Validating Simulation Environments for Automated Driving Systems Using 3D Object Comparison Metric*

Albert Wallace, Siddartha Khastgir, Xizhe Zhang, Simon Brewerton, Benoit Anctil, Peter Burns, Dominique Charlebois and Paul Jennings

Abstract— **One of the main challenges for the introduction of Automated Driving Systems (ADSs) is their verification and validation (V&V). Simulation based testing has been widely accepted as an essential aspect of the ADS V&V processes. Simulations are especially useful when exposing the ADS to challenging driving scenarios, as they offer a safe and efficient alternative to real world testing. It is thus suggested that evidence for the safety case for an ADS will include results from both simulation and real-world testing. However, for simulation results to be trusted as part of the safety case of an ADS for its safety assurance, it is essential to prove that the simulation results are representative of the real world, thus validating the simulation platform itself.**

In this paper, we propose a novel methodology for validating the simulation environments focusing on comparing point cloud data from real lidar sensor and a simulated lidar sensor model. A 3D object dissimilarity metric is proposed to compare between the two maps (real and simulated), to quantify how accurate the simulation is. This metric is tested on collected lidar point cloud data and the simulated point cloud generated in the simulated environment.

I. Introduction

The recent technological advancements in Automated Driving Systems (ADSs) are driven by several key benefits, such as increased road safety [1], increased traffic efficiency [2], reduced emission levels [3] and decreased driver workload [4]. Taking road safety as an example, it is reported that over half a million traffic accidents occurred in the UK between 2015-2020 alone [5], averaging a figure of over seventeen hundred on-road fatalities annually despite the current safety measures. While the potential benefits of ADSs could be immense, there is a significant challenge associated with their safe introduction in the society. From their conceptual design to deployment, a key and necessary stage within the development process of ADSs is their safety assurance. Ensuring safe introduction and public trust are essential requirements for their acceptance, scalable adoption and their correct use [6]. Traditionally, distance-based metrics have been used as safety arguments for vehicle safety. However, for CAVs, it is estimated that they need to be driven for over 11 billion miles to prove that they are 20% safer than human drivers due to the complexity associated with such systems, and the tasks they perform and the environments in which they operate [7]. This has led to a hazard based testing approach been proposed [8], which focuses on the quality of miles rather than quantity, concentrating on how a system fails rather than how it works. Coupled with a scenario-based testing approach [9], which is widely accepted as a verification and validation (V&V) approach, simulation based testing has emerged as a key tool in the V&V processes [10]. While the number of scenarios used to test an ADS will be many, each scenario will describe a set of scenery, environmental, and dynamic conditions within which the ADS will be tested [11].

A scenario-based evaluation workflow as shown in Figure 2, divides the whole process into scenario creation, scenario description format, database storage, plan and execution, safety evidence and validation [12]. The creation of scenarios can be both data driven (e.g. accident data analysis [13]) and knowledge driven (e.g. analytical hazard-based approach [14]). Upon creation, the scenario information will then be represented using an established format that is readable, executable, common and exchangeable [11], and consequently be stored in a scenario database, e.g. Safety Pool™ Scenario Database [15]. The execution of scenarios can be achieved in a virtual environment, real world, or a hybrid of the two such as hardware in the loop testing. As the quantity of scenarios increase, being able to decide how they are distributed across the available test environment settings for their execution becomes crucial. Real world execution ensures that the surrounding environment is consistent with the deployment environment, it can also ensure that identical system (i.e. hardware and software) is tested between testing and deployment.

However, real world execution can be expensive, risky, and time-consuming. This has led to a simulation-based development and testing approach being used in the ADS industry. Simulation based testing can be used in conjunction

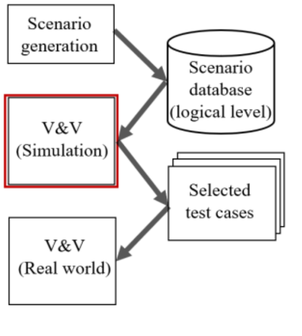


Figure 1. Scenario based evaluiation workflow

*Research supported by Transport Canada and UK Research & Innovation (UKRI) Future Leaders Fellowship.

Albert Wallace, Siddartha Khastgir, Xizhe Zhang and Paul Jennings are with WMG, University of Warwick, Coventry, UK, CV4 7AL. (E-mail: {albert.wallace, s.khastgir.1@warwick.ac.uk, jason.zhang, paul.jennings}@ warwick.ac.uk).

Simon Brewerton is with RDM Group Ltd., 33, Bilton Industrial Estate, Coventry CV3 1JL, UK. (e-mail: sbrewerton@rdmgroup.co.uk).

Benoit Anctil, Peter Burns and Dominique Charlebois are with Transport Canada, 330 Sparks St., Ottawa, ON K1A 0N8, Canada. (E-mail: {benoit.anctil, peter.burns, dominique.charlebois}@ tc.gc.ca).

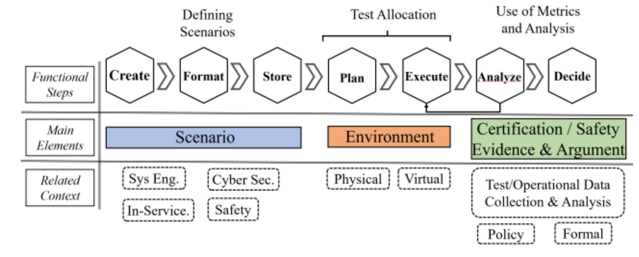


Figure 2. Scenario based evaluiation workflow

with the more *'expensive'* hybrid or real world testing [12] to form a comprehensive testing strategy, as shown in Figure 2. In this case it acts as an initial phase of the testing to identify a small set of highly critical scenarios from a large number of input scenarios (which might be impossible to execute in real world). This small set of critical scenarios can then be executed in real world environment. However, to enable such a simulation-based workflow, or rather to utilize any simulation environment for testing, a fundamental assumption is that simulation and real-world environments are comparable or correlated, and that the simulation results are valid. Without this validity, simulation-based activities carry little meaning in the ADS safety assurance process.

