Saarland University

Master Thesis

Improving Self-Driving Car Testing Through Feedback-Driven Procedural Content Generation

Author:
Marc Müller, B.Sc.

Supervisor:
Prof. Dr.-Ing. Andreas Zeller

Advisor:
Alessio Gambi, Ph.D.

Tuesday 26th September, 2017

Abstract

Self-driving cars have become a very relevant part of the automotive industry and will only become more important in the near future. Ensuring safety is naturally a big concern when handing over full control of the vehicle to software, but the industry has not settled on a standard way to certify self-driving cars. Simulated, or virtual, tests have been introduced as a way to expose problems before deployment, but traditional software testing techniques cannot cope with the massive amount of situations a self-driving car faces. For limited cases, more advanced techniques like search-based testing show more promising results. This work will continue in that direction, utilising procedural content generation and genetic algorithms to evolve full driving scenarios from roads to traffic meant to test the system in control of a car. Meaningful metrics to characterise input scenarios, output behaviour of the car, and how well tests cover the possibilities of both input and output will be introduced to guide test suite evolution towards stressing the vehicle in interesting and relevant settings.

1 Introduction

Over the past few years, the topic of automating cars with the help of software has become an important part of the industry. The automation ranges from systems with limited control over the car, such as advanced driver-assistance systems that help the driver operate a vehicle, to fully self-driving cars that take full control of the car, correctly navigate through roads and obey traffic laws. There are several ongoing projects within this domain (e.g. Mobileye [1], Waymo [45]), some of which mature enough to be currently under testing in everyday life [19].

Testing of such automated vehicles is of obvious importance: Failures in software that's meant to operate vehicles with human passengers can easily lead to crashes, which would – in the worst case – be deadly. Thus, before fully self-driving cars or even just partially automated systems can enter the mainstream, their functionality needs to be rigorously tested. Besides avoiding danger, the quality of driving is another important factor. A car that might be able to travel from point A to point B without an accident could still be uncomfortable for the user simply due to the car driving poorly (e.g., sharp turns, sudden de-/acceleration, etc.)

Despite the importance of testing, the industry has yet to settle in on some standardised procedures to test automated vehicles [26]. Existing methods to ensure the safety of non-automated cars before deployment don't translate well into the space of autonomous cars [29] and how to apply traditional software testing techniques to autonomous vehicles is a fairly open topic (see Section 2.) Putting autonomous vehicles on trial in the real world or tailored driving challenges is a common approach [3, 39, 11, 19], but trials like that would need to cover hundreds of billions of kilometres driven to reliably compare their failure rates to that of human-driven vehicles [25].

An alternative to Naturalistic Driving Studies, where the vehicle is tested in regular traffic, are simulated tests. To avoid the cost of setting up a physical testing environment or the risks associated with testing self-driving cars in normal traffic, the software controlling an autonomous vehicle is fed with simulated data to see how it responds [33]. These simulations vary in scope, some focusing on simulating a single or few sensors to test specific functions of the vehicle [22, 21, 34, 38, 47], going for a realistic simulation of visual data to test lane and object detection capabilities [35, 37, 23], or evaluating behaviour at a higher level of abstraction to see how the vehicle acts in traffic or other scenarios [13, 10, 8, 20, 46, 40, 5]. It's important to note that these simulations don't replace real world tests and come with the problem of the simulation gap: How well the simulations translate into real world scenarios [41].

Depending on the type of simulation, requirements for the simulation framework can be very stringent, as the simulator needs to provide data such as realistic graphics, run complex physics computations to accurately model movements that are difficult to generate efficiently, and model driving scenarios to be executed in the simulation. Some of these problems have been worked on in the context of driving simulators for humans [30, 17, 44], but also video game engines, which take on the task of simulating 3D worlds with physics in real time well enough to have already been used in testing of self-driving cars [16]. What's open is *the automatic* creation of driving scenarios to be bested by the car. The field of procedural content generation, which is used to automatically create content such as levels for video games, is of help here, but generating scenarios for testing requires more than that (e.g., pedestrians, traffic, setting goals, etc.) There exist works to dynamically create city-like road networks [42] or race tracks meant to challenge the player based on their behaviour [32]. These works focus on generating content that is engaging to humans, while my work will build on them to produce driving scenarios meant to challenge self-driving AIs and the metrics required to characterise those challenges.

