Logical Scenario Derivation by Clustering Dynamic-Length-Segments Extracted from Real-World-Driving-Data

Jacob Langner[1], Hannes Grolig[1], Stefan Otten[1], Marc Holzäpfel[2] and Eric Sax[1]
[1] *FZI Research Center for Information Technology, 76131 Karlsruhe, Germany*
[2] *Dr. Ing. h.c. F. Porsche AG, 71287 Weissach, Germany*

Keywords: Scenario Extraction, Logical Scenarios, Vehicle Environment, Real-World-Driving-Data.

Abstract: For the development of Advanced Driver Assistant Systems (ADAS) and Automated Driving Systems (ADS) a change from test case-based testing towards scenario-based testing can be observed. Based on current approaches to define scenarios and their inherent problems, we identify the need to extract scenarios including the static environment from recorded real-world-driving-data. We present an approach, that solves the problem to extract dynamic-length-segments containing a single scenario. These segments are enriched with a feature vector with information relevant for the feature under test. By clustering these scenarios a logical scenario catalog is created, containing all scenarios within the test data. Corner cases are represented as well as common scenarios. An accumulated total length can be calculated for each logical scenario, giving a brief understanding about existing test coverage of the scenario.

1 INTRODUCTION AND RELATED WORK

Original Equipment Manufacturer (OEM), research institutes and start up companies are currently putting a lot of effort into realizing autonomous driving. Many have succeeded in first trials, however, the first autonomous vehicle is yet to be released. Current ADAS slowly develop into conditionally automated, SAE level 3 features, that can take over a driving task completely in explicitly restricted driving environments, e.g. a traffic jam assist on highways up to 60 km/h. Such features are not yet globally available, because OEM lack the proof of safety needed for clearance from the National Motor Vehicle Registration Authorities. Different standards and regulations require a separate approval in each country.

According to the ISO26262 (ISO, 2011) safety goals are set via a risk analysis. To achieve the highest possible safety and security on public roads, the fulfillment of these safety goals has to be proven. The proof can be brought forward analytically or with the help of test cases (Henzel et al., 2017). For system-level testing of ADAS and ADS analytic proof is not feasible due to the complexity and extent of the software code and input parameter space. Therefore, common practice is to use test cases.

In order to achieve sufficient testing, the test cases need to cover all relevant situations for the Feature Under Test (FUT). Test cases are usually derived from the feature requirements manually by domain and test experts. Each functional requirement is covered by at least one test case. During system level testing all test cases are evaluated and a final test result concludes the fulfillment of each requirement. However, determining all relevant situations and declaring all feature requirements, from which the complete set of corresponding test cases can be derived, is not a trivial task.

With the technological advances in environmental perception, car-to-x communication as well as automated driving, the complexity of new ADAS and ADS is steadily increasing. Due to new sensors being integrated in today's vehicles, sensor ranges being increased and background services, e.g. digital map providers, being utilized, the environmental information available in the vehicle has grown significantly. While these technological advances certainly enhance the quality of these features and the possibilities towards more autonomous driving, they also introduce a manifold of new situations, a vehicle can encounter, and thereby increase the required testing efforts. The definition of testable, ISO-conform requirements - and, thereby, the test case-based approach - is no longer feasible for the next generation of ADAS and ADS.

Langner, J., Grolig, H., Otten, S., Holzäpfel, M. and Sax, E.
Logical Scenario Derivation by Clustering Dynamic-Length-Segments Extracted from Real-World-Driving-Data.
DOI: 10.5220/0007723304580467
In *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2019)*, pages 458-467
ISBN: 978-989-758-374-2

For ADAS and ADS development it is best practice to additionally test the feature on public roads in order to experience as many different situations as possible. Huge efforts are made to drive as many kilometers as possible during test campaigns and release approvals, e.g. 36 million test kilometers for the release of the Mercedes Benz E-Class (W212) in 2009 (Deppe, 2009) and 12 million test kilometers for the next generation E-Class (W213) in 2015 (Christiansen, 2015).

Since there is no checklist of situations to be seen during testing, the required test kilometers are derived from statistical calculations. Due to the combination of different environmental situations and traffic participant's maneuvers, the sheer amount of kilometers needed to achieve a statistically valid assertion sums up to a vast number. (Wachenfeld and Winner, 2016) have calculated a theoretical 6.62 billion test kilometers to prove, that an interstate pilot meets the current standards for release approval. The number is derived from fatal accidents drawn from annually crash statistics in Germany and the total sum of driven highway kilometers in Germany per year - assuming that the absence of accidents is the approval criterion for highly automated driving. (Amersbach and Winner, 2017).

Simulation enables the selective testing of for example critical situations, possibly reducing the required test kilometers to a fraction of the statistically calculated 6.62 billion kilometers in case of the interstate pilot. Thereby, simulation is an important tool for OEM to even come close to the required test coverage. However, the validity of the simulation has to be ensured by providing detailed models of the vehicle, environment, sensors and traffic. Efforts for providing these models are limiting the theoretical scalability of the simulation approaches.

Figure 1: The 5-level structure for scenario description from the PEGASUS[1] project.

