

VIRTUAL REALITY FOR LOW-COST AUTOMOTIVE DRIVING SIMULATOR IN VEHICLE ENGINEERING RESEARCH

Kang Hooi-Siang^{a,b*}, Mohamad Kasim Abdul Jalil^b, Lee Kee-Quen^c

^aInstitute for Vehicle Systems and Engineering (IVeSE), Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

^bFaculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

^cDepartment of Mechanical Precision Engineering, Malaysia-Japan International Institute of Technology (MJIT), Universiti Teknologi Malaysia Kuala Lumpur, 54100 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur, Malaysia

Article history

Received

15 March 2017

Received in revised form

27 June 2017

Accepted

10 September 2017

*Corresponding author

kanghs@utm.my

Graphical abstract



Abstract

Interactive simulation in automotive driving has enhanced the studies of driver behaviors, traffic control, and vehicle dynamics. The development of virtual reality (VR) technology leads to low cost, yet high fidelity, driving simulator become technically feasible. However, a good implementation of high realism and real-time interactive three-dimensional (3D) virtual environment (VE) in an automotive driving simulation are facing many technical challenges such as accessibility, dissimilarity, scalability, and sufficiency. The objective of this paper is to construct a virtual reality system for an automotive driving simulator. The technology with variations of terrain, roadway, buildings, and greenery was studied and developed in the VE of the simulator. Several important technical solutions in the construction of VE for driving simulation had been identified. Finally, the virtual reality system was interactively used in a driver-in-loop simulation for providing direct road elevation inputs to the analysis of vehicle dynamics model (VDM). The results indicated identical matching between the VDM inputs and the VE outputs. The outcomes of this paper lead to a human-in-the-loop foundation of a low-cost automotive driving simulator in the vehicle engineering research.

Keywords: Virtual reality, driving simulator, automotive, transportation, virtual environment

Abstrak

Simulasi secara interaksi dalam memandu automotif dapat meningkatkan hasil penyelidikan dalam bidang tingkah laku pemandu, kawalan trafik, dan dinamik kenderaan. Perkembangan teknologi realiti maya (VR) membolehkan simulator memandu yang berkos rendah dan mempunyai kenyataan tinggi menjadi lebih mantap dari segi teknikal. Namun, pelaksanaan simulasi memandu automotif dalam masa interaksi sebenar dalam persekitaran maya (VE) tiga-dimensi (3D) masih menghadapi cabaran-cabaran teknikal seperti kebolehcapaian, ketidaksamaan, kebolehskalaan, dan kecukupan. Tujuan kertas jurnal ini adalah untuk menghasilkan sistem realiti maya bagi sebuah simulator memandu automotif. Teknologi seperti variasi rupa bumi, jalan, bangunan, dan tumbuh-tumbuhan hijau telah dikaji dalam persekitaran maya dalam simulator ini. Penyelesaian teknikal yang penting dalam pembinaan persekitaran maya bagi simulasi memandu telah dikenalpastikan. Akhirnya, sistem realiti maya telah digunakan dalam simulasi pemandu-dalam-lingkaran

bagi menghasilkan input data ketidakrataan jalan yang digunakan seterusnya dalam sistem dinamik kenderaan. Keputusan kajian menunjukkan bahawa pepadanan serupa antara input dinamik kenderaan dan output persekitaran maya. Hasil pengkajian ini membawa kepada satu asas perkembangan selanjutnya simulator automotif yang berkos rendah bagi kejuruteraan kenderaan.

Kata kunci: Realiti maya, simulator memandu, automotif, pengangkutan, persekitaran maya

© 2017 Penerbit UTM Press. All rights reserved

1.0 INTRODUCTION

A driving simulator is a sophisticated system to study the vehicle engineering [1], traffics flow [2], and driver behavioral issues [3,4]. Driving simulators were initially designed at full-scale by the car manufacturers [5-7] and the authorities of transportation [8-10] with which a complex instrumentation system was directly connected to a real car cockpit. Some of the full-scale driving simulators are equipped with large hexapod motion base or huge moving system for lateral and longitudinal motions [9]. The cost for these full-scale driving simulators was ranged from few hundred thousand to multi-millions US dollars [8-10]. This situation has been changing along with the rapid development of personal computer where the high-speed computing power can be available with more affordable cost. Hence, the down-scaled, yet consisting of all the necessary subsystems for driving simulation can be developed at low-cost by the research community for multiple purposes with a reasonable fidelity in the driving simulation [11-14].

A good implementation of high realism and a real-time interactive three-dimensional (3D) virtual environment in driving simulation depends on the computational capability. Virtual reality system is significant in the training for supporting interaction between the avatar and its virtual environment (VE) [15-17]. In an automotive driving simulator designed in Universiti Teknologi Malaysia (UTM) for transportation researches [18-21], the virtual reality system provides road geometrical information to the corresponding vehicle dynamics system for the analysis of car dynamic responses.

The virtual reality system that the vehicle drives on can be modeled by using polygonal and splines road descriptions. In many research works, the road segments are modeled flat, level, and smooth with some road bumpers [2,22-23]. This design allows researchers to neglect the effects of road grade and variations of tire load caused by road surface unevenness and inclination [24]. However, this assumption could be unrealistic for the reason that road slope and unevenness are unavoidable in the real world. The objects in VE such as buildings, road, and greenery are important to enhance the driver-in-

loop immersion during automotive driving simulation. Some driving simulators use 2D photographs and these images are mapped on simple geometry polygons or boxes to represent the interactive virtual environment [22]. This approach can create a realistic image in the virtual environment. However, the user's viewpoint will become less realistic when tracking through the VE because this approach does not support interactive orientations between movable objects in the same photograph [25]. Hence, 3D objects with high fidelity and supports for interactive viewing is an appropriate solution in the construction of a VE in virtual reality system.

The objective of this paper is to construct a virtual environment for automotive engineering research based on virtual reality technology with variations of terrain, roadway, buildings, and greenery. Several important technical issues in the construction of a virtual environment for driving simulation will be discussed. Finally, the virtual reality system was interactively used in a driver-in-loop simulation for providing direct road inputs to the analysis of vehicle dynamic responses.

2.0 METHODOLOGY

The UTM driving simulator for vehicle engineering research is designed in a collaborative simulation with which is consisted of two computers connected over a Local Area Network (LAN) and interact in real-time to share the simulation information. The schematic diagram of system interface and their data flow are illustrated in Figure 1.

