https://doi.org/10.2352/ISSN.2470-1173.2018.17.AVM-164

Virtual Simulation Platforms for Automated Driving: Key Care-About and Usage Model

Prashanth Viswanath, Mihir Mody, Soyeb Nagori, Jason Jones[*] and Hrushikesh Garud;
Automotive Processor, Texas Instruments Incorporated (India, [*]USA)
Email : { p-viswanath, mihir, soyeb, jasonatjones, hrushikesh.garud}

Abstract

Autonomous driving is an active area of research in the automotive market. The development of automated functions such as highway driving, autonomous parking etc. requires a robust platform for development and safety qualification of the system. In this context, virtual simulation platforms are key enablers for development of algorithms, software and hardware components. In this paper, we discuss multiple virtual simulation platforms such as open source car simulators, commercial automotive vendors and gaming platforms that are available in the market. We discuss the key factors that make the virtual platform suitable for automated driving function development. Based on the analysis of various simulation platforms, we end the paper with a proposal of two stage approach for the automated driving functionality development.

Introduction

Virtual simulation platforms have been used for many years in the driver assistance market. Various functionalities such as auto emergency braking, cruise control etc. are typically tested using the virtual simulation platforms. With the race to deploy fully autonomous cars, virtual simulation platforms are gaining popularity now, more than ever. Most car manufacturers have disclosed the usage of virtual platforms for modelling the environment and testing their systems. Their systems are made more robust by varying the simulation dynamics such as weather, lighting, object behavior etc.

The block diagram for autonomous driving is as shown in Figure 1. On the left of the block diagram, there are various sensor modalities such as cameras, RADAR, LIDAR which are input to the system. On the right side of the block diagram, we have the actuator signals such as steering, throttle and brake which are the output of the system.

The autonomous driving system consists of three main blocks: Sense, Plan and Act [12][13]. Sense block pertains to the perception of the environment around the ego-vehicle. The sense block also uses inputs such as GPS, maps and Inertial Navigation Systems (INS) which allow for precise localization of the ego-vehicle. The sense block provides an environment model around the ego-vehicle. Plan block pertains to the navigation planning of the ego-vehicle. It involves finding a suitable path which is safe, secure and in the direction of the destination. Act block generates the necessary actuator signals needed to navigate the ego-vehicle as desired by the planning block. The sense block is also referred to as perception task. The plan and act blocks are referred to as the navigation task. The perception and navigation tasks are inter-related. Based on the navigation commands, the perception of the world changes and based on the perception of the world, navigation commands are issued. Hence, it is important to test both these tasks together which can be achieved using virtual simulation platform, with no safety liability.

Virtual simulation platforms have several advantages compared to real world data. Few of them are as listed below:

- **Data generation:** Virtual simulation platform allows us to obtain huge amount of data along with the ground truth/ labelled semantics. With the advent of machine learning/deep learning based algorithms for tasks such as object detection,

