Efficient Characterization Method for Big Automotive Datasets used for Perception System Development and Verification

PAWEŁ KOWALCZYK¹, (Student Member, IEEE), MATEUSZ KOMORKIEWICZ², (SENIOR MEMBER, IEEE), PAWEŁ SKRUCH³, (SENIOR MEMBER, IEEE) AND MARCIN SZELEST⁴, (Senior Member, IEEE)

¹Aptiv Services Poland S.A. ul. Podgórki Tynieckie 2, 30-399 Kraków, Poland, ORCID: 0000-0002-5594-9278
²Aptiv Services Poland S.A. ul. Podgórki Tynieckie 2, 30-399 Kraków, Poland, ORCID: 0000-0002-0672-5163
³AGH University of Science and Technology, ul. Mickiewicza 30, 30-059 Kraków, Poland, ORCID: 0000-0002-8290-8375
⁴Aptiv Services Poland S.A. ul. Podgórki Tynieckie 2, 30-399 Kraków, Poland, ORCID: 0000-0002-0522-1270
Corresponding author: Paweł Kowalczyk (e-mail: pawel.2.kowalczyk@aptiv.com).

ABSTRACT

THE paper proposes a formal approach for describing and evaluating the datasets that are used in automotive applications for machine learning, testing, and validation purposes. Proper, that is, qualitative and quantitative characterization of the datasets can simplify the analysis, evaluation, and comparison of perception-based algorithms designed for highly automated vehicles. Such formalism is also needed to achieve compliance with the automotive industry safety standards that have been recently introduced. Characterization in the form of size or type of raw data, number of recognized and classified objects, and environmental parameters is not perfectly suitable for describing both the static and dynamic aspects of automotive datasets; therefore, another approach is required. In this paper, an efficient method based on an object tracking mechanism, grid representation of the sensor field of view, heatmap concept, and Wasserstein metric is proposed. The efficiency of the method is demonstrated by its ability to handle both the size, properties, and diversity of the dataset, including static and time-varying aspects. The presented description can also be used to compare different datasets and to define the amount of data to be collected.

INDEX TERMS

Autonomous Vehicles, Big data, Perception, Trajectory

I. INTRODUCTION

A. MOTIVATION

The development of new generation cars with higher level of automation will require solving many research problems. One problem can be formulated as adding eyes and a brain to the vehicle so that it is aware of its driving environment to first assist and then to replace the driver. This research problem is equivalent to equipping a car with a type of artificial intelligence. As special skills and the ability to properly perceive the surrounding environment are required by human beings driving cars, artificial intelligence (AI) together with machine Learning learning (ML) are essential to achieve higher levels of autonomy. In other words, achieving higher levels of driving automation with appropriate levels of quality and reliability will not be possible without the use of AI-based methods. As a reminder, the Society of Automotive Engineers (SAE) distinguishes six levels of autonomy [1], that range, from level 0, which means no automation, to level 5, which means full automation. It should be noted that most of the currently produced models are at level 1 or 2 on this scale.

In the design and development process of automotive control systems, one of the most important issues is related to the safety aspects of the systems. This is in particular valid for systems that utilize AI components. It is clear that testing all aspects of such systems is impossible as there are an infinite number of important road scenarios. The process of selecting just a few of the many possible scenarios is a difficult and challenging task and currently is most often based on qualitative best engineering judgment. Verification of automotive control systems requires large, variable, and diverse datasets to guarantee a proper safety level. By large datasets, we mean collections containing several dozen peta bytes (PB) of data as in the present state. In the last few years,
these requirements for the amount of data collection have changed by several orders of magnitude, which illustrates the technological progress in this area. It is also expected that without virtual simulators, it will not be possible to comprehensively verify driving algorithms for autonomous vehicles. Virtual simulators supported by AI can supplement datasets collected from real test drives. The extended datasets can then be used for learning, testing, and validation purposes. The data from many sensors recorded in a vehicle with a distribution that should reflect the normal user use case statistics are named real-world user profiles (RWUPs). Virtual world user profiles (VWUPs) represent data generated from virtual environments.

One of the most challenging problems in the development of automotive control systems is defining the performance indicators to be used to evaluate and compare different algorithms on a given dataset. This is particularly important for solutions that are based on machine learning techniques where the data are categorized into learning, test, and validation subsets. The learning dataset is used to obtain and tune the neural network parameters. The test dataset is used to verify learning outcomes. Validation datasets are used to assess the algorithm performance. Here, the problem is that with two datasets, we can train a network substantially differently. Moreover, when the network is already trained, we can have two validation datasets that produce different quality metrics. There is a need for an overview of different results with unified assumptions, as presented, for example, in [2]. Thus, there is a need to formally describe the datasets that are used for learning, testing, and validation purposes. Having this formality in place, we can develop a measure that allows us to conclude if the datasets are similar in terms of defined criteria and use so that we can compare the results of the algorithms. It is also clear that to reliably compare two different algorithms, similar datasets must be used. If the datasets are not comparable, we cannot draw any conclusions regarding the comparison. The authors of [3] claim that data design deserves at least the same amount of attention as the model design. Their work underlines that public datasets in the academic community differ substantially from those in industrial settings. Evaluation of active driving assistance systems in newest cars performed by researchers at AAA shows that on over 4000 miles of real-world driving on average some type of issue occurs every eight miles [4]. This discrepancy shows that validation datasets do not reflect reality correctly. More open problems and current research challenges connected to datasets used in computer vision for autonomous vehicles are described in [5].

B. CONTRIBUTION

In this paper, we propose an formal approach for the description, evaluation, and comparison of datasets that are used in automotive applications for machine learning, testing, and validation of perception-based systems. We use the tracks of detected and classified objects as the main sources of information. The tracks are projected on the grid representing the field of view of the sensors monitoring the vehicle’s surroundings. Then, the coverage of the grid elements by the tracks is used to create heatmaps. In our approach, the heatmap is considered a formal representation of the dataset. We propose the use of a Wasserstein metric to compare heatmaps. In addition, we describe track similarity measures that can be used to select unique and diverse test scenarios in the data collection process. The theoretical analysis is verified on publicly available datasets.

C. ORGANIZATION OF THE PAPER

The paper is organized as follows. The next section is a literature review focused on available methodologies of big data description that are useful for automotive data. The following section contains background information related to specific aspects of automotive perception systems, sensors, object tracking, safety, grid representation, and heatmaps. In the following two sections, similarity measures for comparing tracks and test scenarios are introduced. Next, the concept of dataset characterization is presented. The final section contains some comments and conclusions. The pipeline of the proposed methodology is depicted in figure 1.

II. RELATED WORK

Formal descriptions of datasets can be connected to the determination of different types of meta-data and building meta-models that can characterize the datasets. Meta-data and meta-models play key roles in the meta-learning process [6], which is the field of research that aims at improving learning performance. Metadata in the case of automotive datasets can be considered a supplement to raw data logged directly from the vehicle sensors. The meta-data are usually logged at the same time as raw data by the vehicle logging system during data collection (e.g., GPS position, speed, yaw rate, weather conditions, etc.) or are added later on as a result of the postprocessing of raw signals by the system functions (e.g., height of the sun, number of pedestrians in a frame, etc.) [7]. The extensive research done thus far shows that a strong correlation exists between learning datasets and classification algorithms [8]. In [9], the authors investigated such correlations and similarities between datasets. In this paper, the following two approaches were described: clustering datasets using error correlations and clustering datasets using ranks.
In [10], complexity measures were proposed to characterize datasets and to assess the performance levels of learners. Further study on the prediction of classification performance using complexity measures was carried out in [11].

### A. BIG DATA CHARACTERIZATION AND VISUALIZATION

One of the most comprehensive characterization methodologies for big data workloads is the metric importance analysis (MIA) found in [12]. It presents a methodology for choosing and presenting metrics from wide sets that represent data at multiple layers. The authors state that MIA reduces the complexity of the analysis without losing information. The most important metrics characterizing the data are then shown with Kiviat plots.

Regarding the analysis and synthesis of large datasets, much space in the literature is devoted to visualization methods. In [13], the authors presented a tool called time-tunnel for the visualization of time series multidimensional data in a web browser using 3D charts. The approach relies on the exploration and comparison of patterns to find a strong relationship between them. To do so, the data-wings (2D subchart) in time-tunnel are selected and arranged based on a genetic algorithm that produces mutations of potentially optimal solutions to find the best one.

