2020 IEEE Intelligent Vehicles Symposium (IV)
October 20-23, 2020. Las Vegas, USA

Clustering of the Scenario Space for the Assessment of Automated Driving

Jonas Kerber*, Sebastian Wagner*, Korbinian Groh, Dominik Notz,
Thomas Kühbeck, Daniel Watzenig, Alois Knoll

***Abstract*—Assessment and testing are among the biggest challenges for the release of automated driving. Up to this date, the exact procedure to achieve homologation is not settled. Current research focuses on scenario-based approaches that represent driving scenarios as test cases within a scenario space. This avoids redundancies in testing, enables the inclusion of virtual testing into the process, and makes a statement about test coverage possible. However, it is unclear how to define such a scenario space and the coverage criterion.**

This work presents a novel approach to the definition of the scenario space. Spatiotemporal filtering on naturalistic highway driving data provides a large amount of driving scenarios as a foundation. A custom distance measure between scenarios enables hierarchical agglomerative clustering, categorizing the scenarios into subspaces. The members of a resulting cluster found through this approach reveal a common structure that is visually observable. We discuss a data-driven solution to define the necessary test coverage for the assessment of automated driving. Finally, the contribution of the findings to achieve homologation is elaborated.

***Index Terms*—Autonomous vehicles, Vehicle safety, Testing, Risk analysis, Performance analysis**

I. Introduction

Automated driving is one of the most anticipated future technologies and multiple automotive brands have announced the release of automated vehicles in the near future [1], [2]. However, the assessment method to ensure a safe operation and achieve homologation is not clear yet, which delays the release of this technology [3]. Existing approaches in the automotive domain require an infeasible amount of real-world testing [4]. Hence, current research focuses on scenario-based approaches [5], [6]. These promise to put relief on the required test amount by inclusion of cross-verified virtual testing [7], [8] and avoidance of test redundancies [9]. Then again, the representation of scenarios within a scenario or test space enables variation of the test cases and a statement about the test coverage [10].

Despite these efforts, the success of scenario-based testing heavily depends on how well scenarios are defined within a test space. Consequently, this work focusses on numerically describing the difference between scenarios in the test space, followed by a proof of concept on how scenario clustering has to be established to enable an analytical estimate of the test coverage for automated driving.

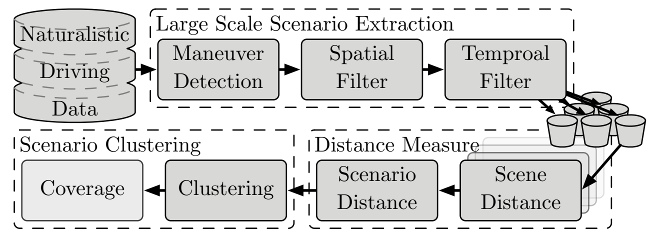

Fig. 1. The methodology presented in this paper is divided into three components scenario extraction, distance measure, and clustering. These also follow the general structure of this paper.

A similar approach applies random forest methods to simulated driving data [11], [12]. The approach shows promising results can be generated with clustering. The way the features are manually chosen limits the mapping of the temporal development of a scenario. Additionally, manually choosing features does not guarantee to map the characteristics of any scenario completely and the impact of using simulated data is not elaborated. There is, to the best of our knowledge, no further work directly addressing the clustering of scenarios. For that reason, and because of inherent differences in the underlying data, the method presented in this work cannot be benchmarked against existing approaches. Rather, a qualitative analysis of the results is conducted.

We present a novel approach to the definition of scenario space. Its major components are depicted in Figure 1 and they structure the remainder of this work. As a foundation, a large amount of driving scenarios is extracted from naturalistic real-world driving data. Section II provides the underlying maneuver detection and spatiotemporal filtering is used for the identification of scenarios within the large amount of data. All scenarios are presorted into buckets for comparability depending on structural information. A custom distance metric for measuring the similarity of scenarios is first described based on scenes or single time steps in Section III. The accumulation to a single measure for a pair of scenarios follows. In Section IV, the results from the distance measure are used for hierarchical agglomerative clustering. Based on the found clusters, discussions about potential coverage statements are held. Lastly, Section V provides conclusions and future work.

