Close to reality surrounding model for virtual testing of autonomous driving and ADAS

F. Chucholowski, C. Gnandt, C. Hepperle
TESIS DYNAware GmbH

Sebastian Hafner

ABSTRACT: For the development of Advanced Driver Assistance Systems (ADAS) and Autonomous Driving, cost efficient virtual methods to understand the vehicle behavior within a complex environment are needed. These methods require an easily configurable virtual world as well as realistic illumination, reflection and weather conditions. Further the connectivity to existing ADAS development toolchains is significant for the success of these methods. To face these challenges, the paper presents an approach that uses technology of real-time computer graphics originally developed for computer games in combination with the execution of ADAS simulation systems. To highlight the value of this alternative approach, driving scenarios for snowy and rainy weather conditions are presented. Further simulation of ultrasonic sensors for testing parallel parking assist functions and the setup of a virtual development environment for a lane keeping assist are shown.

1 INTRODUCTION

The development of ADAS leading to autonomous vehicles is proceeding rapidly. By sensor fusion of vehicle data with vehicle surrounding data a complete and actual environmental model for the systems is provided to realize more and more functions intervening in the driving of vehicles, avoiding dangers and freeing drivers from tedious tasks. When in September 2013 a Mercedes-Benz S 500 INTELLIGENT DRIVE from the Daimler AG drove 100 km from Mannheim to Pforzheim fully automated, the control algorithms have to manage 3 major cities, 23 towns, 18 traffic circles and 155 traffic light crossings on the route. Among systems targeting highly automated driving (VDA 2015) there are also driver assistance systems, such as traffic sign recognition or lane keeping assist (Fu & Huang 2010, Gunia & Schüling 2011). Due to the complexity of the multitude of possible scenarios the requirements for development and test methods are vastly increasing.

Besides typical time and cost issues there are several other aspects that favor the usage of virtual development methods. Testing of system behavior in life-endangering situations, repeatability of maneuvers and what-if analysis are just some of the major advantages development methods based on simulating the real world in comparison with development purely based on measurement data derived from field tests.

Simulations of the environment in common ECU vehicle simulations e.g. in the field of vehicle dynamics and drivetrain are based on tire-road-contact, external forces and driver input. For interaction with the vehicle's full surrounding, a complex environmental model – a complete virtual world – is needed.

Most often the capabilities of established simulation approaches are exceeded. In modern vehicles, cameras and other sensors based on time-of-flight technology gather information of the surrounding environment. In this process, position and motion coordinates, color information of the road and infrastructure, obstacles and other traffic are measured and interpreted. The more accurate the measurement is, the closer the results of interpretation will be to the real world. For high simulation accuracy, elemental geometrical and optical details, as well as object visibility considering sensor physics and interferences are important. Thus idealization and simplification for virtual testing in real-time are not an adequate option anymore.

In this paper an approach that uses technology of real-time computer graphics originally developed for computer games is presented. The solution is evaluated for typical ADAS and autonomous vehicle development use-cases regarding feasibility and economic efficiency.

The most basic requirements are summarized in section 2. Details about the transition from pure computer graphics to technical simulation can be found in section 3. Section 4 describes the integration of the presented solution into vehicle simulation tool chains based on two typical use-cases and section 5 summarizes the results of the paper.

2 REQUIREMENTS

In this section typical but important and challenging requirements concerning environmental modeling in the field of ADAS development are described.

2.1 *Configurable Virtual World*

Control algorithms in the field of ADAS and automated driving become more and more intelligent. Early applications were Adaptive Cruise Control with idealized radar sensors (Reif 2010), lane detection on a camera Hardware-in-the-Loop test bench (Gunia & Schüling 2011) or development of a park assistance system based on ultrasonic sensors (Jeong et al. 2010). The environmental model was rigid and characterized by technical objects. This was an acceptable accuracy for the systems being tested due to their relatively simple logic.

Contrary to this, a human driver uses a lot more information. He can e.g. predict the intention of a pedestrian by its viewing direction, facial expression and moving behavior. Control algorithms for autonomous vehicles are adopting these skills e.g. for interaction with unprotected traffic participants. This implies that a lot of details should be part of the surrounding model. Animation of traffic participants should be possible to fulfill this requirement with minimum effort in scenario generation.

2.2 *Realistic illumination, reflection and weather conditions*

As long as human drivers are steering vehicles, they rely on good sight and road illumination at night. The different sensors that are used also rely on information they gather from their surroundings. Therefore it is crucial that all inputs for sensors are as realistic as required. Light, reflection but also sight impairment caused by different weather conditions such as rain or snow have to be included in the environmental modeling approach. Also the brightness of variable message signs, which is dependent on the viewing angle, should be simulated to reduce their time span of visibility as they are passed.