From the automotive point of view, sensor data is classified as massive spatiotemporal data by the authors of [14]. This enables automatic geospatial analysis and visualization. Another example of visualization of big data comes from [15], which describes how to use the geometry of parallel coordinates to present high dimensional data and enable visual patterns. The author states that complex data cannot be shown in a single visualization alone, but multiple views of different types can provide different perspectives on the same data and are key for deep understanding. To emphasize this effect, it is good to coordinate multiple views. Brushing and linking is given as an example, and [16] extends the parallel approach to the selection of scatterplot techniques. Different types of visualization of high-dimensional data can be found in [17], which shows how to decide which two variables should be selected to investigate their data distribution on the basis of a 3D correlation matrix. Custom properties can be represented as clustered 3D spheres. Their radius is based on the cluster size and color based on a variable independent of clustering. Additionally, we can create cubes with sorted custom properties. The paper [18] describes a visualization method of spatially fixed multivariate volumetric data that focuses on two goals, i.e., maintaining the see-through property while providing easily interpretable color-coded scalar values. The authors find a compromise between these opposing goals by redistributing the opacity of a voxel by making part of it more opaque (making the other parts more transparent). The paper [19] draws attention to the fact that dealing with multivariate big data can mean an analysis of the same topic and even the same source datasets, there are different attributes being measured. The publication presents Robinson-Foulds tree distance metrics [20] and variation of information [21] as similarity metrics that can be adapted to compare dimensionally inconsistent multivariate data which is a subclass of the big data variety research. The authors of [22] stated that directly processing voluminous data is inefficient, but instead of juxtaposition of different visualization types, they proposed division of data into regular blocks using data dimensionality reduction techniques such as linear discriminant analysis (LDA) [23], principal component analysis (PCA) [24] and multidimensional scaling (MDS) [25]. The LDA is a dimensionality reduction method that attempts to find a linear combination of variables to categorize or separate two or more groups. MDS is used to visualize given high-dimensional data in a low-dimensional space by generating a configuration of the given data utilizing Euclidean low-dimensional space. PCA transforms the data into a new coordinate system by projecting the original data in the direction where the variance between data points is maximized. The principal components are derived from the eigenvectors of the covariance or correlation matrix of the dataset. The authors show a novel extension of the last method based on k-means block segmentation. Based on this we can state that although ways to describe multidimensional data and spatiotemporal automotive data related to single scenarios exists, there is a lack of ability to synthesize and summarize information that is important from the point of learning and testing automotive detectors and trackers. The desired methodology should describe not only the quantity of data but also their quality, taking into account the diversity of the whole dataset.

#### B. SIMILARITY OF OBJECTS IN TIME

Description of the similarity between tracks with time signatures can be also carried out simultaneously with the analysis of object cubature, velocity and other properties dynamically assigned to each position of the track generated by object. The most natural way to do this is the method described in the Davis challenge [26] but other options also exists. Examples are variations of the OSPA metric [27], [28], [29] which represents the possibility dividing information by its three components, each separately focusing on localization, cardinality and labeling errors. Weighted state correlation similarity [30], tracking difficulty [31], information theory-based metrics [32], shape association measures based on Minkowski distance [33] and other metrics ( [34], [35], [36]) are event evaluation metrics. The Jaccard index is also a tool that can be used for both local similarity and to summarize trajectories overlapping in time. It is widely used in various tasks that involve the evaluation metric. In [37], the authors use it to study the influence of small errors in ground truth on the tracking algorithm output. In similar cases, the precision and possibility of decomposing information are essential. The authors in [38] showed a combined performance estimation criterion by considering the veracity, real-time demand, and implementation on hardware of the tracking algorithm. In [39] we can find an algorithm that applies fuzzy rules to merge the detected bounding boxes...
into a unique cluster bounding box that covers a unique object. The abovementioned publications show that there are many existing methods to describe the similarity between bounding boxes, but none of them directly translate into the analysis of the trajectories created by the systems of these data. Therefore, we want to analyze methods directly aimed at trajectory analysis.

C. TRAJECTORY ANALYSIS

Motion trajectory analysis is frequently mentioned in the literature. One of the best approaches is presented in [40] and is called TrajAlign. This method presents a methodology for the alignment of trajectories by using their representative distance matrices and Needleman and Wunsch’s dynamic programming algorithm [41]. In [42], the authors optimized the trajectory distance calculation by employing the multiscale transform method and using vector fields on the manifold instead of the Euclidian distance. The work in [43] focuses on trajectory smoothing by clustering with Euler distance and two-dimensional wavelet transform performed on received groups. The reconstructed trajectory can be treated as denoised data. The authors of [44] defined a novel similarity measurement that considers spatiotemporal and semantic features simultaneously using the multidimensional semantic matrix of spatiotemporal trajectory and singular value decomposition to mine the essential characteristics of trajectories. Spatiotemporal matching of trajectories is also considered in [45]. Reference [46] provides a review of existing approaches to trajectory similarity functions and evaluates them in terms of computational cost and memory. The authors of [47] presented a similarity measurement between trajectories based on Pearson’s correlation coefficient and the coefficient of determination. In [48], the authors proposed a similarity measure for trajectories based on longest common subsequence (LCS) method and tested it on a database consisting of tracks of moving vehicles in cities. Reference [49] presents a robust-to-noise version of the LCS method. The authors of [50] show the pose normalization process for a trajectory matching framework that is translation, rotation, and scale invariant. All of these metrics can be useful for trajectory analysis but our aim is to present trajectory comparison metrics regarding the position on the occupancy grid and the time that the area on the grid was occupied by a given object. Additionally, none of those methods have the ability to compare systems of trajectories, which would provide whole scenario comparison functionality.

D. WASSERSTEIN METRIC CAPABILITIES

Reference [51] focuses on metric learning that satisfies the data geometry. The goal of the metric learning algorithm is to learn a metric that assigns a small distance to similar points and a relatively large distance to dissimilar points. Obtaining a metric for high dimensional data which is often difficult to even visualize, is difficult because the metric has to satisfy the data geometry. The authors propose the use of the Mahalanobis distance as the ground distance for the Wasserstein metric calculation and automatically learn the corresponding matrices by the alternative iterative approach. The author presents the ability of different methods to accurately classify information in traffic video databases, showing that their methods outperform others such as the K-Nearest Neighbors method using Euclidean distance or Support-Vector machine (SVM) models [52]. Publication [53] shows efficient algorithms for the calculation of Wasserstein Barycenter while simultaneously showing that it is a well suited tool for the generalization of sets of 2D pictures with various translations of content. Additionally, the author of [54] constructed a two-sample test based on the Wasserstein metric that is designed to detect structural breaks in data with complex geometries. The results published in [55] show that the Wasserstein distance performs well in tasks that require assessing the similarity of objects of similar types. Publication [55] describes a multivehicle tracking methodology based on the Wasserstein association metric. The described results show that the Wasserstein distance performs well in tasks that require assessing the similarity of objects of similar types and is robust to partial occlusion. The authors of [56] describe how to use the Wasserstein generative adversarial network (WGAN) to denoise and improve the quality of positron emission tomography imaging to reduce patient radiation exposure. Using the Wasserstein distance as a loss function solves the problem of excessive smoothness and loss of detailed information regarding classic image reconstruction approaches. Reference [57] proves that the Wasserstein metric can be used to critically score the quality of reconstructed magnetic resonance imaging. All this research provides arguments that this metric proves useful in tasks involving analyzing the of intensity of a given phenomenon. Trajectory data that is understood as the continuous occupation of the field around the ego vehicle generated by objects that appeared in test scenarios require the use of tools that are sensitive to changes in the intensity and localization of data. Based on this, we propose, a methodology that allows the analysis of large automotive data consisting of trajectories. It should allow the occupancy generated by them to be described with use of grid based versions of similarity metrics and Wasserstein distance. These tools make it possible to treat sets of trajectories as single objects without losing key information by unsuited generalization.

III. PRELIMINARIES

A. SENSORS

Cameras, radars, and lidars are currently essential elements for many automotive control systems. The raw data provided by these sensors constitute an input to the perception algorithms aimed at monitoring and interpreting what is happening outside the vehicle.

A camera sensor takes images of the vehicle’s surroundings that are further processed by AI based algorithms to detect and classify objects. The analysis is performed either in a camera (so-called smart sensor) based on a custom SoC (System-on-Chip) or within a centralized processing unit (a
so-called multi-domain controller) with either dedicated or shared computing resources. The output is high-level information about the detected objects (objects IDs, bounding box positions and sizes, etc.).