Several simulations platforms are available for the development and testing of ADSs. They vary in their fidelity level and focus at different testing objectives [16]–[21]. From a fidelity perspective, they range from visually compelling photo realistic physics engine-based simulation to minimalistic rendering of the environment, to purely text-based simulation. From a testing objective perspective, they range from microscopic traffic-based simulation to vehicle and pedestrian level simulation, to system level simulation, and sensor level simulation. At the sensor level simulation, the quality and accuracy of the simulation is dependent on the level of fidelity of the simulation environment (i.e., the world), the same is true when sensors are considered within the system and vehicle level simulation. However, if sensors are to be treated independently from the system and vehicle level simulations, the quality of such simulations becomes less restricted on the fidelity of the environment. Although vast amount of effort has been input into the development and testing techniques of ADSs within simulation environments, there is a lack of emphasis on the validation of the simulation environment up until recently. For the past year, there are ongoing discussion within the United Nations Economic Commission for Europe (UNECE) Validation Method for Automated Driving (VMAD) Sub-group 2 on developing a credibility assessment framework for simulation and virtual testing [22]. The credibility assessment framework being discussed is not only a function of the test objective (as discussed earlier), but also a function of the Operational Design Domain (ODD) of the ADS. The ODD of the ADS will influence the modelling requirements for the simulation platform in order to test the ADS both inside and outside its ODD conditions. For example, if the ODD of an ADS includes *"light rainfall"* but excludes *"heavy rainfall"*, the simulation environment and the sensor models should have appropriate fidelity to be able to distinguish between the rainfall intensities.

The study captured in this paper aims to contribute further on the simulation validation approach, with the use of a lidar system to devise a similarity metric between maps generated through real world-based lidar scan and a virtual world generated through a lidar sensor model in a simulation environment, by comparing the 3D objects and mapped data. The end goal is to have the ability to quantify how similar a map created by a virtual lidar sensor model on a simulated vehicle travelling through a virtual map is to the map generated by a real lidar on a real-world vehicle travelling through the real-world area the virtual map was built from. Though the metric will also be able to generalize to all 3D objects to allow validation for other ADS components, such as conventional cameras and radar. The proposed approach has three-stages: (i) collect the real-world data (ii) synthesize the virtual map and virtual testing environment, (iii) develop a method for comparing the virtual and real world lidar point clouds.

II. Review of Comparison Methods

There are a variety of methods for 3D object comparison. The four main criteria in influencing the selection of method to use are as follows:

- **Suitability for the data type**: The object is created from a lidar environment scan. This means that they are large in scale and complex, are unlikely to be convex and will be unable to be described through a single shape. This is generally different to the type of model created through using CAD systems.
- **Automated:** The method should be able to function without manual interference or apriori knowledge of the model. This will allow for the comparison to be done on many pairs quickly and will allow validation to completed without human bias.
- **Shape preserving**: Ideally the method should maintain the shapes' original structure without deformation. This is to avoid loss of relevant information and fidelity with the comparison.
- **Practicality:** This mainly includes complexity, runtime, and robustness. Meaning it will be able to handle large and complex objects quickly and without breaking even if the data is not exactly as expected.

With lidar and point cloud data, the most direct comparison to make is a 1 to 1 comparison using the mean distance between corresponding points [23]. This was used on airborne lidar data which would make it suitable for this use case. The point comparison is either done on a point-by-point basis, known as the chamfer distance, or by using a

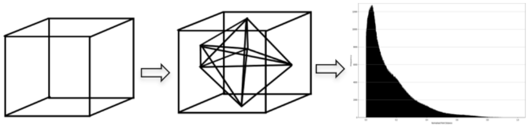


Figure 3. Demonstration of cube to histogram conversion

local model that takes a least squares average plane of all the points in the other cloud that are located within a fixed sphere around the original point and finds the shortest distance between this plane and the original point. This should avoid discrepancies caused by individual points being missing or slight differences in lidar calibration. The major disadvantage of this method is that it requires manual rotational alignment of the two clouds to allow corresponding points to be compared.

Two methods described in [24] and [25] allow for the objects to be aligned automatically through identifying key points using principal curvatures and through using the inertia tensor as a principal axis, respectively. Though these methods would need to be completely reliable to avoid discrepancies in the point cloud comparison, as a small discrepancy in alignment would have a large effect on the score.

Another similar method to this uses an underlying surface created using a 3D Fisher vector known as DPDist [26]. However, this uses a neural network which would require many sample models to train that are not necessarily available for our use case. It is also more suited to sparse point clouds.

The method in [27] suggests turning a 3D object into many 2D silhouettes from different perspectives, the most similar silhouette pairs are identified between the two objects and the dissimilarity is calculated and summed over all pairs. The dissimilarity measure between two 2D silhouettes is given by measuring the distortion of a bijection between the two images. This method would work best with convex objects that have distinct silhouettes whilst the point cloud maps will have lots of detail that will be obscured by the silhouette process (obstructed by buildings etc.).