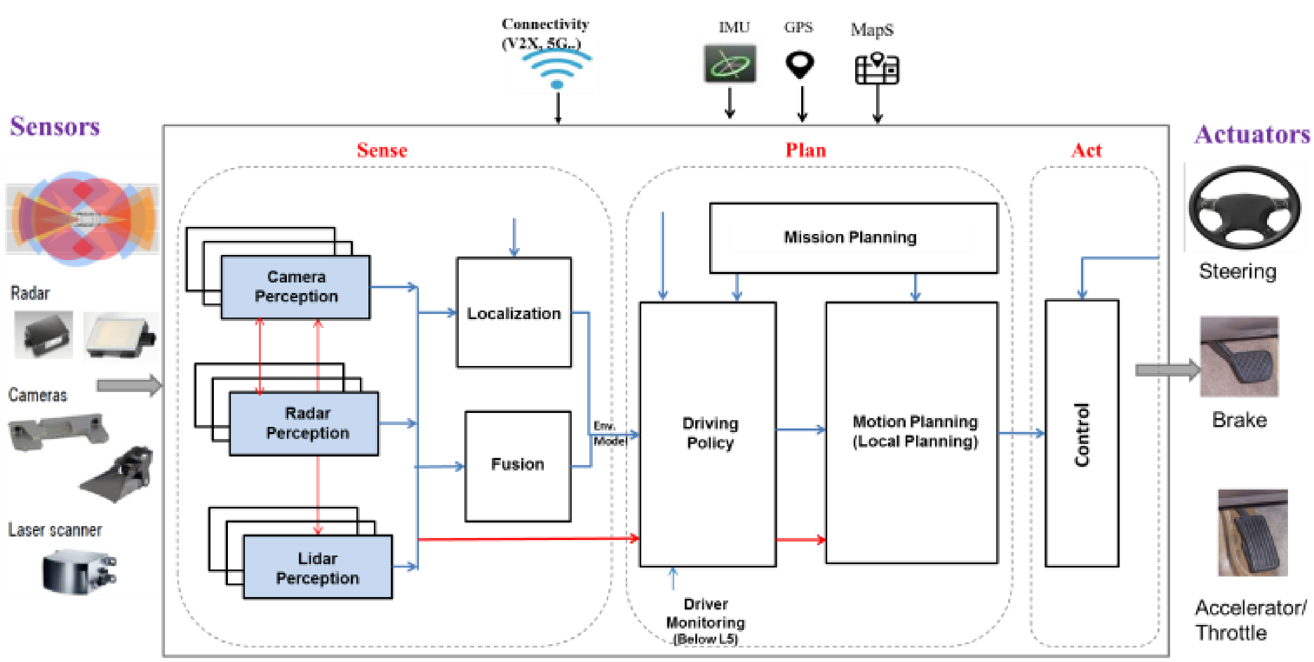

Figure 1: Block Diagram of Autonomous Driving

semantic segmentation etc., input data with ground truth label is critical for training the networks [1] [2].

- **Scenario creation:** Virtual simulation platform provides flexibility to generate specific scenarios and corner cases, which would otherwise take decades to obtain in real world. These corner cases are more interesting for algorithm development and testing. The virtual simulation also allows obtaining variety of data across different weather conditions, lighting conditions, and road conditions etc., which are critical for automotive safety validation of systems.
- **Time to prototype:** Virtual simulation platform allows quick prototyping from concept to product. In a real world, one has to first mount the sensors at the appropriate locations on the car, calibrate these sensors, tap into the drive-by-wire to obtain the actuator signals from the car and decode the CAN messages, which is a very laborious and iterative activity. Also, once real car is involved, appropriate license needs to be obtained from the governing bodies.
- **Inexpensive compared to real car:** Virtual simulation platform is inexpensive compared to a real car which has various sensors mounted on it, and the infrastructure to tap into the relevant data from the car.
- **Reinforcement Learning based policy learning:** Reinforcement learning based driving policy is becoming an important step of autonomous driving. In the past, state machine based approach was used to derive the driving policy that the car has to follow. However, with the growing traffic complexity, reinforcement learning based mechanisms are being pursued for learning a suitable driving policy in complex traffic. Virtual simulators are a great platform for reinforcement learning based algorithms with no safety liability [3] [4] [5] [6] [7].

However, virtual platforms can only help in achieving certain level accuracy, which can never be 100%. In order to obtain 100%, real world data is necessary, unless the simulation data is very close to real data in itself.

In this paper, we discuss the key factors that make virtual platforms suitable for automated driving development. We also provide analysis on some available virtual platforms. Based on this study, we propose a two stage approach for the automated driving functionality development.

The rest of the paper is organized as follows: Section *Virtual Simulation Environment* provides the key care-about of simulation platforms and analysis, Section *Proposed Solution* provides insight into our two stage approach of validating automated driving functionality and Section *Conclusion* provides conclusion.

Virtual Simulation Environment

Virtual Simulation platforms have been used by car manufacturers for decades now. Car manufacturers use virtual simulation to run simulations of their algorithms such as Auto Emergency Braking (AEB), Automatic Cruise Control (ACC) etc. and compute estimated time to collision and tune their algorithms accordingly, to make it robust, efficient and safe.

Figure 2 shows the various blocks involved in the virtual simulation platform for testing autonomous vehicles. Virtual simulation involves modelling of various components that make the platform as realistic as possible. The various blocks involved in modelling virtual platform are as follows:

- **World:** Modelling the world involves modelling various stationary components of the world such as trees, buildings, traffic lights, traffic signs, detailed road layout such as intersections, lane markings etc., rules specific to various countries such as left/right hand drive etc., weather conditions and lighting conditions and how the environment would change based on it.
- **Sensors:** Various sensors such as camera, LIDAR, RADAR, INS, GPS, ultrasound etc. that are fitted on the ego-vehicle should be modelled. These sensors should be as close to the real sensors that are available in the market as possible. The sensors play a critical role in the perception block as

