v_i)$  and  $R_{utility}$  are defined, where  $R_{safety}(v_i)$  qualifies safety of ego vehicle with respect to vehicle  $v_i$  and  $R_{utility}$  calculates the reward according to ego vehicle velocity. The final reward given to the RL algorithm is calculated as weighted average of these two factors:

$$R_{total} = \lambda_s [\min_{0 < i \leq n} R_{safety}(v_i)] + \lambda_u R_{utility} \quad (7)$$

We will explain each of these two reward factors in the remaining parts of this section.

### 3.3.1 Utility Reward

The utility reward will only qualify the velocity of the ego vehicle:

$$R_{utility} = \frac{v_{ego}}{v_{des}} \quad (8)$$

### 3.3.2 Safety Reward

Utility factor of the reward function will motivate the RL agent to drive as fast as possible (i.e.  $v_{ego} = v_{des}$ ) at the intersection. However, in order to yield for other vehicles which are close to the intersection, safety reward will penalize critical situations and force the RL agent to reduce the speed. If the ego vehicle is leaving the intersection, safety reward will penalize situations where the ego vehicle is not able to leave before other vehicles enter the intersection. Therefore, two safety conditions are defined to penalize each of these situations. It should be noted that only one of these two conditions need to be valid for the ego vehicle in order to be safe:

- **Safe Stop (SS):** If the ego vehicle has the possibility to stop before entering collision zone.
- **Safe Leave (SL):** If the ego vehicle has the possibility to leave the collision zone before other vehicles can enter there.

A safety gap for each condition is calculated which should be bigger than a minimum value to be completely safe and the safety reward for that condition become 0 (maximum safety reward). If the safety gap is smaller than critical value, it means the ego vehicle is completely unsafe and the safety reward would be -1 (minimum safety reward). For other cases where the ego vehicle is not completely safe or unsafe, the reward will be calculated as below:

$$R_{SS(v_i)} = \begin{cases} -1 & \text{if } d_{SG} < d_{critical} \text{ (unsafe situation)} \\ 0 & \text{if } d_{SG} > d_{min} \text{ (fully safe situation)} \\ -\left(\frac{d_{SG}-d_{e,stl}}{d_{e,stl}-d_{critical}}\right)^2 & \text{else} \end{cases} \quad (9)$$

$$R_{SL(v_i)} = \begin{cases} -1 & \text{if } t_{SG} < t_{critical} \text{ (unsafe situation)} \\ 0 & \text{if } t_{SG} > t_{min} \text{ (fully safe situation)} \\ -\left(\frac{t_{SG}-t_{min}}{t_{min}-t_{critical}}\right)^2 & \text{else} \end{cases} \quad (10)$$



Figure 3: Top view images from the simulation during one episode. Ego vehicle (red vehicle) stops behind stop line in order to yield to other vehicles (image left), starts driving through the intersection (middle image) and reaches the goal point (right image).

Where  $d_{SG}$  and  $t_{SG}$  are the safety gaps for SS and SL conditions respectively. Also  $d_{min}$  and  $t_{min}$  are minimum required safety gap to be completely safe for each condition.

For each vehicle ( $v_i$ ), one of the safety conditions should be valid to make it a complete safe situation, i.e., ego vehicle should either be able to stop behind collision zone with  $v_i$  or leave that collision zone before  $v_i$  can enter there. Therefore, maximum of  $R_{SS}(v_i)$  and  $R_{SL}(v_i)$  is selected as final safety reward according to that vehicle:

$$R_{safety}(v_i) = \max(R_{SS}(v_i), R_{SL}(v_i)) \quad (11)$$

Finally, we take minimum safety reward for all vehicles as the total safety reward:

$$R_{safety} = \min_{0 < i \leq n} R_{safety}(v_i) \quad (12)$$

In this way we make sure that for each vehicle at least one of the safety conditions is valid and otherwise the ego vehicle is not safe.

## 4 Training and Evaluation

### 4.1 Simulation Environment

In order to learn the proposed DQN approach and also evaluate it, we use Carla simulator [13] to simulate automated driving through an un-signaled intersection with random vehicles driving at the other sides of intersection (figure 3). At the beginning of each training episode, ego vehicle and random number of other vehicles are positions at random distances from the intersection. Each vehicle has random desired speed and is randomly assigned to drive on one of intersection lanes. The position of stop line and also geometry of all intersection lanes are mapped to be used for situation representation as explained in section 2.1.

### 4.2 Training Setup

Figure 4 shows overall structure of DQN network. In the input, current state ( $s_t$ ) is processed using  $h_{ego}$  and  $h_{vi}$  hidden layers for motion features of ego vehicle and other vehicles accordingly. Similar to [4], we used shared weights for processing all sub layers of other vehicles ( $w_{veh}$ ) in order to be independent from the order of vehicle features in

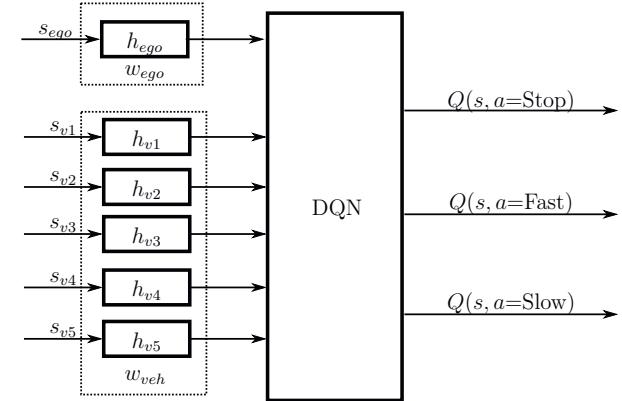


Figure 4: Structure of proposed feature extraction and DQN network. All hidden layers for processing intersecting vehicles ( $h_{vi}$ ) share same weights ( $w_{veh}$ ) and ego vehicle data is processed by a separate hidden layer ( $h_{ego}$ ). All extracted features are then concatenated as the input of DQN network.

the state. After first hidden layer, all extracted features for ego vehicle and other vehicles are concatenated and fed into the DQN network in order to estimate expected cumulative reward for each action at the current state ( $Q(s_t, a_t)$ ).

