Recent Advances on the OSU Virtual Environment System Test-Bed and Its Applications [1]

Coşku Kasnakoğlu
Department of Electrical Engineering
The Ohio State University
Columbus, OH 43210 USA
kasnakoglu.1@osu.edu

Ümit Özgüner
Department of Electrical Engineering
The Ohio State University
Columbus, OH 43210 USA
ozguner.1@osu.edu

Abstract

This paper describes The OSU Virtual Environment System (OSU-VES), a tool that has been developed to be used for design and testing in automotive research. First, RoadEZ, The Virtual Environment Builder, and VeSim, The Virtual Environment Simulator will be presented, followed by new concepts of triggering events, generic modules and real world tracking with transition. Next, three applications of OSU-VES will be explained, which are, Control Authority Transition System (CAT) development and testing; human factors research on driver distractions; and remote presence providing.

1 Introduction

Virtual environment simulation is an extremely valuable tool for automotive research, since it provides the opportunity of testing and verifying systems under consideration, without the time, expense and inconvenience of conducting test track experiments using real motor vehicles.

The Ohio State University Virtual Environment System (OSU-VES) is one such system, developed at OSU. The system was introduced in 1999 (see [1]), although the start of development goes back to 1996. In 2000, two application of the system namely, image processing sensor evaluation and sensor/data fusion design/evaluation, were reported in [2] and [3] respectively, together with brief explanations of the system and the developments until that date. The goal of this paper is to give an overview of the current status of The Ohio State Virtual Environment System, with emphasis on the recent developments and applications.

First, a general overview of the OSU Virtual Environment System will be given, explaining the two parts that make up the system: The Road and Environment Generator (RoadEZ) and the Virtual Environment Simulator (VeSim), followed by detailed explanation of RoadEZ and VeSim and then three new and innovative concepts introduced to OSU-VES will be presented: Triggering events and event triggered modules; generic modules for decoupling of module algorithms from the simulation interface; and real world tracking with transition.

The advances in the OSU Virtual Environment System will be followed by the applications in which the system was used successfully after those reported in [2] and [3]: Building and simulation of virtual environments and scenarios for human factors research on driver distractions [5]; Control Authority Transition System development and testing (CAT) [8]; and, a remote presence provider for a mobile system, which is also related to the shared driving concept and involves real-time synchronization of simulations with actual data from a remote vehicle [6].

2 The OSU Virtual Environment System Overview

Figure 1 shows a general overview of the system and its two main elements, RoadEZ [4] and VeSim [1], which will be described in sections 3 and 4.

3 The Virtual Environment Builder - RoadEZ

The block diagram of the virtual environment generator RoadEZ appears at the bottom left of Figure 1. As it can be seen from the figure, RoadEZ accepts inputs through its graphical user interface (GUI), which is how the roads, terrain, vehicle paths and the positions and orientations of the 3D objects in the environment are defined. A screenshot of the RoadEZ GUI can be seen in Figure 2.

[1]This work was supported through a project ("Control Authority Transition System, Half-Car Platform and The Virtual Environment System as a test-bed for ITS Technologies", 2002-2003) by the OSU CAR-IT Consortium, ITS Thrust Area.

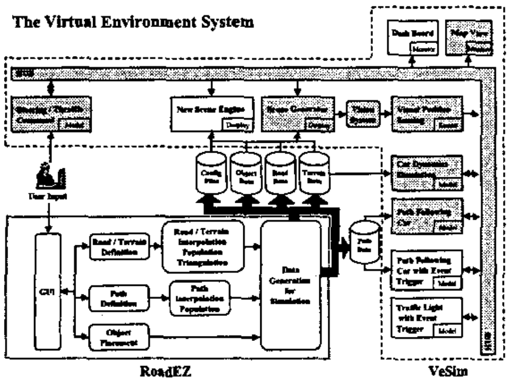


Figure 1: The block diagram of the virtual environment system, showing the operation and interfacing of its two main elements: RoadEZ, The Virtual Environment Builder and VeSim, The Virtual Environment Simulator.