2.3 *System connectivity and modularity*

Different development tools and approaches are available for virtual ADAS development. Each one of them contributes its specific features to the overall development toolchain. Similar to ADAS and functions that are connected in the vehicle these tools must also exchange data in a synchronized way. Therefore the environmental model should provide standardized interfaces and should be able to receive and send information from and to other systems.

Control algorithms are using sensor signals. Due to different development stages and scenarios at OEM and Tier-One companies the sensors can be part of the overall simulation approach or are integrated as HiL components. Thus the system architecture of an environmental model should support all different scenarios. Particularly regarding sensor and control algorithm information the design should be as modular as possible.

3 SURROUNDING MODEL TECHNOLOGY

Computer games process user inputs and draw images on screens in short intervals to allow interaction of the player with a virtual world. Instead of sending the images to a monitor, the data of a virtual vehicle camera in a traffic scene can be used as input for a control unit. Real-time computer graphics use numerous techniques, which have been optimized over decades, to achieve one goal: photo realistic rendering with high image rate. Therefore, only effects which have a direct impact on any visible pixel are considered for the rendering process. The sample space finally is a RGB-tuple for every pixel per time step. The more important elements are the background calculations which are essential for the image quality and level of correlation to reality. The intermediate rendering processes must be radically modified to emulate physical behavior which is different for every kind of sensor. Additionally they provide information which is directly used for technical simulations. The approach presented in this paper is based on the game development platform Unity (Unity Technologies 2016).

3.1 *Surface shading*

Visible images originate from propagation of light and spectral reflection on surfaces. The various kinds of reflections defined by material parameters are calculated by shaders. Shaders are programs that describe the characteristics of either a vertex or a pixel for a desired appearance. These rendering effects are calculated on the GPU providing high performance and a high degree of flexibility (Randima, & Kilgard 2003). Therefore it is possible to model several physical effects, which are disregarded in most computer games in favor of appealing graphics. According to the required model accuracy, objects and surfaces are defined by varied detail of meshes, surface mapping, or pre-rendered textures. To solve conflicting performance and accuracy demands, these methods are switched automatically during simulation based on view distance and other parameters (Unity Technologies 2016).

3.2 *Fluids*

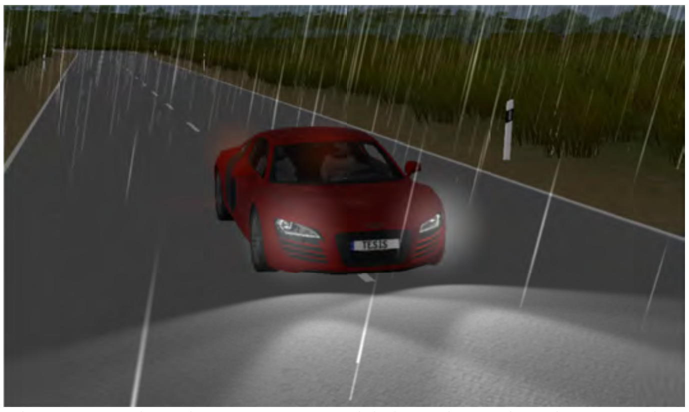
Figure 1. Virtual test scene - driving through rainy weather conditions

A fluid can be simulated using shaders if it has a considerable size of surface like a wet spot on a road or water beside the road. Another important feature is precipitation, like rain fall or a cloud of leaves. These are thousands of small, nearly identical and randomly spread objects with similar movement pattern. Particle systems from game engines facilitate simulation of these effects. The particles can be any size and shape, thus different kind of snowflakes, for example, can be configured and used in a winter scene. Existing configurations of weather elements from games are optimized for human perception including motion blur and other secondary effects.

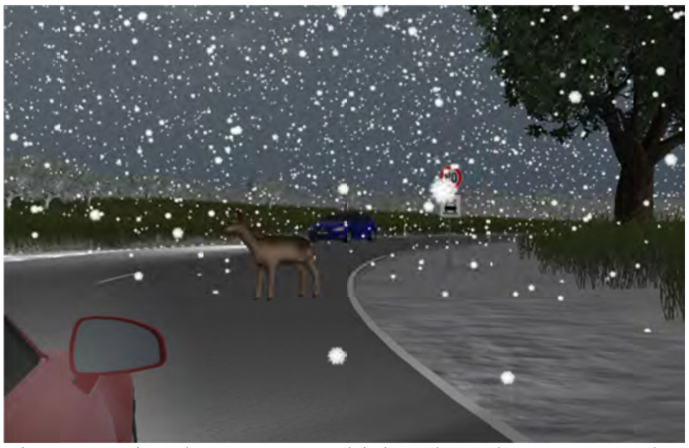
Figure 2. Virtual test scene - driving through snowy weather conditions