* These authors contributed equally to this work

J. Kerber, S. Wagner, and A. Knoll are with the Chair of Robotics, Artificial Intelligence, and Embedded Systems, Department of Informatics, Technische Universität München, Munich, Germany

K. Groh, D. Notz, and T. Kühbeck are with BMW NA, Mountain View, USA

D. Watzenig is with the Institute of Automation and Control at Graz University of Technology, Graz, Austria

II. Large Scale Scenario Extraction

Prior to the definition of a scenario distance metric for the clustering algorithm, a significant number of scenarios need to be generated and classified according to their characteristics into so-called scenario buckets. The data source used for this extraction is the *highD* dataset [13], which provides drone recordings of 110,516 vehicles' naturalistic trajectories on German highways with two or three lanes. Therefore, this extraction method and also the distance measure is created specifically for highway situations and is not suited for urban scenarios without modifications.

The construction of a single scenario from the trajectory dataset requires three steps:

1) Maneuver detection: Every scenario needs to include a certain type of maneuver, e.g. a lane change (LC), that defines the structure of the scenario.
2) Spatial filter: Scenarios are always subjective and refer to one vehicle which is from now on referred to as EGO. A scenario must only include the trajectory information of other traffic objects (TOs) that are relevant for the behavior of the EGO in the vehicle pool.
3) Temporal filter: The duration of a scenario is chosen to fully include all detected maneuvers over their full length in the relevant environment.

These three components are described in the following subsections and the results are summarized.

A. Maneuver Detection

The goal of this component is to detect defined maneuvers as they are later needed for the temporal filter. Following the scenario definition of Ulbrich et. al. [14], a scenario consists of multiple scenes that describe a short temporal snapshot. Scenarios have a longer duration and they contain actions and events. We define actions as specific maneuvers and focus on LC maneuvers as they are one of the primary sources of risk in highway traffic. The duration of a maneuver is specified by the start and end event. A detection algorithm starts with identifying the lane crossing times by continuously checking for a change in the lane ID as provided in the *highD* dataset. After that, the start and end times are determined by iterating forward and backward in time until the lateral velocity drops below a threshold of $0.03\frac{m}{s^2}$. The maneuver detection is repeated for every TO in the dataset and the start and end events of every maneuver are saved with the corresponding TO ID.

In this work, maneuver detection is only done on LCs as a proof of concept. Certainly, an extension to more maneuvers is required for completeness but exceeds the scope of this work.

B. Spatial Filtering

The dataset contains multiple vehicles existing at the same time and certainly not all of them are relevant for the EGO. Therefore, spatial filtering determines the relevant environment that influences the behavior of the EGO. For every recorded scene of the EGO, a list of relevant TOs needs to be determined. The spatial filter has to be designed in a way that this list is as short as possible while still including all relevant TOs.

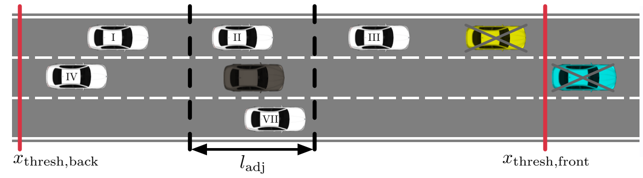

Fig. 2. Spatial filtering through an eight-vehicle-model identifies all TOs that are relevant for the EGO in a current scene. Vehicles II and VII are in the adjacent slot depicted by the dashed lines. The blue TO is discarded because it is past the red relevancy threshold and the yellow vehicle is discarded because there is already TO III in the slot which is closer to the EGO. Three slots stay empty because there is no vehicle within the area.