Radar sensors analyze the reflections of transmitted electromagnetic waves from different object types to find detection (single point reflection from an object). A single detection carries information not only about $x$, $y$, the position of the reflection point, but also many other interesting parameters, such as target speed (from the Doppler effect). In general, there are four levels of radar data abstraction [58]: raw analog to digital converter (ADC) values logged from antennas, data cube representations (CDC), detection levels, representation objects, or object tracking levels, where multiple detections are assigned to one object. The radar systems currently used are equipped with algorithms that, in addition to clustering, tracking, and providing range and velocity information, are also able to classify some objects.

Lidar works on the same principle as radar, but instead of microwaves, it uses light pulses. Due to its high resolution, lidar can be used to classify objects as well as to measure their positions. Unfortunately, lidar technology is still not mature enough (both in cost per unit and lifetime durability) to be mounted on a mass scale in vehicles. There are some examples of successful projects using solid-state lidar technology. In most cases, however, lidars are used as a reference sensors to enable precise labeling and positioning of objects around data logging vehicles. Nevertheless, the most popular devices due to quality vs. price factors are rotating devices rather than solid-state lidars.

It should be mentioned that the sensors provide data in different coordinate systems, for example, radars in the polar coordinate system, cameras in the planar coordinate system and, usually with different frame rates. Moreover, different sensors may have different accuracies, ranges, and angles of view.

To increase the performance of vehicle perception, car manufacturers and designers increase either the number of sensors mounted in the vehicle or increase their resolution.

B. AUTOMOTIVE PERCEPTION SYSTEMS
The perception system analyzes the data provided by a sensor to detect and classify objects existing in the sensor field of view. Video-based motion detection applications are still a challenging task, and reference [59] describes, a memory-efficient texture descriptor to overcome real-world problems. In works [60], [61] and [62], the authors consider this task from the point of background subtraction. The authors of [63] focus on various types of methods connected to human activity recognition. In addition, the system is responsible for determining the static and dynamic parameters of the detected objects. An example of moving object detection (MOD) problem is taken up with graph convolutional neural networks in reference [64]. Typical classes of objects that need to be distinguished by this and other similar perception systems are: passenger cars, pedestrians, bicycles, motorcycles, traffic signs, barriers, tunnels, etc. A typical set of object parameters includes: position - either in 2D or in 3D space, motion parameters such as velocity, acceleration, heading, and turn rate, parameters that determine the shape and size of the objects and the orientation of the shape. For some applications, the determination of the allowable (so-called freespace) or not allowable (so-called occupancy space) driving space is also required [65]. The perception system is exposed to many types of uncertainties. This means that with each object being visible and detectable in the sensor field of view, a certain level of probability of existence is associated. To increase the confidence of the detected objects, features from different types of sensors are fused together (Fig. 2). Recent reviews of tracking and object detection systems in automotive applications can be found in [66], [67] and [68].

C. SAFETY OF THE INTENDED FUNCTIONALITY
Verification of automotive perception systems is difficult to achieve. Such systems rely on sensing the external environment, which is uncertain and contains an infinite number of possible road scenarios. Due to the limitations imposed by the design and implementation of the perception algorithms, as well as the performance of the hardware platform executing these algorithms, there can be potentially hazardous behavior in the system. In addition, due to the high complexity of the system and the specificity of the operation of sensors and associated detection systems, the appearance of errors is inevitable. A properly designed active safety system should take into account the probability of these distortions. System architecture, redundancy, fuzzy logic elements, and fusion algorithms must be designed in a way that guarantees non-propagation and possibly quick suppression of emerging distortions. Proving that the applied methods guarantee the expected quality of operation of the entire system is a major problem. The norm ISO/PAS 21448:2019 Road vehicles — Safety of the intended functionality [69] provides examples...
of such an approach, including the inability of the perception function to correctly comprehend the situation and operate safely, as well as insufficient robustness of the function with respect to sensor input variations and diverse environmental conditions. The norm [69] also provides an example methodology for the definition and validation of an acceptable risk level needed to prove the safety of the system. The methodology includes: (1) partitioning of system failures; (2) modeling of hazardous events; (3) analysis of traffic statistics; and (4) definition of test scenarios. To support this methodology, the data collection should include relevant driving situations derived from the analysis of sensor limitations and feature-specific limitations, such as those described in [70] or [71], regarding LiDAR performance verification in different conditions. In such an analysis, a formal description of the representative dataset is an essential element.

**D. OBJECT TRACKING**

Object tracking is defined as the problem of extracting the motion of the object from a sequence of scans and estimating its trajectory. The scans can be in the form of images or clouds of radar or lidar points. The object state, in addition to its position in the 2D or 3D plane, usually includes additional variables related to its geometric and kinematic state. For example, the state vector representing the state of the detected object on a 2D plane can have the following form (Fig. 3):

$$q_k^{(j)} = \begin{bmatrix} x_k^{(j)} & y_k^{(j)} & v_k^{(j)} & \phi_k^{(j)} & \theta_k^{(j)} & l_k^{(j)} & w_k^{(j)} \end{bmatrix}^T,$$

(1)

where $q_k^{(j)} = q^{(j)}(kT)$, $q^{(j)}(t) = q^{(j)}(kT_s)$ for $t \in [kT_s, (k+1)T_s)$, $(x_k^{(j)}, y_k^{(j)})$ is an object position at time step $k$, and $j$ stands for the object’s identifier. The kinematic state of the object is determined by its velocity $v_k^{(j)}$, heading $\phi_k^{(j)}$ and turning angle $\theta_k^{(j)}$. The object’s shape is assumed to be a rectangle with length $l_k^{(j)}$ and width $w_k^{(j)}$ that may vary over time due to its visibility in the sensor field of view. $T_s$ denotes the sampling time.

**Definition 1**: A track $T^{(j)} : [kT_s, mT_s] \rightarrow \mathbb{X}^{(j)} \times \mathbb{Y}^{(j)}$, of object $j$ detected by the perception system is a motion trajectory extracted from a sequence of scans, that is, $T^{(j)}(t) = [x^{(j)}(t) y^{(j)}(t)]^T, t \in [kT_s, mT_s], k < m$, $x^{(j)}(kT_s) = x^{(j)}_k$, $x^{(j)}(mT_s) = x^{(j)}_m$, $y^{(j)}(kT_s) = y^{(j)}_k$, $y^{(j)}(mT_s) = y^{(j)}_m$, $(y^{(j)}(t) - y^{(j)}_n)(x^{(j)}_{n+1} - x^{(j)}_n) = (y^{(j)}_n - y^{(j)})(x^{(j)}(t) - x^{(j)}_n)$ for $n = k, \ldots, m - 1$.

**E. GRID SPACE**

$\mathbb{X} \subset \mathbb{R}^2$ denotes a state space consisting of all points on the 2D plane that are located in the sensor field of view. This means that any object $j$ in the vehicle’s surroundings with coordinates $[x_k^{(j)}, y_k^{(j)}]^T \in \mathbb{X}$ shall be detected by the perception system.

**FIGURE 3.** Illustration of the car state parameters

**Definition 2**: A grid cell $G_h(i)$ with size $h$ in space $\mathbb{R}^2$ is a set of points defined as follows:

$$G_h(i) = \left\{ [x y]^T \in \mathbb{R}^2 : \left\lfloor \frac{x}{h} \right\rfloor = i_1, \left\lfloor \frac{y}{h} \right\rfloor = i_2 \right\},$$

where $h = [h_x, h_y]^T$, $h_x > 0$, $h_y > 0$, $i = [i_1 i_2]^T \in \mathbb{Z}^2$, $\left\lfloor \frac{x}{h_x} \right\rfloor$ is the largest integer not greater than $x/h_x$, $\left\lfloor \frac{y}{h_y} \right\rfloor$ is the largest integer not greater than $y/h_y$, and $\mathbb{Z}$ stands for the set of integers.

Using this definition, state space $\mathbb{X}$ can be transformed into space $\mathbb{X}_h$, as indicated by the following definition.