A direct comparison between the curvature of corresponding points on the surface is demonstrated in [28]. This method however requires a spherical transformation of the object which deforms the shape.

Another method includes the comparisons between skeletal Reeb graphs [29]. This method works by reducing the 3D object surface to a directed graph based on the distance from the centre of mass. Similarity is calculated by finding the number of edges in the largest common subgraph of both objects. This method however is much more suited to CAD generated 3D objects instead of those generated by point cloud because of the use case the method was built around and the need for geometric attributes to be known about the object.

The method described in [30] uses a comparison of the distribution of pairwise distances between a sample of points on the surface of the object. The distribution for each shape is found by taking a sample of points on the surface of the object, finding the pairwise distances between them and storing those in a histogram as shown in Figure 3. This method was used on objects created from voxels of point clouds, which is similar to our use case. It is also indifferent to the alignment of the object as the distribution is inherent to the shape and it is shape preserving. The method's runtime can be controlled by the number of points sampled and is unaffected by the size of the object. Its main drawback is that due to the random nature of the point sampling the score is not always consistent. However, modifications to this method to improve the consistency of this approach are described in later sections. As this this method fits our four selection criteria, a modified version of the approach has been be implemented in our study for point cloud comparison.

III. Methodology

A. *Data Collection*

The data was collected by mounting a 360-degree lidar upon an Aurrigo pod (i.e. a low speed automated driving system) [31]. This was driven on the route pictured in Figure 5 on University of Warwick campus. It was then driven through this environment multiple times, at two different speeds and with three different lidar rotation rates which resulted in a total of six different lidar scans of the environment which are taken at: 2m/s 1200rpm, 2m/s

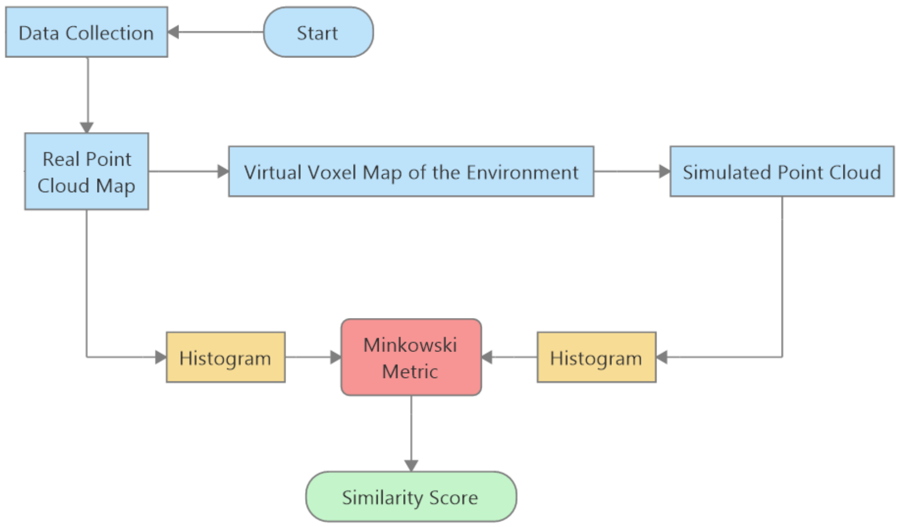


Figure 4: Methodology flowchart

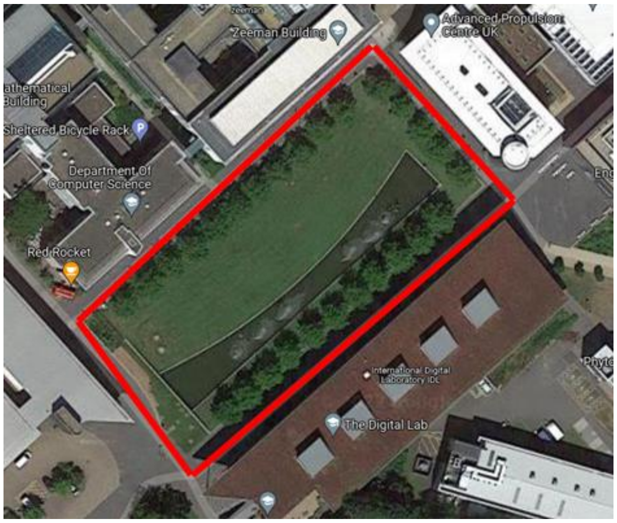

Zeeman Building
Advanced Propulsion Centre UK
athematical Building
Sheltered Bicycle Rack
Department Of Computer Science
Red Rocket
Eng
International Digital Laboratory IDL
Phys
The Digital Lab

Figure 5: Route of the pod data collection on University of Warwick campus

600rpm, 2m/s 300rpm, 4m/s1200rpm, 4m/s 600rpm and 4m/s 300rpm. The best quality scan is taken at the lower speed with the fastest lidar rotations as this allows the most data (points) to be collected as the vehicle moves through the environment. The raw Lidar data was then converted into a 3D point cloud map of the environment.

B. Creation of the Simulated Lidar Map

Before the creation of the simulated lidar map, the point cloud data of the real environment was transformed into a voxel environment map. Voxelization was performed using a 3D cubic grid of specified dimensions, which in our case was 5 x 5 x 5 cm. For every cube in the grid if a point cloud point is present inside then it is filled in as a voxel, which is a solid cube that can be thought of as a three-dimensional pixel.