In this driving simulator's architecture, the vehicle dynamics model (VDM) is operated on the server side, while the virtual reality system is located on the client side. Data communication channels are set up in between the driver testbed and the VDM and in between the server-based VDM and the client-based VE to transfer data under a synchronized simulation time step. The data packages a_{sig} is referred to a 3x1 array which consists of throttle input, brake pedal displacement, and steer wheel angle, b_{sig} is a 6x1 array with the 6 degree-of-freedom motions of the car model, and c_{sig} is a 4x1 array with the road

elevation from VE which interface with the wheels of the car model.

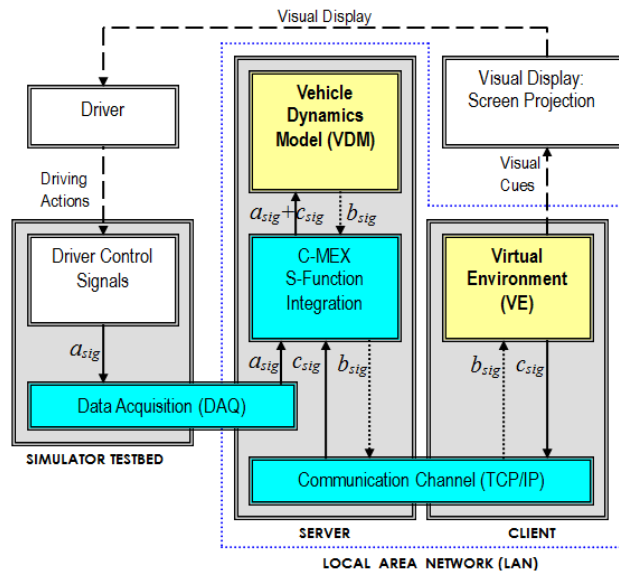


Figure 1 Data flow diagram of UTM driving simulator

The VDM was developed in SIMULINK. It interacts with external applications, such as VE, through C-MEX S-function integration method, which was built as an external function incorporated with SIMULINK. It initiates data acquisition software to receive the signals from the testbed's sensors. The driver maneuvers the car model in VE by controlling the input devices (steering wheel, throttle, and brake pedal in this case) in the testbed based on the visual cues projection on the screen from the virtual reality system. The sensor signals detected by the server-based VDM are fed to update the input variables of vehicle dynamics in each time step. Furthermore, the C-MEX S-function integration block collects the VDM results (three translational motions and three rotational motions in x, y, and z-axes) to transfer them to the client-based VE via Transmission Control Protocol/Internet Protocol (TCP/IP). The car motions calculated from VDM are transferred to the client-based VE to update the vehicle model's position in the VE corresponding to the driver's actions. The road inputs are extracted from VE and fed back to VDM for simulating the car dynamic responses in the next time step. In this paper, the development of virtual reality system was studied. The synchronization of data packages transfer in between the VDM and VE was evaluated to determine the accessibility and sufficiency of the driving simulator.

2.1 Virtual Reality

Virtual reality is a high-end computer interface that involves real-time simulation and interactions through multiple sensorial channels, such as visual, auditory and tactile in which interactive graphical system is

used to create a realistic-looking world, or so-called Virtual Environment (VE) [26].

There are numerous technical difficulties in the development of VE for driving simulation in vehicle engineering research. Firstly, the human driver needs feel that the entire VE is in good accessibility and any maneuvering actions can directly and immediately affect the VE [27]. Secondly, differences in graphical capabilities of the interface attributes in VE and the real world may cause dissimilarity. The dissimilarity needs to be reduced by implementing a high fidelity graphical representation in VE. Thirdly, the number of objects that may simultaneously participate in the system (scalability) is important in order to create a comprehensive VE. The complexity of VE increases exponentially with the number of participating objects because of the possible interactions among them [28]. Fourthly, graphical generation in automotive driving simulator must occur at frame rate with high sufficiency. The real-time processing of human driver's interaction in VE is needed to maintain its immersion.

One of the considerations for combining virtual reality (VR) with automotive driving simulation is its ability to enhance the physical fidelity. The VE provides realistic road inputs data interactively to the VDM analysis integrated with the human driver's maneuvers in good accessibility. In order to reduce the dissimilarity, the VE was developed based on campus view of Universiti Teknologi Malaysia (UTM) at Johor Bahru, Malaysia. Various buildings and infrastructures models were included in the VE to increase its scalability and fidelity. On the other hand, the rendering speed was maintained at a frame rate of adequate sufficiency because the complexity of the VE objects will always slow it down. Hence, appropriate approaches were studied to improve the VE by modeling techniques such as scene-graph, level-of-details (LOD), texturing mapping and model simplifications.

The 3D objects in the VE were modeled by using Computer-Aided-Design (CAD) packages (AutoCAD and 3D Studio Max) and assembled in a VE by using WorldToolKits (WTK) program [29]. The virtual objects were modeled according to their scaled-dimension in CAD programs and used to create the VE. The dynamic responses of movable objects of the vehicle (such as wheels) need to be simulated separately in the virtual reality system. Therefore, these models were created and stored in separate files. The AutoCAD and 3D Studio object models were then converted into a format which is compatible with WTK for the real-time driving simulation.

2.2 Modeling of Virtual Environment

2.2.1 Topographical Terrain Modeling

Modeling issues embrace both building up meshes description from scratch and editing or changing the shape of an existing description. A comprehensive VE in the driving simulation includes terrain [30],

roadway, buildings, foliage, and greenery. A topographic map as shown in Figure 2(a) provides the realistic contour information of the UTM campus area. A terrain model based on realistic altitude was constructed from contour map [31]. The contour lines, as shown in Figure 2(b), of UTM terrain profile were extracted from contour map by using AutoCAD. Spline was applied to obtain curvature of contour lines. Each contour line is adjusted according to their heights to form a 3D contour profile. The 3D contour profile was exported to 3D Studio Max to create a 3D terrain model which contains realistic elevation data to be utilized in the automotive driving simulation. The terrain model based on realistic topographic data is illustrated in Figure 3.

2.2.2 Roadway Modeling

A UTM roadway model was created using several steps. The two-dimensional (2D) roadway was outlined in AutoCAD based on the UTM road map, as shown in Figure 4(a). The 2D roadway model was utilized as a base-plane and was extruded to a 3D model in AutoCAD. The 3D roadway model was exported to 3D Studio Max and integrated with the 3D terrain model. A Boolean operation was applied to obtain intersection between roadway and terrain. The output of the intersection is illustrated in Figure 4(b), which is a 3D roadway model orientated according to the surface of the uneven terrain.