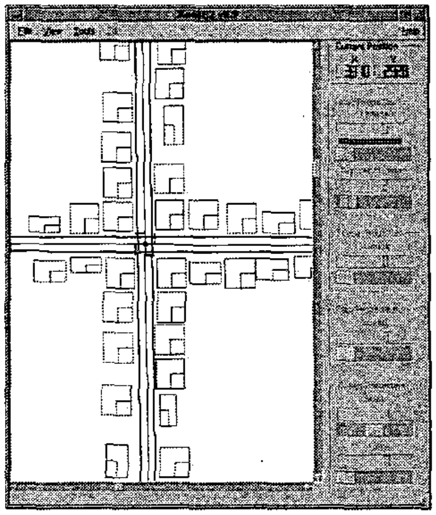


Figure 2: Screenshot from the main window of RoadEZ, illustrating the process of environment building, through the GUI.

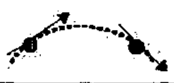
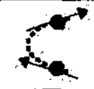
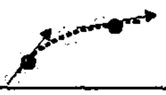
Change in North–South Direction		*Fit function:* $R(x) = Ax^3 + Bx^2 + Cx + D$
Change in East–West Direction		*Fit function:* $R(y) = Ay^3 + By^2 + Cy + D$
No direction change		Fit function can be *R(x)* or *R(y)*: selected to favor *low slope change*

Figure 3: Illustration of polynomial fitting in RoadEZ, using location and slope information. The function type is either dictated by the data or to favor low slope change.

The outputs of RoadEZ are a number of configuration files and 3D descriptions of the roads, terrains and the paths, which can directly be used by VeSim for the simulation of the virtual environment. The design goal of RoadEZ is to automate the environment generation process as much as possible so that a large number of environments can be generated in a short amount of time, with minimal input from the user through the GUI. The only inputs that the user needs to provide for environment generation are a small number of control points for the road, terrain, vehicle paths and object locations. RoadEZ can then complete all of the remaining work, which includes: Function fitting to road data for interpolation (Figure 3); detection and handling of intersections and merges; interpolation, population and triangulation of road data; function fitting to terrain data for interpolation; interpolation, population and triangulation of terrain data; interpolation and population of path data; 3D object placement and editing; and data and configuration file generation for simulation with VeSim.

4 The Virtual Environment Simulator - VeSim

The block diagram of the virtual environment system VeSim is shown in the top and right portions of Figure 1. Observing the figure, one can see that the simulator is made up of a number of modules, which are shown as rectangular blocks in the figure. The modules communicate with each other through a simulation network provided by the specialized process, named the hub.

4.1 Modules of VeSim

4.1.1 Hub: Hub is the central process that extracts information about modules, starts the simulation by executing modules, requests, receives, sends data to/from modules during simulation and ends the simulation by terminating the modules.

4.1.2 Steering/Throttle/Break Command: This module is responsible for generation of the steering, throttle and break commands for the Car Dynamics Simulation (Section 4.1.3). The commands can be generated automatically by a control algorithm or manually from user inputs, such as keyboard, joystick and "The Half Car" at CAR-IT. The last item is a recent addition to the system which greatly increases the realism provided to the driver, by giving the driver the

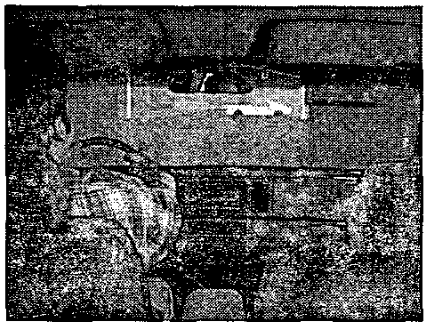


Figure 4: The Half Car platform being used in a scenario being simulated by The OSU Virtual Environment System.