### 4.3 Evaluation

After training more than 1000 episodes with 600,000 experiences, the learned policy reached collision free results for 10 test scenarios. As an example, figure 6 shows which actions have highest  $Q$  values using the trained policy at different distances and velocities of ego vehicle ( $d_{e,goal}$  and  $v_e$ ) at an intersection without any other vehicle. According to this figure, the policy drives slow when the ego vehicle is far from the intersection (to be careful if any vehicle enters the intersection area) and when it gets closer starts to drive fast. For better evaluating the trained policy, we compared it with a rule-based policy for 10 evaluation scenarios. The rule-based policy selects the highest velocity at each decision time which is safe according to the safety conditions explained in section 3.3.2. As it is visible in figure 5, performance of learned agent is close to the rule-based agent and in some cases it outperforms the rule based decisions.

## 5 Conclusions

In this paper, a DQN network was proposed as a decision making module in order to learn optimal high level actions for automated driving through un-signaled intersections. Defining safety and utility terms in the reward function, we tried to learn actions which are safe and also not too much conservative as the output of DQN network. Results show that the agent can learn optimal behaviors in order to drive as fast as possible. However,

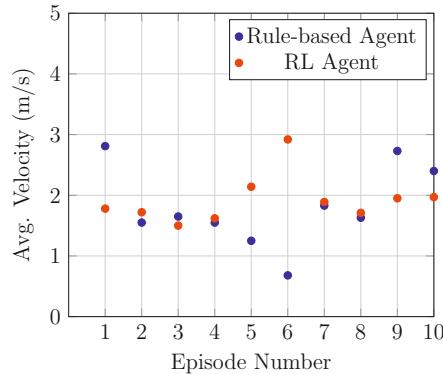
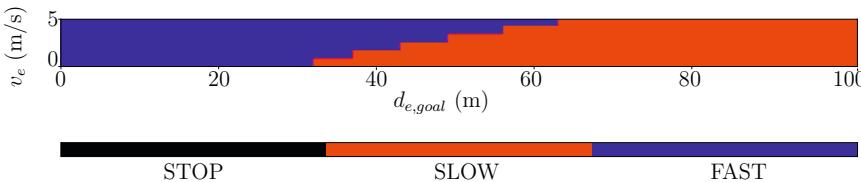


Figure 5: Comparing trained policy and rule based policy for 10 test scenarios.

Figure 6: Evaluating the best action ( $\arg \max_a Q(s, a)$ ) for different distances to the goal and velocities ( $d_{e,goal}$  and  $v_e$ ) of ego vehicle assuming no other vehicle at the intersection.

there are still some challenges that should be addressed in the future works specifically about safety verification of the DQN decisions in order to provide a completely safe and also not overcautious policy.

## 6 Acknowledgment

This research is accomplished within the project “UNICARagil” (FKZ 6EM00287). We acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF).

## References

- [1] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, *et al.*, “Making Bertha Drive—An Autonomous Journey on a Historic Route,” *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.
- [2] R. van der horst and J. Hogema, “Time-to-collision and collision avoidance systems,” Jan. 1994.

- [3] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, “Navigating occluded intersections with autonomous vehicles using deep reinforcement learning,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [4] T. Tram, A. Jansson, R. Grönberg, M. Ali, and J. Sjöberg, “Learning negotiating behavior between cars in intersections using deep q-learning,” *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.
- [5] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, “High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.
- [6] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving,” *arXiv e-prints*, arXiv:1610.03295, 2016.
- [7] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in, ser. Psychology of Learning and Motivation, G. H. Bower, Ed., vol. 24, Academic Press, 1989, pp. 109 –165.
- [8] D. Isele and A. Cosgun, “Selective experience replay for lifelong learning,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 2012.
- [10] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, *et al.*, “Lanelet2: A high-definition map framework for the future of automated driving,” in *Proc. IEEE Intell. Trans. Syst. Conf.*, Hawaii, USA, 2018.
- [11] C. J.C. H. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.
- [12] V. Mnih and D. Silver, “Playing Atari with Deep Reinforcement Learning,” 2013. arXiv: 1312.5602.
- [13] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

# Systemarchitektur und Fahrmanöver zum sicheren Anhalten modularer automatisierter Fahrzeuge

Stefan Ackermann\* und Hermann Winner†

**Zusammenfassung:** Maschinelle Systeme übernehmen einen immer größer werdenden Anteil der dynamischen Fahraufgabe automatisierter Fahrzeuge. Funktionale Degradationen können die Fähigkeiten dieser Systeme negativ beeinflussen, sodass sie die Fahraufgabe nicht weiter erfüllen können. In diesen Fällen wird bei höher automatisierten Systemen die Fahraufgabe von einer maschinellen Rückfallebene übernommen. Im Rahmen des Forschungsprojekts UNICARagil wird eine modulare und dienstbasierte funktionale Fahrzeugarchitektur entwickelt, für die in diesem Beitrag die Anforderungen und die Systemarchitektur einer geeigneten funktionalen Rückfallebene vorgestellt werden und der weitere Forschungsbedarf hinsichtlich der erforderlichen Fähigkeiten der Teilstrukturen, ihrer gegenseitigen Abhängigkeiten und der Absicherung der Teil- und Gesamtstrukturen erläutert wird.