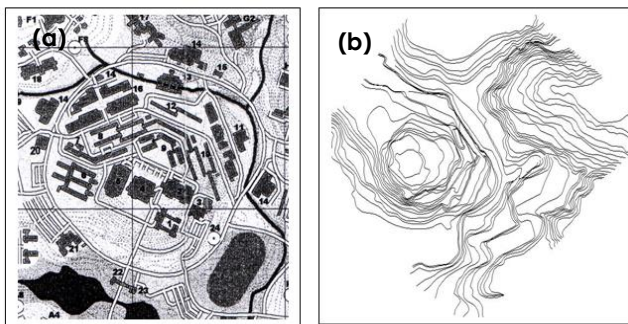


Figure 2 (a) UTM contour map, (b) contour line extracted from the map by using AutoCAD

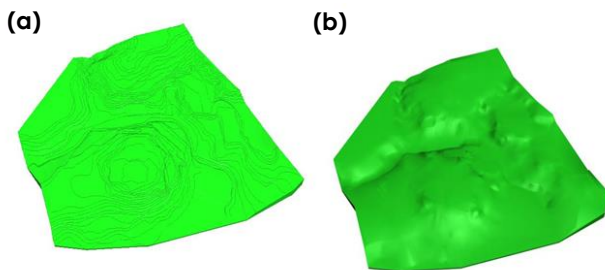


Figure 3 Terrain model generated by 3D Studio Max based on a 3D contour map, (a) terrain contour line image, and (b) rendered image of terrain

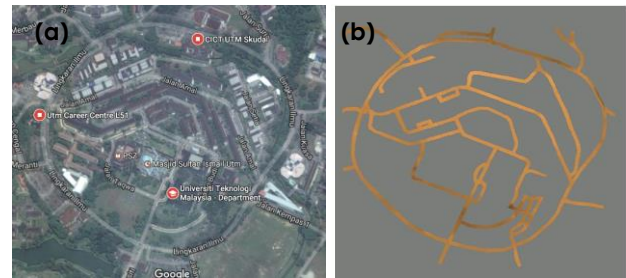


Figure 4 Roadway model, (a) UTM road map (from Google Maps), and (b) intersection output of roadway based on terrain information

2.2.3 Building Modeling

The building models in UTM were modeled in AutoCAD and placed in their corresponding locations in VE based on the map in Figure 2(a). By using AutoCAD, the models can be constructed according to their dimension. One of these building models is the UTM Chancellery Building as shown in Figure 5 which was sketched using AutoCAD. The UTM Chancellery Building model was exported into 3D Studio Max for texturing and material assignment. All building models created in AutoCAD were loaded into 3D Studio Max for polygon editing (Figure 5(a)) and texturing. The model before material assignment and texturing was plain as illustrated in Figure 5(b). The building models were arranged in their actual position as shown in Figure 6, based on the campus map.

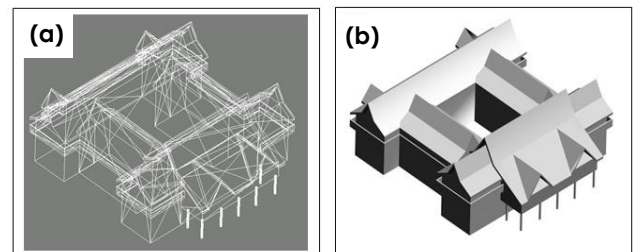


Figure 5 UTM Chancellery Building model, (a) loaded into 3D Studio Max for polygon editing and texture mapping, and (b) rendered image before texture mapping

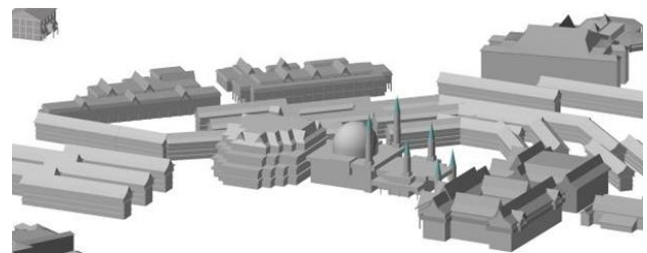


Figure 6 The building models are located on actual positions based on UTM map

2.3 Fidelity of Virtual Environment

2.3.1 Texture Mapping

The realism of VE in low polygons can be enhanced by adding photography surface texture to its polygons, which is called as texture mapping. The texture is an image applied to the surface of the polygon and transformed in VE along with the polygon. The location of an object vertex is stored at modeling time in object coordinate space. The texturing process uses a function to change object parametric coordinates to texture space coordinates [26]. The color of the corresponding texture is then retrieved by the rasterizer and used to change the shaded model pixel color.

Textures increase the level-of-detail (LOD) and scene realism to provide better 3D spatial cues owing to perspective transformations. Besides that, textures substantially reduce the need for a high number of polygons in the scene and therefore increase the frame rate [32].

The Unwrap UVW Mapping was applied to the plain object models in 3D Studio Max as illustrated in Figure 7. For instance, the UTM academic building was originally created in low-polygon mode with the detailed window parts of this building was simply modeled in low-polygon, as shown in Figure 7(a). The photography image of this building was captured by digital camera. This image is assigned with a material file and loaded into Unwrap UVW modifier in 3D Studio Max. The polygons of object model were edited to the desired texture as shown in Figure 7(b). The advantage of texture mapping is many detailed parts can be represented only by a photography image in the VE. Thus, computational overhead can be significantly reduced by rendering a texture image rather than processing the entire detailed polygons. Utilization of a photography image also helps to enhance the fidelity of the virtual reality environment.

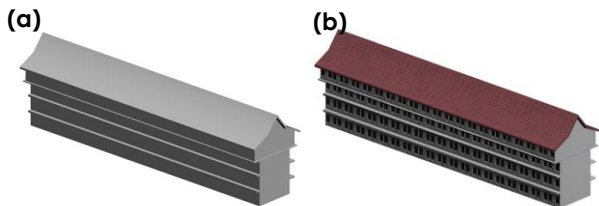


Figure 7 The academic building model, (a) in plain model, (b) after texture mapping

2.3.2 Greenery Enhancement

One of the most important ways to improve the realism in outdoor VE is greenery enhancement. Natural objects give a better visualization effect on VE outdoor-tracking. However, the computational

power required in greenery rendering is very high. For instance, Figure 8(a) shows a tree model in which its trunk and foliage were modeled in 31,752 vertices and 51,535 polygons. It requires very high memory usage for the virtual reality system to duplicate this model in a large population and render them in real-time by using normal work-station.

An appropriate approach to handle this technical difficulty in low-cost virtual reality system is the billboard method. By billboard method, a texture map is considered as a 3D entity and placed in the scene [33]. This method utilizes a 2D image in a 3D scene by rotating the plane of the image so that it is always normal to its human driver's viewpoint in the VE.