For the spatial filter, we use an eight-vehicle-model with three parameters as depicted in Figure 2 to judge whether a TO is relevant for the EGO or not. Here, the proximity of the EGO is divided into eight slots on the current and the neighboring lanes of the EGO. There are three slots in front of and three behind the EGO, each on the lanes left and right to the EGO lane, as well as the EGO lane itself. Additionally, there are the two adjacent slots directly left and right to the EGO. Every slot must only contain one vehicle which is why only the TO closest to the EGO is considered while others are discarded such as the yellow TO in Figure 2. Note that the longitudinal distances to the EGO are measured from the geometric center point of both vehicles. Any TOs that are two or more lanes to the left or right to the EGO are ignored for the spatial filtering as they only become relevant once they are in a neighboring lane. The slot length in front of and behind the EGO is limited by the relevancy thresholds shown by the red lines. They imply that any vehicle beyond does not influence the behavior to the EGO because it is too far away such as the blue TO in Figure 2. If there is no TO in a slot it stays empty and thus TOs I, II, III, IV, and VII are judged relevant for that scene.

The widths of the slots are given by the width of the respective lane and the lengths of the slots have to be parameterized. For the front and back relevancy thresholds $x_{thresh,front}$ and $x_{thresh,back}$, we choose $100m$ and $50m$. The length of the adjacent slot l_{adj} is chosen to be $10m$ long as it is highly unlikely that two vehicles drive within one slot on a highway.

Lastly, the spatial filtering process defines every vehicle in the dataset as the EGO once. The filter is applied for all scenes of its appearance which results in a list of relevant TOs for every EGO choice. This list can also be longer than eight vehicles as TOs can enter and leave the cells of the eight-vehicle-model over time.

After the spatial filtering and the maneuver detection process, every vehicle in the dataset has information over the relevant environment as well as the timestamps of the start and end event of maneuvers. This is the basis for the temporal filter that determines the scenario duration and vehicle pool for the final scenario extraction.

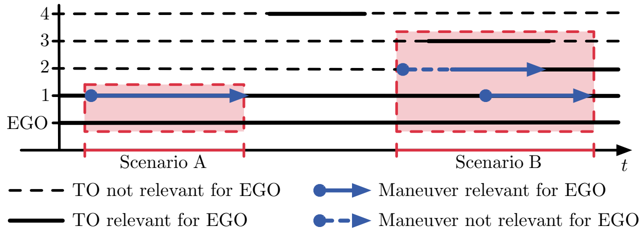

Fig. 3. Temporal filtering determines the duration and the vehicle pool of a scenario. The horizontal lines show the state information of four exemplary TOs over the lifetime of an EGO. The scenario duration, denoted by the width of the red frame, is determined with the help of the start and end events of the maneuvers and the relevant TOs, denoted by the height of the red frame, which are chosen with the help of the relevancy information.

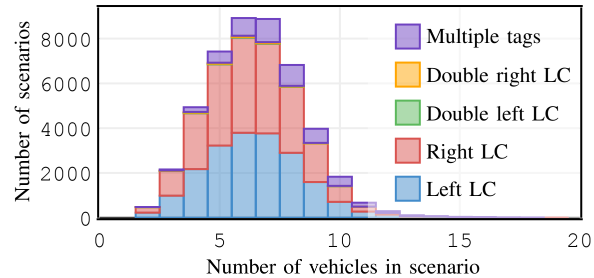

Fig. 4. The extracted scenarios are assigned to scenario buckets according to the number of vehicles included in the scenario. Also, the maneuver types are shown through tags. Most scenarios only contain one specific maneuver tag but with rising vehicle pool size the ratio of scenarios with multiple tags increases.

C. Temporal Filtering

As a final step of the scenario extraction, the detected maneuvers and the relevancy information from the two previous components are used to determine:

a) the duration of the scenario
b) the TOs that are relevant within this time.