These metrics will be a major contribution of the work, as they will provide testers with a way to evaluate their test suites' completeness. They will focus two aspects of testing: the input the vehicle is confronted with and the output behaviour it produces. Input, in this case, is referring to the driving scenario a vehicle has to traverse, while output covers the things the vehicle actually does (e.g. path taken, amount of turns, turn angles, etc.) Those metrics will then be used as part of a genetic algorithm that will evolve test suites towards targeted metrics, both for in- and output.

2 Related work

The field of testing self-driving cars by means of simulations is fairly large and open; what follows is a closer look at the works most relevant to the goals of this thesis.

One rather comprehensive framework for virtual testing is described in [40]. The system is presented as a construction kit for simulations, allowing users to define their own testing scenarios through selecting and combining elements provided by the framework. Besides specification of test cases, the user can also determine success and failure conditions, and which measurements the system needs to provide in order for the user to evaluate it. This makes it easier to analyse failures, because in a sufficiently complicated simulation, the accumulated data can simply become too vast. However, the test case specification here is done manually through a limited set of building blocks and the author stresses that these blocks are "representative", meaning they cover everyday scenarios. Going by real world tests with self-driving cars, uncommon situations have been a cause for accidents [15, 24] and one of the strengths of simulations is being able to easily create new situations. As such, there's merit in a system that tries to exhaust the possibility space and uncovers failures of a vehicle in edge cases.

This naturally leads to the topic of automatic test case generation, which has been applied to autonomous vehicles in various ways. Model-based test generation can be seen in [27, 5], but the complexity of the scenarios to be covered makes test case specification difficult, because essentially it boils down to having to model the real world [7]. Search-based approaches show more promise and one is shown in [36], which generate tests based on the observed behaviour of the SUT and tries to cover so-called "performance boundaries" – basically partitioning the search space by what causes significant changes in behaviour. This touches on another important part: Making individual tests meaningful, because automatically generating many tests that more or less cover the same thing is inefficient [28]. The topic of procedural content generation has also found its use in automatic test case generation already, with works like [6, 43] creating environments to challenge robot navigation. In those, the robot had to navigate through a mostly open terrain with obstacles and not complex road networks like in our case, but the works also include an evaluation of a challenge's difficulty for the robot, similar to my approach.

Regardless of manual or procedural generation, a simulation suite for autonomous driving will invariably include roads. Modelling roads in virtual space has been done many times for various applications, ranging from games to mapping [2]. In case of autonomous vehicles, the CommonRoad [4] project provides a standard data format to model roads that enables benchmarking motion planning. Roads are defined as a combination of customisable “lanelets”. Additionally, obstacles that have a movement trajectory are defined as part of the model. The goal of the test is specified via a goal region the car has to reach. From this specification, an optimal way to succeed can be inferred and the authors provide a representative dataset of roads for benchmarking motion planning. While their goal was to provide such a dataset, the data format they use will serve as the basis for my thesis' specification of roads. Furthermore, the authors outlined the conversion from CommonRoad to OpenStreetMap, making them compatible with more tools.

Besides the test scenario, test oracles, i.e. how a test is failed/passed, also need to be specified. While there are obvious binary criteria like the vehicle crashing, a more nuanced way of evaluation is presented in [31]. Here, on top of the aforementioned binary criteria, the authors define three scores: Smoothness, safety, and smartness. The three values summarise driving behaviour. Smoothness represents how comfortable the ride was, penalising sudden ac-/decelerations or sharp turns; safety measures how risky behaviour the car engaged in, such as getting too close to other vehicles during manoeuvres; and smartness indicates higher level functions, such as how early a vehicle detects traffic signs. These measurements give much more insight into the performance of the vehicle than the binary criteria and allow specification of more meaningful test oracles. In my thesis, I plan to elaborate those metrics to define fitness functions guiding search-based testing methods.