The VE of UTM driving simulator is populated with 200 tree models to create its greenery. The billboard method enables every tree models to face to the driver's viewpoint in the normal direction all the time. However, it causes high computational overhead in checking the rotation angles of each tree model in the virtual reality system. A more effective solution was implemented by using the diagonal-crossed billboard approach. The 2D billboard image was duplicated and rotated on its vertical axis to create diagonal-crossed planes as shown in Figure 8(b). This approach has ensured that at least one image plane can be viewed from any angle in the VE during the simulation. Though this graphical-based solution has doubled the billboard polygons, however, additional virtual reality program routines used to rotate all tree models can be excluded.



Figure 8 Tree model, (a) 3D model with foliage and trunk, and (b) modified billboard method with diagonal-crossed planes

2.4 Virtual Environment for Interactive Driving Simulation

2.4.1 Polygons Simplification

Since the terrain model will be rendered in real-time, this model must be constructed in low polygons for avoiding phase lag. The vertices and polygons of terrain model are shown in Figure 9(a). The complexity in polygons has increased the total rendering time. A suitable approach to handle this technical difficulty in the automotive driving

simulation is to simplify the polygon meshes. An approach known as the *quarter (1/4) of point's simplification* in 3D Studio Max was utilized in both horizontal and vertical directions to simplify the meshes of the terrain to be exported to VE. Figure 9(b) shows the total polygons in terrain model were significantly reduced to only 6.6% of the original number of polygons after simplification. The rendered images in Figure 9(c) and Figure 9(d) show that the main attributes of terrain model are maintained, at an acceptable level, after the simplification.

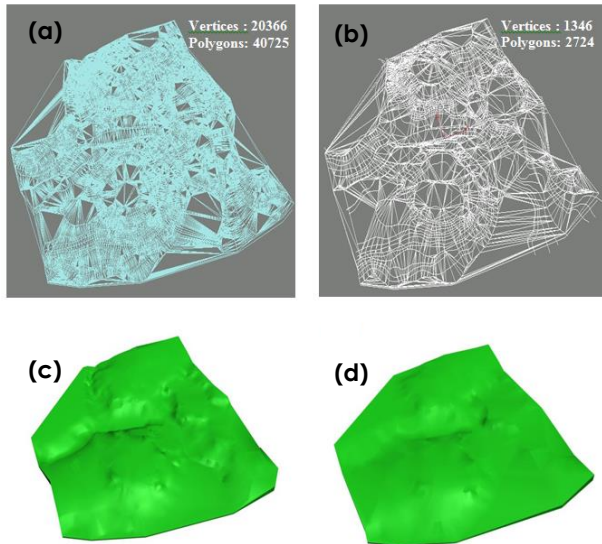


Figure 9 Terrain model, (a) polygons before simplification; and (b) polygons after $\frac{1}{4}$ of point's simplification in horizontal and vertical direction, (c) rendered model before simplification, and (d) rendered model after simplification

2.4.2 Removal of Overlapped Polygons

The roadway model is co-existed with parts of the terrain model in the VE. Modeling two or more surfaces on a coplanar causes the surfaces overlapping. WTK program which generates virtual environment may produce unexpected outputs, called as flashing effect, due to polygons overlapping during the automotive driving simulation, as shown in Figure 10(a).

The solution for polygon overlapping is to construct the model either (a) the polygons are not located on the same plane, or (b) by using Subtraction Method. In the former approach, it requires two polygons to be modeled on different planes. The distance between these two polygons must be sufficiently distinguished to avoid image overlapping [29]. The advantage of this solution is extra polygons are not produced. Nevertheless, the accuracy of polygon position is reduced after adjustment to a different plane. In the later approach by using Subtraction Method, a negative surface (hole) is generated in the original model, and the second model is located

identically on the coordinate of the hole. This approach has the advantage that surfaces of both models can appear coplanar. The drawback of this approach is that creating the hole will generate extra polygons. In this paper, the Subtraction Method was implemented to create the roadway model coplanar with the terrain model because the accuracy of model coordinate is important in the driving simulation. As illustrated in Figure 11, a negative surface is created on the terrain model along with the roadway location, while the wire frame mode shows that these two models are coplanar but not in overlap.

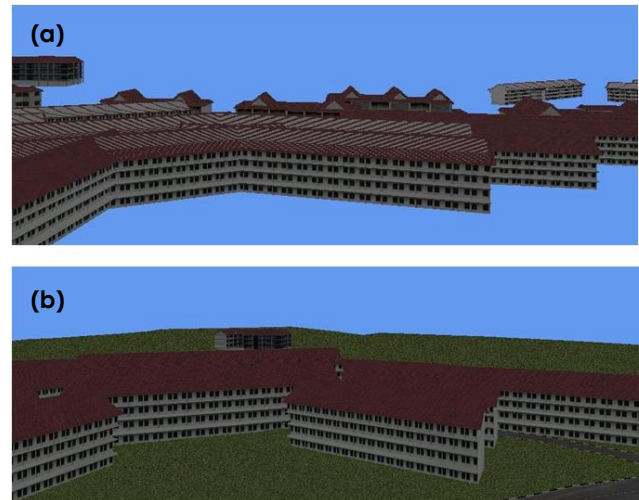


Figure 10 Overlapping object surfaces, (a) flashing effect on the roof of building in VE, and (b) after correction

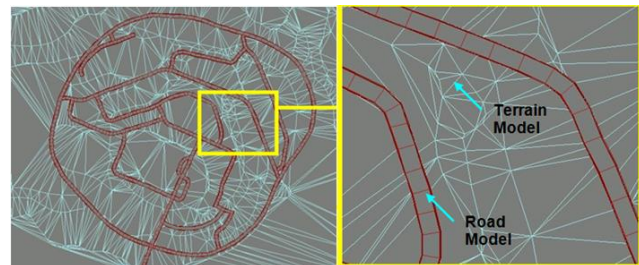


Figure 11 Negative surface (hole) is created on terrain model along with roadway to avoid polygons overlapping

2.4.3 Rendering in Virtual Environment

In this paper, the VE is generated by WTK program in virtual reality technology. VE is composed of various objects, such as lights, positional information, fogs and geometries [29]. WTK launches an interactive automotive driving simulation by loading these object nodes into the VE. The human driver maneuvers the car model in the VE. The viewpoint of the human driver is interactively updated with respect to the driver's actions in the VE. Hence, the visual display system of VE is important to feedback the visual cues for the driver's further maneuvering

actions. The VE can be managed systematically by arranging the virtual reality objects in a hierarchical structure called as the scene-graph in WTK. Scene-graph increases the rendering speed by applying level-of-details (LOD) and back-face culling.