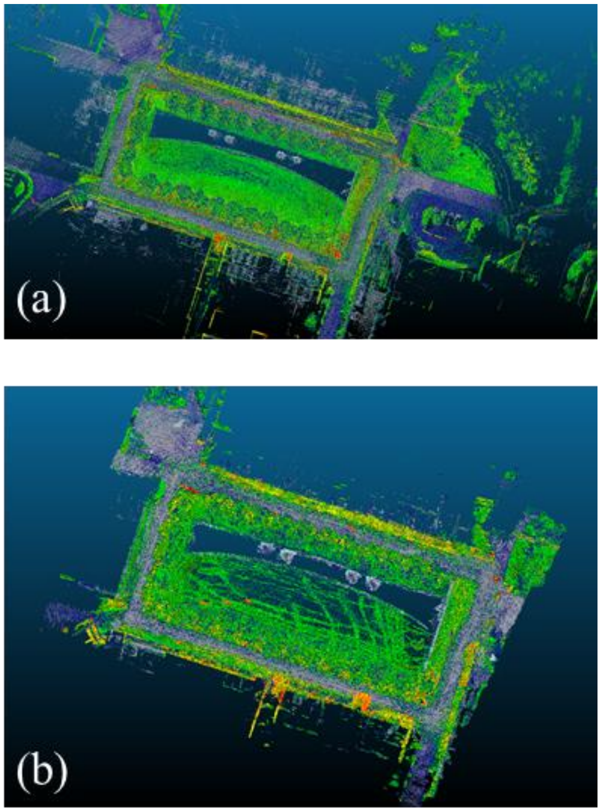


Figure 6: (a) Real map (b) simulated map

Then a simulated lidar traversed the same route through the voxel environment as the vehicle did through the real world, which then produces a simulated lidar scan. Direct comparison is then possible between any two maps (real map and simulated map). It must be noted that the fidelity of the simulated map will depend on the virtual lidar model and the fidelity of the world in the simulation. Thus, the methodology enables us to validate the simulation platform (both the world and the sensor model).

This was performed for all different configurations of the lidar sensor and vehicle speed. Pictured in Figure 6 is the real world and simulated point cloud maps for the highest quality lidar scan at 2 m/s and 1200 rpm.

C. Comparison Method

The method used to compare the real and simulated point cloud maps was adapted from [30]. This method takes a sample of points on the surface of each object. The pairwise Euclidean distances for all these points are then calculated and stored in a 2-dimensional matrix. These distances are then normalized with respect to the maximum distance of each object and converted into a histogram with a fixed number of evenly spaced bins. Instead of storing frequency in the histogram, probability density is used instead.