It has been shown, that for certain ADAS recorded driving data from real world tests can be utilized to substitute most of the models needed for the simulation, for an example see (Zofka et al., 2015). Open-loop features can easily be tested using this data. In some cases, e.g. predictive features, it even entails the benefit of the ground truth in form of the recorded events (Langner et al., 2017). Even closed-loop longitudinal control features have been simulated using the recorded data from test campaigns (Bach et al., 2017) (de Gelder and Paardekooper, 2017).

The real-world-driving-data provides a large amount of realistic driving situations including many critical ones, which offer better test coverage than synthetic data and scale better than manually defined test cases. However, the value of the real-world-driving-data tests is diminished by redundancies. Simply simulating all of the recorded data offers the same control over the tested situations as real world test drives.

The calculations made by Wachenfeld and Winner are based on a poisson probability distribution of crashes from German traffic data. The statistical calculations hold true as long as there is no semantic interpretation and selection of the test drives. The contribution towards the proof of safety has to be measured by the amount of different situations tested, not by the mere amount of kilometers simulated (Junietz et al., 2017). An abstract description and the possibility to measure the coverage of these situations is needed.

Recent research focuses on scenario-based approaches, where different situations a feature can encounter are described on a higher abstraction level. A scenario contains the static environment, the dynamic environment including other traffic participants and their actions as well as the actions of the ego vehicle for a certain period of time (Elrofai et al., 2016). (Schuldt et al., 2013) have proposed a 4-level scenario structure for the description of scenarios: the basic road network, situational modifications, traffic situation and environment. In the PEGASUS project the basic road network is split into street level and traffic infrastructure, resulting in a 5-level scenario structure (see Figure 1). (Bagschik et al., 2017) within the PEGASUS project have suggested three different abstraction layers for scenarios to offer the needed flexibility in each development phase (see Figure 2).

Concrete scenarios are individual instances of logical scenarios, where the complete static and dynamic environment as well as the state of the ego vehicle are described in detail. They offer little flexibil-

[1]https://www.pegasusprojekt.de/files/tmpl/linebreak PDF-Symposium/04_Scenario-Description.pdf

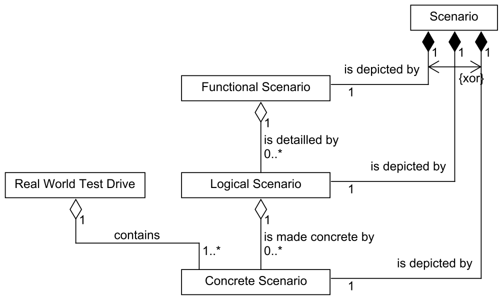


Figure 2: Scenario abstraction layers (Steimle et al., 2018).

ity and their abstraction level is quite similar to test cases. By abstracting from the explicit signal values of these concrete scenarios some generalization can be achieved. Logical scenarios describe a group of concrete scenarios by parameter ranges for each signal, whereas functional scenarios textually describe logical scenarios based on a functional view. This allows for a high-level description of different scenarios early on, which can be specified in later stages of the development process. Test cases can be derived from concrete scenarios, which in turn can be derived from logical scenarios. Therefore, test coverage can be measured on the logical scenario level.

Still, all possible scenarios a feature can undergo must be known and explicitly specified to achieve complete test coverage. There is a lot of ongoing research in the field of scenario definition and standardization. However, a common scenario catalog representing all scenarios, that can occur while driving in the real world, is not yet available. Up until now, attempts to create such a common scenario catalog are limited to certain features and environmental conditions, for example a traffic jam assist with scenarios that are altogether on highways (Pütz et al., 2016). Furthermore, it is to be debated, if a universal scenario catalog exists, since relevant scenarios strongly depend on the FUT, e.g. urban test drives do not matter for testing of a highway pilot. It can be argued, that there exists a universal scenario catalog for level 5 autonomous driving and that current level 2 and level 3 features can derive relevant subsets from this catalog. However, the argument leads nowhere, since this catalog not yet exists.

Instead of explicitly defining all scenarios, they can also be extracted from recorded real-world-driving-data (see Figure 2). Several approaches to extract driving maneuvers of the ego vehicle or other traffic participants have been published (Gerdes, 2006), (Hülnhagen et al., 2010). In most cases, basic maneuvers are explicitly defined and mapped onto physical parameters and parameter ranges or parameter curves. These are then used as a pattern to detect further instances of the same or similar maneuver. (Elrofai et al., 2016) enhance this approach with data analytics to improve the performance of the detectors. Another approach was published by (Minnerup et al., 2015), who use triggers for scenario extraction from real-world-driving-data. However, only explicitly specified and searchable scenarios like critical distance to vehicle in front or amount of steering per second can be found. (Junietz et al., 2017) derive an abstract criticality metric to identify critical scenarios based on relative metrics between the ego vehicle and other dynamic objects. Calibration is done with manually classified scenarios and the metric only derives a pre-selection for manual filtering. There are no approaches known to the authors, that focus on extracting scenarios including the static environment of the vehicle.