**Schlüsselwörter:** Automatisiertes Fahren, Risikominimaler Zustand, Sicheres Anhalten, UNICARagil

## 1 Einleitung

Die technische Entwicklung intelligenter Transportsysteme führt zu automatisierten Fahrfunktionen, die einen immer größer werdenden Anteil der dynamischen Fahraufgabe übernehmen. Doch was passiert, wenn die Fähigkeiten dieser Fahrfunktionen durch funktionale Degradationen nicht mehr für die Erfüllung der Fahraufgabe ausreichen? Nach SAE Norm J3016 [1] werden für höher automatisierte Fahrzeuge (SAE Level 3 und höher) maschinelle Systeme als Rückfallebene der primären Fahrfunktionen vorausgesetzt. Auch vonseiten der Gesetzgebung zeichnen sich Forderungen nach entsprechenden Rückfalllösungen ab [2]. Für Fahrzeuge mit SAE Level 3 wird beispielsweise das ECE-Homologationsverfahren [3] von der UNECE-Arbeitsgruppe "Automatically Commanded Steering Function" (ACSF) [4] überarbeitet.

Im Projekt UNICARagil [5] wird die modulare und dienstbasierte Softwarearchitektur "Automotive Service-Oriented Software Architecture" (ASOA) zur Erfüllung der Fahrfunktionen eines automatisierten Fahrzeugs entwickelt. Dieser Ansatz ermöglicht die Aufteilung der funktionalen Architektur in separate Dienste. Für diese, im Kontext der Fahrzeugautomatisierung neuartige, modulare und dienstbasierte Softwarearchitektur werden in diesem Beitrag die Anforderungen und die davon abgeleitete Systemarchitektur ei-

\*Stefan Ackermann ist Wissenschaftlicher Mitarbeiter am Fachgebiet Fahrzeugtechnik der TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt, (E-Mail: ackermann@fzd.tu-darmstadt.de).

†Hermann Winner ist Leiter des Fachgebietes Fahrzeugtechnik der TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt (E-Mail: winner@fzd.tu-darmstadt.de).

ner geeigneten funktionalen Rückfallebene "Sicheres Anhalten" vorgestellt, die bisherigen Erkenntnisse darüber zusammengefasst und der weitere Forschungsbedarf erläutert.

### 1.1 "Sicheres Anhalten"

Die Bezeichnung "Sicheres Anhalten" wurde im Projekt "Automatisch fahrerlos fahrendes Absicherungsfahrzeug für Arbeitsstellen auf Autobahnen" (aFAS) [6] für die Benennung eines Betriebszustands verwendet. In der Dissertation von Reschka [7] wird die Bezeichnung dagegen für ein Fahrmanöver genutzt. Im Rahmen dieses Beitrags wird die maschinelle Rückfallebene der Fahrfunktionen eines automatisierten Fahrzeugs "Sicheres Anhalten" (engl.: Safe Halt) genannt.

### 1.2 Das Forschungsprojekt UNICARagil

Für das Gemeinschaftsprojekt UNICARagil [5] hat sich ein Konsortium aus acht deutschen Universitäten und mehreren Industriepartnern gebildet. Ziel des Projekts ist die Erforschung einer disruptiven modularen Architektur für vielfältige, agile und vollautomatisierte Fahrzeugkonzepte. Die Modularität erstreckt sich hierbei sowohl über die Hard- als auch die Software-Architektur. Entwickelt werden agile Fahrplattformen mit integrierten Dynamikmodulen, die Antriebs-, Brems- und Lenkfunktion kombinieren. Im Verlauf des Projekts werden vier prototypische vollautomatisierte Fahrzeugausprägungen aufgebaut.

### 1.3 Struktur des Beitrags

Die Struktur dieses Beitrags gliedert sich in fünf Kapitel. In Kapitel 2 wird die Hardwarearchitektur des automatisierten Fahrzeugs im Projekt UNICARagil vorgestellt. In Kapitel 3 werden Anforderungen an eine dienstbasierte maschinelle Rückfallebene formuliert. Eine von diesen Anforderungen abgeleitete Systemarchitektur und ihre Einzelfunktionen werden in Kapitel 4 vorgestellt. In Kapitel 5 wird schließlich der Erkenntnisstand zusammengefasst und ein Ausblick auf die zukünftige Forschungsarbeit gegeben.

## 2 Hardwarearchitektur im Projekt UNICARagil

Die Hardwarearchitektur des automatisierten Fahrzeugs im Projekt UNICARagil [5] ist in die drei Ebenen Großhirn, Stammhirn und Rückenmark gegliedert, von denen das Stammhirn und das Rückenmark ausfallsicher (Fail-Operational) gestaltet sind [8]. Die Bezeichnung der Ebenen orientiert sich an den jeweiligen Entsprechungen im menschlichen Körper. Jedes Steuergerät im Fahrzeug ist einer dieser Hardwareebenen zugeordnet. Abbildung 1 zeigt das Konzept der Hardwarearchitektur des automatisierten Fahrzeugs im Projekt UNICARagil. Das Großhirn übernimmt die Aufgaben der Umfeldwahrnehmung und der Verhaltens-, Manöver- und Trajektorienplanung. Zur Umfeldwahrnehmung sind vier Sensormodule an den Fahrzeugecken angebracht. Jedes dieser Sensormodule erzeugt mithilfe von Lidar, Radar sowie Kameras Umfeldinformationen, die im Großhirn zu einem Umfeldmodell fusioniert werden. Basierend auf diesem Umfeldmodell und Informationen

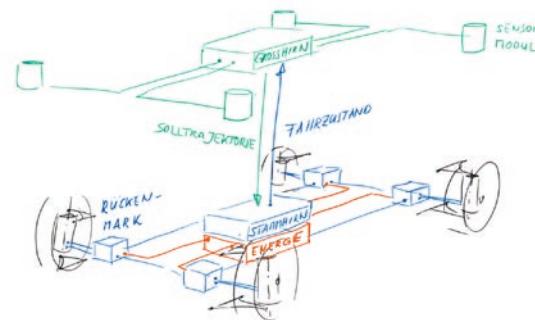


Abbildung 1: Skizze der Hardwarearchitektur des automatisierten Fahrzeugs im Projekt UNICARagil [5]

aus einer digitalen Karte wird eine Solltrajektorie geplant und an die Ebene des Stammhirns übertragen.