3 Method

This thesis will procedurally generate test suites that try to find flaws within a self-driving car's behaviour by confronting it with a diverse set of scenarios which try to elicit a broad set of manoeuvres by the car. For this, two new coverage metrics to evaluate the completeness of test suites and the behaviour of the car will be introduced, both of which will reward diverse input scenarios and output behaviour of the car with higher coverage values. These are then used to evolve test suites towards higher coverages. Very broadly, the routine will be:

1. Generate road layout the ego-car has to traverse
2. Import road layout into game engine for real-time simulation
3. Execute a test suite in simulation with the SUT to gather measurements
4. Evaluate how the SUT performed, computing output coverage
5. Generate a new road layout based on performance (and go back to step 2.)

For the generation of roads, I will rely on the polyline format employed in both the CommonRoad [4] and OpenStreetMap projects, which make the generated inputs easily compatible with multiple engines. As mentioned in the related work, the CommonRoad format is very conducive to benchmarking and was built to be convertible to the OpenStreetMap format, making it both useful and easily applicable to our purposes.

The first metrics introduced will be for the input space. It will take into account the possible combinations of lanelets, obstacles on them, and the manoeuvres they optimally require for traversal. What exactly goes into the formula with how much weight is unclear, but will obviously be examined and deliberated as part of the thesis. While the goal is to produce a test set that's meaningfully diverse, it's important to note that the generation will ensure the test cases are plausible and require achievable goals by the SUT. The measurements generation focus on should also be intuitive to humans to make it possible for testers to easily characterise a desired test suite through these metrics and then have my genetic algorithm evolve one converging towards those.

The thesis will try to stay engine-agnostic, mainly relying on standard formats for the roads and measurements that can easily be added to any engine for performance evaluation. However, during the thesis, BeamNG.drive [9] will be used, depending on which one turns out to be more suitable. The main requirement for any engine would be accepting roads as polylines or even OpenStreetMap format and being able to provide simple measurements such as the location of the car in regular intervals to be compatible with the tool. In our case, game engines such as BeamNG.drive are used because they are easily extendible and already provide a good simulation of driving, and works like [16] argue for game engines' merit in self-driving car testing. However, if this work proves to be promising, the same approach could be adapted to much more sophisticated industrial-grade simulation suites.

Tests will be executed within the game engine used, but the main task of the thesis happens before and after: Generating a road network and evaluating the car's performance. The evaluation metrics will build on the aforementioned CommonRoad benchmarking, but also the three measurements described in [31]. Since we're treating the vehicle's software as a black box, however, some of the factors going into the measurements are impossible to observe, such as how early a car detects a sign. How this affects the measurements/results will be examined in more detail. Identifying how certain road layouts affect the performance and how that can be used in search-based testing will also be part of the thesis.

After execution, the second coverage introduced becomes relevant. This metric will be regarding output coverage, i.e. what the test actually made the car do. The idea behind this is that, even if our input provides the car with an extremely diverse set of challenges which would require all manner of manoeuvres, that doesn't mean all of them will actually be performed by the vehicle. Since the goal is to cover as much behaviour as possible, this second metric will be used to indicate how much coverage was achieved.

The evaluation will then be used to evolve the test suite based on the existing tests and how the SUT performed in a way that improves the fitness of test suites. How the fitness for a test suite should be calculated and how changes to the road layout relate to changes in fitness will be investigated as part of this thesis. At the lowest level, the work will build on prior research in procedural content generation [18, 12, 14] to generate roads, but how those roads are laid out, how candidates are chosen, and so on, will be researched.

4 Evaluation

The goal of this thesis is to introduce new metrics for the input and output space of autonomous vehicles, with a focus on their motion planning and examine methods to generate test suites described by these metrics. How test suite evolution converges towards "fit" driving scenarios will be evaluated with a series of experiments running both the generation and simulation automatically. This will involve an implementation of the algorithms and adapting the BeamNG.drive [9] engine to execute test cases defined in the thesis and provide required outputs. For full automation, the simulation naturally requires a self-driving AI to be tested. However, it will likely not be possible to hook up an actual self-driving AI to the system, as the very few openly available AIs can't easily be adapted to run in such a virtual environment. Instead, an AI BeamNG.drive ships with will be used. One key difference between BeamNG.drive's AI and a real world self-driving one would be the possibility of perfect knowledge, since the BeamNG.drive AI can query the virtual world for information a proper self-driving AI has to infer, but this should have no effect on the evaluation of my thesis: no matter how a vehicle obtains the information used to make decisions, my fitness function and genetic algorithm should still observe the ego-vehicles behaviour and converge generation to fit test suites. The method is system-agnostic, so to speak, in a way, because, even for a human driver, the system should generate challenging scenarios using the driver's behaviour. As such, BeamNG.drive's AI suffices for my evaluation. With promising results, future work can look into adapting an AI used in the industry.