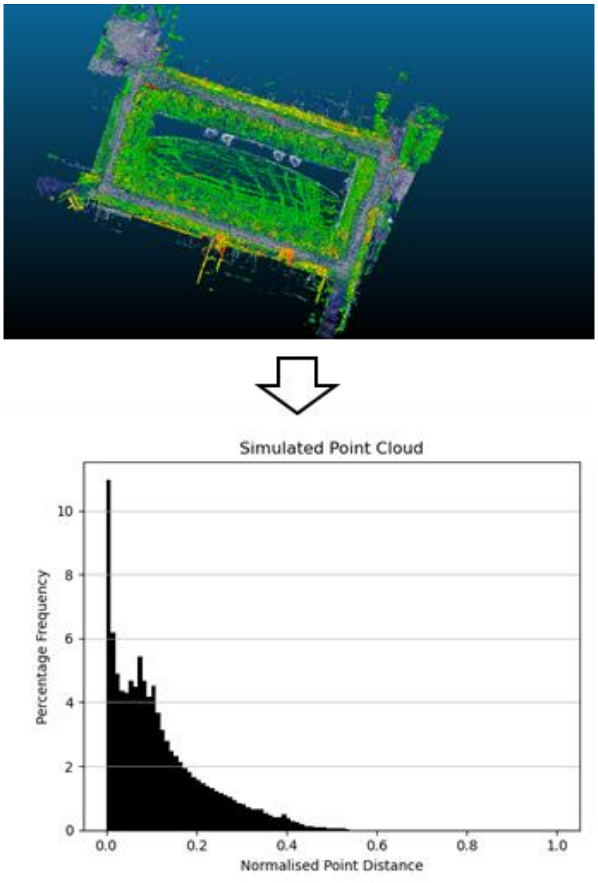


Figure 7: Conversion of simulated map into histogram

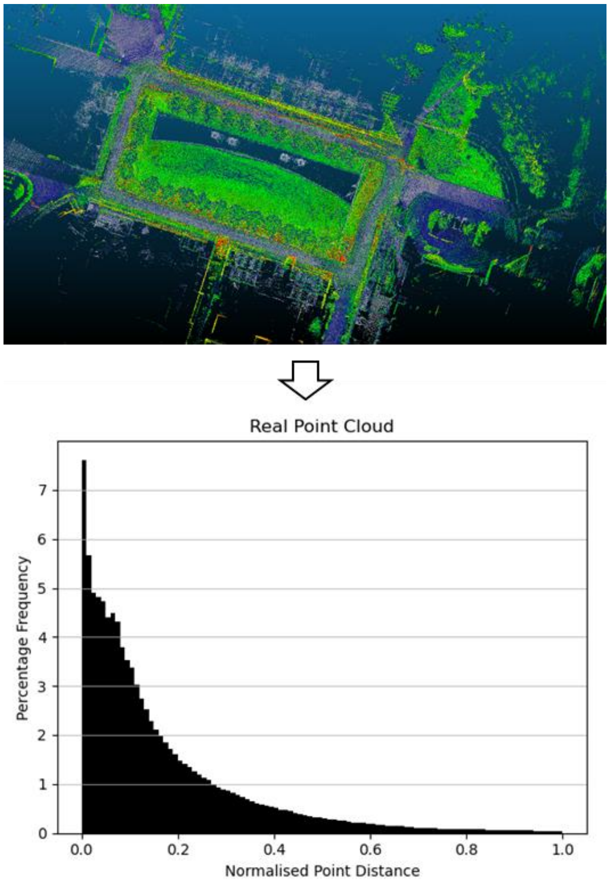


Figure 8. Conversion of a real map into a histogram

The two histograms are then created. These are discrete probability distributions that represent the shapes. These distributions are then compared using the Minkowski distance shown in *Eqn 1*. for two general distributions. The Minkowski distance is the sum of the absolute differences between the columns of the histograms. This metric was chosen because of its robust and clear nature; though there are many other options for this comparison e.g., Bhattacharya and Chi-Square Distances.

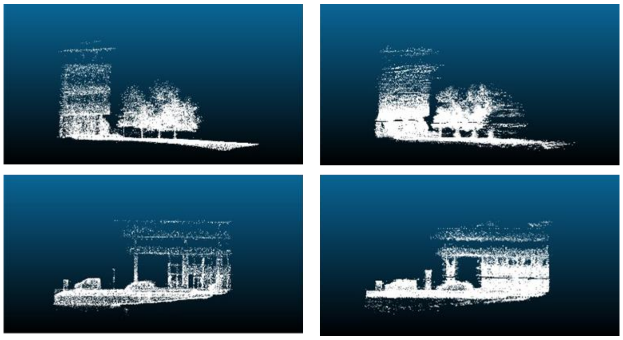


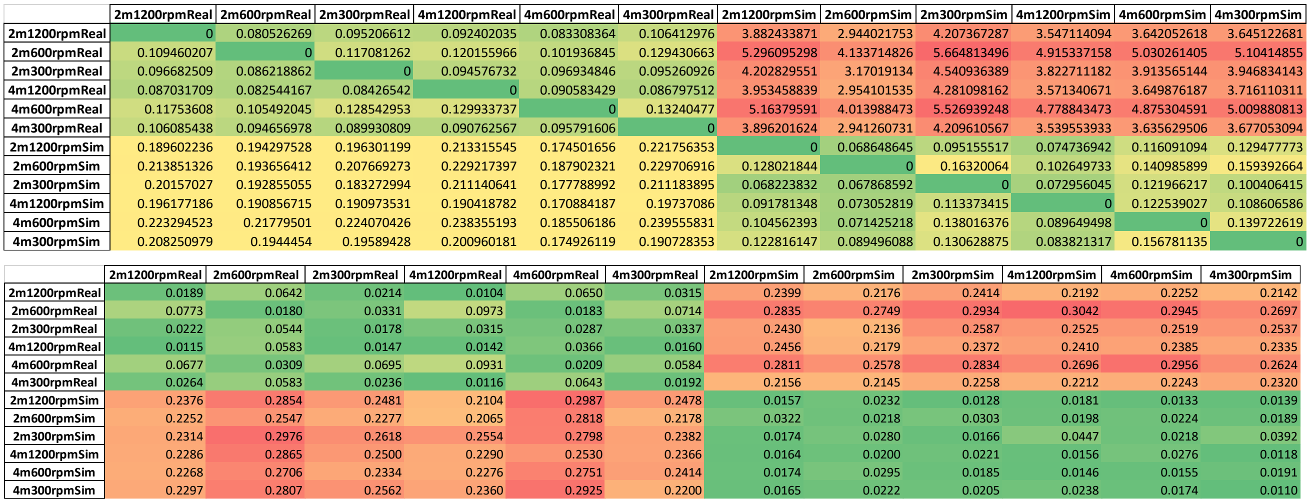
Figure 9. Different areas of real (left) and simulated (right) maps

$$D(X,Y) = \sum_{i=1}^{n}|x_i - y_i| \tag{1}$$

The original method would have normalized the distances for each object by dividing by the maximum distance found for that object, meaning that the total size of the map would be irrelevant. This is useful in the general case as it allows comparison between shapes without having to manually control the volume beforehand. It also allows the histogram bins to be aligned by simply picking the number of evenly spaced bins required.

However, in our use case the transformation from real point cloud to simulated point cloud needs to be distance preserving. This means that the similarity metric should consider the objective distances instead of the distances relative to the size of the object.

This was done by using the largest distance out of both objects to normalise the distances. The histogram was then created for the object with the larger distance. The same histogram bins were then used to create the second histogram for the other object to ensure that they aligned, and the Minkowski metric could be used on them appropriately. For instance, in Figure 8, the real point cloud has the larger distance which is why the histogram has normalised distances present up to one on the x-axis, whilst in the simulated point cloud in Figure 7, the histogram does not. This then allows the histograms to be directly comparable w.r.t distance.



