Virtual Stochastic Testing of Advanced Driver Assistance Systems

Stephanie Prialé Olivares, Nikolaus Rebernik, Arno Eichberger and Ernst Stadlober

Abstract With Advanced Driver Assistance Systems becoming increasingly complex, testing methods must keep up to efficiently test and validate these systems. This paper focuses on a method of testing vision-based Advanced Driver Assistance Systems on a state-of-the-art hardware-in-the-loop test bench. Virtual driving scenarios are being used for functional testing. This paper suggests a framework where the driving scenarios are constructed using a stochastical approach. This allows the testing of the parameter combinations that might otherwise be forgotten or disregarded by a human creating the scenarios. The first step of this framework, a road generator, is introduced. Generic courses of roads are created using the Markov Chain and Markov Chain Monte Carlo methods reconstructing real-life scenarios by analyzing map data.

Keywords Markov Chain · Markov Chain Monte Carlo · Stochastic road generator · Virtual driving scenario generator

S. Prialé Olivares (✉)
BMW Group and Institute for Real-Time Computer Systems, Technical University of Munich, Grillparzerstraße 25, 81675 Munich, Germany
e-mail: stephanie.priale@gmail.com

N. Rebernik
BMW Group and Institute of Automotive Engineering, Graz University of Technology, Luisenstraße 45, 80333 Munich, Germany
e-mail: nikolaus.rebernik@gmail.com

A. Eichberger
Institute of Automotive Engineering, Graz University of Technology, Inffeldgasse 11/II, 8010 Graz, Austria
e-mail: arno.eichberger@tugraz.at

E. Stadlober
Institute of Statistics, Graz University of Technology, Kopernikusgasse 24/III, 8010 Graz, Austria
e-mail: e.stadlober@tugraz.at

T. Schulze et al. (eds.), *Advanced Microsystems for Automotive Applications 2015*, Lecture Notes in Mobility, DOI 10.1007/978-3-319-20855-8_3

1 Introduction

As Advanced Driver Assistance Systems (ADAS) are becoming more and more complex, testing and validation processes and methods must keep up. The automation of driving functions is key for future mobility, facilitating greener cars with reduced fuel consumption, as well as improved road safety, by the minimization of hazardous situations in assisting the driver to react more quickly. However, automation of safety critical driving functions requires a minimum risk of failures, as described by norm ISO 26262. According to literature, a valid validation of automated driving functions in compliance with ISO 26262 demands a very large amount of testing kilometers [1], which will make the system expensive and hinder market penetration.

As a step towards a more efficient validation process, this paper looks at a new approach for subsystem integration using a state-of-the-art Hardware-In-the-Loop (HIL) test bench with a focus on vision-based ADAS demonstrated by the example of a lane detection system. HIL testing is a vital step in subsystem integration. It offers a test platform in early development stages and is cost-efficient, since the development can be started earlier. It also helps reduce the number of prototype vehicles needed.

In HIL environments, a variety of approaches are being employed to test and validate ADAS today: A fixed set of scenarios is being used and recorded scenes are being reprocessed or transferred to virtual environments: for example, using recorded sensor data to automatically transfer and create a scenario [2]. Augmented reality, as proposed in [3], is yet another approach that is being used as a Vehicle-In-The-Loop (VEHIL) setup. These approaches are either labor intensive or cannot adequately cope with the high complexity of future systems. Until now, the designing principle of intervening systems is not to try to cover all possible scenarios where a system could help but, rather, to focus on situations where a system will prove beneficial [4]. This, however, will change with increasing automation in driving, where vehicles will have to cope with many situations without the help of their driver. When looking into the future towards autonomously driving vehicles, a fixed catalogue will have to include thousands of tests to cover a sheer infinite number of scenarios. Creating them by hand will take much time and labor. It is said to take 50×10^8 km [1] of road testing to reliably validate autonomous driving functions. With this high amount of required kilometers, more efficient methods of validation must be implemented along the entire validation process throughout the development stages.