5 Schedule

Work on the thesis will start in the first quarter of 2018 and has an expected completion date around August. Most of the early efforts will be work on basic generation and import into BeamNG.drive. With that in place, work can continue to characterise the input and output spaces required for test suite evolution. Possible fitness functions and generation algorithms will be examined and implemented. These would be refined until they satisfy the required goals and then evaluated as described earlier. Depending on the results of the evaluation and time left, further refinement of the algorithms and fitness functions will be performed. While the actual thesis document will be written in parallel with the work, roughly one month of "full time" writing will be done at the end of the available six months.

Table 1: Thesis Schedule

Weeks	**Task**
8	Basic road generation and driving evaluation with BeamNG.drive
4	Devising input/output metrics
4	Test suite evolution based on SUT performance
4	Simulation refinement
4	Evaluation (both regarding metrics and generation)
4	Writing the thesis

6 Success criteria

The most important contributions of the thesis will be novel metrics for road networks and the manoeuvres of cars, a method of generating test cases for vehicles as road networks, and a way to adapt the road network based on the car's performance. Alongside those follow requirements such as generating tests that can actually be finished successfully and naturally evaluations of the work. It's important to note the limits, however, as a term like "driving scenario" can encompass an almost infinite amount of things. The base requirement for the scope of my thesis would be the generation of road networks themselves, with possible turns, intersections, and various numbers of lanes. If time allows, other vehicles that drive along the road as obstacles to avoid by the ego-vehicle will be added as well. Future work could take this generation even further, introducing pedestrians on the side of the road and the simulation required for those. My thesis, however, will mainly require diverse road networks and optionally include other traffic.

Features that would be nice to have like being able to run multiple tests in parallel or accelerating tests will be done depending on how much time is available.

What's explicitly not required is testing on an actual self-driving AI, for reasons discussed in the evaluation section. Since the focus of the work is on motion planning on roads, the generation and simulation will also only focus on aspects relevant to that.

Table 2: Summary of the Expected Thesis Features

Feature	**Must-Have**	**May-Have**	**Must-Not Have**
Metrics to characterise input (road network)	X	–	–
Metrics to characterise output (driving)	X	–	–
Dynamic road network generation	X	–	–
Dynamic traffic generation	–	X	–
Vertically overlapping networks	–	X	–
Support standard formats for roads	X	–	–
Generating tests that can actually succeed	X	–	–
Driving evaluation	X	–	–
Gathering measurements in modern game engine	X	–	–
Evaluation of coverage metrics	X	–	–
Evaluation of generation algorithm	X	–	–
Parallel test case execution	–	X	–
Performing tests faster than real time	–	X	–
Focus on irrelevant parts of simulation	–	–	X
Evaluation using real autonomous AI	–	–	X

References

[1] Mobileye | Autonomous Driving & ADAS (Advanced Driver Assistance Systems).

[2] OpenStreetMap.

[3] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen. Experience, Results and Lessons Learned from Automated Driving on Germany's Highways. *IEEE Intelligent Transportation Systems Magazine*, 7(1):42–57, 2015.

[4] M. Althoff, M. Koschi, and S. Manzinger. CommonRoad: Composable benchmarks for motion planning on roads. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 719–726, June 2017.

[5] Anneliese Andrews, Mahmoud Abdelgawad, and Ahmed Gario. Towards world model-based test generation in autonomous systems. In *Model-Driven Engineering and Software Development (MODELSWARD), 2015 3rd International Conference on*, pages 1–12. IEEE, 2015.

[6] James Arnold and Rob Alexander. Testing Autonomous Robot Control Software Using Procedural Content Generation. In *Computer Safety, Reliability, and Security*, Lecture Notes in Computer Science, pages 33–44. Springer, Berlin, Heidelberg, September 2013.