**Definition 3**: Space $\mathbb{X}_h$ transformed from space $\mathbb{X}$ using grid $G_h$ is defined as follows:

$$\mathbb{X}_h = \left\{ i \in \mathbb{Z}^2 : \exists [x y]^T \in \mathbb{X} : [x y]^T \in G_h(i) \right\}.$$  

(2)

The number of elements in the transformed state space $\mathbb{X}_h$ depends on the parameter $h$ and can be chosen accordingly based on the system requirements. The number of states of the transformed state space $\mathbb{X}_h$ covered by track $T^{(j)}$ can be calculated as follows:

$$V_h(T^{(j)}) = \left\{ i \in \mathbb{X}_h : \exists t \in [kT_s, mT_s] : [x^{(j)}(t) y^{(j)}(t)] \in G_h(i) \right\}.$$  

(3)

As a scenario $S$, we understand a set of tracks $\{T^{(j)} : j \in J \subset \mathbb{N}^+\}$. For scenario $S$ a histogram of occupancy is as follows: $H : \mathbb{X}_h \rightarrow \mathbb{N}$, where

$$H(i) = \sum_{T^{(j)} \in S} |\{ t \in [kT_s, ..., mT_s] : T^{(j)}(t) \in G_h(i) \}|,$$

(4)

where $| \cdot |$ stands for the cardinality of the set.
It is a great challenge to properly log, store and use these product (forward collision warning camera, autopilot, etc.) needed for the development and verification of the final thousands of kilometers of driving to gather the material driving (ADAS, AD) project requires logging hundreds of A typical advanced driver-assistance system or autonomous

\[ \mu(i) = \sum_{j \in \mathcal{X}_h} H(j), i \in \mathcal{X}_h. \]  

(5)

The most convenient method for visualizing histograms and two-dimensional distributions is heatmap. A heatmap is a data visualization technique that shows the magnitude of a phenomenon using a variable color intensity that changes with increasing value of the function, assigning a value to a two-dimensional vector.

\[ \text{Heatmap of grid occupancy based on cumulative histograms of occupancy for all scenarios in database. Radar data provide information about all the tracks that appeared around the host vehicle. The position of the host is constantly in the center, and its heading is set to the right side of the image.} \]

\[ \text{FIGURE 4. Heatmap of grid occupancy based on cumulative histograms of occupancy for all scenarios in database. Radar data provide information about all the tracks that appeared around the host vehicle. The position of the host is constantly in the center, and its heading is set to the right side of the image.} \]

F. DISTANCE IN THE SPACE OF SCENARIOS

The distribution of occupancy (Definition 2) is a discrete probability density function. \((\mathcal{X}_h, d)\) is a metric space, where \(d\) is the induced Euclidean metric.

Definition 5: For any two distributions of occupancy \(\mu\) and \(\nu\) we can calculate the Wasserstein distance between them as follows:

\[ W(\mu, \nu) := \left( \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{\mathcal{X}_h \times \mathcal{X}_h} d(x, y)^2 d\gamma(x, y) \right)^{\frac{1}{2}}, \]

(6)

where \(\Gamma(\mu, \nu)\) is a set of all probability measures defined on space \(\mathcal{X}_h \times \mathcal{X}_h\) such that \(\mu\) and \(\nu\) are their marginals.

G. AUTOMOTIVE DATASETS

A typical advanced driver-assistance system or autonomous driving (ADAS, AD) project requires logging hundreds of thousands of kilometers of driving to gather the material needed for the development and verification of the final product (forward collision warning camera, autopilot, etc.). It is a great challenge to properly log, store and use these data due to the very large number of files and required disk space (often exceeding tens of petabytes). Recorded material is usually considered IP and is not revealed to anyone outside the project team.

Some car manufacturers and universities have published their own automotive databases, which can be used by researchers to train algorithms and benchmark their performance [72]. In addition to the various raw sensor data, the databases contain object annotations in different formats (sometimes just 2D bounding box on camera images, sometimes a 3D bounding box of objects in world coordinates). One of the most well-known database is KITTI [73], which was released by Toyota and Karlsruhe Institute of Technology (7481 annotated frames + 7518 unlabeled frames). Another database was released by the Audi Autonomous Driving Dataset (12499 frames with 3D bounding box annotations + 390000 unlabeled frames). There are also many other datasets, such as the Waymo Open Dataset [74], Berkeley DeepDrive, BDD100k [75], Baidu ApolloScapes [76], and Cityscape Dataset [77], etc. The dataset that we decided to use to test the methodology described in this article is the nuScenes dataset [78] provided by Motional (Hyundai Aptiv joint venture). The full dataset includes approximately 1.4 M camera images, 390 k Lidar sweeps, 1.4 M radar sweeps, and 1.4 M object bounding boxes in 40 k frames. It was chosen mainly because it contains a large number of annotated scenarios from two different countries (left- and right-hand traffic).

In our work, we narrow the meaning of an automotive database, which is the collection of data logged by a vehicle with various radar, lidar, or camera sensor setups.

We understand a dataset (database) as a set of scenarios (recording chunks providing a set of tracks) that were ground truthed (either by manual or automated labeling) so that at least the unique id and 2D positions of objects (vehicles, bicycles, pedestrians etc.) in relationvto the recording vehicle is known. Databases without annotation cannot be used for algorithm training/performance computation and as such are almost useless.

The unique IDs and positions in every frame allow the creation of a movement trajectory for every object instance (in the recording vehicle coordinate system). If the database also contains object types (classes), it is possible to further subdivide the trajectories, e.g., trajectories for vehicles and trajectories for pedestrians.

IV. TRACK SIMILARITY DESCRIPTION

In this section, we provide several metrics in the form of modified distance functions and similarity measures that can be used to compare two tracks \(T^{(i)}\) and \(T^{(j)}\). State-of-the-art approaches such as those described in [79] and others researched in section II make use of the direct positions and properties of tracked objects. Our approach is focused on analyzing the occupancy that objects generate on a grid around car, and calculations are performed on this basis. These tracks can have different lengths and can belong to the same or different classes of detected objects. If the distance is less

\[ \text{VOLUME 4, 2016} \]

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
than a defined threshold, then the tracks can be considered equivalent; otherwise, they are different from each other.

A. TRAJECTORY COMPARISON EXPERIMENT

The first definition is based on the Minkowski distance between two points in the norm vector space.

Definition 6: The distance between two tracks \( T^{(i)} \) and \( T^{(j)} \) in the \( L^p \)-norm sense can be defined using the following formula:

\[
D_{L^p}(T^{(i)}, T^{(j)}) = \left( \sum_{k} \left( |x_k^{(i)} - x_k^{(j)}|^p + |y_k^{(i)} - y_k^{(j)}|^p \right) \right)^{\frac{1}{p}},
\]

for all \( k \) such that \( x_k^{(i)} \in X^{(i)} \cap X^{(j)} \) and \( y_k^{(i)} \in Y^{(i)} \cap Y^{(j)} \). Using Definition 6, we can manipulate the value of \( p \) and calculate the distance using different means, as follows: if \( p = 1 \), then the distance is similar to the Manhattan distance; if \( p = 2 \), then the distance formula reflects the Euclidean distance metric; and if \( p = \infty \), the distance is based on the Chebyshev maximum metric. In Table 1, we calculated the distances between examples of tracks presented in Fig. 5 according to this definition with \( p = 2 \). Definition 6 can also be extended to the function space \( L^p \).

Definition 7: The distance between two tracks \( T^{(i)} \) and \( T^{(j)} \) in the \( L^p \)-norm sense can be defined using the following formula:

\[
D_{L^p}(T^{(i)}, T^{(j)}) = \left( \int \int_{X^{(i)} \cap X^{(j)}} |T^{(i)} - T^{(j)}|^p dx dy \right)^{\frac{1}{p}}.
\]

The next definition is based on the Jaccard similarity index, which is a measure of similarity for two sets of data to determine which elements are shared and which are distinct. The definition also utilizes the concept of the transformed state space introduced in the previous section.

Definition 8: The similarity measure between two tracks \( T^{(i)} \) and \( T^{(j)} \) in the Jaccard sense can be defined using the following formula:

\[
D_J(T^{(i)}, T^{(j)}) = \frac{|V_h(T^{(i)}) \cap V_h(T^{(j)})|}{|V_h(T^{(i)}) \cup V_h(T^{(j)})|},
\]

where \( |X| \) stands here for the cardinality of set \( X \). Examples of comparisons with this measure are in Table 2.

We can also apply a similar scheme as in the calculation of the Hamming distance, which is a metric for comparing two binary data strings. As the Hamming distance for two strings of equal length is the number of bit positions in which two bits are different, the distance between two tracks can be the number of grid cells that are not covered by two tracks.

Definition 9: The distance between two tracks \( T^{(i)} \) and \( T^{(j)} \) in the Hamming sense can be defined as follows:

\[
D_H(T^{(i)}, T^{(j)}) = \frac{|V_h(T^{(i)}) \cup V_h(T^{(j)})| - |V_h(T^{(i)}) \cap V_h(T^{(j)})|}{|X|},
\]

where \( |X| \) stands here for the cardinality of set \( X \). Examples of comparisons with this metric are shown in Table 3.

B. APPLICATION

As the results show, those metrics are focused on indicating whether the trajectories occupy similar positions in the field around the car. This can be applied to search for similar maneuvers on the road in different scenarios to create unique and diverse synthetic scenarios. Those and other trajectory metrics can also be used to describe the diversity of scenarios. Although similarity measures that take into account the shape
of trajectories while ignoring their relative positions exists, we will focus on the diversity description of scenarios and datasets in terms of grid coverage and intensity of grid occupancy.