The goal is to automatically create virtual driving scenarios for testing image processing functions of vision based ADAS with an optimized parameter distribution to focus on more critical parameter ranges. One approach is to use combinatorics for the creation of test scenarios, as discussed in [5]. Scenarios are created by calculating all possible combinations of parameters and using equivalency class formation to reduce the number of scenarios to be tested.

This paper, however, proposes a stochastical approach. Since HIL testing is usually done in real time, it is not economically reasonable to test millions of possible

outcomes and scenarios. Thus, a selection process has to be implemented. The parameters' influence on ADAS is analyzed using a design of experiment (DOE) approach. By doing this, the influence of parameters and their ranges on the correct functionality of the image processing algorithms of ADAS are evaluated. The likelihood of occurrence of parameter values for generating virtual driving scenarios will be higher in ranges where faults are more likely to occur, thereby focusing on critical areas while still testing the entire parameter ranges. As will the described in Chap. 2, real world statistical data to form distribution functions of road characteristics is obtained by analyzing map data as a basis for road generation. These functions describing road characteristics are then influenced towards the obtained critical ranges. This will be done in a future step, as described in the conclusion. Virtual environments have the advantage of knowing their properties. In that sense, fundamental information is known to which measurements of the ADAS can be compared. To evaluate the large number of generated scenarios, an automatic evaluation process must also be implemented. By testing a significant amount, depending on the ADAS function to be tested, of scenarios a qualitative statement can be made about the correct functionality of an ADAS function. Thereby a fast overview about an ADAS system can be obtained during the development stages. The method will be tested on vision-based ADAS; more specifically, on a lane detection system as a base for expansion towards more complex functions like lane keeping and congestion assistance systems, which, again, are a base for automated driving functions.

To reach this goal, the presented research will implement and test a number of steps. The first step is a road generator, which is the focus of this paper. It is the basis for creating driving scenarios and to establish and test this framework. This road generator will be explained in more detail in the next chapter.

2 System Model

In this section, the system model in charge of generating stochastic scenarios is described. Stochastic scenarios are the representation of the road's static elements, traffic, and surrounding elements, which are modeled according to probabilities computed by statistical analysis using different types of statistical methods. These scenarios seek a good approximation to scenarios that attempt to reflect reality as precisely as possible so that the ADAS can be properly tested.

Since close-to-reality scenarios are being pursued, the stochastic scenarios are based on statistical analysis on routes which actually exist. Essential characteristics such as geometry, type, and number of lanes, among others, are deduced from routes selected in OpenStreetMap. The system model takes this information, develops a database, and produces a probability density function (pdf) or conditional probabilities, depending on the attribute, for each aspect that needs to be evaluated to generate a scenario. Pdfs and conditional probabilities are a part of the statistical methods, thereby creating a statistical model in charge of generating roads with specific attributes.

A road is designed according to probabilities computed by the statistical analysis using different types of statistical methods. The following two methods are used for the evaluation and generation of scenarios: the Metropolis algorithm and the Markov Chain.[1]

These algorithms are directly influenced by the route chosen in OpenStreetMap. They are also indirectly influenced by constraints added to the system inside these algorithms to ensure its right functionality. Each time a new route is selected as base for the statistical analysis in OpenStreetMap, it results in a new pdf. This models the probability of attributes being sampled in the statistical methods, in turn influencing the resulting scenarios. Meanwhile, the constraints can also impact the road features depending on how they are set.

2.1 Building Statistical Analysis

The building of a statistical analysis, or, more precisely, the creation of a Bayesian statistical analysis, usually requires some fundamental analysis steps [6]. However, some posterior distributions are too complex to be calculated analytically and they are too time-consuming to sample. Using Markov Chain Monte Carlo (MCMC) algorithms, such as the Metropolis Hastings algorithm, can be very helpful for tackling both of these problems.