	2m1200rpmReal	2m600rpmReal	2m300rpmReal	4m1200rpmReal	4m600rpmReal	4m300rpmReal	2m1200rpmSim	2m600rpmSim	2m300rpmSim	4m1200rpmSim	4m600rpmSim	4m300rpmSim
2m1200rpmReal	0	0.080526269	0.095206612	0.092402035	0.083308364	0.106412976	3.882433871	2.944021753	4.207367287	3.547114094	3.642052618	3.645122681
2m600rpmReal	0.109460207	0	0.117081262	0.120155966	0.101936845	0.129430663	5.296095298	4.133714826	5.664813496	4.915337158	5.030261405	5.10414855
2m300rpmReal	0.096682509	0.086218862	0	0.094576732	0.096934846	0.095260926	4.202829551	3.17019134	4.540936389	3.822711182	3.913565144	3.946834143
4m1200rpmReal	0.087031709	0.082544167	0.08426542	0	0.090583429	0.086797512	3.953458839	2.954101535	4.281098162	3.571340671	3.649876187	3.716110311
4m600rpmReal	0.11753608	0.105492045	0.128542953	0.129933737	0	0.13240477	5.16379591	4.013988473	5.526939248	4.778843473	4.875304591	5.009880813
4m300rpmReal	0.106085438	0.094656978	0.089930809	0.090762567	0.095791606	0	3.896201624	2.941260731	4.209610567	3.539553933	3.635629506	3.677053094
2m1200rpmSim	0.189602236	0.194297528	0.196301199	0.213315545	0.174501656	0.221756353	0	0.068648645	0.095155517	0.074736942	0.116091094	0.129477773
2m600rpmSim	0.213851326	0.193656412	0.207669273	0.229217397	0.187902321	0.229706916	0.128021844	0	0.16320064	0.102649733	0.140985899	0.159392664
2m300rpmSim	0.20157027	0.192855055	0.183272994	0.211140641	0.177788992	0.211183895	0.068223832	0.067868592	0	0.072956045	0.121966217	0.100406415
4m1200rpmSim	0.196177186	0.190856715	0.190973531	0.190418782	0.170884187	0.19737086	0.091781348	0.073052819	0.113373415	0	0.122539027	0.108606586
4m600rpmSim	0.223294523	0.21779501	0.224070426	0.238355193	0.185506186	0.239555831	0.104562393	0.071425218	0.138016376	0.089649498	0	0.139722619
4m300rpmSim	0.208250979	0.1944454	0.19589428	0.200960181	0.174926119	0.190728353	0.122816147	0.089496088	0.130628875	0.083821317	0.156781135	0

	2m1200rpmReal	2m600rpmReal	2m300rpmReal	4m1200rpmReal	4m600rpmReal	4m300rpmReal	2m1200rpmSim	2m600rpmSim	2m300rpmSim	4m1200rpmSim	4m600rpmSim	4m300rpmSim
2m1200rpmReal	0.0189	0.0642	0.0214	0.0104	0.0650	0.0315	0.2399	0.2176	0.2414	0.2192	0.2252	0.2142
2m600rpmReal	0.0773	0.0180	0.0331	0.0973	0.0183	0.0714	0.2835	0.2749	0.2934	0.3042	0.2945	0.2697
2m300rpmReal	0.0222	0.0544	0.0178	0.0315	0.0287	0.0337	0.2430	0.2136	0.2587	0.2525	0.2519	0.2537
4m1200rpmReal	0.0115	0.0583	0.0147	0.0142	0.0366	0.0160	0.2456	0.2179	0.2372	0.2410	0.2385	0.2335
4m600rpmReal	0.0677	0.0309	0.0695	0.0931	0.0209	0.0584	0.2811	0.2578	0.2834	0.2696	0.2956	0.2624
4m300rpmReal	0.0264	0.0583	0.0236	0.0116	0.0643	0.0192	0.2156	0.2145	0.2258	0.2212	0.2243	0.2320
2m1200rpmSim	0.2376	0.2854	0.2481	0.2104	0.2987	0.2478	0.0157	0.0232	0.0128	0.0181	0.0133	0.0139
2m600rpmSim	0.2252	0.2547	0.2277	0.2065	0.2818	0.2178	0.0322	0.0218	0.0303	0.0198	0.0224	0.0189
2m300rpmSim	0.2314	0.2976	0.2618	0.2554	0.2798	0.2382	0.0174	0.0280	0.0166	0.0447	0.0218	0.0392
4m1200rpmSim	0.2286	0.2865	0.2500	0.2290	0.2530	0.2366	0.0164	0.0200	0.0221	0.0156	0.0276	0.0118
4m600rpmSim	0.2268	0.2706	0.2334	0.2276	0.2751	0.2414	0.0174	0.0295	0.0185	0.0146	0.0155	0.0191
4m300rpmSim	0.2297	0.2807	0.2562	0.2360	0.2925	0.2200	0.0165	0.0222	0.0205	0.0238	0.0174	0.0110

Figure 10. Results tables for comparisons with Chamfer distance (top) and Histogram method (bottom), grading is green to red, with red meaning a higher score and therefore more dissimilar maps

Outlier points that were far from the vehicle route were removed from the real point cloud as part of the pre-processing.

IV. Results

Firstly, every possible combination of the different lidar setting were compared with each other and a score was generated. The same was then done using the chamfer distance which is the mean distance between nearest points in the two clouds. The top table in Figure 10 shows the chamfer distance results. There is a large difference depending on whether the points of the simulated cloud were considered first and matched against the real-world points compared with considering the real-world points first. There are also several outliers while comparing between two simulated maps, for example, 2 m/s 300 rpm Sim compared with 2 m/s 600 rpm Sim which has a score of 0.163, which is a similar score to most of the real and simulated scores in the lower left quadrant meaning that the metric can't always differentiate between two clouds of different types.