The goal of the temporal filter is that the duration and the vehicle pool of the scenario are chosen so that the extracted scenario contains its cause and effect. An example of this is a scenario, as shown later in Figure 6, where the EGO follows another TO A in the left lane at a higher speed (cause) which results in A merging to the left lane in front of TO B so that the EGO can pass (effect). Accordingly, every scenario should always include the full duration of the maneuvers that caused it. The scenario duration and vehicle pool are determined by two rules:

Rule A: A scenario always starts with the first start event and ends with the last end event of all ongoing maneuvers.

Rule B: If a TO is relevant for at least once within the scenario duration its relevancy will be extended over the whole duration.

These rules are further elaborated with the use of Figure 3. Therein, the temporal evolution of an EGO and four relevant TOs identified by the spatial filter are shown. Dashed lines indicate irrelevancy for the EGO in the current scene while solid lines state relevancy. Further, blue arrows show the maneuvers found with the presented maneuver detection. During the first maneuver of TO_1, no other TO is relevant and no other maneuver happens. The temporal context is hence given through Rule A and only this TO and the EGO are included in the scenario. Rule A means further that two overlapping maneuvers result in only one scenario, as shown for Scenario B on the right. During that time, also TO_3 is marked as relevant once and therefore included over the whole duration of the two maneuvers through Rule B. As a conclusion of both of these rules, for a TO, that initially becomes relevant while it is in a maneuver, the relevancy needs to be extended so that the start of the maneuver is included in the scenario duration. This can be seen for TO_2 in Scenario B in Figure 3.

Scenarios are extracted by iterating over the scenes of every vehicle in the dataset while considering relevancy information and maneuver events. Note that multiple scenarios can be extracted from a single vehicle if the maneuvers and therefore the scenario durations do not overlap, as shown for the EGO in Figure 3. Having all three components of the extraction defined, they are applied to the *highD* dataset and results follow.

D. Scenario Extraction Results

Using the presented method, 46,681 scenarios are extracted from the 110,516 vehicles included in the *highD* dataset. This section aims to present some statistics over the gathered data to make the filter choices plausible.

For 38.1% of vehicles exactly one scenario is extracted, for 2.0% two scenarios and below 0.1% three to five scenarios are extracted. No scenarios are extracted for 59.8% of vehicles because no maneuver happens in the relevant environment. The distribution of scenarios according to their vehicle pool size is shown in Figure 4. Furthermore, every scenario is tagged with the maneuver type it contains and the vertical bars are divided accordingly. As a result, 56.13% of scenarios contain a right LC, 52.59% a left LC, 1.36% a double right LC, and 0.80% a double left LC. The total percentage of scenarios that contain more than one tag and therefore multiple maneuvers is 10.48%. This number rises with increasing vehicle pool size form 2.1% at three, over 6.7% at five, and up to 22.3% at ten vehicles as more maneuver happen with more vehicles. Currently, these tags are only used for statistical plausibility check of the results.

It is crucial for the scenario distance measure (Section III) and the clustering (Section IV) that the vehicle pool size and the scenario duration should be kept as small as possible to prevent an increase of the scenario space dimensions. The restrictiveness of the spatial filter causes a small total average vehicle pool size of 6.51. This is small compared to the average number of vehicles that are on screen at a time (15.77) and during the whole duration per EGO (30.26). Also, the temporal filtering approach reduces the average scenario duration from an average $14.38s$ on-screen time to $8.36s$.

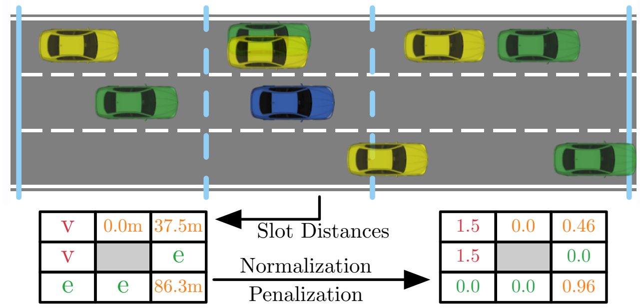

Fig. 5. The figure shows two scenes in yellow and green from two different scenarios that are compared. The scene distance is calculated by summing the eight surrounding slot distances. First, the slot distances are calculated in meters. Slots with only one TO are marked as vacant and slots where neither of the two scenarios has a TO are marked as empty. In a second step, the metric slot distances are normalized to a distance between 0 and 1, empty slots are set to 0, and vacant slots are set to a penalty factor of 1.5.