All these approaches are motivated by ADAS like an Adaptive Cruise Control (ACC) or Emergency Brake Assist (EBA) and are focused on extracting critical situations for the release approval. Most of the critical situations for these features occur, when interacting with other traffic participants. Thereby, according to (Schuldt et al., 2013) the relevant scenarios are also mainly based on the dynamic objects. But, this does not hold true for predictive features, whose predictions mainly rely on the sensed environment as well as upfront information about the static environment from digital maps. The maneuver extraction approaches, therefore, do not yield a viable scenario catalog for these features.

Furthermore, testing critical situations alone is not sufficient either. Especially features taking over longitudinal or lateral control also have a functional quality aspect besides the functional safety - the driver or passenger comfort. Merely avoiding accidents is not sufficient for the passenger acceptance of the ADS. Comfort ratings will become increasingly important with further advances towards autonomous driving - and comfort has to be evaluated in all situations, not only the critical ones.

In this paper we present a data-driven approach to segment the real-world-driving-data into scenarios tailored to the FUT. In contrast to the maneuver extraction approaches, we focus on the static environment - including the street level, traffic infrastructure and environment conditions - and its correct segmentation. The scenarios are automatically extracted based on selected signals relevant for the FUT. The approach combines the description of the static environment and the dynamic objects and derives a feature-based scenario catalog from the driving data. By using a dynamic clustering approach we determine a flexible amount of logical scenarios, which contain all found concrete scenarios within the driving data. The aggregation of discrete signal values leads to logical scenarios, which offer a more comprehensible scenario catalog for system-level testing. Further interpretation of the logical scenarios can result in functional scenarios for the FUT.

2 SCENARIO-EXTRACTION CONCEPT

Real-world-driving-data consists of many driving situations, which uninterpreted and on their own offer little information towards the coverage of all possible situations. However, extracting all these driving situations as self-contained concrete scenarios and aggregating them to logical scenarios does. The problem is to find and extract these scenarios within the test drives.

Our goal is to automatically extract all existing scenarios relevant for the FUT implicitly from the data in order to create the feature's logical scenario catalog. An example of two logical scenarios is illustrated in Figure 3. The 5-level scenario structure from the PEGASUS project is used for the scenario description. Figure 4 reveals the necessary steps: The first step is to select the relevant signals for the FUT and pre-process these signals for further usage. The scenario dividing signals are then used to slice the test drives into segments with dynamic lengths. Then, a feature vector describing the static and dynamic environment for each segment is extracted from the driving data. The feature vectors are clustered and each cluster is interpreted as a logical scenario. The feature vector is also used as a scenario description.

2.1 Data Selection and Pre-Processing

As already stated - for system level testing - the data selection has to be done with regard to the FUT. For ADAS and ADS the amount of possible different situations, accrued from the combination of the static and dynamic vehicle environment, is a critical part for testing. Therefore, signals describing the relevant vehicle environment should be used. Not all information, however, can be derived from a single signal. Some have to be transformed or gathered from additional information sources. The temperature, wetness or traffic density for example can be extracted from weather or traffic data providers, if they are relevant for the FUT. More abstract values like the time to collision (TTC) can be calculated as well.

When working with recorded real-world-driving-data, signal pre-processing is required. The data is noisy and for statistical uses, some assumptions have to be made. Error and initial values have to be handled. Especially the GPS signal and the velocity have to be considered carefully as they are the foundation for further signal processing. Noise in these signals propagates throughout the analysis.

Real-world-driving-data is commonly recorded time-based. This time-dependency leads to a distortion in the representation of the static environment due to differing velocities during the test drives. In order to compare the test drives, the signals have to be converted to a track position-based sampling. The sampling should be equidistant and consistent cross-samples and with a reasonable distance between two data points.

Further pre-processing includes derivation of new signals or reshaping of existing data.

2.2 Scenario Extraction

For predictive features, which rely heavily on the sensed static environment, it is necessary to identify all environmental situations, that can be experienced, and to validate the correct behavior in these situations. Contrary to the maneuvers, which have clear start and end points, the challenge for the static environment is to detect the change points for seemingly smooth transitions between different environmental situations.

Common machine learning approaches, that need to partition real-world-driving-data, use either static segment lengths or time intervals (Langner et al., 2018) or up- or downsample the sequences to a defined length (Wang et al., 2018) in order to normalize and partition the test drives, which than can be processed further. By using a static segment length, single scenarios within the test drives might be separated into two subsequent segments (see Figure 5). A single segment might also contain more than one scenario. With segments containing more than one scenario and scenarios being divided into multiple segments, grouping these segments by their feature vectors to generate a scenario catalog is pointless. Up- or downsampling on the other hand distorts the static environment and reduces comparability of different sequences. Consequently, the test drives need to be cut along their natural scenario transitions.