Comparatively, the histogram method results in the bottom of Figure 10 shows consistency within each quadrant and is much more symmetrical. The lidar configurations that gave consistently the worst results were when the 600 rpm real scans were compared with the simulated point cloud. The 600 rpm real scans for both 2 m/s and 4 m/s configurations contained over 6 million points, whilst the rest of the point clouds had around 5 million. When the open3D statistical outlier removal function was used on these clouds the most points were removed from those with 600 rpm, indicating they are the noisiest. This also explains the higher score (poorer matching as compared to 300 rpm) between the point clouds at 600 rpm as intuitively they should have a better match due to higher number of points in the raw point cloud.

The method was also used on various areas of the map. The average score between all configurations of the real point cloud and their simulated point clouds was 0.247. Whilst removing areas of the real point clouds which were not present in the simulated gave an average of 0.123. The method also showed that the best lidar configuration to use is 4m/s 1200 rpm with an average dissimilarity score of 0.2275 over comparisons with all simulated maps though 2 m/s 1200 rpm was only marginally worse with an average score of 0.2299 as shown in Table 1

An area containing many trees was compared (illustrated in the top of Figure 9) and had an average of 0.259 whilst an area that contained no trees but did contain two vehicles and a complex building. However, the areas pictured in the bottom of Figure 9, had a lower average score of 0.184. This shows that the simulation method is currently more viable for vehicles and buildings than it is for trees. This also illustrated the influence of scene complexity and ODD attributes on the fidelity requirements for the simulation environment.

V. Discussion and Future Work

The main drawback of this method is the element of inconsistency created through the random sampling. This has been mitigated by taking a large enough sample of points which will give a more stable distribution of distances for the object. Another factor that effects the consistency of the method is the number of bins created for the histograms, as having a greater number of smaller bins will allow for a higher fidelity comparison as smaller differences in distance will be accounted for, this however does also lead to a greater effect on the score from the random sampling.

Since the algorithm has $O(n^2)$ complexity with respect to the number of sample points it is important to find the highest number of sampled points that still has an acceptable runtime. For our purposes 10000 sample points with 100 bins gives consistent scores to 2 decimal places and a good runtime of approximately 15 seconds.

The other issue caused by the random sampling is that it means a comparison between two identical objects does not give a dissimilarity score of 0. This problem is easy to control however because it's possible to find the similarity of two identical objects. This can then be used to benchmark what a perfect score is as well as quantifying the irregularity caused by the sampling.

Future work would involve compared the solid voxels which are generated from the point clouds (both real and simulated) and to compare the results with the point cloud comparison results. Additionally, the methodology can also be applied to dynamic elements (both in real-world and simulation) and will be part of future work. This will provide further insight into the robustness of the comparison and the methodology. Potential future work could also include the development of metrics for point cloud/voxel comparison as a function of the ODD of the ADS.

TABLE I. AVERAGE SCORE OF REACH REAL MAP COMPARED WITH THE SIMULATED MAP

Configuration	**Average score**
2 m/s 1200 rpm real	0.2299
2 m/s 600 rpm real	0.2793
2 m/s 300 rpm real	0.2462
4 m/s 1200 rpm real	0.2275
4 m/s 600 rpm real	0.2802
4 m/s 300 rpm real	0.2336

VI. Conclusion

Simulation-based testing of ADSs is widely accepted as a key building block towards generating evidence for safety assurance of the ADSs. In order to trust the results of the simulation for use as safety evidence as part of the ADS safety case, it is essential that the simulation is validated, i.e., representative of the real-world. Up until now, there has been limited focus on validation of simulation used for ADS testing. In this paper, we propose a novel approach of validation simulation environment using 3D object mapping in a real-world point cloud map and simulated point cloud map.

We propose a histogram comparison method for point cloud data which has tested using various configurations of the lidar sensor in real world and in simulation. Due to the generality of the method, it can be extended to other types of simulation data and is not restricted to use on lidar point clouds. As the histogram method can sample points off the surface of any 3D object, this means that comparison can be used on maps generated through combined means such as lidar, radar and camera. The method presented in this paper will allow simulation maps to be benchmarked against their real-world counterparts and validate whether they are a good approximation of the real world, to enable their use in generating safety evidence for ADSs. The results suggest that the best lidar configuration to use is 4m/s 1200 rpm with an average dissimilarity score of 0.2275 over comparisons with all simulated maps. However, the 2 m/s 1200 rpm configuration was marginally worse with an average score of 0.2299. Results also suggested that rotations per second has a higher impact on simulation fidelity as compared to speed of the ADS.

ACKNOWLEDGMENT

The work presented in this paper has been supported by Transport Canada, UKRI Future Leaders Fellowship (Grant MR/S035176/1) and Department of Transport, HM Government. The authors would like to thank the WMG center of HVM Catapult, and WMG, University of Warwick, UK for providing the necessary infrastructure for conducting this study.

REFERENCES

[1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles : opportunities , barriers and policy recommendations," *Transp. Res. Part A*, vol. 77, pp. 167–181, 2015, doi: 10.1016/j.tra.2015.04.003.

[2] S. Le Vine, X. Liu, F. Zheng, and J. Polak, "Automated cars : Queue discharge at signalized intersections with ' Assured-Clear-Distance-Ahead ' driving strategies," *Transp. Res. Part C Emerg. Technol.*, vol. 62, pp. 35–54, 2016, doi: 10.1016/j.trc.2015.11.005.

[3] D. J. Fagnant and K. M. Kockelman, "The travel and environmental implications of shared autonomous vehicles , using agent-based model scenarios," *Transp. Res. Part C Emerg. Technol.*, vol. 40, pp. 1–13, 2014, doi: 10.1016/j.trc.2013.12.001.