Mapping the virtual reality objects to camera coordinates is the first task in the geometry stage of the rendering task lines. It is followed by lighting, perspective projection, clipping and screen mapping [26]. The display process includes projection onto a view plane. Moreover, only a portion of the view plane can be displayed on a projected screen at one time. A rectangular region of interest, called a viewpoint, is selected for display. A finite pyramid is formed by the eye point as vertex and four planar sides, each side containing the eye point and an edge of the viewpoint. Both parallel planes to the view plane result in a view volume called the view frustum [33].

It is always not practical to completely process all the models in VE at the same time display them on a screen. The processing of rendering the virtual reality objects can be optimized by the view frustum. As illustrated in Figure 12, only virtual reality objects inside the view frustum are processed for displaying by WTK on the projected screen, whereas virtual reality objects outside the view frustum are culled [33].

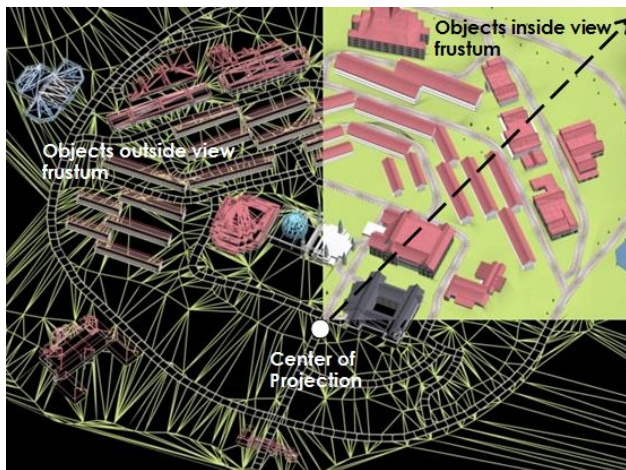


Figure 12 Objects inside the view frustum are processed for display on the computer screen and objects outside the view volume are culled

2.4.4 Level-of-Detail Nodes

The level-of-detail (LOD) node is a specialized switch node in WTK that selects its currently active virtual reality objects in VE automatically based on its distance from the egocentric viewpoint. A LOD node is dynamically selected between a set virtual reality objects with different levels of detail. After a LOD node was created, several sub-nodes were added to it in hierarchical form. A simpler model will be progressively swapped in and freeing up memory

and system resources when the node recedes into a certain distance from viewpoint [33]. In this paper, the LOD solution is employed to construct a sequence of models whose polygon count diminishes over the sequence. Figure 13 shows the LOD nodes of the UTM Library model, with three models in different complexities, were added as sub-modes in the LOD node.

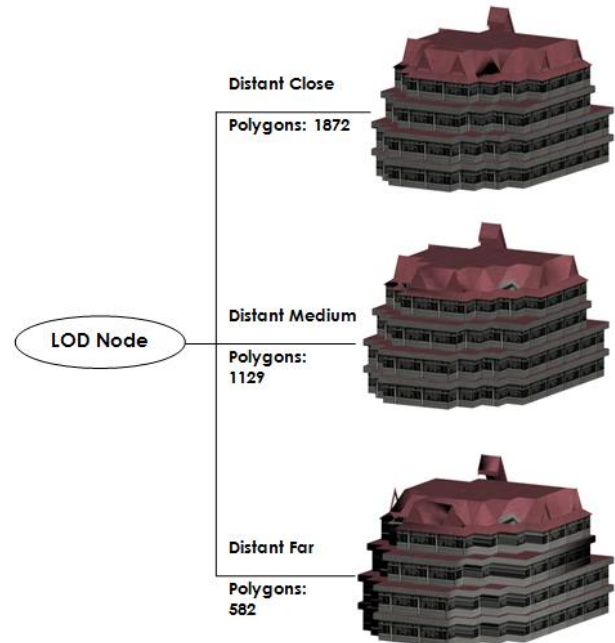


Figure 13 Hierarchical structure of LOD nodes

3.0 RESULTS AND DISCUSSION

In this section, the dissimilarity and scalability of virtual reality environment, together with the accessibility and sufficiency in between vehicle dynamics model (VDM) and virtual environment (VE), are presented.

Firstly, the virtual environment in WTK for interactively automotive driving simulation is presented against the actual street views. As shown in Figure 14, the VE scene is enriched with building models and road model after texture mapping, greenery enhancement, polygon simplification, removal of overlapped polygons, LOD, and diagonal-crossed planes modification. Moreover, an egocentric viewpoint, as shown in Figure 15, is implemented in the UTM driving simulator in order to improve the sense of the presence of human driver during the driving simulation. It can be found that the VE in the UTM driving simulator has been modeled in an adequately detailed for reducing the dissimilarity and enhancing the scalability of the VE.

Secondly, the synchronization in between vehicle dynamics model (VDM) and virtual environment (VE) was evaluated to determine the accessibility and sufficiency of the driving simulator. As the automotive driving simulator is designed for vehicle engineering research, the car object motions in the virtual reality

system which responds to the realistic road elevation in VE is needed to transfer back to the server-based VDM to update the input variables of vehicle dynamics in each time step. In addition, in a collaborative driving simulation, the responses of VE objects with respect to human driver's maneuvering must be displayed in real-time condition.

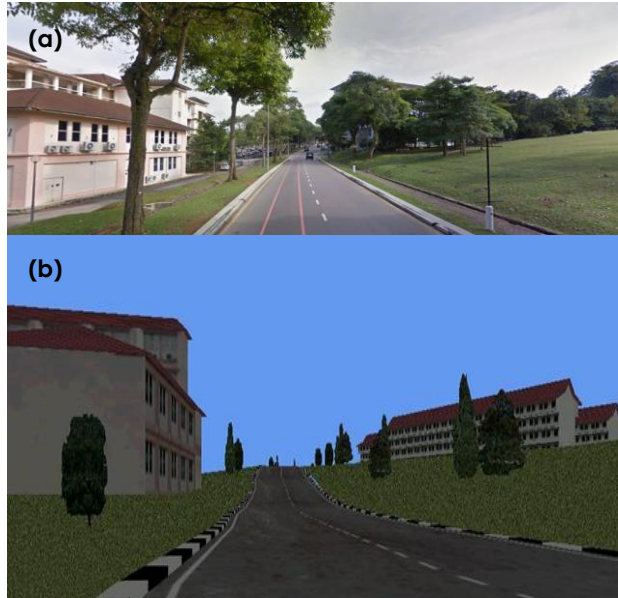


Figure 14 The driving environment, (a) actual street view (from Google Maps), (b) virtual environment after incorporated with multiple graphical techniques as proposed in the Methodology