The algorithms used for the statistical analysis of the scenario's attributes are chosen according to the degree of complexity by the calculation of the posterior distribution and its high dimensional sampling. Two methods have been selected. One of them is the Metropolis algorithm (the special case of the Metropolis Hastings algorithm for symmetrical distributions) and the other is the Markov Chain. The Metropolis algorithm is in charge of creating the road geometry. This algorithm generates a random walk of points distributed according to a target distribution [7]. In this case, the points are being distributed according to the pdf that is a result of the analysis of the curvatures of the selected OpenStreetMap road. Since the resulting distribution for the geometry factor requires high dimensional sampling and can be too time-consuming with conventional Monte Carlo algorithms or Markov Chains, a MCMC algorithm is selected. Further points regarding this decision on the statistical methods are explained in 2.2.

On the other hand, a Markov Chain is required for those attributes where the Markov property also has to be fulfilled, but where sampling is not that time-consuming (e.g. setting the number of lanes on the road) and where algorithms for high dimensional distributions are not required. There are also attributes that possess constant values or are user-defined-data and, therefore, no algorithm is needed for their analysis.

[1] "Markov Chain" is used as reference to indicate a Markov process with a finite number of states.

2.1.1 Metropolis Algorithm Construction Steps

This algorithm works according to the structure presented in Fig. 1. The algorithm works by choosing an initial position x(0) [Step 1], then a proposed move x* is generated from the proposal distribution [Step 2]. This move will be either accepted or rejected according to an acceptance criterion $A(x^{(i)}, x^*)$, where x(i) represents the last accepted move and x* represents the proposed move. After performing a number of steps, the Metropolis Hastings algorithm generates a number of points, which construct the Markov Chain. These points will be distributed according to the desired distribution. The proposal distribution in the algorithm impacts the acceptance criterion. Therefore, it should be chosen in such a manner that it covers the target distribution completely. However, it is also important that it converges quickly and effectively. For continuous components, the Gaussian distribution or heavier-tailed distributions, e.g. Student's t distributions with low degrees of freedom, are commonly used [8]. Both of them are symmetric, which makes the acceptance criterion easy to overcome and produces a quick convergence. Another requirement of the Metropolis algorithm is to specify the target distribution.

The target distribution is fixed and calculated according to the data collection of the feature being analyzed. The calculation in this model is done in 3 steps:

1. Calculation of histogram based on data exported from OpenStreetMap.
2. Estimation of the pdf using kernel density estimation.
3. Specification of the (target) probability distribution according to the calculated pdf.

Once all requirements are provided, the algorithm is run and the samples related to the curvature of the road are generated. Unlike the features calculated with Markov Chains, these samples are not taken directly as attributes for the objects implemented in the system, but they have to be recalculated so they can be part of the objects belonging originally to the geometry of the road. This is required, since the road geometry is defined by spiral, arc, and line elements. These attributes are created according to the curvature sampled by the algorithm.

Figure 2 is an example of the curvature pdf for the highways around the city of Munich, calculated via kernel estimation.

The target distribution results from taking the resulting function of the kernel density estimation and setting the input values of the function as undefined. These values will be set once the algorithm is running. These undefined values take the

Fig. 1 Metropolis algorithm

1. Initialise $x^{(0)}$
2. For i = 0 to N-1
   - Sample $u \sim U_{[0,1]}$
   - Sample $x^* \sim q(x^* | x^{(i)})$
   - If $u < A(x^{(i)} | x^*) = \min\{1, \frac{p(x^*)}{p(x^{(i)})}\}$