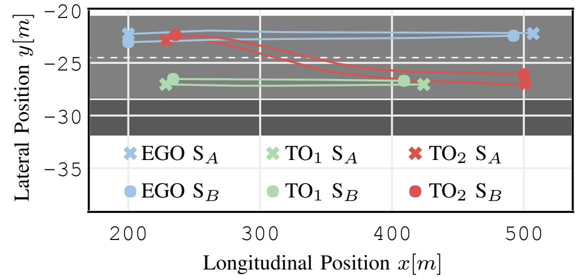

Fig. 6. A scenario comparison plot enables visual inspection of the similarity of a scenario pair. The trajectories of both scenarios are superimposed, whereby crosses mark scenario A (S_A) and circles mark scenario B (S_B). This example shows a scenario pair with one of the closest distances in scenario bucket on a *highD* recording location with two lanes and three vehicles in the pool. Note that the scaling of longitudinal to lateral coordinates is 1:8 for better visibility.

To be able to compare two scenarios, they must both take place at the same location and their vehicle pool must be of the same size. Hence, they are divided into scenario buckets of comparable instances. With this scenario dataset, a scenario distance measure that enables a clustering of scenarios is derived in the following section.

III. A Scenario Distance Measure

A central requirement for clustering scenarios is the definition of a distance measure that describes the similarity of two scenarios. As previously mentioned, it is only meaningful to calculate a distance on comparable scenarios, meaning with the same number of vehicles in its pool and on the same location. Therefore, a bucket with three vehicles including the EGO on a two-lane highway is used for all results in the following.

The derived distance measure samples multiple scenes in fixed time intervals, calculates the scene distance for each sample, then cumulates the scene distances, and normalizes it by its length. In the first subsection, the calculation of the scene distance utilizing the eight-vehicle-model is explained. After that, scene distances are accumulated and normalized by their length to get the scenario distance. Finally, results of the derived distance measure are shown and future improvements are discussed.

A. Scene Distance

The calculation of the scene distance is based on the slots of the eight-vehicle-model, as shown in Figure 5. Herein, the TOs in the proximity of the EGO are shown in yellow and green for the current scenes of two scenarios to compare. Similar to the spatial filter in the extraction, all surrounding vehicles are first mapped to the eight slots around the EGO.

Whenever both scenarios have a TO in one slot the longitudinal distance between the vehicles' geometrical centers is measured and saved for that slot as referred by the orange values in the figure. We define slots as empty (green e) when neither scenario contains a TO in that slot and as vacant (red v) when only one of both scenarios has a vehicle in that slot.

We use the same parameterization as in the extraction with $100m$ front relevancy threshold, $50m$ back relevancy threshold, and $10m$ adjacent slot length which results in a maximum slot lengths of $95m$. All eight slots are now normalized with this value so that all distances have a value range of $[0, 1]$. The slots marked as empty are set to 0 as there is no vehicle in both scenarios and therefore no difference between them. For a vacant slot, a penalty of 1.5 is introduced, which is even worse than the maximal difference. This is motivated by the fact that removing or adding a vehicle in one of these slots structurally changes the scenario. Certainly, a cluster should only contain scenarios with similar structure and hence such a dissimilarity is penalized.

In conclusion, every slot distance can have a value between 0 and 1.5 which sums up to a scene distance value range of $[0, 12]$. Yet, alone the distance of single scenes is not enough for comparing scenarios. Thus, its extension to scenarios is presented in the following.