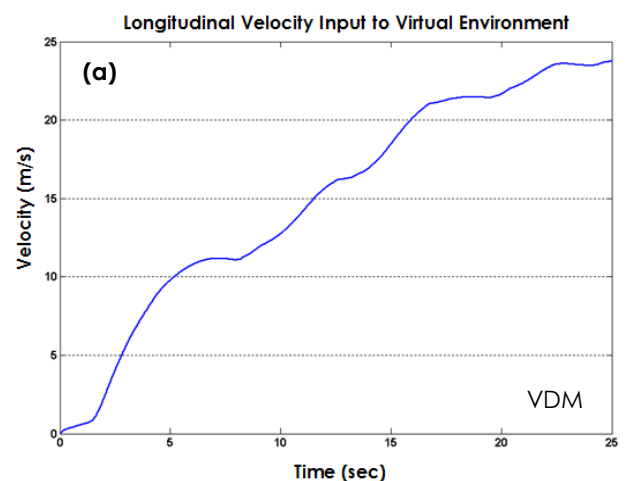


Figure 15 Ego-centric viewpoint is implemented in driving simulation, (a) actual street view (from Google Maps), (b) virtual environment

A set of longitudinal velocity, lateral velocity and yaw motion of virtual reality car model was transferred from VDM to VE. The test is to determine the accessibility and sufficiency of collaborative automotive driving simulation when vehicle's local and global motion transferring over multi-platform. A gradual increasing longitudinal velocity of virtual reality car model from VDM was assigned to VE, as shown in Figure 16(a). The received data in VE was retrieved and shown in Figure 16(b), which represents an identical match between these two data sets. The sampling time step in the VE was 0.25 sec to collect the longitudinal velocity of the virtual car model. It is noteworthy that all the longitudinal velocity inputs from VDM were transferred properly to VE in this automotive driving simulator.

The second variable to be validate is the lateral velocity. The lateral velocity of virtual reality car model from VDM is shown in Figure 17(a). It was increased from steady state to 10.5 m/s in the direction of car body left-side in the first 10 seconds. After that, the lateral velocity varied in the left-side of the car model in different rates. The data received in the VE is shown in Figure 17(b), which presents an identical match with the inputs from the VDM.

The third variable to be validate the synchronization of VDM and VE is the yaw motion of car body. The yaw motion was transferred to VE from VDM as shown in Figure 18. The car body yaw angle with respect to global axes increased from zero-baseline to 0.7 rad (in the counter clock-wise direction with respect to the car body geometrical center of gravity) at around eight seconds. Then, its value reduced to the range of 0.25-0.55 radian until the end of the test.



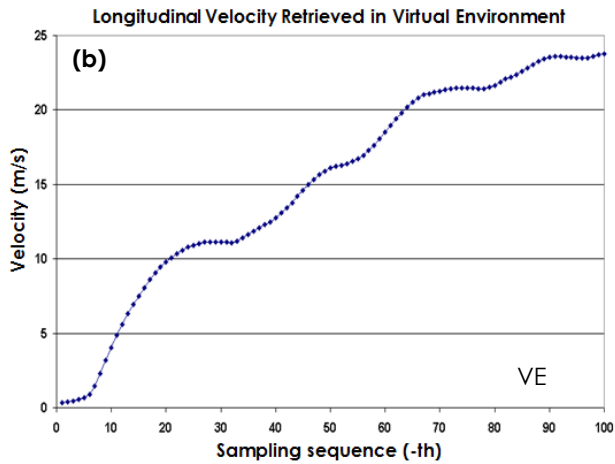


Figure 16 Longitudinal velocity (a) input to the virtual environment from VDM; (b) retrieved in the virtual environment

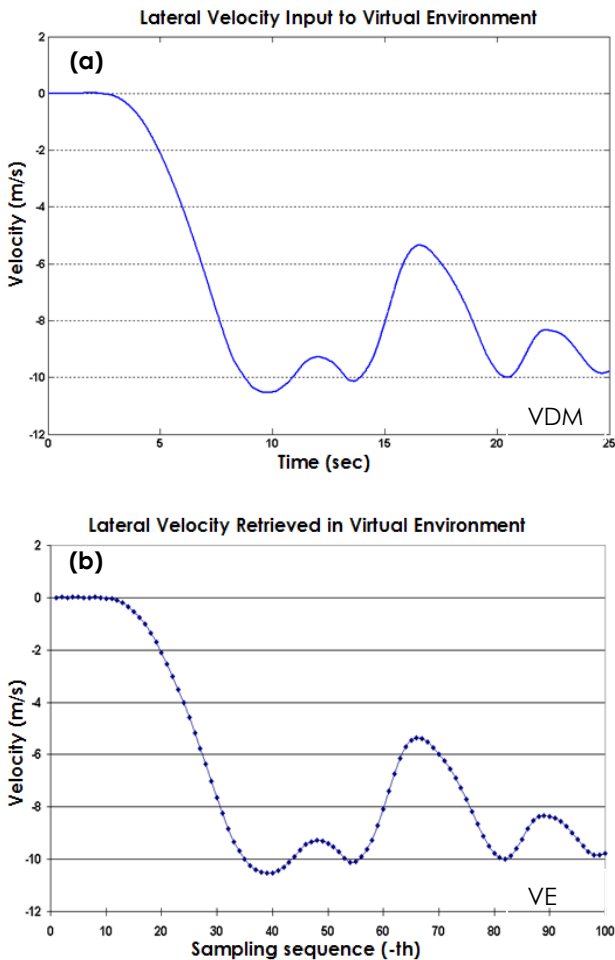


Figure 17 Lateral velocity (a) input to the virtual environment from VDM; (b) retrieved in the virtual environment

Again, the plots are perfectly matched before and after transferred to the VE.

From the highly synchronization between the VDM and the VE, it can be found that the accessibility of

human driver maneuvering after the calculation of VDM can be transferred properly into the VE.

As illustrated in Figure 19, the car body heading angle was maneuvered from its zero-baseline to 0.65 radian (in the counter clock-wise direction) at around 10 seconds (where the sampling sequence at the 40th time). Its value then reduced to the range of 0.3-0.6 radian (in the counter clock-wise direction) until the end of the test. The virtual reality car model's heading angle during this maneuvering test was retrieved from WTK virtual environment and illustrated in Figure 19(b). The results indicate that the virtual reality car model in VE was updated identically according to the VDM inputs. Hence, the sufficiency of data transfer in between the VDM and VE can be adequately achieved.