     $x^{(i+1)} = x^*$

     else

     $x^{(i+1)} = x^{(i)}$

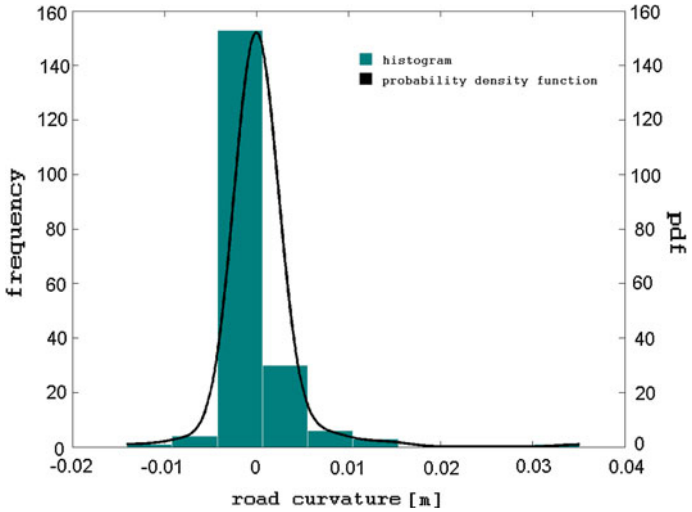

Fig. 2 Pdf estimation based on the highways around the city of Munich

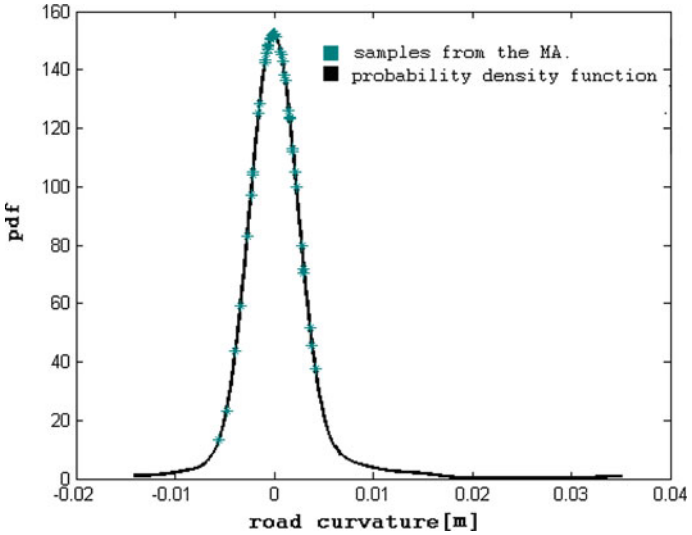

Fig. 3 Metropolis algorithm example for the highways around the city of Munich

sample values of the Markov Chain, which are built by the Metropolis algorithm (MA). Figure 3 shows an MA example for the highway around the city of Munich. This figure represents "x" values being sampled from the resulting pdf of the kernel estimation according to the Metropolis algorithm being associated with its corresponding pdf(x) value.

2.1.2 Markov Chain Construction Steps

Each Markov Chain is based on states that correspond to the possible values that the attribute being analyzed can take. Every time the Markov Chain is run, transitions between states occur according to probabilities given by the transition matrix P. These transition matrices are created according to the conditional probabilities of the attribute being analyzed, where each ijth entry represents a particular conditional probability: p (*moving to state* $j|$ *in state* i).

These conditional probabilities are calculated from the database of road features specified in OpenStreetMap. After setting the transition matrices, the stochastic row vector can be calculated through definition (1), where P represents the transition matrix and $x^{(n)}$ the stochastic row vector at step time n [6], and samples can be drawn. These resulting samples represent the attributes of the stochastic scenario, where the Markov property is fulfilled but where the sampling does not consume a great amount of time.

$$x^{(n+1)} = x^{(n)}\mathrm{P} \tag{1}$$

2.2 *Statistical Methods*

The objective of statistical methods is to make the process of scenario generation as efficient and productive as possible; hence, a proper selection of statistical methods is necessary. The most important part of choosing the correct test is to ensure that the test is appropriate for the type of data that has been collected [9], to observe adequate distributions of the variables, and to reflect what kind of relationship exists between them. The data collected for the system model is made up of the features taken from the route(s) chosen in OpenStreetMap, which are, in turn, the attributes required for the scenario generation. Some of these feature variables are correlated, meaning that the probability of a next event can increase or decrease based on the current event. These variables can be seen in the form of states and can be described with the help of Markov Chains. This is because a Markov Chain is generated based only on the previous state making new states likely to be correlated with the preceding state.