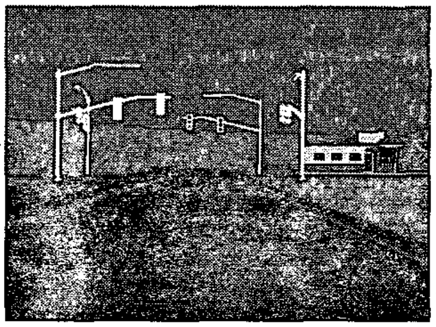


Figure 5: A view from the New Scene Engine of VeSim, showing an intersection, traffic lights, a building and the terrain with mountains behind.

possibility to sit inside an actual car cut into half, and use its steering wheel and pedals to control the simulation vehicle. In addition, The Half Car also provides additional details of a real driving experience such as the instrument panel, radio, adjustable car seat, mirrors and so on. A picture of The Half Car being used in a simulation scenario can be seen in Figure 4.

4.1.3 Car Dynamics Simulation: This module is responsible for the physics of the vehicle, and solves the mathematical equations of the car dynamics. This module is typically built using MATLAB/Simulink and compiled with Real Time Workshop to become a simulation module.

4.1.4 Scene Generator/New Scene Engine: These modules are responsible for the 3D rendering of the environment in real-time to produce a graphical display. The New Scene Engine in a recent addition to the system and although its major function is the same as that of the original Scene Generator, New Scene Engine overcomes many preexisting size and performance limitations by making use of two excellent open source tools, Open Scene Graph [10], and Demeter [11]. See Figure 5 for a view from the New Scene Engine.

4.1.5 Vision System/Visual Position Sensing: This module estimates, in real time, the centerline offset and distance to the vehicle ahead, by processing the image generated by the Scene Generator or New Scene Engine.

4.1.6 Path Following Car (with Event Trigger): Path Following Cars model a vehicle (or any object) that follows a predefined path. In the Path Following Cars with Event Trigger, the path following process is controlled by a number of trigger events, which are described in Section 4.2.1. Also see Figure 6 for a path following car with event trigger in action.

4.1.7 Traffic Light (with Event Trigger): These modules model a traffic light that changes its state based on a given schedule. In Traffic Lights with Event Trigger, the light changing process is controlled by a number of trigger events.

4.1.8 Map View/Dash Board/Data Logger: These are three monitor modules. Map View keeps track of the environment and vehicle state and presents it in map form. Dash Board in a new addition to the system, and emulates the dashboard of the vehicle and can be used to display the states of various instruments. The Data Logger is also a new and very important module responsible for the efficient recording of desired information from other modules into a log file, as the simulation progresses, for later post processing.

4.2 New Concepts in VeSim

4.2.1 Triggering Events and Event Triggered Modules: One of the most important extensions to VeSim is the addition of the concept of triggering events, which are generated when a certain condition in the environment is satisfied, and can be used to trigger an action in certain models of the simulation that accept these triggering events as input.

A triggering event creating condition can be built by any combination of data available in the simulation network. The following situations that create triggering events are very typical: Time Triggers, i.e. events are generated when a certain amount of time has elapsed; Distance Triggers, i.e. events are generated when an object is at a certain position; Distance In Time Triggers, i.e. events generated based on the distance of the object to the trigger point, measured in terms of time units.

With the introduction of triggering events into VeSim, it became possible to trigger the actions of models as

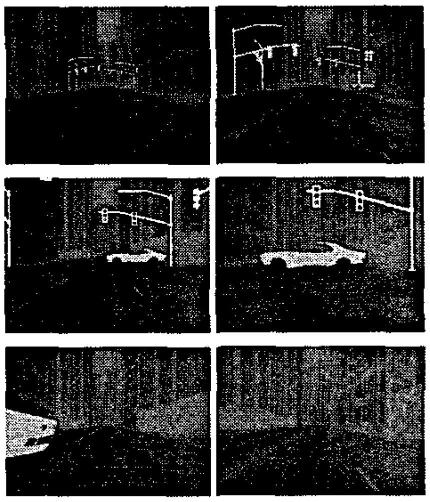


Figure 6: Screenshots taken during a scenario in the city, implemented with trigger events. The order of the screenshots is from left to right, and top to bottom. Our car approaches a green light and another vehicle suddenly comes out from the back of a building and runs the red light.