V. SCENARIO SIMILARITY DESCRIPTION

In this section, we propose a concept to characterize scenarios using object tracks projected on a grid space.

A. GRID COVERAGE APPROACH

A metric that can represent the degree how a set of trajectories \( T_{\text{set}} = \{ T^{(j_1)}, T^{(j_2)}, \ldots, T^{(j_N)} \} \) belonging to \( N \) detected objects corresponds to the whole grid in the field of view can be defined as follows:

\[
C_h(T_{\text{set}}) = \frac{\bigcup_{k=1}^{N} V_h(T^{(j_k)})}{|X_h|}.
\]  

(8)

The proposed measure is defined using a partition of the system state space. The partition forms a rectangular grid and, roughly speaking, metric (8) is defined by the number of grid boxes covered by the tracks. The boundedness of the space \( X \) assures that the numerator and denominator in Formula (8) are finite numbers. A similar concept was used in [80] to define a test coverage measure for continuous-time software systems.

In the example presented in Fig. 6 one track was recorded. This track covers 27 cell grids out of 415 in the whole field of view, which gives \( C_h(T_{\text{set}}) = 0.065 \) (6.5%). Full coverage of the sensor field of view by the recorded tracks is illustrated in Fig. 7. An acceptable level of coverage can be achieved by selecting tracks according to Algorithm 1. Fig. 8 illustrates the tracks of the radar system consisting of several radar sensors that deliver a 360-degree view around the vehicle. The tracks were recorded in a real experiment and then selected according to Algorithm 1. Out of 93 tracks recorded in that scenario Algorithm 1 selects 72 tracks that cover the equivalent area. Additionally, out of 4443 recorded tracks that create the occupancy heatmap in Fig. 4 Algorithm 1 selects only 2391 tracks that cover the equivalent area.

Algorithm 1 Selection of tracks to guarantee an acceptable level of coverage of the sensor field of view

\begin{itemize}
  \item \textbf{Inputs:} Size of the grid \( h = [h_1, h_2]^T \), \( h_1, h_2 > 0 \), acceptable coverage level \( \delta \in (0,1] \), transformed state space \( X_h \) determined by the elements, \( G_h(i) \), set of available tracks \( \{ T^{(j_k)} \} \).
  \item \textbf{Outputs:} set of tracks \( T_{\text{set}} = \{ T^{(j_1)}, T^{(j_2)}, \ldots, T^{(j_N)} \} \) such that \( C_h(T_{\text{set}}) \geq \delta \)
  \begin{enumerate}
    \item \( T_{\text{set}} := \emptyset \), \( V_h(T_{\text{set}}) := \emptyset \), \( C_h(T_{\text{set}}) = 0 \), \( k := 0 \)
    \item \textbf{while} \( C_h(T_{\text{set}}) < \delta \) \textbf{do}
    \begin{enumerate}
      \item Find new track \( T^{(j_k)} \) from the available tracks that cover at least one yet do not cover the cell grid from the set \( X_h \setminus V_h(T_{\text{set}}) \), \( k := j_k \)
      \item if \( k == \emptyset \) then
        \begin{enumerate}
          \item return \( T_{\text{set}} \)
        \end{enumerate}
      \end{enumerate}
    \end{enumerate}
    \item \textbf{end if}
    \item \( T_{\text{set}} := T_{\text{set}} \cup T^{(k)} \), \( k := \emptyset \)
  \end{enumerate}
  \item \textbf{return} \( T_{\text{set}} \).
\end{itemize}

B. SCENARIO COMPARISON EXPERIMENT

Based on the nuScenes database [81], we created occupancy distribution for a sample of 10 scenarios chosen from this database. Their occupancy heatmaps based on the vehicle trajectories are presented in Fig. 9. We calculated the Wasserstein distance between each pair from this set of scenes (Table I) to compare the values of this metric with the desired sense of similarity.

The calculation of the Wasserstein metric in the case of two distributions of occupancy \( \mu \) and \( \nu \) can be considered the
The largest value, i.e., 28.7, is achieved for the comparison of scenarios 60 and 539. In scenario 60, most vehicles appeared in front of the host vehicle, while for scenario 539, most vehicles appear behind the host vehicle. In scenarios 87 and 202, occupancy is equally distributed around the host vehicle, and the value for this comparison is relatively low, i.e., 5.4. Scenarios 273 and 274 are based on data collected at intersections. The Wasserstein distance between them is equal to 4.9. Crossroad examples are almost twice as distant as those from the scenario that comes from a straight road (Scenario 305). This scenario is also significantly distanced from the data based on a road with perpendicular movement in relation to the host (Scenario 169) with a Wasserstein distance equal to 15.6. Scenarios with a form similar to that achieved in examples 256 and 403 arise when the host makes major turns. The distance between the given examples is equal to 4.9. Based on this, we can state that this measure representatively describes the similarity between scenarios in terms of the intensity of occupancy and the wider context in which they are recorded.

VI. DATASET SIMILARITY DESCRIPTION

Datasets that are used in automotive systems are specific in terms of their size, properties, and diversity. The formal description should therefore be in the form of a set of parameters that describe these features. Moreover, the description should include both static and dynamic aspects that charac-

### TABLE 4. Wasserstein distances between scenarios from the nuScenes database.

| Scene | 60 | 87 | 169 | 202 | 256 | 273 | 274 | 305 | 403 | 539 |
|-------|----|----|-----|-----|-----|-----|-----|-----|-----|-----|
| 60    | 0  | 15.3 | 14.0 | 16.5 | 21.1 | 14.1 | 15.6 | 18.2 | 20.0 | 28.7 |
| 87    | 15.3 | 0  | 11.2 | 5.4  | 9.3  | 10.2 | 11.7 | 11.5 | 9.5  | 17.4 |
| 169   | 14.0 | 11.2 | 0   | 10.5 | 15.7 | 12.2 | 13.7 | 15.6 | 16.3 | 24.8 |
| 202   | 16.5 | 5.4 | 10.5 | 0   | 11.1 | 11.5 | 12.9 | 11.3 | 11.9 | 17.5 |
| 256   | 21.1 | 9.3 | 15.7 | 11.1 | 0   | 12.0 | 12.8 | 12.1 | 4.9  | 14.1 |
| 273   | 14.1 | 10.2 | 12.2 | 11.5 | 12.0 | 0   | 4.9  | 8.8  | 10.7 | 20.3 |
| 274   | 15.6 | 11.7 | 13.7 | 12.9 | 12.8 | 4.9 | 0   | 8.3  | 10.3 | 21.4 |
| 305   | 18.2 | 11.5 | 13.6 | 11.3 | 12.1 | 8.8 | 8.3 | 0   | 10.4 | 19.0 |
| 403   | 20.0 | 9.5 | 16.3 | 11.9 | 4.9 | 10.7 | 10.8 | 10.4 | 0   | 15.9 |
| 539   | 28.7 | 17.4 | 24.8 | 17.5 | 14.1 | 20.3 | 21.4 | 19.0 | 15.9 | 0   |

The value can be calculated from the following formula:

$$W(\mu, \nu) = \sqrt{\sum_{k=1}^{Xhl} \sum_{j=1}^{Xhl} f^*(k, j)d(k, j)}.$$  (14)

minimum cost flow problem (MCFP) and can be written as a linear program. For any given Xhl, we can define bijective function \( \eta : M \rightarrow Xhl \), where \( M = \{1, \ldots, |Xhl|\} \). To calculate the Wasserstein distance \( W(\mu, \nu) \) we have to find the optimal \( f^* : M \times M \rightarrow [0, 1] \) that minimizes the sum

$$\sum_{k=1}^{Xhl} \sum_{j=1}^{Xhl} f(k, j)d(k, j),$$  (9)

where \( d(k, j) \) is squared Euclidean distance between vectors \( \eta(k) \) and \( \eta(j) \).