Depending on the variable being analyzed, a Markov Chain can be easily constructed and sampled. This happens by calculating the conditional probabilities from the database of road features specified in OpenStreetMap and setting the transition matrices. For the construction of a suitable Markov Chain, every possible state should be specified. Therefore, a Markov Chain is useful if the number of states is countable.

However, features for the analysis of the geometry construction of the road exist where the probability distribution is difficult to calculate and to simulate due to the high number of parameters. This can be solved by using Markov Chain Monte Carlo (MCMC) algorithms.

There are different kinds of MCMC algorithms. Among them, the most commonly used are the Metropolis Hastings algorithm and the Gibbs sampler. In the system model, the Metropolis Hastings algorithm is used instead of the Gibbs sampler, since, with the Metropolis Hastings algorithm, it is required to know the full conditional distribution up to a constant [10], while, with the Gibbs sampler, it is necessary to provide the full conditional distributions of the model in closed form [11]. This has to take place; otherwise, it is not possible to use the Gibbs sampler. Hence, the Metropolis Hastings algorithm becomes a more suitable method for the model. In the system model, the proposal distributions are symmetric; therefore, the simpler Metropolis algorithm is used instead.

2.3 Constraints

For both statistical methods (the Markov Chain and the Metropolis algorithm), constraints have to be set in order to ensure the right function of the system model. Conditional requirements vary depending on the attribute being analyzed.

These requirements are included in the algorithm in charge of creating the attribute during the generation process of stochastic scenarios such as, for example, to avoid a transition from a two lane-road to a four-lane road to be part of the stochastic scenarios. This constraint is implemented as part of the Markov Chain of the driving lanes' attributes in order to disable this possibility.

3 Performance Study

Different types of scenarios are constructed by varying the chosen route in OpenStreetMap. To inspect the performance of the system model even closer, the number of samples in the algorithms and repercussions of constraints in the generation of stochastic scenarios are evaluated.

3.1 OpenStreetMap Route Selection

When a new route in OpenStreetMap is selected for evaluation in the system model, the inputs for the algorithms change; in other words, the *transition matrix* required for the construction of the Markov Chain and the *pdf* taken as target distribution for the Metropolis algorithm change. For example, if a route is selected, where rather few curves but more straight stretches of road are presented, a *pdf* similar to Fig. 2 is expected to occur. On such a route, the highest probability resides in the curvature with value 0, and the algorithm tends to sample towards this value. In the case of routes with a higher probability of curvature, resulting roads with strong curvatures have a higher probability to appear.

The *transition matrix* necessary for the calculation of the Markov Chain changes when new routes are selected. However, since some of the non- road's geometry features, such as road markings, are based on fixed road specifications, their probabilities do not vary among different selected routes.

3.2 Constraints

Constraints used in the algorithms for the correct functionality can affect the road's features in the stochastic scenario depending on how well they are tuned. In the case of the geometry, a maximum or minimum duration of a curvature can be established in order to generate stochastic scenarios that fulfill specific characteristics expected by the user.

For features such as speed limit, constraints can be set so that jumping between values is not possible but a linear increment is given. This is also a feature set according to user desire.

Figure 4 shows how varying the constraint set for the geometry attribute can influence the smoothness of the curve despite maintaining the same probability density. The three resulting roads represent different curvature constraints where the spiral in charge of creating the curvature uses less iteration steps on each figure from left to right. By using a lesser number of iterations to diminish the curvature, the curvature ends up having a higher tangent and a stronger curve.