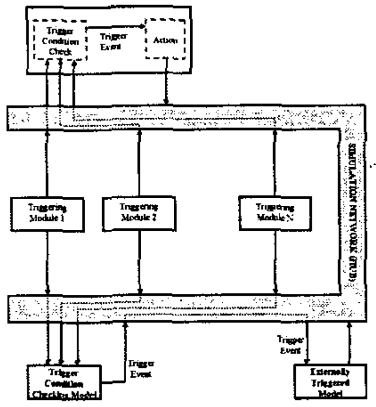


Figure 7: Internally and Externally Triggered Models. In the case of internal triggering, trigger condition checking is done locally whereas for external triggering, a separate model receives inputs from the triggering models, evaluates the triggering condition check and sends the result of the check to the model to be triggered.

a result of a certain condition being satisfied in the environment. There are two ways of incorporating the triggers to control actions within the models, as shown in Figure 7: Internal Triggering and External Triggering.

Also see Figure 6 for an illustration of the triggering event concept in action.

4.2.2 Generic Modules for Decoupling of Module Algorithms from the Simulation Interface: Although its structure makes the system very extendable, there is still the issue that for each module, the programmer of the modules has to write code to communicate with the hub, which brings along the problems that the module implementer has to fully understand the communication protocol, and has to code the communication related tasks in addition to the actual algorithm, causing his efforts to split between the two tasks. Also since the simulation interface has to be re-implemented again and again in every module, this causes a great deal of redundancy and a potential source for errors.

To be able the address these issues and make VeSim even more flexible and extendible, we have implemented a set of four generic modules, one for each type of module (model, display, monitor, sensor), to decouple the simulation interface from the actual algorithms (Figure 8).

As one can observe from the figure, a generic module takes care of all the simulation interfacing; it waits until data is available, then makes it available to a custom defined block, where the programmer can use this data to implement his algorithm and pass the outputs to the generic module, which then sends them to the hub to be used by other simulation modules. In this way, the programmer can fully concentrate on the algorithms, without having to worry about the details of the simulation interface, significantly reducing programming effort and module production time.

4.3 Real World Tracking With Transition

This is also a new and innovative concept that has been recently introduced, which provides a better realism to risk ratio than real world driving, entirely mathematical simulation or entirely virtual simulation. The approach is to generate a representative environment, which can be tracked by the simulation until a time point of transition, when the mode can be switched to simulation, which implements the technology to be tested. This way more realism can be provided than the previous cases since the point of transition corresponds to a real world state so the algorithms receive control in a situation that has arisen in an actual scenario. This also has the advantage that the results can directly be compared to the actual world result, without having to predict what would have happened in the absence of the system in reality.

See sections 5.1 and 5.3 for information on how this concept can be applied, and also Figure 10.

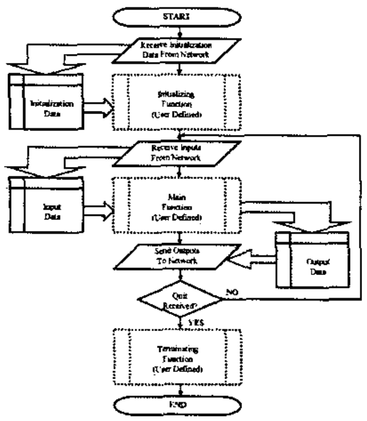

START
Receive Initialization Data From Network
Initialization Data
Initializing Function (User Defined)
Receive Inputs From Network
Input Data
Main Function (User Defined)
Output Data
Send Outputs To Network
Quit Received?
NO
YES
Terminating Function (User Defined)
END

Figure 8: Flowchart illustrating the algorithm - simulation interface decoupling with generic modules. Generic module functions take care of data sending and receiving (solid), so the module programmer can concentrate on implementing the algorithm related parts (dashed).