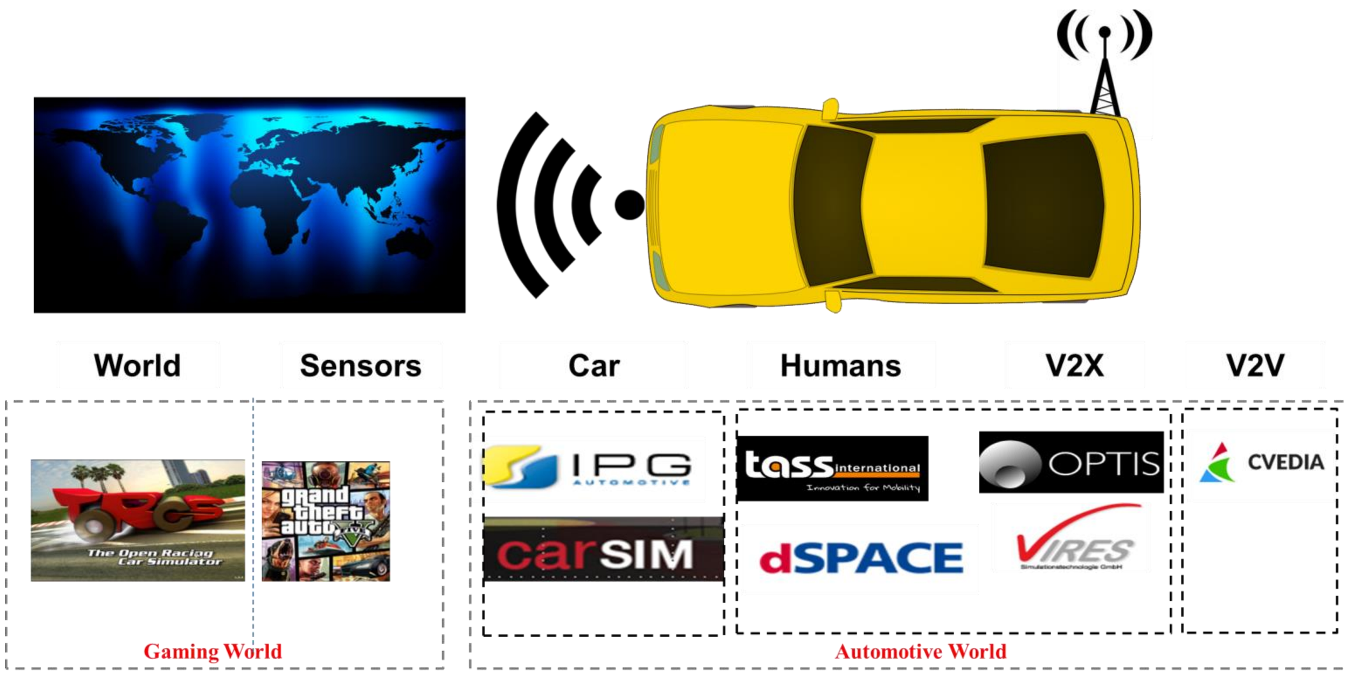

Figure 2: Block Diagram of Virtual Simulation Environment

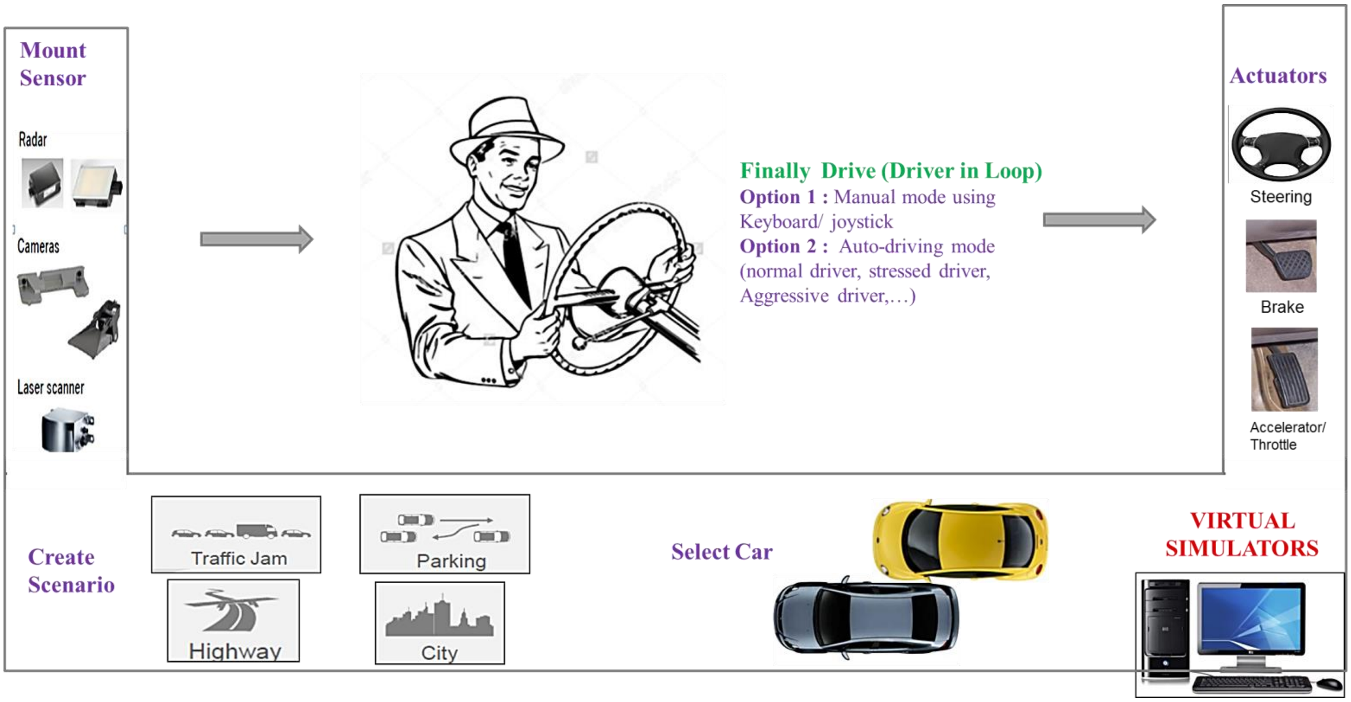

Figure 3: Default Usage of Virtual Simulation Platform

discussed earlier.

- **Car:** Modelling the vehicles in the scene including the ego-vehicle is a crucial factor in analyzing a particular scenario. This involves modelling the vehicle dynamics such velocity, acceleration, trajectory, tire pressure, aeroynamics etc.
- **Humans:** Humans are an integral part of the simulation environment. The ultimate goal behind autonomous driving is to save human lives and make their commute productive. Modelling humans involves modelling pedestrians and animals crossing the street etc.
- **V2X/V2I:** Vehicle to vehicle infrastructure connectivity can simplify the problems for autonomous cars. For example, if the vehicle is connected to traffic light/traffic signs, then detecting them may not be very critical as the vehicle would be informed of their position and state well in advance.

There are various virtual simulation platforms available as shown in Figure 2.