[7] J. Bach, S. Otten, and E. Sax. Model based scenario specification for development and test of automated driving functions. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1149–1155, June 2016.

[8] Gerrit Bagschik, Till Menzel, and Markus Maurer. Ontology based Scene Creation for the Development of Automated Vehicles. *arXiv:1704.01006 [cs]*, March 2017.

[9] BeamNG. BeamNG.drive.

[10] Christian Berger, Michel Chaudron, Rogardt Heldal, Olaf Landsiedel, and Elad M. Schiller. Model-based, composable simulation for the development of autonomous miniature vehicles. In *Proceedings of the Symposium on Theory of Modeling & Simulation-DEVS Integrative M&S Symposium*, page 17. Society for Computer Simulation International, 2013.

[11] M. Bertozzi, A. Broggi, A. Coati, and R. I. Fedriga. A 13,000 km Intercontinental Trip with Driverless Vehicles: The VIAC Experiment. *IEEE Intelligent Transportation Systems Magazine*, 5(1):28–41, 2013.

[12] Luigi Cardamone, Pier Luca Lanzi, and Daniele Loiacono. TrackGen: An interactive track generator for TORCS and Speed-Dreams. *Applied Soft Computing*, 28:550–558, March 2015.

[13] J. V. Casas, A. Torday, and A. Gerodimos. Combining Mesoscopic and Microscopic Simulation in an Integrated Environment as a Hybrid Solution. *IEEE Intelligent Transportation Systems Magazine*, 2(3):25–33, 2010.

[14] Guoning Chen, Gregory Esch, Peter Wonka, Pascal Müller, and Eugene Zhang. Interactive procedural street modeling. In *ACM transactions on graphics (TOG)*, volume 27, page 103. ACM, 2008.

[15] P. Fairley. Self-driving cars have a bicycle problem [News]. *IEEE Spectrum*, 54(3):12–13, March 2017.

[16] M. Feilhauer and J. Häring. A multi-domain simulation approach to validate Advanced Driver Assistance Systems. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1179–1184, June 2016.

[17] Jesus Felez, Joaquin Maroto, Jose M Cabanellas, and Jose M Mera. A full-scale simulation model to reproduce urban traffic in real conditions in driving simulators. *SIMULATION*, 89(9):1099–1114, September 2013.

[18] T. Georgiou and Y. Demiris. Personalised track design in car racing games. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8, September 2016.

[19] Jonathan M. Gitlin. Volvo's autonomous Drive Me research project gets underway, January 2017.

[20] D. Gruyer, O. Orfila, V. Judalet, S. Pechberti, B. Lusetti, and S. Glaser. Proposal of a virtual and immersive 3d architecture dedicated for prototyping, test and evaluation of eco-driving applications. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 511–518, June 2013.

[21] Dominique Gruyer, Sebastien Glaser, Benoit Vanholme, and Bertrand Monnier. Simulation of automatic vehicle speed control by transponder-equipped infrastructure. pages 628–633. IEEE, October 2009.

[22] T. Hanke, N. Hirsenkorn, B. Dehlink, A. Rauch, R. Rasshofer, and E. Biebl. Classification of sensor errors for the statistical simulation of environmental perception in automated driving systems. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 643–648, November 2016.

[23] D. Hospach, S. Mueller, O. Bringmann, J. Gerlach, and W. Rosenstiel. Simulation and evaluation of sensor characteristics in vision based advanced driver assistance systems. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2610–2615, October 2014.

[24] Shiyu Huang and Deva Ramanan. Expecting the Unexpected: Training Detectors for Unusual Pedestrians with Adversarial Imposters. 2017.

[25] Nidhi Kalra and Susan M. Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, December 2016.

[26] S. Khastgir, S. Birrell, G. Dhadyalla, and P. Jennings. Identifying a gap in existing validation methodologies for intelligent automotive systems: Introducing the 3xd simulator. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 648–653, June 2015.

[27] Martin Kienle, Benjamin Franz, Hermann Winner, Klaus Bengler, Marcel Baltzer, Frank Flemisch, Michaela Kauer, Thomas Weißgerber, Sebastian Geyer, Ralph Bruder, Stephan Hakuli, and Sonja Meier. Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance. *IET Intelligent Transport Systems*, 8(3):183–189, May 2014.