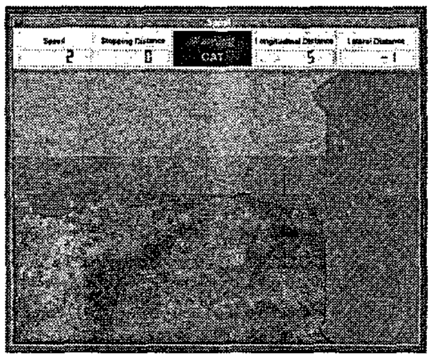


Figure 9: The virtual environment system being used for CAT System development and testing.

5 Recent Applications of The OSU Virtual Environment System

The previous sections have described The OSU Virtual Environment System and its components in some detail. Two successful applications of the system were given in [2] and [3]. We will now talk about three recent applications of the system that illustrate the usefulness of the system as a development tool and test-bed.

5.1 Control Authority Transition Sytem - CAT

CAT is a new driver assistance system introduced for safety enhancement, with the goal of investigating the concept of transferring some of the control authority of the driver to an automated system. For this purpose, a virtual test environment and modules implementing the CAT algorithm were built, and the operation was tested on the OSU Virtual Environment System to verify the design goals.

All of the modes of the CAT system were tested in the virtual environment system including emergency braking, obstacle avoidance (see Figure 9) and lane departure warning/avoidance. The system has also provided a means to analyze the switching effects between the driver and automated system, as described in [9]. Also for the lane departure avoidance/testing the real world tracking with transition concept described in Section 4.3 was utilized to track a real world driving with lane departure from a video, with a transition to the simulation running the lane departure/avoidance algorithm, where the point of transition is detected automatically by the CAT system [7]. Also see Figure 9.

5.2 Human Factors Research on Driver Distractions

OSU-VES was also used in the building and simulation of an environment and its scenarios to make it possible to conduct a human factors research, with the goal of measuring how certain distractions affect the driving responses of human subjects. The entire environment of this study occupies an area that is slightly larger that 8000 m^2, and contains a total number of 777 objects. The complexity of the environment illustrates the capability of the system in handling large and complex environments with dynamic scenarios in real-time. Figures 5 shows a screenshot of a country region road and Figure 6 shows one from the city during a event triggered scenario.

5.3 Remote Presence Providing

This is an ongoing research with the purpose of obtaining measurements from a remote vehicle being driven on an actual road and use these measurement to generate, in real-time, a virtual world corresponding to the vehicle and environment states. Currently, the vehicle state estimation stage has been completed, where the idea is to feed the measurements to an Extended Kalman Filter (EKF) based observer containing sufficiently complex models of the vehicle and the human driver. The estimation results have been verified using mathematical simulation, followed by the virtual environment system. The latter is important because it allows an actual driver to control the simulation, just as it would be the case in practice, and also, allows us to observe phase delays and jitters that are not visible in a plot but might nevertheless be irritating to human eye, which might be an important factor depending on the application in which the observer is being used, such as a shared driving experience sort of concept. Figure 10 provides a brief illustration of the idea.