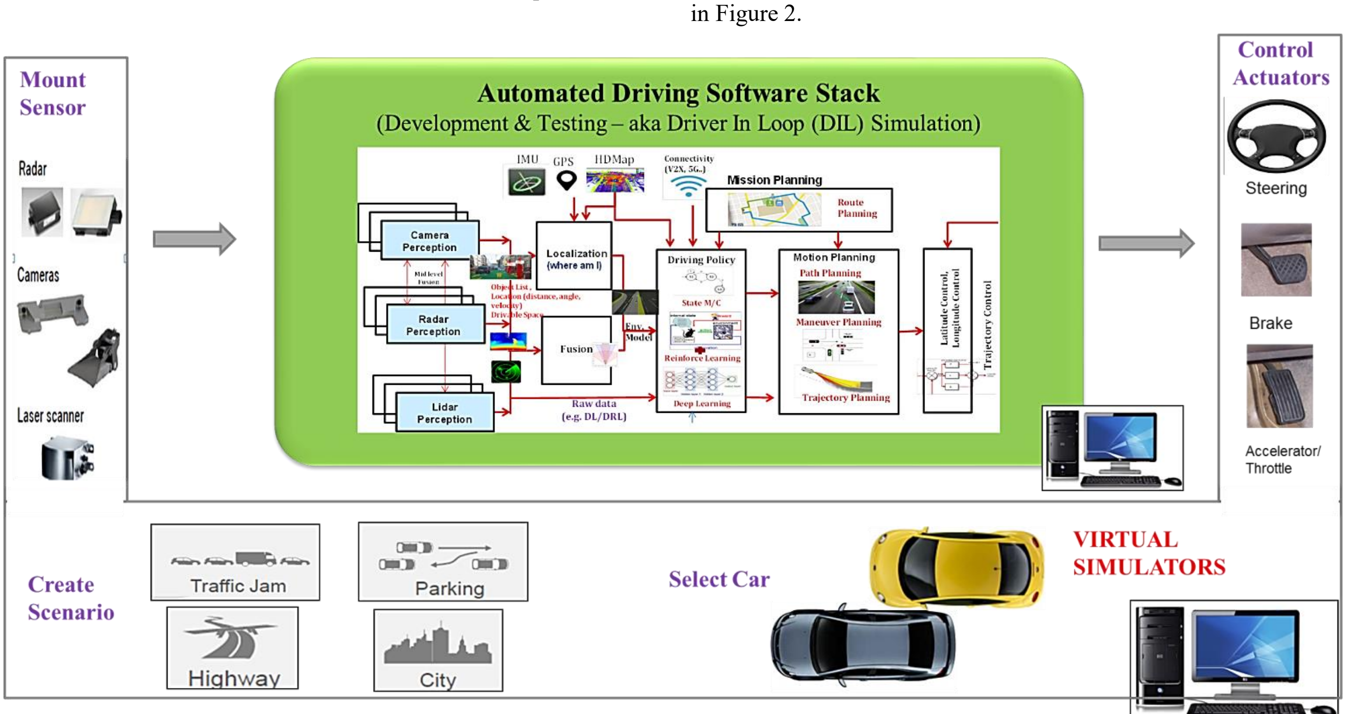

Figure 4: Algorithm Usage Model in Virtual Simulation Platform

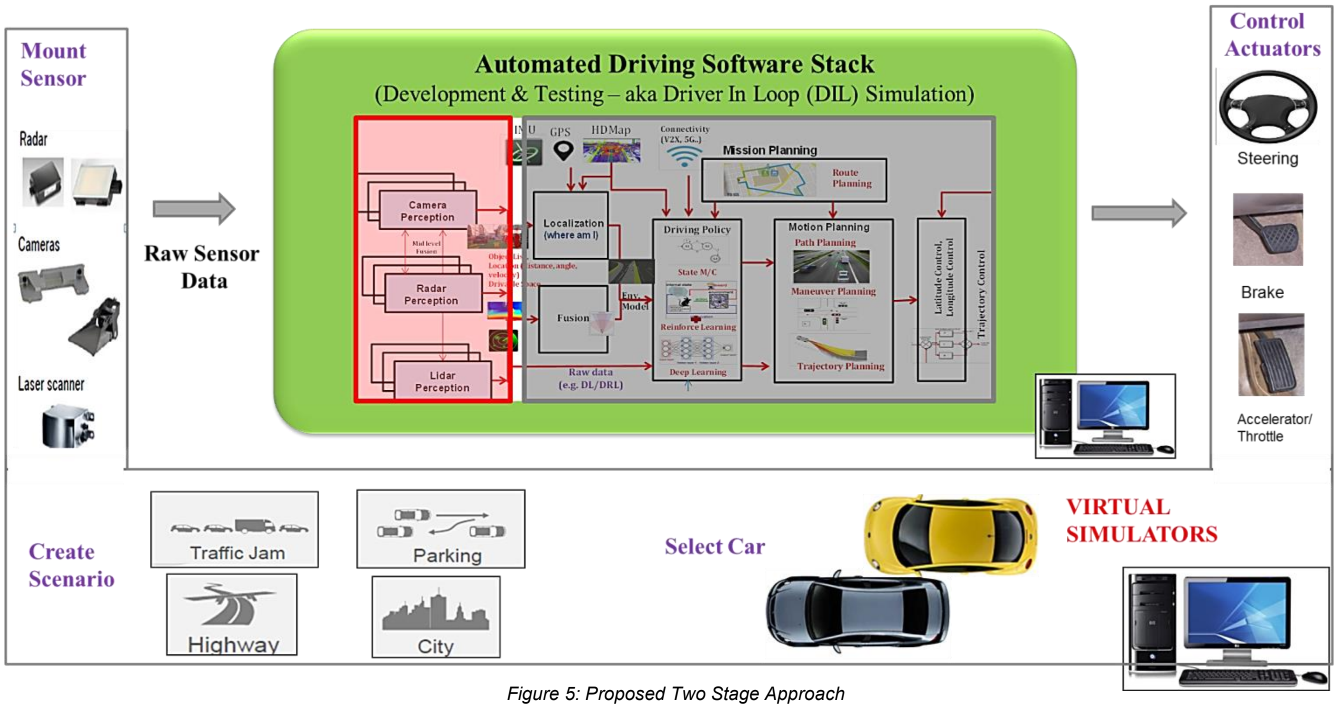

Figure 5: Proposed Two Stage Approach

- **Gaming Platforms:** Gaming platforms have grown in popularity due to recent advancements in reinforcement learning based autonomous driving algorithms [4] [5] [6] [7]. The Open Racing Car Simulator (TORCS) is an open source platform which is very popular in the research community and has been widely used for developing and testing machine learning and reinforcement learning based algorithms [3] [6] [7]. Although the graphics of the game is not very realistic, it has been used to train networks as it provides abundant data for training. Grand Theft Auto – V (GTA-V) is another popular platform used in the research community for developing machine learning and reinforcement learning based algorithms for autonomous driving [1]. The gaming community has developed various mods which allows to get access to various data from the game such as camera feed, depth buffer, object annotations, vehicle controls such as steering, throttle, braking etc. [8].
- **Automotive Simulator Vendors:** There are multiple automotive vendors who have developed simulation platforms with varied focus areas. IPG Automotive's *Carmaker* platform and *Carsim* platform's key focus area include modelling the vehicle dynamics. Vendors such as TASS International, Optis, Vires and dSpace platform's key focus area include modelling the sensors such as cameras, radar, ultrasound etc. and the environment. There is a provision of using multiple platforms in tandem. For example, *Vires* platform can be integrated with *Carsim* and *Carmaker* to obtain a precise vehicle model from them and use it with accurate sensor model from *Vires*. Also, there are many startups who see the value in providing a good simulation platform such as CVEDIA's *Syncity*. Recently, there have been open source platforms such as Carla and Airsim, which are getting popular in the research community. There could be many more simulation platforms, which are not discussed here, due to brevity.