B. Scenario Distance

The scenario distance is calculated by sampling scenes at $5Hz$, summing the scene distance values, and normalizing the sum by the number of scenes. When comparing scenario pairs of different length, longer scenario pairs do not automatically receive a higher scenario distance score than shorter ones. The advantage of the normalization is that the scenario distances also have a value range of $[0, 12]$ which gives an intuitive understanding of the distance score. For example, if a scenario pair has a distance score under 1.0 it usually has a similar structure because over the majority of the sampled scenes there is no vacant slot which results in a slot distance value of 1.5.

Finally, symmetry and non-negativity of the scenario distance are given which is a requirement for most linkage criteria of hierarchical clustering as described in Subsection IV-A. Exemplary results given through this parameterization follow.

C. Distance Measure Results

Every possible pair within one scenario bucket is calculated and stored in a distance matrix. Results from the scenario distance measure are analyzed with the help of a scenario comparison plot that shows the trajectories of both scenarios superimposed in Figure 6. This example shows a scenario where the EGO follows another TO_2 on the left lane and both drive by a TO_1 on the slower right lane. TO_2 then changes to the slower right lane so that the EGO passes TO_1. This also shows that the extraction method successfully includes cause (EGO follows TO_2 at a higher speed) and effect (TO_1 merges to the right lane to let the EGO pass). Based on this comparison plot, one can validate the similarity of both scenarios as this example, with a distance value of 0.14, is in the 0.77 percentile of all scenario pairs in its vehicle pool.

While the current implementation of the distance measure works well for a proof of concept of scenario clustering, as shown in Section IV, there are some possible improvements to make for the future. The current implementation only describes differences in the surrounding of the EGO and does not reflect the EGO movement. Thus, scenarios with a velocity difference across all TOs including the EGO and scenarios that are shifted by lanes are considered similar. Additionally, distance measures of scenario pairs with different lengths are only calculated over the duration of the shorter scenario. This works fine for minor duration differences but gets more problematic as they get bigger. One option to solve this would be to introduce a penalty for duration differences or even to filter out very short scenarios.

Summarizing, the presented distance measure assigns low values to scenario pairs whose similarity is confirmed visually. Downsides to the current implementation reveal suggestions for improvements that are to be considered for an unrestricted application. For this proof of concept, the current implementation is sufficient to provide the foundation for a clustering following in the next section.

IV. Proof of Concept for Scenario Clustering

This section introduces a proof of concept for the actual clustering of scenarios utilizing the distance measure presented before. First, hierarchical agglomerative clustering is applied to the pre-computed distance matrix of a scenario bucket. After that, the resulting clusters are validated by visual inspection. Finally, a concept is introduced to how scenario clustering can be used in the future to estimate test coverage for automated driving.

A. Application of Hierarchical Clustering

We choose a connectivity-based clustering approach as it groups instances into clusters based on their relative distances. Hierarchical agglomerative clustering follows a bottom-up approach where all instances start in their own cluster and are continuously merged following a linkage criterion until a threshold is reached [15]. This procedure is visualized through a dendrogram, as shown in Figure 7.

Experimenting with different linkage criteria and thresholds reveals best results with complete-linkage also known as Farthest Point Algorithm or Voor Hees Algorithm [16] for the introduced distance measure. The exact mathematical

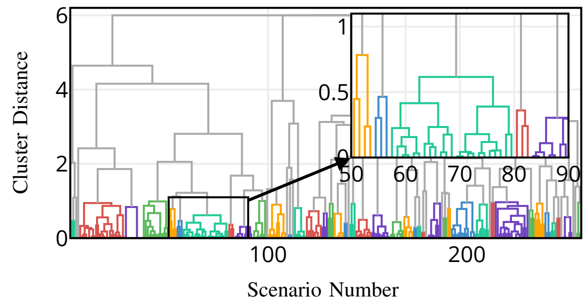

Fig. 7. The dendrogram visualizes the bottom-up hierarchical clustering process. Scenarios shown on the x-axis are continuously merged into clusters according to an increasing linkage criterion depicted on the y-axis. The merging stops at a determined threshold and the resulting clusters are formed, depicted by the different colors. The example cluster that is depicted in Figure 8 is shown in greater detail in the top right corner.