The core problem is to find these transition points. One solution is to use predefined scenarios, specified by experts, to detect the conditions, that start or end a scenario. While this approach might work well for single maneuvers, there is no guarantee, that the experts thought of every possible scenario. Most cer-

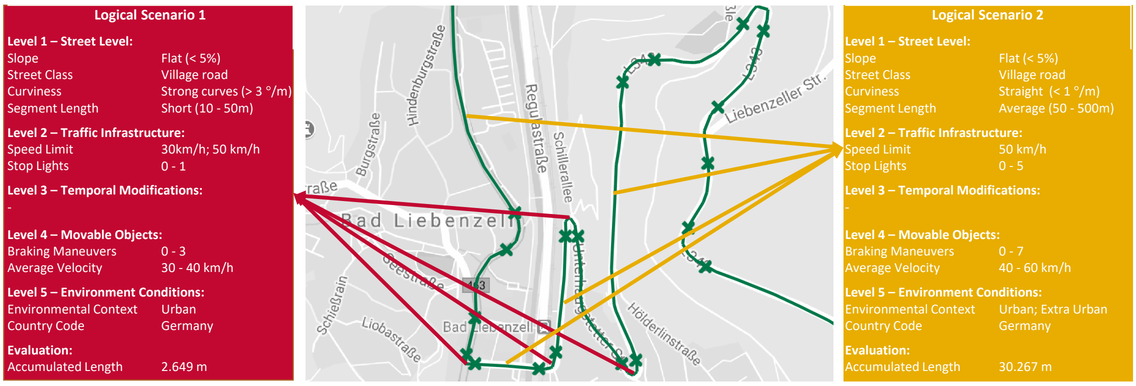


Figure 3: Exemplary scenario clusters for a PCC feature and their aggregated description.

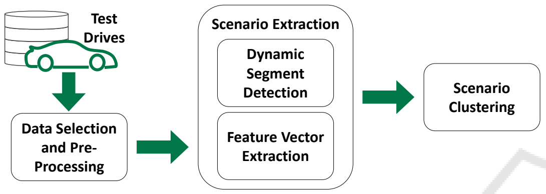


Figure 4: Required steps for the scenario extraction approach.

tainly even, there are corner cases, the experts did not think of. For the larger parameter space of complete scenarios, including the static vehicle environment, an approach is needed, that does not rely on explicit scenario definitions beforehand. By extracting the scenarios from the data pool, scenario specification is no longer necessary. All occurring scenarios are implicitly derived, if the data pool is sufficiently large and diverse.

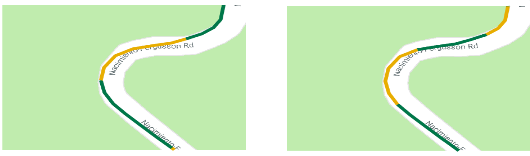


Figure 5: Static segment length (left side) vs. dynamic segment length (right side) with transition points derived from the curviness. The curve is separated into two segments with the static approach whereas the dynamic approach is able to detect the curve as a coherent segment.

In order to derive the transition points, we separate the relevant signals in scenario-dividing signals and scenario attributes. A feature's input space consists of many signals. Out of the subset of these signals, which describe the vehicle environment, some signals are categorical or ordinal, others are continuous.

Aggregation for categorical and ordinal signals can oftentimes only be done via counting or percent-based, which are not ideal for a distinct differentiation. We argue, that - in most cases - these signals are also scenario dividing and, therefore, changes in these signals naturally lead to scenario transition points. For example a change in the street type from highway to country road marks a significant change in the environmental situation and thereby separates two different scenarios. Furthermore, the percent-based representation, e.g. 64% highway and 36% country road, is not a useful description for the street type of a segment.

Continuous signals, e.g. slope or the ego vehicle's velocity, can be handled differently. For the velocity a mean value can be calculated, the slope can be represented as two values, a positive and negative value, so that these do not negate each other. In most cases, these signals describe a segment in the data and do not clearly divide it. Thus, they are categorized as scenario attributes. However, if a continuous signal is crucial for the FUT, it can also be used as a scenario dividing signal, e.g. the ego vehicle's velocity for the traffic jam assist. Via binning or thresholding the velocity can be transformed into a low dimensional space, where the boundaries of the velocity bins or the thresholds operate as the transition points. Then, a possible scenario segmentation could include differentiation of vehicle standstill, stop-and-go traffic, slow driving and driving close to the feature's velocity limit.

In general, it proved practical to use low dimensional and categorical signals as scenario dividing signals. Continuous and numeric signals on the other hand, can be utilized as scenario attributes.

2.3 Scenario Clustering

Each extracted feature vector describes a concrete scenario within the test data. These concrete scenarios shall be clustered in order to create groups of concrete scenarios - the logical scenarios. Scenarios within one cluster shall be similar and scenarios in different clusters shall be distinguishable. The clusters shall be interpretable and shall represent relevant scenarios in the real world. Since the actual number of different scenarios is unknown, the number of clusters can not be explicitly defined. The optimal number of clusters shall be derived implicitly.

3 EXEMPLARY IMPLEMENTATION

The proposed approach was implemented for a PCC feature, which not only considers the front vehicle like an ACC system does but also incorporates additional environment information into the velocity prediction (Albrecht and Holzäpfel, 2018). The corresponding signals for the environment description were selected from the real-world-driving-data. The main focus for the scenario extraction was to create a scenario catalog for functional system level feature evaluation. The PCC feature considers curves and slopes, speed limits, road topology, street class, crossings and the front vehicle for the velocity prediction.