The default usage modality of the virtual simulation platform is shown in Figure 3. This stage is referred to the data collection phase. This stage does not involve running any external software or algorithm. As shown in Figure 3, there are two options of driving in the simulation platform.

1. Manual model: In this mode, the user should be able to control the ego-vehicle in the simulation platform using a keyboard/joystick. This is important in collecting ground truth data for behavior cloning type algorithms.
2. Auto-driving mode: In this mode, an in-built engine should be able to drive the ego-vehicle in the simulation platform automatically. This is useful in collecting ground truth data for reinforcement learning and other supervised learning based algorithms.

Figure 4 shows the second phase of usage of the virtual simulation platform. In this phase, the algorithms/software shown in Figure 1 is responsible for controlling the ego-vehicle. This mode is critical for validation of the algorithms and software components. It is also referred to as Software-In-Loop (SIL) testing. The software and algorithms can be running on PC or an embedded platform and thus, same model can be used for Hardware-In-Loop (HIL) testing.

The key requirements that make a simulation platform viable are:

1. Sensor support: The simulation platform should support various sensor types such as camera, RADAR, LIDAR, ultra-sound, GPS, INS etc. The platform should also provide flexibility in tuning the parameters for these sensors such as placement around the ego-vehicle, the Field of View (FoV) etc.
2. Photo-Realism: The camera images rendered by the simulation platform should be close to real world images. Some of the vendors support physics based camera model