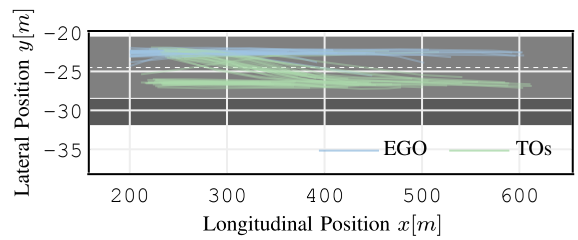

Fig. 8. The cluster validation plot overlays all trajectories of one cluster to ascertain that the clustered scenarios are similar. This example is the same cluster as shown in Figure 7 and illustrates a scenario where the EGO drives on the left lane behind another vehicle A that then merges in front of vehicle B so that the EGO can pass.

procedure shall not be explained further at this point as it strictly follows the implementation provided by *scipy*'s hierarchical clustering module [17]. The threshold where the merging stops and the final clusters are specified are determined manually. An initial threshold is set at the biggest gap in the dendrogram which is then fine-tuned by iteratively decreasing the threshold until the validation plots of the clusters showed homogeneous scenario types.

The results provided in Figure 7 show the clusters for the previously mentioned bucket including three vehicles on a two-lane road. A cluster in green is highlighted as it is used for visual validation in the following.

B. Clustering Results and Validation

After creating clusters through the application and parameterization of the chosen clustering method we need to validate that the resulting clusters indeed group scenario of similar type. The validation is done by a qualitative inspection of the plot shown in Figure 8 where a clear structure is recognized. In fact, the scenario pair example shown earlier in Figure 6 is part of the cluster that is highlighted in the dendrogram in Figure 7 and this validation plot. With these results, we prove that the derived distance measure succeeds at describing the similarity between scenarios and hierarchical agglomerative clustering with complete-linkage can be successfully applied to the extracted dataset.

When comparing the scenario buckets on the same location but with bigger vehicle pool size the increased dimensionality, which requires exponentially more data, is evident. While the average distance between all scenario pairs increases from 3.49 for three vehicles to 4.68 for five vehicles the average relative cluster size decreases from 1.89% to 0.72%. This means that there are more clusters, which are more spread out, as dimensionality increases.

In summary, this proof of concept shows a possible solution for scenario clustering which can be used to determine test coverage for automated driving as presented in the following subsection.

C. Towards Estimation of Test Coverage

Finally, we propose that scenario clustering can be applied to find a data-driven solution to the required test coverage for autonomous driving. The idea is, that the test volume has to be equal to all scenarios that are possible on a specific road. This is done by continuously collecting scenario data until an exit criterion is met.

We abstract the data collection into three (partly overlapping) phases:

1) Discovery: New types of scenarios with the same structure are discovered and form a new cluster
2) Expansion: Variations of that scenario type are found and expand the size of the cluster
3) Convergence: The density of clusters increase while no new clusters are discovered and expansion stops

When the third phase has been reached, a density threshold must be defined as an exit criterion.

However, using scenario clustering, one must test for every location individually as global homologation can not be given. Thus, locations are divided into reference and validation locations. On a reference location, the collection process will be done once until convergence. Every validation location is linked to a reference location that is very similar to the former with regards to road topology, speed limits, etc. Therefore, only a subset of the data volume of the reference location has to be tested on validation locations to ensure a safe operation.

V. Conclusion and Future Work

A scenario clustering approach for automated driving situations is proposed. It first creates a scenario dataset through maneuver-based extraction with a spatiotemporal filter using an eight-vehicle-model. After that, a novel scenario distance measure is derived that describes the similarity of a scenario pair based on their trajectories. Finally, a proof of concept shows the application of hierarchical clustering with the custom distance measure to the scenario dataset and resulting clusters are validated.