Eine der Aufgaben des Stammhirns ist die Bewegungsregelung des Fahrzeugs. Der hierfür benötigte Fahrdynamikzustand wird durch eine mit dem Stammhirn verbundene Fahrdynamikzustandsschätzung erfasst. Die Bewegungsregelung vergleicht diesen Fahrdynamikzustand mit dem durch die Solltrajektorie beschriebenen Sollzustand und erzeugt entsprechende Stellgrößen für die Fahrzeugaktorik. Die Solltrajektorie wird dabei nur bei einer Änderung des dynamischen Umfelds, nicht aber bei einer Regelabweichung angepasst. Zusätzlich zur Bewegungsregelung ist dem Stammhirn die funktionale Rückfallebene „Sicheres Anhalten“ zugeordnet. Mithilfe dieser Rückfallebene wird gewährleistet, dass das automatisierte Fahrzeug jederzeit in einen risikominimalen Zustand überführt werden kann. Die Rückfallebene erzeugt hierfür eine kollisionsfreie Solltrajektorie für die Bewegungsregelung.

Die von der Bewegungsregelung erzeugten Stellgrößen werden an die Ebene des Rückenmarks übermittelt und dort für die Regelung der Fahrzeugaktorik genutzt.

Auf allen drei Ebenen wird die dienstbasierte Softwarearchitektur ASOA eingesetzt. Diese Architektur ermöglicht eine ebenen- und hardware-übergreifende Kommunikation zwischen den Diensten des automatisierten Fahrzeugs.

### 3 Anforderungen an eine dienstbasierte maschinelle Rückfallebene

Die maschinelle Rückfallebene eines automatisierten Fahrzeugs wird aktiviert, wenn die Fähigkeiten (engl.: abilities) der primären Fahrfunktionen durch funktionale Einschränkungen nicht länger für die Erfüllung der dynamischen Fahraufgabe ausreichen. In diesen Fällen überführt die Rückfallebene das Fahrzeug in einen risikominimalen Zustand. Reschka beschreibt in seiner Dissertation [7] drei verschiedene Fahrmanöver, mit denen sich ein Fahrzeug in einen risikominimalen Zustand überführen lässt. Beim ersten Fahrmanöver wird das Fahrzeug mit einer starken Verzögerung (bis zu  $6 \text{ m/s}^2$ ) bei gleichzeitig konstan-

tem Lenkwinkel in den Stillstand überführt. Beim zweiten Fahrmanöver wird ohne eine weitere Umfeldsensierung die zuletzt geplante Bahn durch Nutzung der Querführung bei gleichzeitiger Verzögerung in den Stillstand abgefahren. Beim dritten Fahrmanöver wird das Fahrzeug an einer geeigneten Stelle angehalten. Das automatisierte Fahrzeug muss hierfür noch in der Lage sein, Hindernisse wahrzunehmen und die umliegende Geometrie und Fahrstreifenmarkierungen zu erkennen.

Im Rahmen dieses Beitrags wird ein viertes Fahrmanöver vorgeschlagen. Solange die Fähigkeiten der Planungsebene Großhirn für die Erfüllung ihrer Funktionen ausreichen, wird zusätzlich zur Solltrajektorie eine Notbahn geplant und an die ausfallsicher gestaltete Rückfallebene auf dem Stammhirn übertragen. Im Gegensatz zum ersten und zweiten Fahrmanöver endet diese Notbahn an einem risikominimalen Ort, der sich möglichst außerhalb des fließenden Verkehrs befindet. Sobald die Rückfallebene aktiv ist, wird anstatt der Solltrajektorie die vorausberechnete Notbahn abgefahren. Im Unterschied zum dritten Fahrmanöver beschränken sich die hierfür benötigten Fähigkeiten auf die Zwischenspeicherung der Notbahn, die Erfassung von Hindernissen und der Planung eines kollisionsfreien Geschwindigkeitsprofils zur Abfahrt der Notbahn. Das vorgeschlagene Fahrmanöver beginnt mit einer Anpassung auf eine niedrigere Fahrzeuggeschwindigkeit, um das Risiko für die Fahrzeuginsassen und das Fahrzeugumfeld zu minimieren. Die Abfahrt der Notbahn bis zu ihrem Ende erfolgt auf Basis eines örtlichen Geschwindigkeitsprofils. Werden während des Manövers Hindernisse auf der Notbahn erkannt, so wird die Fahrzeuggeschwindigkeit zur Kollisionsvermeidung entsprechend angepasst. Der örtliche Verlauf der Notbahn wird dabei nicht verlassen.

Die für die Ausführung des beschriebenen Fahrmanövers benötigten Funktionen werden in Form von Diensten in der Softwarearchitektur ASOA implementiert. In dieser Architektur existiert ein Dienst für die Bewegungsregelung des Fahrzeugs. Dieser setzt die von der Planungsebene Großhirn berechnete Solltrajektorie in Stellbefehle für die Fahrzeugaktorik um. Um auf die Implementierung eines zweiten Bewegungsreglers speziell für das „Sichere Anhalten“ verzichten zu können, wird in der Rückfallebene eine Solltrajektorie in der gleichen Kodierung wie die der Solltrajektorie der Planungsebene erzeugt. Mit diesem Vorgehen kann der existierende Bewegungsregler sowohl von der Planungs- als auch von der Rückfallebene genutzt werden.