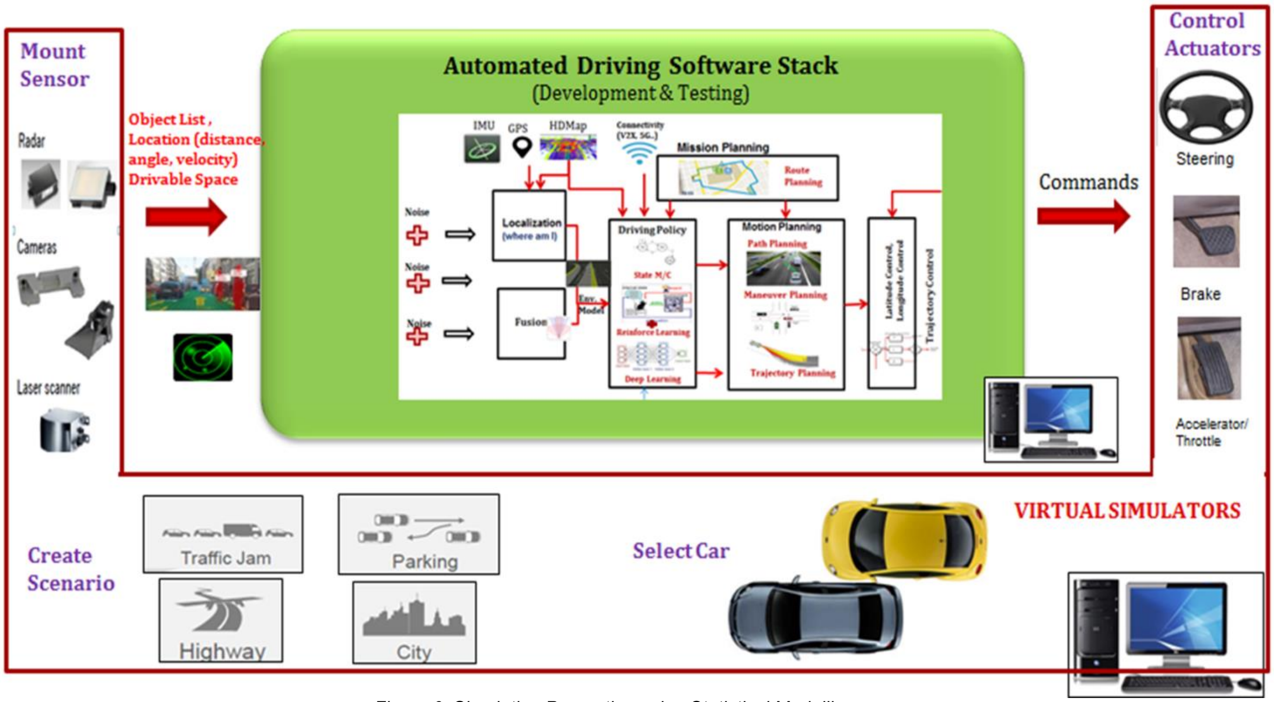

Figure 6: Simulating Perception using Statistical Modelling

which provides realistic simulation of camera images. The neural networks which are trained using the data from simulation platform should be generalizable to real world data.

3. Scenario Generation: The simulation platform should provide flexibility to create various scenarios. It should also include certain pre-built scenarios such as highway driving, parking etc. in order to save time and effort to prototype algorithms.
4. Real-time Usage: The simulation platform should be able to run real-time and allow for SIL and HIL testing in real-time.
5. Ground truth generation: The simulation platform should have ability to generate ground truth data for object segmentation, depth, car control signals, maps etc.

Currently all the virtual simulation platforms available in the market have gaps in meeting the above requirements. The traditional simulation platforms are good at scenario generation, sensor support and ground truth generation. However, the environment modelling is not very realistic and does not allow for generalization to real world data. If there is any improvement in the photo realism like physics based camera model integration, then the simulation platform can no longer meet the real-time constraints. The gaming platforms do not provide flexibility to create scenarios and does not support various sensor modalities. However, they provide realistic data and can be used for real-time demonstration. Various startups are promising simulation platforms which meet all the requirements listed above and also provide demonstrations of the same. Unfortunately, none of them are fully developed and available for usage yet, but should be available soon.

Proposed Solution

Since there is no single simulation platform available to test the perception and navigation modules together, we propose to use a two stage approach as shown in Figure 5. We propose to develop and test the perception task of the block diagram separately from the rest of the chain. The perception algorithm block can be developed and tested using raw sensor data from real world. This enables the use of existing datasets from public domain such as KITTI [9], Cityscapes [10], Oxford RobotCar [11] etc. This also expedites the development time of the perception algorithms as these datasets are well maintained and robust. However, the labelling of these data would have to be done if it has to be used for supervised learning type of algorithms.

The rest of the chain can be developed using any of the simulation platforms already discussed above. The input to the simulation module would be an object list (which includes object position, velocity), drivable space (semantics), which is typically the output of the perception module. These variables can be statistically modelled along with noise terms and fed as input to the rest of the blocks such as fusion, localization etc. as shown in Figure 6. Almost all simulation platforms available today allow for statistical modelling in their simulation environment.

Another approach is to combine simulation platform with real world data. Many simulation platforms allow capturing real world data and simulating this in the virtual environment. This approach allows capturing the detailed environment of the real world such as lane markings, intersections, traffic signs, traffic lights, sidewalks, parking garages etc. in the virtual environment. These approaches are suitable for applications like autonomous parking, where the area is limited and contain mostly stationary obstacles. Since the area is limited, the environment can be made more realistic as it would require less time and effort. Hence, this approach can allow end-to-end testing of both perception and navigation modules. However, this approach would be laborious to scale to larger areas.

The proposed two stage approach allows us to develop all the perception and the navigation blocks independently, without having to test on a real car. Once a single simulation platform is made available, the algorithms developed using the above approach can be put together and tested end-to-end as it would be done in a real car.

Conclusion

In this paper, we have presented the key requirements for using virtual simulation platforms in developing software and algorithms for autonomous driving. We have presented the details that need to be modelled to make a virtual simulation platform a viable solution for software development and testing. We have also presented the usage models of the virtual simulation platform, both in data collection or development phase and testing phase. Since there are gaps in meeting all the requirements by virtual simulation platform available in the market today, we have proposed a two stage approach for developing software and algorithms for autonomous driving to achieve near term goals. As the industry focus and innovation increases in this domain, a simulation platform which can meet all the requirements should be available very soon.

References

[1] S. R. Richter, V. Vineet, S. Roth and V. Koltun, "Playing for Data: Ground Truth from Computer Games", European Conference on Computer Vision (ECCV), vol. 9906, pp. 102-118, 2016.