3.1 Data Selection and Pre-Processing

A time-based signal s can be mapped onto the track position p as $s(t) \rightarrow s(p)$. Equation 1 displays the mapping for every measuring point i. Δt refers to the sampling interval and v_j to the measured ego velocity in point j.

$$s_\mathrm{i}(p) = \sum_{j=0}^{i} \Delta t * v_\mathrm{j} \tag{1}$$

The differing velocity leads to a non-equidistant position-based sampling of the formerly equidistant time-based sampling. Besides, signals on the bus system and thereby in the recorded test data are sampled with different frequencies up to 50 Hz and more. This leads to distances of 0.56 m for a velocity of 130 km/h. Such short distances are not required to detect scenarios, that can be several hundred meters long, e.g. elongated curves on highways. We decided to sample the data at every 2 m. Though, this is a design choice, we are convinced, that a more detailed resolution is not required for the scenario extraction. With the mapping of the signals to the track position, they are now independent from the altering ego vehicle's velocity during the test drives. The static environment is represented consistently.

Since real-world-driving-data is noisy, signal filtering has to be applied in order to accurately describe the vehicle's environment. Initial and error values have to be handled with respect to the actual signal values. Oftentimes, these values are written as ones in the bit-encoding of the CAN- or flexray message. Therefore, when converting the bit-stream into decimal numbers, initial and error values are converted to large decimal numbers. They can not be left unchanged or set to zero, since both could have unforeseen effects towards the clustering of the signal. For example, there is a difference between a missing speed limit being represented as 300 km/h, 0 km/h or NaN, since cluster algorithms use relative distance functions. For missing speed limits we chose NaN and for initial values the first occurring speed limit was extrapolated. In unlimited areas the speed limit was set to 210 km/h, which is the maximum supported velocity for the PCC feature.

3.2 Scenario Extraction

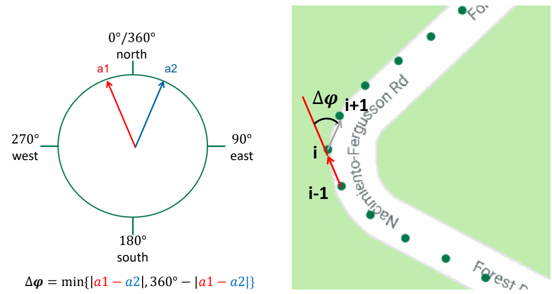
3.2.1 Dynamic Segment Detection

The street type, the speed limit and the classification of urban, extra urban or highway drive were used as scenario dividing signals. However, for the PCC there is, besides these low dimensional signals, one continuous signal, that has a major influence on the system's behavior: the curvature. Curves limit the maximum possible and the maximum tolerated velocity and hence depict unique scenarios for the longitudinal control, especially in terms of passenger comfort. They shall be evaluated separately from other road segments. By transforming the curvature into a summable value, the curviness, and then detecting significant changes in the incline of the accumulated curviness over the track position, curves can be included in the scenario separation. The curviness offers a good abstraction of the overall curvature of a certain road segment and is calculated as shown in equation 2

$$Curviness = \frac{angular\ change}{distance} = \frac{\Delta\gamma}{\Delta d} \tag{2}$$

The angular change can be derived from multiple sources, e.g. the yaw rate, the steering wheel angle or wheel angle. However, in our case all these signals were too noisy to derive a viable curviness. The best results were achieved by calculating the azimuth from the GPS points as shown in Figure 6a and 6b. With the azimuth as the angular change in each GPS point, the curviness can be calculated by dividing the angular change $\Delta\gamma$ by Δd, the sum of half the distance between $p_\mathrm{i\text{-}1} \rightarrow p_\mathrm{i}$ and half the distance between $p_\mathrm{i} \rightarrow p_\mathrm{i+1}$.

Since the GPS is noisy as well, some filtering has to be applied. The calculated curviness can be used as an indicator for implausible GPS points by calculating the maximum possible curviness that complies



(a) Azimuth as the horizontal angular change of the cardinal direction.

(b) Calculation of the azimuth of each GPS point based on the two lines $p_{i-1} \rightarrow p_i$ and $p_i \rightarrow p_{i+1}$.

Figure 6: Calculation of the curviness based on the azimuth of each GPS point.

with traffic regulations. In Germany the traffic regulations state, that vehicles must be able to drive through a turning circle with a radius r of 12.5 m. With the help of equation 3 this leads to a circumference C of 78.54 m for the minimal possible turning circle. The curviness of this turning circle can be calculated as shown in equation 4. This maximal possible curviness can be used as a threshold for filtering the GPS. A curviness above this threshold is very likely caused by a GPS error (see Figure 7 for different detected GPS errors). Since the velocity, which is used to calculate the curviness, can be noisy as well, the threshold was set to $Curviness_{\max} = 9\frac{\circ}{m}$.

$$C = 2 * \pi * r = 78.54\ m \quad (3)$$

$$Curviness_{\max} = \frac{\Delta\gamma}{\Delta d} = \frac{360\ ^\circ}{78.54\ m} = 4.58\frac{\circ}{m} \quad (4)$$

Additional filtering of the GPS can be achieved by comparing the distance between two GPS points and the driven distance derived from the velocity. Again, due to possible noise, the thresholds were set to $0.5 < \frac{d_{\text{GPS}}}{d_{\text{driven}}} < 2.0$.

Figure 7: Filtered GPS errors with thresholds derived from the curviness and relation of GPS distance to driven distance.