Figures 1 and 2 show typical virtual test scenes where automated driving systems are challenged with different weather and ambient conditions. Since motion of particles affects all kinds of sensors and cameras differently, the particle systems must also be modified according for each of them. The particles fail to be realistic if they are too close to a virtual camera, especially after a collision with the windshield or the sensor. This is why noise emulation is used for modelling of the pollution.

3.3 *Complete toolchain*

Precipitation is one kind of animation, but there is a lot more motion in the vehicle's surrounding. Besides the obvious proper motion of the vehicle, there are traffic participants, infrastructure, branches of trees waving in the wind, and much more that might affect the algorithms interpreting the situation. This represents a particular advantage when using a game engine: Motion of ductile surfaces like clothing of pedestrians, branches and leaves are supported by default. Object oriented design allows to add any number of objects to one scene and to let them interact. Finally the visualization of instruments for driver assistance, characteristic values and simulation data is essential for interpretation of development results. Figure 3 shows a typical setup for validation of a parallel parking system.

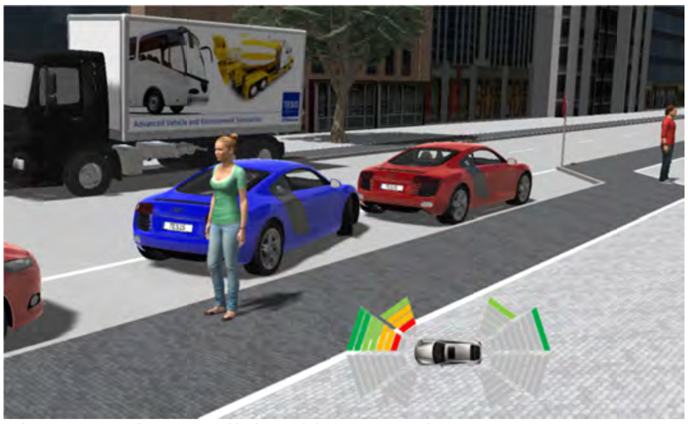
Figure 3. Urban parallel parking scenario.

4 INTEGRATION IN VEHICLE SIMULATION TOOL CHAINS

Today's driver assistance systems are mostly integrated into the vehicle control systems via closed-loop. This is why simulation of these systems needs a complete vehicle model. Existing vehicle models are based on common simulation platforms like Simulink. Nevertheless the vehicle must interact with the surroundings simulated with computer graphics. Therefore the surrounding model is connected to existing simulation platforms in order to provide a complete virtual development environment. Vehicle and traffic simulation deliver precise results of vehicle motion. Secondary motion patterns, like spinning wheels and walk cycles of pedestrians, are calculated by the animation of the game engine.

Figure 4 illustrates a common system setup. UDP and TCP/IP network connections are used for connection and synchronization. Via its synchronized data exchange the animation can be connected to MiL, SiL or HiL systems for development and testing of control unit algorithms. The connection with other simulation systems such as DYNA4 (TESIS DYNAware 2016) is realized by common interfaces like Simulink s-functions, FMI or XML. When connected with DYNA4 scenario setup, open-drive road definitions and automatic terrain generation can be utilized. The overall system provides different possibilities for ADAS and autonomous vehicle virtual development and can be enhanced for future challenges as well.

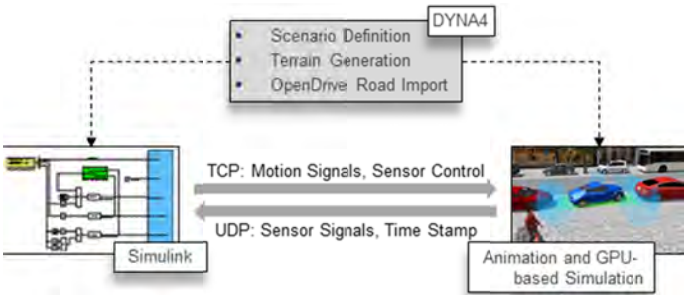
DYNA4
Scenario Definition
Terrain Generation
OpenDrive Road Import
TCP: Motion Signals, Sensor Control
UDP: Sensor Signals, Time Stamp
Simulink
Animation and GPU-based Simulation