Um die Latenz für die Aktivierung der Rückfallebene zu minimieren, arbeiten die Dienste des „Sicheren Anhalts“ im „Hot Standby“-Betrieb. Zur kollisionsfreien Abfahrt der Notbahn wird somit auch in anderen Betriebsmodi das Fahrzeugumfeld von der Rückfallebene erfasst, eine kollisionsfreie Trajektorie erzeugt und den weiteren Diensten der ASOA zur Verfügung gestellt. Mit diesem Ansatz kann die dienstbasierte Softwarearchitektur latenzminimal auf die Rückfallebene umschalten. Zusätzlich wird mit diesem Vorgehen die Funktionsfähigkeit der Rückfallebene nachgewiesen, indem funktionale Degradationen des Systems bereits im inaktiven Betrieb aufgedeckt werden.

### 4 Systemarchitektur der maschinellen Rückfallebene „Sicheres Anhalten“

Zur Sicherstellung der störungsfreien Verfügbarkeit der Rückfallebene ist die relevante Hardware- und Kommunikationsarchitektur des automatisierten Fahrzeugs im Projekt

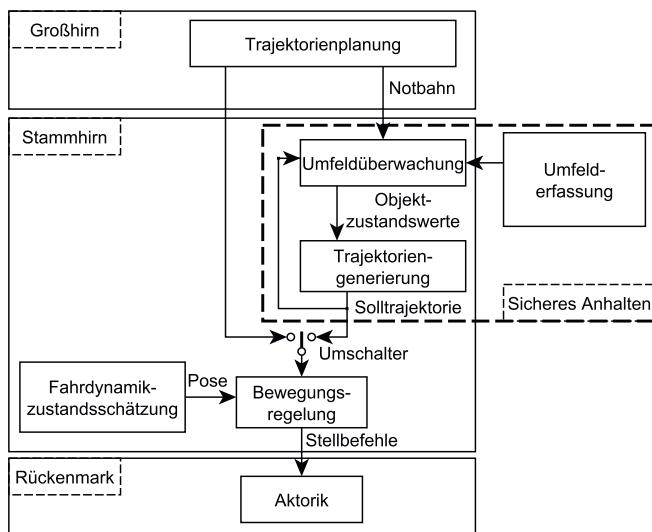


Abbildung 2: Zuordnung der Umfelderfassung und der relevanten Dienste der Rückfallebene “Sicheres Anhalten“ zur Hardwarearchitektur des Projekts UNICARagil

UNICARagil ausfallsicher gestaltet [8]. Das im vorherigen Kapitel beschriebene Fahrmanöver setzt eine dedizierte örtliche Notbahn voraus, die zusammen mit einem örtlichen Geschwindigkeitsprofil geplant wird. Die Aufgabendefinition des Trajektorienplaners auf dem Großhirn wird daher entsprechend erweitert.

Für die Überwachung des Fahrzeugumfelds wird eine Umfeldsensorik benötigt, die unabhängig von der primären Umfelderfassung des Fahrzeugs ist. Mithilfe dieser Sensorik werden potenzielle Kollisionsobjekte auf dem örtlichen Verlauf der Notbahn erfasst. Auf eine Freiflächenerkennung durch die zusätzliche Sensorik kann verzichtet werden, da die Notbahn nicht verlassen werden soll und eine grundsätzliche Befahrbarkeit schon bei der Planung der Bahn abgesichert ist.

Das Gesamtsystem der maschinellen Rückfallebene “Sicheres Anhalten“ umfasst die Umfelderfassung sowie die Dienste “Umfeldüberwachung“ und “Trajektoriengenerierung“. Beide Dienste sind in die ASOA integriert. Auf Abbildung 2 sind die Umfelderfassung und die relevanten Dienste der Rückfallebene “Sicheres Anhalten“ dargestellt und der Hardwarearchitektur des Projekts UNICARagil zugeordnet.

### Notbahn oder Nottrajektorie?

Für das beschriebene Fahrmanöver zur Überführung des Fahrzeugs in einen risikominimalen Zustand wird von der Planungsebene eine Bahn und keine Trajektorie berechnet,

da der zeitliche Verlauf der Fahrzeugposen (Position und Orientierung) einer Trajektorie ungültig wird, sobald Hindernisse auf der Trajektorie eine Anpassung der Fahrzeuggeschwindigkeit erfordern. Aus diesem Grund wird vom Großhirn eine Bahn geplant, die den örtlichen Verlauf der Fahrzeugposen beschreibt und durch ortsbabhängige Geschwindigkeitsvorgaben ergänzt wird. Nach Aktivierung der Rückfallebene und während der anfänglichen Geschwindigkeitsanpassung dienen diese Vorgaben als Auslegungsgeschwindigkeiten für die Trajektoriengenerierung. Nach abgeschlossener Anpassung der Fahrzeuggeschwindigkeit werden sie von der Trajektoriengenerierung als Maximalgeschwindigkeitsvorgaben genutzt.

### 4.1 Umfelderfassung

Zur kollisionsfreien Abfahrt der Notbahn wird das Fahrzeugumfeld mit einer von der restlichen Umfelderfassung des Fahrzeugs unabhängigen Sensorik erfasst. Je nach Anwendungsbereich des automatisierten Fahrzeugs werden geeignete Sensorprinzipien ausgewählt. Die Sensorreichweite ist so zu wählen, dass in jedem Fahrzustand rechtzeitig auf Hindernisse im Fahrzeugumfeld reagiert werden kann. Dies bedeutet, dass insbesondere in den Hauptbewegungsrichtungen des Fahrzeugs eine größere Sensorreichweite gefordert ist.