Constrains for function \( f \) are as follows

\[ \forall k, j \in M : 0 \leq f(k, j) \leq 1, \]
\[ \forall k \in M : \sum_{j=1}^{Xhl} f(k, j) = \mu(\eta(k)), \]
\[ \forall j \in M : \sum_{k=1}^{Xhl} f(k, j) = \nu(\eta(j)), \]
\[ \sum_{k=1}^{Xhl} \sum_{j=1}^{Xhl} f(k, j) = 1. \]  (13)

To solve this setup, we can apply the network simplex method described in [82] or [83]. Then, the Wasserstein distance value can be calculated from the following formula:

$$W(\mu, \nu) = \sqrt{\sum_{k=1}^{Xhl} \sum_{j=1}^{Xhl} f^*(k, j)d(k, j)}.$$  (14)

FIGURE 9. Examples of occupancy heatmaps based on scenarios from the nuScenes database. The Wasserstein distances between those scenes are presented in Table 4.
TABLE 5. Example of dataset collection specification

| Time of Day     | Percentage |
|-----------------|------------|
| Day             | 50%        |
| Night           | 35%        |
| Dusk            | 15%        |

| Vehicle Speed       | Percentage |
|---------------------|------------|
| Speed [m/s]         | Speed [km/h]|
| 0 – 25              | 0 – 40     | 60%        |
| 26 – 50             | 41 – 80    | 40%        |
| > 50                | > 80       | 0%         |

| Type                | Percentage |
|---------------------|------------|
| Dry/Clear sky       | 65%        |
| Rain                | 10%        |
| Fog                 | 5%         |
| Snow                | 5%         |
| Overcast            | 10%        |
| Heavy rain          | 5%         |

The optical path for the automotive vision system consists of a full set of hardware starting from the windshield, through antireflective coatings, lenses, light detectors, and gain control to the serializer. RWUP data collected for a certain optical path are valid only for this specific setup. Eventual reuse of RWUP for different optical paths is possible for indication purposes only. To reduce the cost of data collection and to reduce time-to-market, the test scenarios selected by the presented algorithms can be used as a seed of the corresponding virtual scenario database. An additional benefit of this approach is that the same seed is sufficient to generate variations in RWUP data for a given distribution, for example, the same set of scenes rendered for different countries, the same set of scenes rendered for different sun heights (dazzle simulation), and the same set of scenes rendered for different weather conditions and light illumination. With this approach, many of the SOTIF required scenarios may be virtually generated and virtually tested with relatively little human effort. Algorithm 2 illustrates an approach for selecting representative test scenarios.

Algorithm 2 Selection of representative test scenarios

Inputs: Size of the grid \( h = [h_1, h_2]^T \), \( h_1, h_2 > 0 \), transformed state space \( \mathcal{X}_h \) determined by the elements, histogram \( H: \mathcal{X}_h \to \mathbb{N} \), set of available test scenarios \( \mathcal{T}_{case} \)

Outputs: representative set of test scenarios \( T_{suite} = \{T_{case,1}, T_{case,2}, \ldots, T_{case,m}\} \) containing tracks that span the space \( \mathcal{X}_h \) according to the histogram \( H \)

1. \( T_{suite} := \emptyset \), \( \forall i \in \mathcal{X}_h \), \( h_{i,j} := 0 \), \( k := 0 \)
2. while \( \exists i \in \mathcal{X}_h \), \( h_{i,j} < h_{i,j} \) do
3. Find a new test scenario \( T_{case}^{(j_k)} \) from available set that contains at least one track which crosses a cell \( i \in \mathcal{X}_h \) such that \( h_{i,j} < h_{i,j} \) and is not similar (in the sense of the track similarity measure) to any of the tracks from selected test scenarios, \( k := j_k \)
4. if \( k = \emptyset \) then
5. return \( T_{suite} \)
6. end if
7. Increment values of \( h_{i,j} \) for all \( i \in \mathcal{X}_h \) crossed by tracks from new test scenario \( T_{case}^{(j_k)} \)
8. \( T_{suite} := T_{suite} \cup T_{case}^{(j_k)} \), \( k := 0 \)
9. end while
10. return \( T_{suite} \)

FIGURE 10. Example of the modeling of hazardous events

It is possible to select a set of trajectories for scenarios that represent similar levels of grid occupancy. Below, we propose a comparison methodology for two scenario datasets \( S \) and \( R \) in terms of the intensity of grid occupancy in the scenarios they contain. Each database is a set of scenarios \( \{S_1, \ldots, S_n\} \) and \( \{R_1, \ldots, R_m\} \). Every scenario consists of at least one trajectory. Our goal is to determine whether database \( R \) is a significant extension of database \( S \) in terms of the intensity of occupancy of grid \( \mathcal{X}_h \) generated by every set of trajectories.
This result should be an indication of the most influential scenarios in $R$ and a formal description of the diversity introduced by base $R$ in relation to base $S$.

### A. CUMULATIVE HISTOGRAM COMPARISON

We can also calculate the Wasserstein distance between cumulative histograms of occupancy for sets of scenarios. To do so, we divided nuScenes into two databases. The first one includes the scenarios that were recorded in the USA (467), and the second is made of scenarios recorded in Singapore (383). Then, we created cumulative histograms for both databases separately for pedestrian and vehicle trajectories. The Wasserstein distance between the cumulative occupancy is equal to 3.0989 in the pedestrian case. For vehicle trajectory comparison, this metric takes a value of 4.0773.

**Figure 11.** Heatmap of grid occupancy based on cumulative histograms of occupancy for pedestrian and vehicle trajectories in the nuScenes division based on location.

### B. SCENARIO CLUSTERING

Below we present another, more complex method of dataset comparison. This approach can provide a more detailed description of the relation between two large datasets and makes use of the Wasserstein distance. Thanks to this metric, we can cluster scenarios from databases with the use of an algorithm similar to the one described in [84]. The difference is that we do not cluster one-dimensional quality distributions but two-dimensional distributions of occupancy, and we use the Dunn Index [85] to choose optimal number of groups. Let us assume that we cluster database $S$. As an output from the clustering algorithm, we obtain clusters of scenarios $C_1, C_2, \ldots, C_f$ and the corresponding centroids $\xi_1, \xi_2, \ldots, \xi_f$. Each centroid $\xi_i$ is a Wasserstein barycenter of scenes from cluster $C_i$. In other words centroid $\xi_i$ is a probability distribution on space $X_h$ that averages all occupancy distributions from cluster $C_i$ in a Wasserstein metric sense. The next step is clusterization of the sum of scenes from both databases $S \cup R$. As a result, we obtain clusters of scenarios $D_1, D_2, \ldots, D_f$ and corresponding centroids (Wasserstein barycenters) of those clusters $\pi_1, \pi_2, \ldots, \pi_f$. The difference between those two outputs from the clustering algorithm is the information that database $R$ brings to database $S$. Now, we formally describe this difference; to do so, we have to define the assignment of clusters from the second output to the clusters from the first run of the clustering algorithm.

**Definition 10:** Let $I_e = \{1, \ldots, e\}$ and $I_f = \{1, \ldots, f\}$. We define function $F : I_e \rightarrow I_f$ such that:

$$F(i) = \arg \min_{k \in J} W(\xi_i, \pi_k) \tag{15}$$

where $J = \{j \in I_f : C_i \cap D_j \neq \emptyset\}$.

### C. DIVERSITY COEFFICIENTS

Based on our algorithm from the previous subsection, we define the diversity coefficients of database $R$ in relation to database $S$.

**Definition 11:** Enrichment factor of existing clusters

$$w_1 = \sum_{i=1}^{e} \frac{W(\xi_i, \pi_{F(i)})}{e} \tag{16}$$

**Definition 12:** Cluster collection enrichment factors

$$w_2 = \frac{|f - e|}{e}, \quad w'_2 = \frac{\sum_{i \in I_f - I_e} |D_i|}{|f - e|}, \quad w''_2 = \frac{\sum_{i \in I_f - I_e} \sum_{j \in I_e} \min W(\xi_j, \pi_i)}{|f - e|} \tag{18}$$

**Definition 13:** Mean internal inertia change

$$w_3 = \left| \sum_{i \in I_e} \sum_{\sigma \in C_i} \frac{W(\xi_i, \sigma)^2}{|S|} - \sum_{i \in I_e} \sum_{\sigma \in C_i \cup D_{F(i)}} \frac{W(\pi_{F(i)}, \sigma)^2}{|S \cup R|} \right| \tag{20}$$
D. DATASET COMPARISON EXPERIMENT

Finally, we would like to show an application of this methodology. Our goal is to describe in the terms presented before how a set of 50 scenes from Singapore extends the dataset created from 150 scenes from the USA (both samples are taken from the nuScenes dataset). The clustering results and diversity coefficients are stored in Table 6. Before adding the Singapore scenarios to our database, 150 USA scenarios are decomposed into 3 groups. The new system obtained on the basis of clustering consists of only one additional group but increases the inertia of this system by 34.2%. The mean shift of centroids \( w_1 \) is equal to 4.7, which suggests that the change in existing centroids is similar to the difference between scenarios of the same type (Table 4) and can be interpreted as low. However, the fact that mean size of new groups \( w_2' \) is larger than the sample of Singapore scenarios indicates that the new scenes expand the set of scenes that occurred in the US but were not sufficiently represented to create a separate cluster. The low internal inertia \( w_3' \) with regard to the new final inertia suggests that the new cluster is still highly distorted and has the potential to further decay.