Figure 4. Overall system

4.1 *Application example: parallel parking assist*

Under specific conditions it has been possible to simulate a Time-of-Flight sensor on the GPU by using a special virtual camera together with an adapted shader (Hafner 2015). The approach allows the use of common hardware without the loss of real-time capability. The animation is part of the complete modeling approach and synchronizes simulated sensor data with the rest of the overall model. The significant features and effects of ultrasonic sensors can successfully be reproduced in virtual tests and the new economic efficiency method fills the gap between common serially solved model equations and costly methods like ray-tracing (Hafner 2015). Figure 5 shows the rendered depth by the GPU-supported shader technology. Due to the promising results the approach will be transformed to other sensor physics like radar or laser scanners in the future.

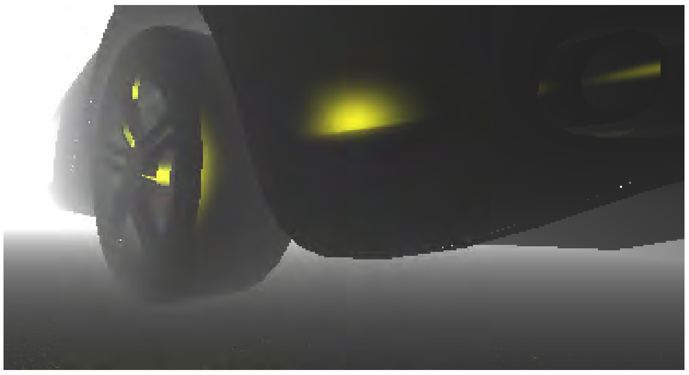

Figure 5. Rendered depth by GPU-supported shader for ultrasonic sensor.

4.2 *Application example: lane keeping assist*

A common approach to test optical control devices is a HiL test bench where virtual scenarios are displayed on a screen. The display is used as input for the optical ECU. This approach has some limitations e.g. for use-cases like driving in the night and blinding effect of headlights from oncoming vehicles.

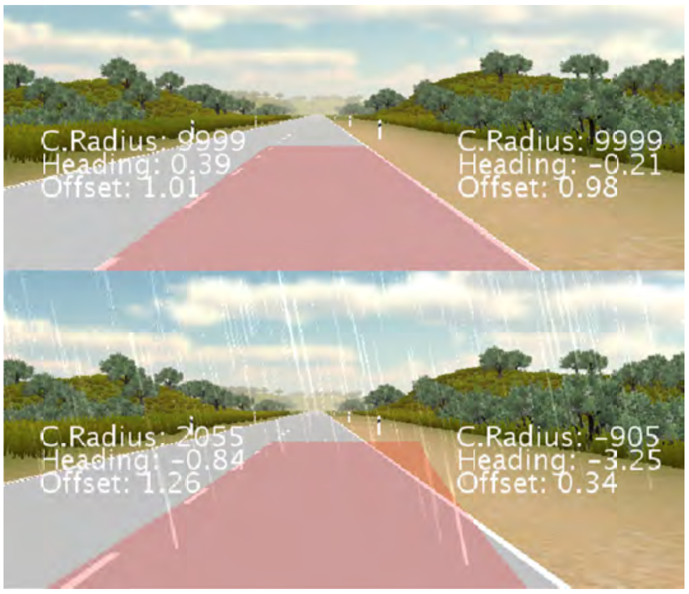

Figure 6. Visual input for lane keeping assist with different weather conditions

Since the environmental simulation is also capable of sending information back to the overall simulation model, the virtual camera now provides visual information for a lane keeping assistant control algorithm developed in Matlab Simulink (Gerke & Roggero 2015). Raw data from the virtual camera mounted at the roof of the vehicle (figure 6) is used as input into the object detection algorithm. The animation is also used to visualize the overall results of the approach (figure 6, red surface). Different weather conditions – with and without rain – are simulated. The detected traffic lane objects are further processed by statistical methods and then grouped by rectangles (figure 7). From this information the final traffic lane information is derived (figure 8).