Im Projekt UNICARagil wird ein Fahrzeug gezeigt, dass eine 360°-Bewegungsrichtung haben kann. Die maximalen Geschwindigkeiten werden dabei in den beiden Fahrzeuggängenrichtungen gezeigt und betragen vorwärts etwa 20 m/s und rückwärts 10 m/s. Die entwickelte Hardware des Stammhirns nutzt eingebettete Systeme (engl.: embedded systems) und stellt somit nicht die von Hochleistungsrechnern mit Grafikkarten bekannte Rechenleistung zur Verfügung. Aus diesem Grund wird eine sensornahen Verarbeitung der Sensorsrohdaten angestrebt. Aufgrund der geforderten Sensorreichweite in den Hauptbewegungsrichtungen und der sensornahen Rohdatenverarbeitung werden für die Rückfallebene im Projekt UNICARagil Radarsensoren in Fahrzeugfront und -heck eingesetzt.

Zur Absicherung der 360°-Bewegungsrichtung des Fahrzeugs werden die beiden Radarsensoren durch eine 360°-Umfelderfassung ergänzt. Fahrten mit großen Schwimmwinkel sind im Projekt nur mit niedrigen Fahrzeuggeschwindigkeiten erlaubt. Es wird somit eine vergleichsweise geringe Sensorreichweite gefordert. Mithilfe dieser zweiten Umfelderfassung wird zusätzlich die Abfahrt der Notbahn mit Auslegungsgeschwindigkeit abgesichert. Aufgrund der geforderten Sensorreichweite und 360°-Überwachung sowie der sensornahen Rohdatenverarbeitung werden in diesem Projekt Ultraschallsensoren und Kameras mit Fischaugenobjektiven als zweiter Bestandteil der Umfelderfassung für das “Sichere Anhalten“ eingesetzt.

### 4.2 Umfeldüberwachung

Die Sensoren der Umfelderfassung sind mit dem Stammhirn und damit mit dem Dienst “Umfeldüberwachung“ verbunden. Dieser fusioniert die Daten der Umfeldsensoren zu einem gemeinsamen Umfeldmodell. Zusätzlich empfängt dieser Dienst die zuletzt geplante Notbahn vom Trajektorienplaner des Großhirns.

Aufgabe dieses Dienstes ist die Prüfung der Notbahn auf mögliche Kollisionsobjekte. Mit dem Wissen über den örtlichen Verlauf der Notbahn und dem geplanten Geschwindigkeitsprofil kann die eigene zukünftige Fahrzeugpose (Position und Orientierung) voraus-

berechnet werden. Zusammen mit der eigenen Fahrzeugausdehnung (Breite und Höhe des Fahrzeugs ergänzt mit einem Sicherheitsabstand) und dem zeitlichen Verlauf der eigenen Fahrzeugposen wird somit ein örtlicher und zeitlicher Raum im Umfeld beschrieben, der als Fahrschlauch bezeichnet wird. Mithilfe der durch die Umfelderfassung detektierten Objekte und deren Prädiktionen werden Kollisionsobjekte aufgedeckt, die mit diesem Fahrschlauch sowohl örtlich als auch zeitlich kollidieren. Wird ein solches Objekt detektiert erfolgt die Planung eines kollisionsvermeidendenden örtlichen Geschwindigkeitsprofils, das von der nachfolgenden Trajektoriengenerierung umgesetzt wird, s.a. [9].

### 4.3 Trajektoriengenerierung

Der Dienst "Trajektoriengenerierung" erzeugt kollisionsfreie und geschwindigkeitsangepasste Solltrajektorien für die Bewegungsregelung auf dem Stammhirn. Die Kodierung der Solltrajektorie ist dabei identisch zur Kodierung der Solltrajektorie, die vom Trajektorienplaner des Großhirns erzeugt wird. Mit diesem Ansatz ist es nicht erforderlich, dass eine dedizierte Bewegungsregelung für die Rückfallebene implementiert wird. Um eine Verwechslung mit der Solltrajektorie des Großhirns zu vermeiden wird im weiteren Verlauf des Beitrags die Solltrajektorie, die von der Rückfallebene generiert wird, als Nottrajektorie bezeichnet.

Der örtliche Verlauf der Nottrajektorie wird vollständig von der zuletzt vom Großhirn übermittelten Notbahn übernommen, da ein örtliches Abweichen von der Notbahn aufgrund fehlender Informationen über das Fahrzeugumfeld nicht abgesichert werden kann. Um aus dem örtlichen Verlauf der Notbahn eine explizite Beschreibung der Nottrajektorie zu erzeugen, wird ein örtliches Geschwindigkeitsprofil gesucht. Dieses Geschwindigkeitsprofil wird während der Abfahrt der Nottrajektorie entsprechend des erfassten Fahrzeugumfelds aktualisiert. Eine Änderung des Geschwindigkeitsprofils bedingt damit auch eine Änderung der Nottrajektorie. Das Geschwindigkeitsprofil wird aus verschiedenen Einzelprofilen erzeugt. Das erste Geschwindigkeitsprofil ist die Auslegungsgeschwindigkeit für die Notbahn, die in der Planungsphase vom Großhirn berechnet wurde. Ein weiteres Geschwindigkeitsprofil wird auf Basis der Objektzustandswerte, die von der Umfelderfassung detektiert wurden, erzeugt. Das Profil wird so geplant, dass das Fahrzeug dynamischen Objekten mit einer definierten Zeitlücke folgt und hinter statischen Objekten mit einem definierten Abstand anhält.