Finally, the transition points can be derived by building a vector for each scenario dividing signal: the street type, the speed limit, the classification of urban, extra urban or highway drive and the curviness. As already mentioned, the transition points for the curviness are derived from significant changes in the incline of the accumulated curviness over the track position (see Figure 8). The segments can then be derived by combining the transition point vectors of all signals: $\rho = \rho_{\text{street type}} \cup \rho_{\text{speed limit}} \cup \rho_{\text{environment}} \cup \rho_{\text{curviness}}$. The combined transition point vector is then sorted and duplicates are removed. The remaining elements depict start and end positions of the segments in the test drive. Merge strategies can be applied, if the segmentation is too compartmentalized due to a large number of scenario dividing signals.

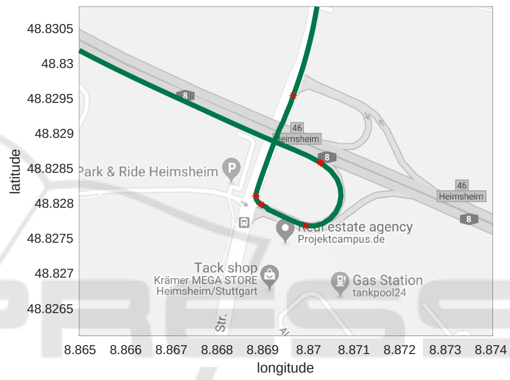


Figure 8: Segmentation via change points of the accumulated curviness.

3.2.2 Feature Vector Extraction

Besides the scenario dividing signals each segment needs an adequate description of its environmental situation in order to be clustered correctly. For each segment a describing feature vector has to be extracted. For several reasons, time or position-based features are not ideal. For one, the sequence length varies, which either leads to a different amount of values in the feature vectors or to a different sampling. Both lead to a decrease in comparability of the sequences. The second reason is, that clustering time-series with different sample sizes is not trivial.

A better solution is to derive single values for every feature by means of aggregation, minima or maxima, mean values, etc. For the showcase, 10 features were selected leading to a feature vector with 10 scalar values for each segment in the test data. For an example see Table 1.

Table 1: Exemplary segments with their corresponding feature vectors (abbrev: Segment Identifier (SID); Speed Limit (SL); Amount of Occurring Braking Maneuvers (BM); Amount of Occurring Traffic Lights (TL)).

SID	Env	Street Type	$Length_{seg}$	SL	Curv	$Slope_{pos}$	$Slope_{neg}$	Vel_{avg}	BM	TL
4	rural	rural road	54	70	2.85	0	-7.26	28.21	0	0
5	rural	rural road	56	70	2.80	0	-7.37	34.46	0	0
8	rural	district road	32	100	4.32	7.84	0	28.56	0	0
31	highway	highway	1578	140	0.02	0	-1.70	132.20	2	0
32	rural	rural road	142	140	1.16	0	-2.28	49.96	0	1
76	rural	rural road	78	140	0.56	0	-3.23	40.39	0	1

3.3 Scenario Clustering

The feature vectors can be used to cluster the segments. Due to the scalar representation common cluster algorithms can be used. For the showcase, the K-Medoids with Partitioning around Medoids (KPAM) algorithms has been implemented (Kaufman and Rousseeuw, 2009).

K-medoids is a common partitioning cluster algorithm, that is stable against outliers. A set of N data points is divided into k clusters, so that the sum of distances between each data point and its corresponding medoid is minimized. The KPAM algorithm is a deviation of k-medoids with two phases. The set of data points or observations O is divided into a subset of medoids or selected objects S and a subset of unselected objects U with $|O| = N$ and $U = O \setminus S$. For every observation o two values are calculated: the distance D_o between o and the nearest medoid in S and the distance E_o between o and the second nearest medoid in S.

During the build phase initial medoids are selected. The first medoid is the observation with the lowest sum of distances to all other observations in O. Iteratively $k-1$ medoids are selected so that the new partitioning is as distinct as possible.

During the swap phase the iteratively selected medoids are swapped in order to improve the partitioning. Every object pair in SxU is evaluated. If the sum of distances for the object in U is lower than for the object in S, both objects are swapped. This process is repeated until the partitioning is stable. For the detailed algorithm we refer to (Kaufman and Rousseeuw, 2009).

We implemented the KPAM and searched for the optimal k by iterating $1 \leq k \leq 30$ with 5 iterations for each k to avoid local optima. The best solution for each k was kept. For the k-medoid algorithm the optimal k was evaluated with the elbow-method (Ketchen and Shook, 1996) - the optimal k = 24 was derived.

4 EVALUATION

As there is no ground truth by means of classified scenarios for the test drives or an existing scenario catalog for the PCC feature, the evaluation has to be done empirically. In a first step, the quality of the clustering has to be evaluated. In a second step the interpretability of the clusters as logical scenarios is validated. Some information about the used data set is given in Table 2.

Table 2: Overview of the used data set for the PCC showcase.