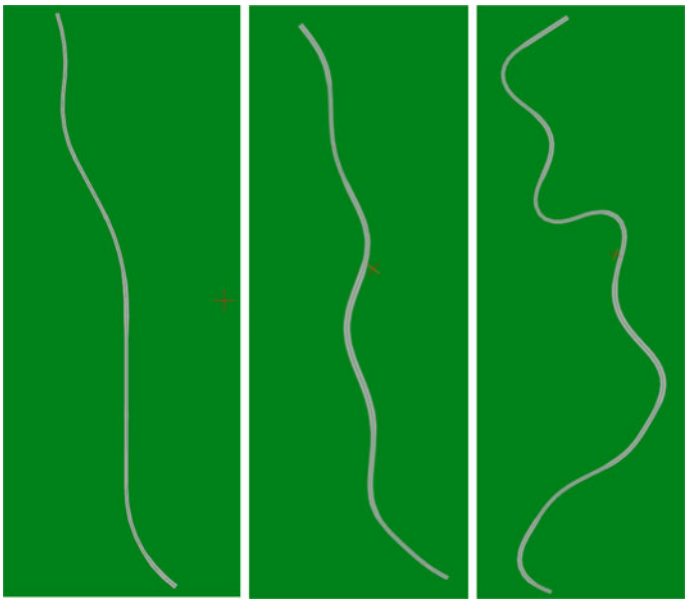

Fig. 4 Output roads with different curvature constraints

3.3 *Number of Samples in Statistical Methods*

The number of samples or steps the algorithms need to create each road attribute has an influence on the resulting roads. If the number of samples for the algorithms increases, the probability of sampling a particular value, which is normally low, increases as well. Furthermore, in case a constraint determines that an attribute needs to remain constant for a defined amount of iterations, these already low-probability values can reach an even lower probability if the number of iterations decreases, and vice versa.

4 Conclusion and Outlook

Using a stochastic approach by employing Markov Chain and Markov Chain Monte Carlo methods, and using real world data to develop the corresponding probability density functions for road-generating algorithms, generic but still realistic roads can be created. The real world data is being obtained by analyzing roads exported from OpenStreetMap. The discussed road generator is the basis for generating virtual driving scenarios and for the suggested framework.

The next steps will include a design of experiment approach to optimize the probability density functions towards critical parameter ranges. With this approach the main influencing parameters on the image processing functions of the ADAS, concerning the road creation, will be evaluated. The critical parameters as well as their critical value ranges, which cause the most failures of the vision based ADAS, are of interest. The pdf's of the road generator will be influenced towards these more critical ranges. Another step is the expansion of the road generator to include different road markings, roadside structures as well as construction zones. Construction zones are challenging for lane detection functions as there are typically many different markings and structures on the road that make detecting the correct driving lane more difficult.

This framework contributes a further approach to tackle the challenges of validating complex ADAS with an outlook on automated driving functions.

References

1. Winner H (2013) Absicherung automatischen Fahrens, 6. FAS-Tagung München, Munich
2. Lages U, Spencer M, Katz R (2013) Automatic scenario generation based on laserscanner reference data and advanced offline processing. In: Intelligent vehicles symposium workshops (IV workshops), pp 146, 148
3. Zofka MR, Kohlhaas R, Schamm T, Zöllner JM (2014) Semivirtual simulations for the evaluation of vision-based ADAS. In: Intelligent vehicles symposium proceedings, IEEE, pp 121, 126

4. Schwarz J (n.d.) Response 3—code of practice for development, validation and market introduction of ADAS—A PReVENT project. DaimlerChrysler AG, Stuttgart
5. Schuldt F, Sausts F, Lichte B, Maurer M (2013) Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen. In: AAET2013—Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel, Braunschweig
6. Gamerman D, Lopes H (2006) Markov Chain Monte Carlo: stochastic simulation for bayesian inference. Taylor & Francis Group, Boca Raton USA
7. Müller P (2009) Monte Carlo methods and bayesian computation: MCMC, vol 10
8. Neal R (1993) Probabilistic inference using Markov Chain Monte Carlo methods, U. Toronto
9. Vowler S (2007) Analysing data—choosing appropriate statistical methods. Hosp Pharmacist 44:12
10. Mengersen KL, Tweedie RL (1996) Rates of convergence of the hastings and metropolis algorithms. Ann Stat 24:101–121
11. Gilks W, Richardson S, Spiegelhalter D (1996) Markov Chain Monte Carlo in practice. Chapman & Hall/CRC, London