Bei der Planung der Auslegungsgeschwindigkeit der Notbahn werden die fahrdynamischen Grenzen des Fahrzeugs beachtet. Die von der Planung gewählten Ausführungsgrenzen sind allerdings nicht statisch, sondern ändern sich sowohl über den Ort als auch über die Zeit. Ein Dienst in der Softwarearchitektur des Fahrzeugs liefert die zum aktuellen Zeitpunkt gültigen Ausführungsgrenzen. Zusammen mit dem örtlichen Verlauf der Notbahn wird auf Basis dieser Ausführungsgrenzen ein Geschwindigkeitsprofil geplant, das vorbeugt, dass das Fahrzeug in einen instabilen Fahrdynamikbereich gerät.

Über einen ortsdiskreten Vergleich der Geschwindigkeitsprofile wird das Minimum für die Trajektoriengenerierung ermittelt.

Aus dem örtlichen Verlauf der Notbahn und dem berechneten Geschwindigkeitsprofil wird schließlich eine von dem Bewegungsregler interpretierbare Nottrajektorie erzeugt. Die vom Großhirn übermittelten Positionen der Notbahn liegen in einem global ellipsoidischen Koordinatensystem. Als geodätisches Referenzsystem wird im Projekt UNICARagil ETRS89 [10] verwendet. Die global ellipsoidischen Koordinaten der Notbahn werden in einem ers-

ten Schritt in global kartesische Koordinaten umgerechnet. Diese weiterhin dreidimensionalen Koordinaten werden im zweiten Schritt in eine zweidimensionale Ebene gedreht. Diese Ebene tangiert das für das Referenzsystem ETRS89 genutzte GRS80 Ellipsoid im Aufpunkt der Notbahn. Für die Drehung der dreidimensionalen Koordinaten werden Vektoren zwischen dem Aufpunkt der Notbahn und allen nachfolgenden Koordinaten der Notbahn erzeugt. Jeder dieser Vektoren wird anschließend in die zweidimensionale Ebene transformiert. Nach der Umrechnung liegt der örtliche Verlauf der Notbahn in einem zweidimensionalen kartesischen Koordinatensystem mit Ursprung im Aufpunkt der Notbahn vor. Unverändert ist die Notbahn nur an diskreten Koordinaten definiert. Für die vollständige Beschreibung der Bahn wird zwischen diesen Koordinaten eine Interpolation durchgeführt. Eine Randbedingung der Nottrajektorie ist, dass der enthaltene Beschleunigungsverlauf stetig sein soll. Ein linearer Beschleunigungsverlauf führt nach zweimaliger Integration auf einen kubischen Positionsverlauf. Aus diesem Grund erfolgt die Interpolation zwischen den diskreten Positionen mithilfe eines kubischen Splines. Das berechnete Geschwindigkeitsprofil ist gleichermaßen nur an diskreten Stellen definiert. Somit wird auch hier eine Interpolation zwischen den diskreten Stellen durchgeführt. Da ein stetiger Beschleunigungsverlauf gefordert ist, wird eine quadratische Interpolation gewählt. Abschließend ist auch die Fahrzeugorientierung nur an den diskreten Aufpunkten der Bahn definiert. Analog findet daher auch hier eine kubische Interpolation statt. Alle Splines nutzen als Splineparameter den Weg über den örtlichen kubischen Spline.

Die Nottrajektorie besteht aus zeitlich äquidistanten Sollzuständen. Mithilfe des interpolierten Geschwindigkeitsprofils werden die Wegpunkte auf dem örtlichen Spline zu den äquidistanten Zeitpunkten ermittelt. Die Posen der Trajektorie werden durch Auswertung der entsprechenden Splines an den berechneten Wegpunkten ermittelt. Analog wird für jedes Trajektorienelement auch ein Geschwindigkeits- und Beschleunigungsvektor berechnet. Ergebnis ist eine zeitlich äquidistante diskrete Beschreibung einer Nottrajektorie in Pose, Geschwindigkeit und Beschleunigung. Diese Trajektorie wird an die Bewegungsregelung übertragen.

## 5 Fazit und Ausblick

Bei automatisierten, fahrerlosen Fahrzeugen steht bei funktionalen Einschränkungen der dynamischen Fahrfunktionen der menschliche Fahrer nicht als Rückfallebene zur Verfügung. In diesen Fällen erfüllt ein maschinelles System die Funktionen einer Rückfallebene, um das Fahrzeug in einen risikominimalen Zustand zu überführen. Für diese Zustandsübergabe wurden aus der Literatur bekannte Fahrmanöver vorgestellt und hinsichtlich ihrer Eigenschaften analysiert. Auf Basis dieser Analyse wurde ein neues Fahrmanöver vorgestellt und dargelegt, dass dieses Manöver die Überführung eines automatisierten Fahrzeugs in einen risikominimalen Zustand außerhalb des fließenden Verkehrs ermöglicht. Im nächsten Schritt wurden Anforderungen an die funktionale Rückfallebene zur Ausführung des neuen Fahrmanövers vorgestellt. Da die zeitlichen Informationen einer Trajektorie ungültig werden, sobald die Fahrzeuggeschwindigkeit aufgrund erfasster Hindernisse angepasst wird, plant das Großhirn anstatt einer Trajektorie eine Bahn mit örtlichem Geschwindigkeitsprofil. Es wurde gezeigt, dass die Modularität der Hard- und Software des im Projekt UNICARagil erforschten automatisierten Fahrzeugs für die Implementierung der für das Manöver notwendigen Funktionen geeignet ist und entsprechende Aufgabende-

finitionen für die drei Hardwareebenen abgeleitet. Für die Hardwareebene des Stammhirns wurden die Funktionen der Rückfallebene in Dienste unterteilt und ihre jeweiligen Teilfunktionen beschrieben.

In zukünftigen Forschungsarbeiten sollten die vorgestellten Teilfunktionen der Rückfallebene näher untersucht werden. Welche Fähigkeiten müssen diese Teilfunktionen erfüllen? Mit welchen Metriken können diese Fähigkeiten bewertet werden? Wie können die Teil-, aber auch die Gesamtfunktion der Rückfallebene abgesichert werden? Welche Abhängigkeiten herrschen zwischen den Teilfunktionen?