Number of test drives	115
Included Countries	10
Overall Length	8432 km
Shortest Test Drive	1.13 km
Longest Test Drive	299.19 km
Average Length	73.09 km
Amount Urban	1271 km
Amount Extra Urban	7161 km

We want to demonstrate the results by showcasing three clusters of the KPAM algorithm, for the simple reason that showing all clusters goes beyond the scope of this article. Figure 9, 10 and 11 show three exemplary clusters of the KPAM clustering algorithm with six randomly chosen concrete scenarios each. Each cluster in itself is supposed to be similar, with larger dissimilarities between the clusters.

Figure 9 depicts concrete scenarios from a logical scenario, that could be described as straight country road drive with little slope. Figure 10 depicts strong curves on steep serpentines. Figure 11 displays longer highway drive scenarios on a straight road with high speed limits. All three clusters each contain similar segments. On closer inspection, scenario 4 in Cluster 19 attracts attention. It is the only scenario not on a highway. However, further investigation has shown, that this road section was formerly defined as a federal highway and is still classified as such in the electronic map, which makes the assignment plausible.

On a feature vector basis the clustering algorithm

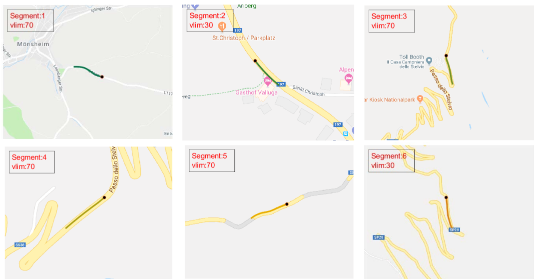


Figure 9: Six randomly selected concrete scenarios from cluster 7 depicting straight segments on country roads.

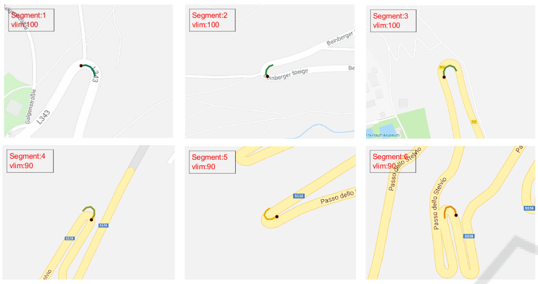


Figure 10: Six randomly selected concrete scenarios from cluster 14 depicting strong curves and serpentines.

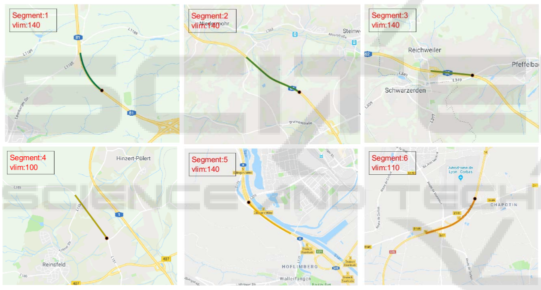


Figure 11: Six randomly selected concrete scenarios from cluster 19 depicting long highway segments.

has build valid clusters. The question remains, if the feature vectors represent valid concrete scenarios and the clusters represent logical scenarios respectively.

Hence, the next step in the evaluation is to examine the interpretability of the found clusters as logical scenarios. Each cluster represents a group of similar feature vectors describing a segment of the driving data for which the parameter spaces can be derived from the global maxima and minima for each feature. In order to represent these clusters in a human-readable format, a functional scenario for each cluster is derived:

All concrete scenarios in Cluster 7 can be classified as country road drives with average velocity on a straight and flat road segment. All scenarios are extra urban.

The concrete scenarios in Cluster 14 are on country roads as well, however, with a steep incline and strong curves. The curves are segmented accurately.

Cluster 19 represents highway test drives. All scenarios are driven at high velocities on highway segments with no noticeable curve or slope.

For the PCC feature all three logical scenarios depict different use cases with different driving challenges as well as driving requirements. It can be argued, that this approach leads to plausible and useful logical as well as functional scenarios directly derived from the data.

5 CONCLUSION & FUTURE WORK

We have shown, that our approach is well suited to find self-contained environmental scenarios within real-world-driving-data. The data-driven approach utilizes the data to extract all occurring concrete scenarios, that are gathered during test drives. A cluster algorithm was used to derive logical scenarios.

The logical scenario catalog can now be used for scenario-based testing and coverage analysis of the different existing environmental scenarios. Functional quality can be assessed for different situations, like different curves, serpentines or corner case clusters. An automated process to identify correlations between the functional quality and the logical scenarios can be set up.

The number of logical scenarios was derived implicitly for the selected signals and the exemplary data set. In future work, we will alter the data set by varying the selected test drives as well as the total length of the included test drives. The convergency point for the number of clusters k will be determined. The minimal accumulated length of the test drives to ensure convergence will be derived and we will discuss, if inferences towards the required test kilometers can be made.

For ADAS features it is common practice to differentiate between highway, urban and extra urban driving, because the driving tasks and environmental situations differ a lot between these three contexts. In order to get more precise clusters within each context, we will investigate, whether dividing the data beforehand into three different groups: highway, urban and extra urban segments, can improve the clustering result.