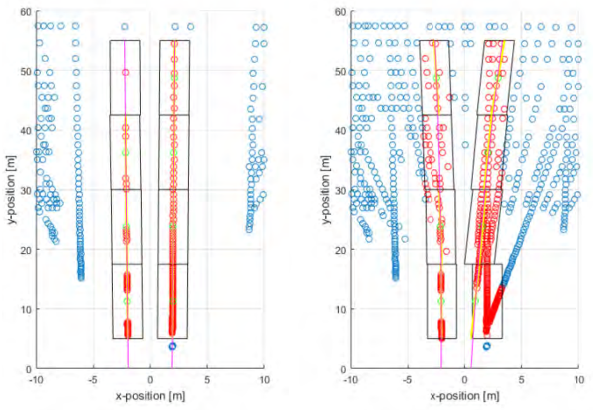

Figure 7. Processing of raw image data with (right side) and without rain (left side)

The traffic lane information is then used as input into the lane keeping control algorithm. As can be seen the overall method provides reliable results with plausible correlation to the virtual world. Using the environmental model without rain produces a quite smooth input for the control algorithm. The single lanes can clearly be recognized. Adding rain as an interfering element to the scene challenges the processing of the raw image (figure 7, ride side) but still produces reliable information for the control algorithm (figure 8, lower part).

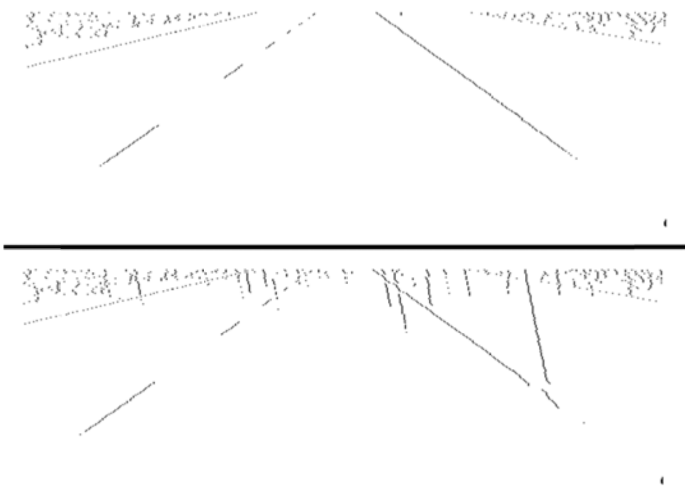
Figure 8. Final traffic lane information with (lower part) and without rain (upper part)

5 CONCLUSIONS

The usage of a game engine for realizing an advanced environmental model is not a trivial endeavor, because most methods are made for human perception and not for technical simulation. By adapting them to physical laws and specific peculiarities of computer vision, a powerful and promising tool is available. The open system is extendable while having a user interface that is easy to operate and to configure at any time. Interfaces are designed according to established standards which allow integration into existing toolchains. It is possible to benefit from the latest technical innovations for computer games while having an economical efficient solution that runs on mid-priced PC systems with no additional hardware requirements. In consequence using the solution within the field of developing and testing ADAS and autonomous vehicles is a reasonable and valuable alternative to current test setups.

REFERENCES

Fu, M. Y. & Huang, Y. S. 2010. A survey of traffic sign recognition. *International Conference on Wavelet Analysis and Pattern Recognition:* 119-124.

Gerke, T. & Roggero, M. 2015. ADAS-Entwicklung mit Model-Based Design. *HANSER automotive (5-6):* 40-43.

Gunia, D. & Schüling, J. 2011. Model-Based Testing of Ford's Lane Keeping System. *ATZ 11/2011* (113)*:* 28-31.

Hafner, S. 2015. Physical modeling of environment detection sensors, based on GPU-supported shader technology. *15. Internationales Stuttgarter Symposium:*1567-1586.

Hafner, S. 2015. Konzeptstudie zur echtzeitfähigen Simulation von Ultraschallsensoren mittels Computergrafik. *Master Thesis*.

Jeong, S. & Choi, C. & Oh, J. & Yoon, P. & Kim, B. & Kim, M. & Lee, K. 2010. Low cost design of parallel parking assist system based on an ultrasonic sensor. *International Journal of Automotive Technology* (11/3): 409-416.

Randima, F. & Kilgard M. 2003. *The Cg Tutorial: The Definitive Guide to Programmable Real-time Graphics*. Boston: Addison-Wesley.

Reif, K. 2010. Adaptive Cruise Control (ACC). *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*: 158-175. Wiesbaden: Vieweg+Teubner

TESIS DYNAware 2016. *DYNA4 Driver Assistance. ADAS Development with Real-Time Simulation*. URL: http://www.tesis-dynaware.com/en/products/dyna4-driver-assistance. Call date: 10.01.2016

Unity Technologies 2016. *Unity. Game engine, tools and multiplatform*. URL: http://unity3d.com/unity. Call date 07.01.2016.

Verband der Automobilindustrie (VDA) 2015. Automatisierung: Von Fahrerassistenzsystemen zum automatisierten Fahren. *VDA Magazin*.