[4] N. Balfe, S. Sharples, and J. R. Wilson, "Impact of automation : Measurement of performance , workload and behaviour in a complex control environment," *Appl. Ergon.*, vol. 47, pp. 52–64, 2015, doi: 10.1016/j.apergo.2014.08.002.

[5] "Road Safety Data - STATS19," *UK Department for Transport*, 2020. https://data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data (accessed Apr. 02, 2020).

[6] S. Khastgir, S. Birrell, G. Dhadyalla, and P. Jennings, "Calibrating trust through knowledge: Introducing the concept of informed safety for automation in vehicles," *Transp. Res. Part C Emerg. Technol.*, vol. 96, pp. 290–303, 2018, doi: 10.1016/j.trc.2018.07.001.

[7] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?," *Transp. Res. Part A Policy Pr.*, vol. 94, pp. 182–193, 2016.

[8] S. Khastgir, S. Birrell, G. Dhadyalla, and P. Jennings, "The Science of Testing: An Automotive Perspective," 2018, doi: 10.4271/2018-01-1070.

[9] C. Neurohr, L. Westhofen, M. Butz, M. Bollmann, U. Eberle, and R. Galbas, "Criticality Analysis for the Verification and Validation of Automated Vehicles," *IEEE Access*, vol. 9, no. i, 2021, doi: 10.1109/ACCESS.2021.3053159.

[10] D. Nalic, H. Li, A. Eichberger, C. Wellershaus, A. Pandurevic, and B. Rogic, "Stress Testing Method for Scenario-Based Testing of Automated Driving Systems," *IEEE Access*, vol. 8, pp. 224974–224984, 2020, doi: 10.1109/ACCESS.2020.3044024.

[11] X. Zhang, S. Khastgir, and P. Jennings, "Scenario Description Language for Automated Driving Systems: A Two Level Abstraction Approach," 2020.

[12] X. Zhang, S. Khastgir, H. Asgari, and P. Jennings, "Test Framework for Automatic Test Case Generation and Execution Aimed at Developing Trustworthy AVs from Both Verifiability and Certifiability Aspects," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2021-Septe, pp. 312–319, 2021, doi: 10.1109/ITSC48978.2021.9564542.

[13] E. Esenturk, S. Khastgir, A. Wallace, and P. Jennings, "Analyzing real-world accidents for test scenario generation for automated vehicles," *IEEE Intell. Veh. Symp. Proc.*, vol. 2021-July, no. Iv, pp. 288–295, 2021, doi: 10.1109/IV48863.2021.9576007.

[14] S. Khastgir, S. Brewerton, J. Thomas, and P. Jennings, "Systems Approach to Creating Test Scenarios for Automated Driving Systems," *Reliab. Eng. Syst. Saf.*, doi: doi.org/10.1016/j.ress.2021.107610.

[15] "Safety Pool Scenario Database." .

[16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," no. CoRL, pp. 1–16, 2017.

[17] "PreScan." https://tass.plm.automation.siemens.com/prescan-services (accessed Feb. 13, 2022).

[18] "OpenPASS," 2022. https://openpass.eclipse.org/ (accessed Feb. 13, 2022).

[19] "SVL simulator." https://www.svlsimulator.com/ (accessed Feb. 13, 2022).

[20] E. Knabe, "SimS. Simulation Scenarios. Public report," 2019.

[21] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO--simulation of urban mobility: an overview," *Proc. SIMUL 2011, Third Int. Conf. Adv. Syst. Simul.*, 2011.

[22] "UNECE VMAD - SG 2 (Virtual testing)." https://wiki.unece.org/pages/viewpage.action?pageId=117508578 (accessed Feb. 13, 2022).

[23] T. Zhou, S. M. Hasheminasab, Y.-C. Lin, and A. Habib, "Comparitive Evaluation of Derived Image and LIDAR Point Clouds from UAV-based Mobile Mapping Systems," *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLIII-B2-2, pp. 169–175, Aug. 2020, doi: 10.5194/isprs-archives-XLIII-B2-2020-169-2020.

[24] J. Huang and S. You, *Point cloud matching based on 3D self-similarity*. 2012.

[25] M. Novotni and R. Klein, "A geometric approach to 3D object comparison," in *Proceedings International Conference on Shape Modeling and Applications*, 2001, pp. 167–175, doi: 10.1109/SMA.2001.923387.

[26] D. Urbach, Y. Ben-Shabat, and M. Lindenbaum, "DPDist: Comparing Point Clouds Using Deep Point Cloud Distance BT - Computer Vision – ECCV 2020," 2020, pp. 545–560.

[27] H. Gotoda, "3D shape comparison using multiview images," *NII J.*, pp. 19–25, Sep. 2003.

[28] H.-Y. Shum, M. Hebert, and K. Ikeuchi, "On 3D shape similarity," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996, pp. 526–531, doi: 10.1109/CVPR.1996.517122.

[29] S. Biasotti and S. Marini, "3D object comparison based on shape descriptors," *Int. J. Comput. Appl. Technol.*, vol. 23, pp. 57–69, Jan. 2005, doi: 10.1504/IJCAT.2005.006465.

[30] V. Muliukha, A. Lukashin, and A. Ilyashenko, "An Intelligent Method for Comparing Shapes of Three-Dimensional Objects," in *2019 25th Conference of Open Innovations Association (FRUCT)*, 2019, pp. 234–240, doi: 10.23919/FRUCT48121.2019.8981528.

[31] "Aurrigo Driverless Technology." https://aurrigo.com/ (accessed Feb. 13, 2022).