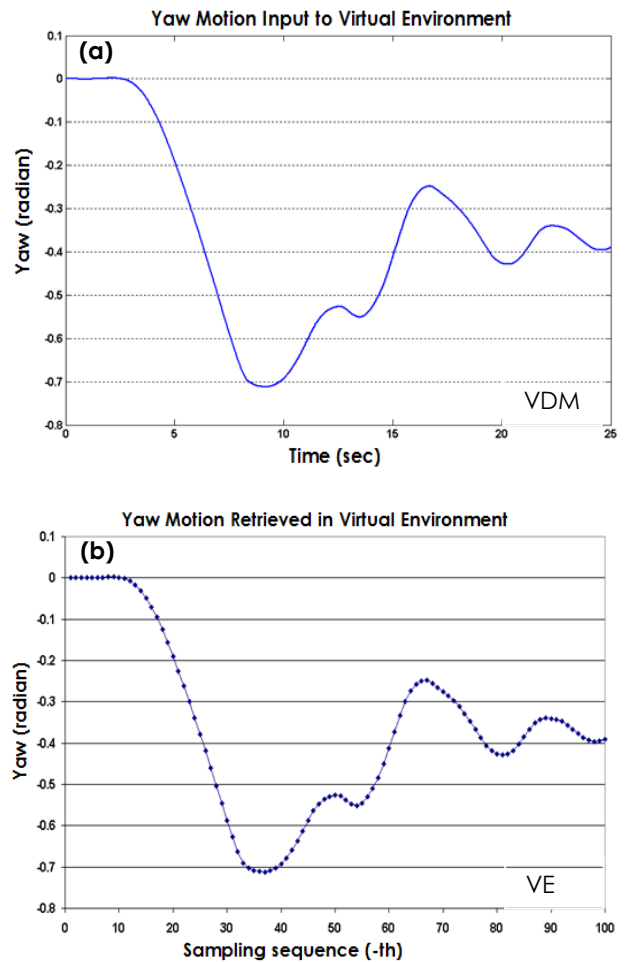


Figure 18 Yaw motion (a) input to the virtual environment from VDM; (b) retrieved in the virtual environment

4.0 CONCLUSION

In this paper, virtual reality system for an automotive driving simulator has been studied based on the variations of terrain, roadway, buildings, and greenery. The virtual reality system was interactively

used in a driver-in-loop simulation for providing direct road inputs to the analysis of vehicle dynamic responses has been studied. A topographical-based realistic terrain based on UTM campus was constructed by using virtual reality technology. Contour information, landscape profile, and buildings location were included as references to generate the VE of the driving simulator. Technical challenges in the VE, such as accessibility, dissimilarity, scalability, and sufficiency were enhanced by modeling techniques such as scene-graph, level-of-details (LOD), texturing mapping and model simplifications. In addition, the balance in between the needs to produce a high level of accuracy of VE in vehicle dynamics simulation and the needs to reduce noticeable lag caused by extensive real-time simulation was studied. The synchronization of VDM and VE was validated through simulation in longitudinal velocity, lateral velocity, yaw motion, and maneuvering heading angle of the virtual reality car model in the VE. The input data and output data in VE had enabled the realization of fully utilized low-cost computing resources in real-time automotive driving simulation for vehicle engineering research.

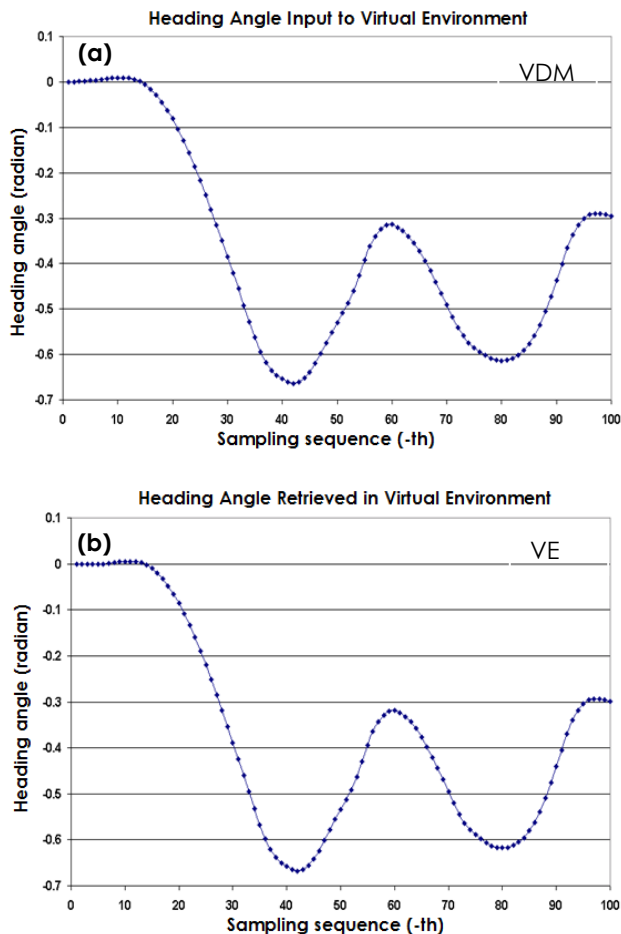


Figure 19 Heading angle data (a) input to the virtual environment from VDM; (b) retrieved in the virtual environment

Acknowledgement

The authors fully acknowledged Ministry of Higher Education (MOHE) and Universiti Teknologi Malaysia Potential Academic Staff Grant Vot 02K43 for the supporting this research.