Furthermore, scenarios may be found in other subspaces of the complete feature space. E.g. to distinguish a curve scenario from a steep incline or a highway scenario, different subspaces need to be looked at. For the highway scenario the street class is relevant, but slope and curviness may not be, since both are usually virtually zero on highways. For steep inclines the street class can be neglected but the slope plays a significant role. As these examples demonstrate, subspaces may play a vital role in detecting different environmental scenarios and, therefore, these subspaces will be investigated by utilizing projective clustering algorithms, e.g. K-Means with Splitting and Merging (KSM). Experiments with the KSM will also include clustering with only a few clusters to investigate the most distinctive features. A low k should result in a distinction of the most separating subspaces. These will be investigated further.

REFERENCES

Albrecht, M. and Holzäpfel, M. (2018). Vorausschauend effizient fahren mit dem elektronischen co-piloten. *ATZextra*, 23(5):34–37.

Amersbach, C. and Winner, H. (2017). Functional decomposition: An approach to reduce the approval effort for highly automated driving. In *8. Tagung Fahrerassistenz*, München.

Bach, J., Otten, S., Holzäpfel, M., and Sax, E. (2017). Reactive-replay approach for verification and validation of closed-loop control systems in early development. In *SAE Technical Paper 2017-01-1671*.

Bagschik, G., Menzel, T., Reschka, A., and Maurer, M. (2017). Szenarien für entwicklung, absicherung und test von automatisierten fahrzeugen. In *11. Workshop Fahrerassistenzsysteme. Hrsg. von Uni-DAS e. V*, pages 125–135.

Christiansen, M. (2015). 5komma6 - mercedes-benz: In-geheimer-mission-auf-abnahmefahrt-mit-der-neuen-mercedes-e-klasse-w213.

de Gelder, E. and Paardekooper, J.-P. (2017). Assessment of automated driving systems using real-life scenarios. In *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pages 589–594. IEEE.

Deppe, P. (2009). Mercedes-benz passion: Mercedes-benz-praesentiert-in-genf-limousine-und-coupe-der-neuen-e-klasse.

Elrofai, H., Worm, D., and den Camp, O. O. (2016). Scenario identification for validation of automated driving functions. In *Advanced Microsystems for Automotive Applications 2016*, pages 153–163. Springer.

Gerdes, A. (2006). Automatic maneuver recognition in the automobile: the fusion of uncertain sensor values using bayesian models. In *Proc. 3rd International Workshop on Intelligent Transportation*.

Henzel, M., Winner, H., and Lattke, B. (2017). Herausforderungen in der absicherung von fahrerassistenzsystemen bei der benutzung maschinell gelernter und lernender algorithmen. *UNI DAS eV (eds.)*, 11:136–148.

Hülnhagen, T., Dengler, I., Tamke, A., Dang, T., and Breuel, G. (2010). Maneuver recognition using probabilistic finite-state machines and fuzzy logic. In *Intelligent vehicles symposium (IV), 2010 IEEE*, pages 65–70. IEEE.

ISO, I. (2011). 26262: Road vehicles-functional safety. *International Standard ISO/FDIS*, 26262.

Junietz, P., Schneider, J., and Winner, H. (2017). Metrik zur bewertung der kritikalität von verkehrssituationen und-szenarien. In *11. Workshop Fahrerassistenzsysteme*.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

Ketchen, D. J. and Shook, C. L. (1996). The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal*, 17(6):441–458.

Langner, J., Bach, J., Otten, S., Sax, E., and Holzäpfel, M. (2017). Framework for using real driving data in automotive feature development and validation. In *8. Tagung Fahrerassistenz*, München.

Langner, J., Bach, J., Ries, L., Otten, S., Holzäpfel, M., and Sax, E. (2018). Estimating the uniqueness of test scenarios derived from recorded real-world-driving-data using autoencoders. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1860–1866. IEEE.

Minnerup, P., Kessler, T., and Knoll, A. (2015). Collecting simulation scenarios by analyzing physical test drives. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 2915–2920. IEEE.

Pütz, A., Zlocki, A., and Eckstein, L. (2016). Absicherung hochautomatisierter fahrfunktionen mithilfe einer datenbank relevanter szenarien.

Schuldt, F., Saust, F., Lichte, B., Maurer, M., and Scholz, S. (2013). Effiziente systematische testgenerierung für fahrerassistenzsysteme in virtuellen umgebungen. *Automatisierungssysteme, Assistenzsysteme und Eingebettete Systeme Für Transportmittel*.

Steimle, M., Bagschik, G., Menzel, T., Wendler, J. T., and Maurer, M. (2018). Ein beitrag zur terminologie für den szenarienbasierten testansatz automatisierter fahrfunktionen. *AAET-Automatisiertes und vernetztes Fahren. Hrsg. von ITS Niedersachsen,(zur Veröffentlichung angenommen)*.

Wachenfeld, W. and Winner, H. (2016). The release of autonomous vehicles. In *Autonomous Driving*, pages 425–449. Springer.

Wang, W., Ramesh, A., and Zhao, D. (2018). Clustering of driving scenarios using connected vehicle datasets. *arXiv preprint arXiv:1807.08415*.

Zofka, M. R., Kuhnt, F., Kohlhaas, R., Rist, C., Schamm, T., and Zöllner, J. M. (2015). Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems. In *Information Fusion (Fusion), 2015 18th International Conference on*, pages 1422–1428. IEEE.