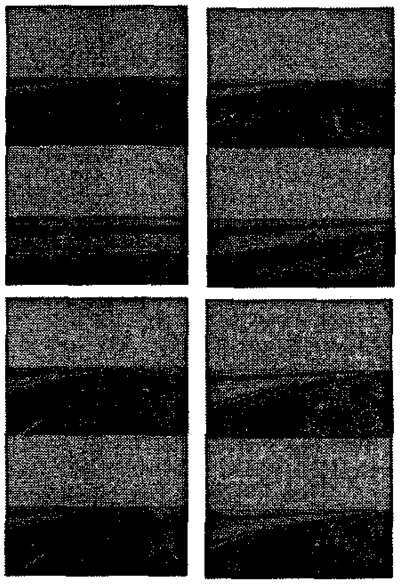


Figure 10: The virtual environment system being used for remote presence providing study. There are four figures with two portions; top portion is actual driving and bottom portion is generated from estimated states. The order in time is from left to right, top to bottom. One sees that the actual and estimated states converge after a while and the remote presence provider tracks the real-world thereafter.

6 Conclusions

We have described the recent advances on the OSU Virtual Environment System (OSU-VES), a tool for development and testing for automotive research. System components, RoadEZ and VeSim were described in some detailed, followed by the introduction of three innovative concepts, which are triggering events and event triggered modules; generic modules for decoupling of module algorithms from the simulation interface; and real world tracking with transition.

This was followed by three applications that the system was used in successfully, namely, Control Authority Transition System (CAT) development and testing; human factors research on driver distractions; and remote presence providing. These applications serve to support our strong belief that the significant changes and improvements OSU-VES has gone through has made it a highly capable test-bed for automotive research, and the tasks that can be achieved by OSU-VES are only limited by the imaginations of automotive researchers.

Our future work includes introduction of additional modules for increased capabilities and completing the work on remote presence providing.

References

[1] J.I. Martin. A Simulation System for Computer Vision Guided Vehicles. Master's thesis, The Ohio State University, 1999.

[2] K.A. Redmill, J.I. Martin and Ü. Özgüner. Virtual Environment Simulation for Image Processing Sensor Evaluation. *2000 IEEE Intelligent Transportation Systems Conference Proceedings*, pp.64-70, Dearborn, MI, 2000.

[3] K.A. Redmill, J.I. Martin, Ü. Özgüner and K. Tamura. Sensor and Data Fusion Design and Evaluation with a Virtual Environment Simulator. *2000 IEEE Intelligent Vehicles Symposium*, pp.668-674, Dearborn, MI, 2000.

[4] C. Kasnakoğlu. RoadEZ - A GUI Virtual Environment Builder. Technical Report, Department of Electrical Engineering, The Ohio State University, 2003.

[5] C. Kasnakoğlu. An Application of the OSU Virtual Environment System: Building and Simulation of Virtual Environments and Scenarios for Human Factors Research on Driver Distractions. Technical Report, Department of Electrical Engineering, The Ohio State University, 2002.

[6] C. Kasnakoğlu. Extended Kalman Filtering Based Tracking of Actual Vehicle States by the OSU Virtual Environment System. Technical Report, Department of Electrical Engineering, The Ohio State University, 2002.

[7] C. Kasnakoğlu. Lane Departure/Avoidance in CAT System Technical Report, Department of Electrical Engineering, The Ohio State University, 2003.

[8] T. Acarman, Y. Pan and Ü. Özgüner. A Control Authority Transition System for Collision and Accident Avoidance. *Journal of Vehicle System Dynamics, The Special Issue on Intelligent Transportation Systems*, 39(2):149-187, 2003.

[9] T. Acarman, C. Kasnakoğlu, Y. Pan and Ü. Özgüner. Control Authority Transition System, Half-Car Platform, Vehicular Simulator and Dynamic Switching Analysis between The Human Being and Driver Assistance System To appear in *2003 IEEE Intelligent Vehicles Symposium*, Columbus, OH, June 2003.

[10] R. Osfield. Open Scene Graph: A Cross Platform C++/OpenGL Library for Realtime Visualization www.openscenegraph.org, 2003.

[11] C. Fowler. Demeter: A Cross Platform C++/OpenGL Terrain Rendering Engine www.terrainengine.com, 2003.