**TABLE 6.** Diversity coefficients for fragments of the nuScenes dataset in mutual relation

| Datasets          | USA     | USA + Singapore |
|-------------------|---------|-----------------|
| Scenarios         | 150     | 150 + 50        |
| Clusters          | 3       | 4               |
| Final Inertia     | 10932.19| 14673.15        |
| \( w_1 \)         | 4.699   |                 |
| \( w_2 \)         | 0.333   |                 |
| \( w_2' \)        | 76.0    |                 |
| \( w_3' \)        | 8.013   |                 |
| \( w_3 \)         | 10921.349|               |

VII. CONCLUSIONS

In this paper, an efficient method to formally characterize automotive datasets used for perception-based system development and verification is presented. The method primarily focuses on datasets used in automotive applications, as these datasets are specific in terms of dynamic aspects that should not be omitted. However, the presented approach can be extended to other types of large datasets used in other robotic applications. By characterizing the datasets in terms of motion trajectories of detected objects appearing in the sensor field of view during vehicle movement, we can compare and evaluate different datasets and relate them to the performance of perception algorithm. Such analytical formalism helps in understanding the data and may also improve the algorithm development process and consequently reduce effort, cost, and time, as in the automotive industry, data collection, testing, and verification activities consume the majority of the project effort. Moreover, a dataset described using mathematical notation, can serve as a base to define a dataset that can accurately represent the vehicle’s entire surroundings in reality with a defined confidence level. Such a representative dataset can be used to validate control and perception algorithms for autonomous vehicles with the support of a strong mathematical explanation. This problem remains an issue to date, and no good solution has been proposed thus far.

REFERENCES

[1] SAE International, Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles, Standard J3016, 201806, www.sae.org/standards/content/J3016_201806, 2018.
[2] S. Royo and M. Ballestra-Garcia, “An overview of lidar imaging systems for autonomous vehicles,” Applied Sciences, vol. 9, no. 19, 2019. [Online]. Available: [https://www.mdpi.com/2073-8994/9/19/4093](https://www.mdpi.com/2073-8994/9/19/4093)
[3] M. Ujičič, D. Hurych, P. Krizek, and S. Yogamani, “Challenges in designing datasets and validation for autonomous driving,” p. 2019.
[4] “Evaluation of active driving assistance systems,” American Automobile Association [https://newsroom.saa.com/asset/report-fact-sheet-evaluating-active-driver-assistance-systems/](https://newsroom.saa.com/asset/report-fact-sheet-evaluating-active-driver-assistance-systems/), 2020.
[5] J. Janai, F. Güney, A. Behl, and A. Geiger, 2020.
[6] R. Vilalta and Y. Drissi, “A perspective view and survey of meta-learning,” Artificial Intelligence Review, vol. 18, no. 2, pp. 77–95, 2002.
[7] P. Skrucha, R. Dhinside, K. Krogst, P. Markiewicz, D. Sasin, and M. Rozewicz, “The simulation strategy and its realization in the development process of active safety and advanced driver assistance systems,” SAE Technical Paper 2015-01-140, 2015.
[8] Y. Sohn, “Meta-analysis of classification algorithms for pattern recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 11, pp. 1131–1144, 1999.
[9] A. Kalousis, J. Gama, and M. Hilario, “On data and algorithms: Understanding inductive performance,” Machine Learning, vol. 54, no. 3, pp. 275–312, 2004.
[10] N. Macia, E. Bernado-Mansilla, A. Orriols-Puig, and T. Ho, “Learning excellence biased by data set selection: A case for data characterisation and artificial data sets,” Pattern Recognition, vol. 46, no. 3, pp. 1054–1066, 2013.
[11] L. Morán-Fernández, V. Bolón-Canedo, and A. Alonso-Betanzos, “Can classification performance be predicted by complexity measures? a study using microarray data,” Knowledge and Information Systems, vol. 51, no. 3, pp. 1067–1090, 2017.
[12] Z. Yu, W. Xiong, L. Eckhout, Z. Bei, A. Mendelson, and C. Xu, “Mia: Metric importance analysis for big data workload characterization,” IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 6, pp. 1371–1384, 2018.
[13] R. Akase and Y. Okada, “Web based time-tunnel: An interactive multidimensional data visualization tool using genetic algorithm,” in 2015 19th International Conference on Information Visualisation, 2015, pp. 95–100.
[14] P. Voland, “Spatio-temporal visualization of automotive sensor data — a conceptual and implementation framework,” in 2017 IEEE Region 10 Symposium (TENSYMP), 2017, pp. 1–5.
[15] J. Heinrich and D. Weiskopf, “Parallel coordinates for multidimensional data visualization: Basic concepts,” Computing in Science Engineering, vol. 17, no. 3, pp. 70–76, 2015.
[16] A. Watanabe, T. Itoh, K. Chiba, and M. Kanazaki, “A scatterplots selection technique for multi-dimensional data visualization combining with parallel coordinate plots,” in 2017 21st International Conference Information Visualisation (IV), 2017, pp. 78–83.
[17] R. Smolka and O. Heisk, “Interactive visualization of high dimensional marketing data in the financial industry,” in Ninth International Conference on Information Visualisation (IV’05), 2005, pp. 814–817.
[18] R. Khlebnikov, B. Kainz, M. Steinberger, and D. Schmalstieg, “Noise-based volume rendering for the visualization of multivariate volumetric data,” IEEE Transactions on Visualization and Computer Graphics, vol. 19, no. 12, pp. 2926–2935, 2013.
[19] N. Micic, D. Neagu, D. Torgunov, and F. Campean, “Exploring methods for comparing similarity of dimensionally inconsistent multivariate numerical data,” in 2018 IEEE 20th International Conference on High Performance Computing and Communications, 2018, pp. 1528–1535.
[20] D. Robinson and L. Foulds, “Comparison of phylogenetic trees,” Mathematical Biosciences, vol. 53, no. 1, pp. 131–147, 1981.
S. B. Needleman and C. D. Wunsch, “A general method applicable to the comparison of bases sequences,” Journal of Molecular Biology, vol. 48, no. 3, pp. 443–453, 1970.

R. A. Fisher, “The use of multiple measurements in taxonomic problems,” 1936.

J. Olliffe, “Principal Component Analysis.” Springer Verlag, 1986.

B. Ristic, B. Vo, and D. Clark, “Performance evaluation of multi-target tracking using the osca metric,” in 2013 13th International Conference on Information Fusion, 2010, pp. 1–7.

B. Ristic, B. Vo, D. Clark, and B. Vo, “A metric for performance evaluation of multi-target tracking algorithms,” IEEE Transactions on Signal Processing, vol. 59, no. 7, pp. 3452–3457, 2011.

X. Shi, F. Yang, F. Tong, and H. Lian, “A comprehensive performance metric for evaluation of multi-target tracking algorithms,” in 2017 3rd International Conference on Information Management (ICIM), 2017, pp. 373–377.

A. Tokta and A. K. Hocaoglu, “A track to track association algorithm based on weighted state correlation similarity,” in 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018, pp. 1–4.

P. Pan, F. Porikli, and D. Schonfeld, “A new method for tracking performance evaluation based on a reflective model and perturbation analysis,” in 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 3529–3532.

E. Loutas, N. Nikolaidis, and I. Pitas, “Evaluation of tracking reliability metrics based on information theory and normalized correlation,” in Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPRA 2004., vol. 4, 2004, pp. 653–656 Vol.4.

I. Batyrshin, “Constructing time series shape association measures: Minkowski distance and data standardization,” in 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, 2013, pp. 204–212.

S. Mori, K. Chang, and C. Chong, “Comparison of track fusion rules and track association metrics,” in 2012 15th International Conference on Information Fusion, 2012, pp. 1996–2003.

D. Roth, E. Koller-Meier, D. Rowe, T. B. Moeslund, and L. Van Gool, “Event-based tracking and evaluation metrics,” in 2008 IEEE Workshop on Motion and video Computing, 2008, pp. 1–8.

T. Vu and R. Evans, “A new performance metric for multiple target tracking based on optimal subpattern assignment,” in 17th International Conference on Information Fusion (FUSION), 2014, pp. 1–8.

G. Liu, S. Liu, M. Lu, and Z. Pan, “Effects of improper ground truth on target tracking performance evaluation in benchmark,” in 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2017, pp. 261–266.