The implementation is done on a proof of concept level. For completeness, the approach still requires extensions to be applicable without limitations. Among these is the extension to urban environments, more maneuvers than just LCs, and open shortcomings of the distance measure.

The distance measure can be used to find similar scenarios in simulation and real-world testing for cross-verification [8]. Most importantly, the clustering approach provides the means for a data-driven as opposed to a statistical [4], [18] estimation of required test-coverage as suggested in Subsection IV-C. Altogether, the presented methodology is a promising candidate to enable scenario-based testing for homologation and assessment of automated driving.

Acknowledgment

The authors gratefully acknowledge the financial support by the BMW Group within the CAR@TUM project.

References

[1] F. Lambert, "Elon Musk clarifies Tesla's plan for level 5 fully autonomous driving: 2 years away from sleeping in the car," Apr. 2017.
[2] M. Poschenrieder, "Automated driving at the BMW Group." May 2017.
[3] L. K. Schoolov, Katie, "Self-driving cars were supposed to be here already — here's why they aren't and when they should arrive," Nov. 2019.
[4] W. Wachenfeld and H. Winner, "Die Freigabe des Autonomen Fahrens," in *Autonomes Fahren*. Springer, 2015, pp. 439–464.
[5] K. Groh, T. Kuehbeck, M. Schiementz, and C. Chibelushi, "Towards a Scenario-Based Assessment Method for Highly Automated Driving Functions," in *8th Conference on Driver Assistance*, Munich, Sep. 2017.
[6] T. Form, "PEGASUS Method for Assessment of Highly Automated Driving Function," Tokio, Nov. 2018.
[7] K. Groh, S. Wagner, T. Kuehbeck, and A. Knoll, "Simulation and its Contribution to evaluate Highly Automated Driving Functions," *SAE International Journal of Advances and Current Practices in Mobility*, vol. 1, no. 2, pp. 539–549, Apr. 2019.
[8] S. Wagner, K. Groh, T. Kühbeck, and A. Knoll, "Towards Cross-Verification and Use of Simulation in the Assessment of Automated Driving," in *Processings of the IEEE Intelligent Vehicle Symposiym 2019*. Paris, France: IEEE, Jun. 2019, pp. 1407–1414.
[9] C. Amersbach and H. Winner, "Functional Decomposition: An Approach to Reduce the Approval Effort for Highly Automated Driving," in *8. Tagung Fahrerassistenz*, 2017.
[10] S. Wagner, K. Groh, T. Kühbeck, D. Watzenig, L. Eckstein, and A. Knoll, "Virtual Assessment of Automated Driving – Methodology, Challenges and Lessons Learned," *SAE International Journal of Connected and Automated Vehicles*, no. Special Issue on Safety and Reliability for Autonomous Vehicles, Oct. 2019, (under review).
[11] F. Kruber, J. Wurst, and M. Botsch, "An unsupervised random forest clustering technique for automatic traffic scenario categorization," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2811–2818.
[12] F. Kruber, J. Wurst, E. S. Morales, S. Chakraborty, and M. Botsch, "Unsupervised and Supervised Learning with the Random Forest Algorithm for Traffic Scenario Clustering and Classification," in *Proceedings of the IEEE Intelligent Vehicle Symposiym 2019*. Paris, France: IEEE, Jun. 2019, p. 8.
[13] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems," in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018.
[14] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference On*. IEEE, 2015, pp. 982–988.
[15] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.
[16] E. M. Voorhees, "Implementing agglomerative hierarchic clustering algorithms for use in document retrieval," *Information Processing & Management*, vol. 22, no. 6, pp. 465–476, 1986.
[17] E. Jones, T. Oliphant, and P. Peterson, "SciPy: Open source scientific tools for Python," 2001.
[18] C. Amersbach and H. Winner, "Defining required and feasible test coverage for scenario-based validation of highly automated vehicles," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 425–430.