[2] G. Ros, L. Sellart, J. Materzynska, D. Vazquez and A. Lopez, "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes, Computer Vision and Pattern Recognition (CVPR), 2016

[3] B. Wymann, E. Espie, C. Guionneau, C. Dimitrakakis, R. Coulom and A. Sumner, "TORCS, The Open Racing Car Simulator", http://www.torcs.org, 2014.

[4] S. Shalev-Shwartz, N. Ben-Zrihem, A. Cohen and A. Shashua, "Long-term Planning by Short-term Prediction", arXiv: 1602.01580 [cs], Feb. 2016.

[5] S. Shalev-Shwartz, S. Shammah and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving", arXiv: 1610.03295, 2016

[6] A. E. Sallab, M. Abdou, E. Perot and S. Yogamani, "End-to-end deep reinforcement learning for lane keeping assist", arXiv: 1612.04340, 2016.

[7] A. E. Sallab, M. Abdou, E. Perot and S. Yogamani, "Deep reinforcement learning framework for autonomous driving", Autonomous Vehicles and Machines, Electronic Imaging, 2017.

[8] A. R. Miralles, S. K. Izquierdo and C. V. Royo, "An open-source development environment for Self-driving vehicles", Master's Thesis Report, Open University of Catalonia, 2017.

[9] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite", Computer Vision and Pattern Recognition (CVPR), 2012.

[10] 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", Computer Vision and Pattern Recognition (CVPR), 2016.

[11] W. Maddern, G. Pascoe, C. Linegar and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset", The International Journal of Robotics Research (IJRR), 2016.

[12] S. Jagannathan, M. Mody, J. Jones and P. Swami, "Multi-sensor fusion for Automated Driving: Selecting Model and Optimizing on Embedded Platform", AVM track, Electronic Imaging, 2018.

[13] M. Mody, J. Jones, K. Chitnis, R. Sagar, G. Shurtz, Y. Dutt, M. Koul, M. G. Biju and A. Dubey, "Understanding Vehicle E/E Architecture Topologies for Automated Driving : System Partitioning and Tradeoff Parameters", AVM track, Electronic Imaging, 2018.

Author Biography

Prashanth Viswanath received his MS in Electrical Engineering from the University of Houston (2009). Since then, he has worked in the automotive group at Texas Instruments Incorporated in Houston and Bangalore. His work is focused on hardware and software development for Advanced Driver Assistance Systems.

Mihir Mody is Senior Principal Architect for Automotive business at Texas Instruments (TI) Incorporated. His domains of interests are image processing, computer vision, video coding, machine learning and deep learning algorithms with focus on embedded HW & SW solution. He holds Master of Engineering (ME) degree from Indian Institute of Science (IISc) in 2000 and Bachelor of Engineering (BE) from GCOEP in 1998.

Soyeb Nagori received bachelor degree from S.V. NIT, Surat and masters from electrical engineering department, IISc, Bangalore. He is with Texas Instruments for over 16 years and working as Senior Member of Group Technical Staff (SMTS). During his stay at TI, he has been involved in realizing various multimedia algorithms on different TI embedded processors. He has been involved in development of Video Encoders on various DSPs and HW Accelerators based video platforms. Last few years he has been focusing on exploring computer vision algorithm space with special interest in 3d reconstruction using monocular camera and object detect using machine learning for advanced driver assistance systems (ADAS). Currently he is working as ADAS Algorithm Lead. He holds 27 US granted patents in the area of video encoding and vision algorithm for embedded system and has 23 other patents pending in USPTO.

Hrushikesh Garud received the B.E. degree in electronics engineering from Dr. B. A. Marathwada University, India in 2005 and M.Tech degree in from Indian Institute of Technology Kharagpur, India in 2008. He is currently working as Technical Lead (Computer Vision Algorithm Development) with Texas Instruments India, Bangalore. Overall, Hrushikesh has 8+ Years of Work/Research experience in optical imaging, image and video signal processing, image analysis and computer vision algorithm development. His current research interests include- image and multidimensional signal processing, computer vision and its automotive applications and medical imaging and analysis. Hruhsikesh has long experience in scientific writing and publication, where he has authored more than 17 research papers published in peer-reviewed journals and conferences, and 6 granted US patents. Hrushikesh has delivered six invited talks at various conferences and workshops on different aspects of image processing and computer vision.

Jason Jones is the Principal Systems Architect for Automotive Business at Texas Instruments (TI) Incorporated, responsible for current and future roadmap solutions encompassing Digital Cockpit, ADAS, and Vehicle Gateway applications. He holds Bachelor of Science (BS) degree from Texas A&M University in 1993.

Free access to this paper is brought to you with the generous support of ON Semiconductor.

All research funding for this paper is referenced in the text; unless noted therein, no research funding was provided by ON.