Y. Fang and X. Fan, “Performance evaluation for ir small target tracking algorithm,” in 2011 Sixth International Conference on Image and Graphics, 2011, pp. 749–753.

J. Tu, A. Del Amo, Y. Xu, L. Guari, M. Chang, and T. Sebastian, “A fuzzy bounding box merging technique for moving object detection,” in 2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), 2012, pp. 1–6.

Z. Ang, K. Sim, and W.S. Ng, “Traj align: A method for precise matching of 3-d trajectories,” in 2010 20th International Conference on Pattern Recognition, 2010, pp. 3818–3821.

S. B. Needelman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” J. Mol. Biol., vol. 48, pp. 443–453, 1970.

H. Mi and M. Wang, “Trajectory similarity metric based on vector fields on manifold and multi-scale space partition,” in 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016, pp. 1600–1604.

X. Li, Z. Liu, Z. Liu, R. W. Liu, and Z. Feng, “Spatio-temporal vessel trajectory smoothing based on trajectory similarity and two-dimensional wavelet transform,” in 2019 5th International Conference on Transportation Information and Safety (ICTIS), 2019, pp. 1500–1505.

C. Jiang, Y. Zhou, and C. Zhang, “Similarity measurement of spatiotemporal trajectories considering semantic features,” in 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), 2021, pp. 423–426.

B. Chen, Y. Liu, and W. Shi, “Research on trajectory similarity matching model based on spatiotemporal trajectory big data,” in 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), vol. 1, 2020, pp. 2208–2215.

M. Riyad, N. Mustapha, and D. Riyad, “Review of trajectories similar measures in mining algorithms,” in 2018 Al-Mansour International Conference on New Trends in Computing, Communication, and Information Technology (NTCIT), 2018, pp. 36–41.

J. Urain and J. Peters, “Generalized multiple correlation coefficient as a similarity measurement between trajectories,” in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 1363–1369.

H. Fashandi and A. M. E. Moghaddam, “A new rotation invariant similarity measure for trajectories,” in 2005 International Symposium on Computational Intelligence in Robotics and Automation, 2005, pp. 631–634.

M. V. Markos, G. Kollie, and D. Gunopulos, “Discovering similar multidimensional trajectories,” in Proceedings 18th International Conference on Data Engineering, 2002, pp. 673–684.

A. Croitoru, P. Agouris, and A. Stefanidis, “3d trajectory matching by pose normalization,” GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, pp. 153–162, 01 2005.

J. Liao, Y. Luo, C. Deng, and H. Huang, “Multi-level metric learning via smoothed wasserstein distance,” in Proceedings of the 27th International Joint Conference on Artificial Intelligence, ser. IJCAI’18. AAAI Press, 2018, p. 2919–2925.

N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, 1st ed. Cambridge University Press, 2000. [Online]. Available: http://www.amazon.com/Introduction-Support-Machines-Kernel-based-Learning/dp/0521803959/\result=1\?ie=UTF8&k=9s=8.

M. Cuturi and A. Doucet, “Fast computation of wasserstein barycenters,” 2014.

J. Ebert, V. Spokoiny, and A. Suvorikova, “Construction of non-asymptotic confidence sets in 2-wasserstein space,” 2017.

Y. Zeng, X. Fu, L. Gao, J. Zhu, H. Li, and Y. Li, “Robust multivehicle tracking with wasserstein association metric in surveillance videos,” IEEE Access, vol. 8, pp. 47 863–47 876, 2020.

Z. Hu, H. Xue, Q. Zhang, J. Gao, N. Zhang, S. Zou, Y. Teng, X. Liu, Y. Yang, D. Liang, X. Zhu, and H. Zheng, “Dip-net: Direct pet image reconstruction based on the wasserstein generative adversarial network,” IEEE Transactions on Radiation and Plasma Medical Sciences, vol. 5, no. 3, pp. 35–43, 2020.

K. Lei, M. Mardani, J. M. Pauly, and S. S. Vasanawala, “Wasserstein gains for mr imaging: From paired to unpaired training,” IEEE Transactions on Medical Imaging, vol. 40, no. 1, pp. 105–115, 2021.

P. Swami, A. Jain, P. Goswami, K. Chinits, A. Dubey, and P. Chaudhari, “High performance automotive radar signal processing on ti’s tda3x platform,” in 2021 IEEE Radar Conference (RadarConf), 2021, pp. 1317–1320.

R. Singh and P. Sharma, “Instance-vote-based motion detection using spatially extended hybrid feature space,” The Visual Computer, no. 34, 2020.

M. Riyadh, N. Mustapha, and D. Riyadh, “Review of trajectories similarity measurement methods,” in 2019 International Conference on Information Management (ICIM), 2017, pp. 423–426.

S. Rahman, M. A. Hossain, and A. H. G. Ahamed, “A survey on handcrafted and learning-based action representation approaches for human activity recognition,” Applied Sciences, vol. 7, no. 1, 2017. [Online]. Available: https://www.mdpi.com/2076-3417/7/1/110.

J. H. Giraldo, S. Javed, N. Werghi, and T. Bouwmans, “Graph cnn for moving object detection in complex environments from unseen videos,” in 2022 IEEE Access.
in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, October 2021, pp. 225–233.
[65] P. Markiewicz, K. Kogut, M. Różywiec, P. Skruch, and R. Starosolski, “Occupancy grid fusion prototyping using automotive virtual validation environment,” in Proceedings of the 6th International Conference on Control, Mechatronics and Automation, 2018, pp. 81–85.
[66] P. Markiewicz, M. Długosz, and P. Skruch, “Review of tracking and object detection systems for advanced driver assistance and autonomous driving applications with focus on vulnerable road users sensing,” in Polish Control Conference, Springer, 2017, pp. 224–237.
[67] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, “A systematic review of perception system and simulators for autonomous vehicles research,” Sensors, vol. 19, no. 3, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/3/6548.
[68] M. Chapel and T. Bouwmans, “Moving objects detection with a moving camera: A comprehensive review,” CoRR, vol. abs/2001.05238, 2020. [Online]. Available: https://arxiv.org/abs/2001.05238.
[69] International Organization for Standardization, ISO/PAS 21448:2019(E): Road vehicles — Safety of the indented functionality, www.iso.org, 2019.
[70] M. Kutula, P. Pykkönen, H. Holzhütter, M. Colomb, and P. Duthon, “Autonomous car lidar performance verification in fog and rain,” in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 1695–1701.
[71] T. Goelbes, B. Schläger, and S. Muckenhuber, “Fault detection, isolation and control,” Acta Automatis, 2016, pp. 1–12.
[72] Y. Kang, H. Yin, and C. Berger, “Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments,” IEEE Transactions on Intelligent Vehicles, vol. 4, no. 2, pp. 171–185, 2019.
[73] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” International Journal of Robotics Research (IJRR), 2013.
[74] S. Eiting, S. Cheng, B. Caise, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. Qi, Y. Zhou, Z. Yang, A. Choudur, P. Sun, J. Nguyen, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, “Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset,” CoRR, vol. abs/2014.10133, 2021.
[75] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, R. Kell, and T. Darrell, “Bdd100k: A diverse driving dataset for heterogeneous multitask learning,” 2020.
[76] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, “The apolloscape open dataset for autonomous driving and its application,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 10, pp. 2702–2719, 2020.
[77] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” 2016.
[78] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” 2020.
[79] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The clear mot metrics,” EURASIP Journal on Image and Video Processing, vol. 2008, 01 2008.
[80] P. Skruch, M. Długosz, and W. Mirkowski, “Mathematical methods for verification of microcontroller-based PID controllers for improving their reliability,” Eksploatacja i Niezawodność – Maintenance and Reliability, vol. 17, no. 2, pp. 327–333, 2015.
[81] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” arXiv preprint arXiv:2003.11027, 2019.
[82] W. Cunningham, “A network simplex method,” Math Program, vol. 11, pp. 105–116, 12 1976.
[83] N. Bonneel, M. van de Panne, S. Paris, and W. Heidrich, “Displacement interpolation using lagrangian mass transport,” ACM Trans. Graph., vol. 30, no. 6, p. 1–12, Dec. 2011. [Online]. Available: https://doi.org/10.1145/2074812.2074192.
[84] M. Kowalczyk, P. Bugiel, I. Izdyryczk, and M. Szelest, “Clustering of test scenes by use of Wasserstein metric and analysis of tracking quality in optically degraded videos,” Mar. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4628038.
[85] J. C. Dunn, “Well-separated clusters and optimal fuzzy partitions,” Journal of Cybernetics, vol. 4, no. 1, pp. 95–104, 1974.