Für die beschriebenen Funktionen der Rückfallebene sind Erweiterungen denkbar. Möglich ist beispielsweise, dass die Schnelligkeit der Geschwindigkeitsanpassung des Fahrzeugs nach Aktivierung der Rückfallebene variabel gestaltet wird und beispielsweise abhängig vom rückwärtigen Verkehr (geringere Verzögerung bei kleiner Zeitlücke zum Folgefahrzeug) und der Art des Transports (bspw. geringere Verzögerung bei stehenden Fahrzeuginsassen) sein kann.

Mit der vorgestellten Systemarchitektur ist es möglich, eine zusätzliche Absicherung der von der Planungsebene berechneten Solltrajektorie durchzuführen. Durch die Einführung der vorgestellten Rückfallebene steht eine zweite, unabhängige Umfelderfassung zur Verfügung. Diese kann genutzt werden, um die Solltrajektorie vor der Ausführung von einer zweiten Instanz überprüfen zu lassen und bei erfassten Hindernissen eine entsprechende Reaktion auszulösen. Möglich wäre außerdem die Aufdeckung von Degradationen der Umfelderfassungssysteme durch widersprüchliche Umfeldmodelle.

Die Absicherung mithilfe der zweiten Umfelderfassung soll auch auf extern geplante Trajektorien oder Bahnen ausgeweitet werden. Im Projekt UNICARagil ist eine Teleoperation geplant, mit der ein menschlicher Operator das Fahrzeug durch eine direkte Steuerung oder durch Vorgabe einer Trajektorie bewegen kann. Auch diese Vorgaben können mithilfe der Umfeldüberwachung der Rückfallebene abgesichert werden.

Im weiteren Verlauf des Projekts wird die vorgestellte Rückfallebene implementiert und die in diesem Kapitel genannten Forschungsfragen bearbeitet.

## Acknowledgement

Diese Forschungsarbeiten wurden im Rahmen des Projekts „UNICARagil“ durchgeführt (FKZ 16EMO0286). Wir bedanken uns für die finanzielle Unterstützung des Projekts durch das Bundesministerium für Bildung und Forschung (BMBF).

## Literatur

- [1] SAE International, *SAE J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, 2018.
- [2] C. Arzt und S. Ruth-Schumacher, *Überführen hoch- oder vollautomatisierter Fahrzeuge in den "risikominimalen Zustand"* in *Recht Automobil Wirtschaft - RAW*, Heft 2/2017. S. 89 - 98, 2017.

- [3] UNECE, *Harmonization of Vehicle Regulations* [Online] Verfügbar unter: [https://www.unece.org/trans/main/wp29/meeting\\_docs.wp29.html](https://www.unece.org/trans/main/wp29/meeting_docs.wp29.html). Zugriff am: Aug. 21 2019.
- [4] UNECE, *Automatically Commanded Steering Function (ACSF) - Transport - Vehicle Regulations* [Online] Verfügbar unter: <https://wiki.unece.org/pages/viewpage.action?pageId=25265606>. Zugriff am: Aug. 21 2019.
- [5] T. Woopen, B. Lampe, T. Böddeker, L. Eckstein, A. Kampmann, B. Alrifaei, S. Kowalewski, D. Moormann, T. Stolte, I. Jatzkowski, M. Maurer, M. Möstl, R. Ernst, S. Ackermann, C. Amersbach, S. Leinen, H. Winner, D. Püllen, S. Katzenbeisser, M. Becker, C. Stiller, K. Furmans, K. Bengler, F. Diermeyer, M. Lienkamp, D. Keilhoff, H.-C. Reuss, M. Buchholz, K. Dietmayer, H. Lategahn, N. Siepenkötter, M. Elbs, E. von Hinüber, M. Dupuis und C. Hecker, *UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts* in *27th Aachen Colloquium*, Aachen, Germany, S. 663 - 694, 2018.
- [6] T. Stolte, G. Bagschik, A. Reschka, M. Maurer, *Automatisch fahrerlos fahrendes Absicherungsfahrzeug für Arbeitsstellen auf Autobahnen (aFAS)* in *AAET 2015 - Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel*, S. 371 - 390, 2015.
- [7] A. Reschka, *Fertigkeiten- und Fähigkeitengraphen als Grundlage des sicheren Betriebs von automatisierten Fahrzeugen im öffentlichen Straßenverkehr in städtischer Umgebung*, 2017.
- [8] D. Keilhoff, D. Niedballa, H.-C. Reuss, M. Buchholz, F. Gies, K. Dietmayer, M. Lauer, C. Stiller; S. Ackermann, H. Winner; A. Kampmann, B. Alrifaei, S. Kowalewski; F. Klein, M. Struth, T. Woopen, L. Eckstein *UNICARagil - New Architectures for Disruptive Vehicle Concepts* in *19th Stuttgart International Symposium - Automotive and Engine Technology*, Stuttgart, Germany, 2019
- [9] S. Ackermann, H. Winner, M. Buchholz *Modul und Verfahren zur Absicherung von Solltrajektorien für automatisiertes Fahren*, Deutsche Patentanmeldung, Patent Anmeldenummer: 10 2019 125 401.9, 2019
- [10] EUREF EUROPEAN TERRESTRIAL REFERENCE SYSTEM 89 (ETRS89) [Online] Verfügbar unter: <http://etrs89.ensg.ign.fr/>. Zugriff am: Dez. 15 2019



---

# **Uni-DAS**

---

## **Herausgeber**

**Lutz Eckstein**

**Klaus Bengler**

**Klaus Dietmayer**

**Markus Maurer**

**Christoph Stiller**

**Hermann Winner**