References

- [1] Mohajer, N., H. Abdi, K. Nelson, and S. Nahavandi. 2015. Vehicle Motion Simulators, a Key Step Towards Road Vehicle Dynamics Improvement. *Vehicle System Dynamics*. 53(8): 1204-1226.
- [2] Liu, J., L. Zhang, S. Xiao, and X. Xin. 2014. Development of Virtual Drive HILS System Based on VR and CarSim. *2014 33rd Chinese Control Conference (CCC)*. Nanjing, China. 28-30 July 2014. 6441-6444.
- [3] Luimula, M., A. Besz, P. Pitkäkangas, T. Suominen, J. Smed, F.R. Izullah, and H. Hämäläinen. 2015. Virtual Evaluation Tool in Driving Inspection and Training. *6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*. Győr, Hungary. 19-21 October 2015. 57-60.
- [4] Atchley, P., A.V. Tran, and M.A. Salehinejad. 2017. Constructing a Publically Available Distracted Driving Database and Research Tool. *Accident Analysis & Prevention*. 99: 306-311.
- [5] Wojcik, C. K. and S. F. Hulbert. 1965. Driving Simulator - A Research Tool. *Mechanical Engineering*. 87(12): 63.
- [6] Lincke, W., B. Richter, and R. Schmidt. 1973. Simulation and Measurement of Driver Vehicle Handling Performance. *SAE Technical Paper*. No. 730489.
- [7] Weir, D.H. and S.M. Bourne. 1995. An Overview of the DRI Driving Simulator. *SAE Technical Paper*. No. 950173.
- [8] Alicandri, E. 1994. HYSIM: The Next Best Thing to Being on the Road. *Public Roads*. 57: 3.
- [9] Salaani, M. K., J. P. Chrstos, and D. A. Guenther. 1997. Parameter Measurement and Development of a NADSdyna Validation Data Set for a 1994 Ford Taurus. *SAE Technical Paper*. No. 970564.
- [10] Bertollini, G. P., C. M. Johnston, J. W. Kuiper, J. C. Kukula, M. A. Kulczycka, and W. E. Thomas. 1994. The General Motors Driving Simulator. *SAE Technical Paper*. No. 940179.
- [11] Mohajer, N., H. Abdi, K. Nelson, and S. Nahavandi. 2015. Vehicle Motion Simulators, a Key Step towards Road Vehicle Dynamics Improvement. *Vehicle System Dynamics*. 53(8): 1204-1226.
- [12] Pieroni, A., C. Lantieri, H. Imine, and A Simone. 2016. Light Vehicle Model for Dynamic Car Simulator. *Transport*. 31(2): 242-249.
- [13] Klüver, M., C. Herrigel, C. Heinrich, H. P. Schöner, and H. Hecht. 2016. The Behavioral Validity of Dual-Task Driving Performance in Fixed and Moving Base Driving Simulators. *Transportation Research Part F: Traffic Psychology and Behaviour*. 37: 78-96.
- [14] Gaspar, J. G., T. L. Brown, C. W. Schwarz, J. D. Lee, J. Kang, and J. S. Higgins. 2017. Evaluating Driver Drowsiness Countermeasures. *Traffic Injury Prevention*. 1-6.
- [15] Clarke, D. B., R. C. D'Arcy, S. Delorme, D. Laroche, G. Godin, S. G. Hajra, R. Brooks, and R. DiRaddo. 2013. Virtual Reality Simulator Demonstrated Use in Neurosurgical Oncology. *Surgical innovation*. 20(2): 190-197.
- [16] Stevens, J., P. Kincaid, and R. Sottolare. 2015. Visual Modality Research in Virtual and Mixed Reality Simulation. *Defense Modeling and Simulation*. 12(4): 519-537.
- [17] Ni, T., H. Zhang, C. Yu, D. Zhao, and S. Liu. 2013. Design of Highly Realistic Virtual Environment for Excavator Simulator. *Computers & Electrical Engineering*. 39(7): 2112-2123.

- [18] Kang, H. S., M. K. A. Jalil, and M. Mailah. 2004. A PC-Based Driving Simulator Using Virtual Reality Technology. *2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry*. Singapore, 16-18 June 2004. 273-277.
- [19] Chiew, Y.S., M.K.A. Jalil, and M. Hussein. 2009, February. Motion Cues Visualisation of a Motion Base for Driving Simulator. *2008 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Bangkok, Thailand. 22-25 February 2009. 1497-1502.
- [20] Fouladinejad, N., N. Fouladinejad, M. K. A. Jalil, and J. M. Taib. 2011. Modeling Virtual Driving Environment for a Driving Simulator. *2011 IEEE International Conference on Control System, Computing and Engineering (ICCSC)*. Penang, Malaysia. 25-27 November 2011. 27-32.
- [21] Fouladinejad N., J. M. Taib, and M. K. A. Jalil. 2015. Development of a Realistic Driving Behaviour by Means of Fuzzy Inference System. *Journal Teknologi*. 74(10): 69-77.
- [22] Huang, A. R. and C. Chen. 2003. A Low-Cost Driving Simulator for Full Vehicle Dynamics Simulation. *IEEE Transactions on Vehicular Technology*. 52(1): 162-172.
- [23] Yu, Y., A. El Kamel, and G. Gong. 2013. Modeling Intelligent Vehicle Agent in Virtual Reality Traffic Simulation System. *2013 2nd International Conference on Systems and Computer Science (ICS)*. Villeneuve d'Ascq, France. 26-27 August 2013. 274-279.
- [24] Demerly, J. D. and K. Youcef-Toumi. 2000. Non-Linear Analysis of Vehicle Dynamics (NAVdyn): A Reduced Order Model for Vehicle Handling Analysis. *SAE Automotive Dynamics & Stability Conference*. Michigan, USA. 15-17 May 2000. No. 2000-01-1621.
- [25] Watt, A. H. and M. Watt. 1992. *Advanced Animation and Rendering Techniques*. New York: ACM Press.
- [26] Burdea G.C. and P. Coiffet. 1994. *Virtual Reality Technology*. London: Wiley-Interscience.
- [27] Donalek, C., S.G. Djorgovski, A. Cioc, A. Wang, J. Zhang, E. Lawler, S. Yeh, A. Mahabal, M. Graham, A. Drake, and S. Davidoff. 2014. Immersive and Collaborative Data Visualization Using Virtual Reality Platforms. *2014 IEEE International Conference on Big Data (Big Data)*. Washington DC, USA. 27-30 October 2014. 609-614.
- [28] Shafieloo, I. 2005. *A Virtual Reality-Based Training Environment Using Haptic Interfaces*. Doctoral dissertation. Montreal, Canada: Concordia University.
- [29] WorldToolKit Reference Manual, Release 9. 1999. Mill Valley, CA: Engineering Animation, Inc.
- [30] Yu, Y., A. El Kamel, G. Gong, and F. Li. 2014. Multi-Agent Based Modeling and Simulation of Microscopic Traffic in Virtual Reality System. *Simulation Modelling Practice and Theory*. 45: 62-79.
- [31] Bidoshi, K. 2003. *Virtual Reality Visualization*. Doctoral dissertation. Columbus, USA: Ohio State University.
- [32] Hearn, D. and M.P. Baker. 1997. *Computer Graphics: C Version*. 2nd ed. New Jersey: Prentice Hall.
- [33] Watt, A. 2000. *3D Computer Graphics*. New York: Addison-Wesley Publishing.