[28] S. Klingelschmitt, F. Damerow, and J. Eggert. Managing the complexity of inner-city scenes: An efficient situation hypotheses selection scheme. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1232–1239, June 2015.

[29] Philip Koopman and Michael Wagner. Challenges in Autonomous Vehicle Testing and Validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, April 2016.

[30] S. Lee, J. Kim, H. Song, and S. Kim. A 3-dimensional real-time traffic simulator considering the interaction among autonomous and human-driven vehicles. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1917–1918, October 2014.

[31] L. Li, W. L. Huang, Y. Liu, N. N. Zheng, and F. Y. Wang. Intelligence Testing for Autonomous Vehicles: A New Approach. *IEEE Transactions on Intelligent Vehicles*, 1(2):158–166, June 2016.

[32] D. Loiacono, L. Cardamone, and P. L. Lanzi. Automatic Track Generation for High-End Racing Games Using Evolutionary Computation. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):245–259, September 2011.

[33] S. Masuda. Software Testing Design Techniques Used in Automated Vehicle Simulations. In *2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 300–303, March 2017.

[34] Pascal Minnerup and Alois Knoll. Testing Automated Vehicles Against Actuator Inaccuracies in a Large State Space. *IFAC-PapersOnLine*, 49(15):38–43, January 2016.

[35] R. Molenaar, A. van Bilsen, R. van der Made, and R. de Vries. Full spectrum camera simulation for reliable virtual development and validation of ADAS and automated driving applications. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 47–52, June 2015.

[36] Galen E. Mullins, Paul G. Stankiewicz, and Satyandra K. Gupta. Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1443–1450. IEEE, 2017.

[37] M. Nentwig and M. Stamminger. Hardware-in-the-loop testing of computer vision based driver assistance systems. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 339–344, June 2011.

[38] Jae-Hyuck Park and Yu-Wing Tai. A simulation based method for vehicle motion prediction. *Computer Vision and Image Understanding*, 136:79–91, July 2015.

[39] E. Royer, F. Marmoiton, S. Alizon, D. Ramadasan, M. Slade, A. Nizard, M. Dhome, B. Thuilot, and F. Bonjean. Lessons learned after more than 1000 km in an autonomous shuttle guided by vision. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2248–2253, November 2016.

[40] Fabian Schuldt. Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen, Towards testing of automated driving functions in virtual driving environments. April 2017.

[41] Christoph Sippl, Florian Bock, David Wittmann, Harald Altinger, and Reinhard German. From Simulation Data to Test Cases for Fully Automated Driving and ADAS. In *Testing Software and Systems*, Lecture Notes in Computer Science, pages 191–206. Springer, Cham, October 2016.

[42] R. M. Smelik, T. Tutenel, K. J. de Kraker, and R. Bidarra. A declarative approach to procedural modeling of virtual worlds. *Computers & Graphics*, 35(2):352–363, April 2011.

[43] Thierry Sotiropoulos, Jeremie Guiochet, Felix Ingrand, and Helene Waeselynck. Virtual Worlds for Testing Robot Navigation: A Study on the Difficulty Level. pages 153–160. IEEE, September 2016.

[44] Daniele Sportillo, Alexis Paljic, Mehdi Boukhris, Philippe Fuchs, Luciano Ojeda, and Vincent Roussarie. An Immersive Virtual Reality System for Semi-autonomous Driving Simulation: A Comparison Between Realistic and 6-DoF Controller-based Interaction. In *Proceedings of the 9th International Conference on Computer and Automation Engineering*, ICCAE '17, pages 6–10, New York, NY, USA, 2017. ACM.

[45] Waymo. Waymo.

[46] C. Zhang, Y. Liu, D. Zhao, and Y. Su. RoadView: A traffic scene simulator for autonomous vehicle simulation testing. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1160–1165, October 2014.

[47] M. R. Zofka, R. Kohlhaas, T. Schamm, and J. M. Zöllner. Semivirtual simulations for the evaluation of vision-based ADAS. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 121–126, June 2014.