2009 International Conference on Advanced Information Networking and Applications Workshops

Development of Immersive Virtual Driving Environment Using OpenCABIN Library

Yoshisuke Tateyama, Tetsuro Ogi and Hidekazu Nishimura
Graduate School of System Design and Management,
Keio University
Kanagawa, Japan
tateyama@sdm.keio.ac.jp

Noriyasu Kitamura and Harumi Yashiro
Tokio Marine & Nichido Risk Consulting Co., Ltd
Tokyo, Japan

***Abstract*— To investigate how aged drivers can drive more safely, we are developing an immersive virtual driving environment. It consists of precise force feedback steering system device, dynamics simulation software and the K-Cave. Using OpenCABIN library system, an accidental scenario is actualized as a combination of small VR applications.**

Keywords-population aging, training driving skill, driving simulator, CAVE, OpenCABIN library

I. Introduction

In Japan, older population is growing. Population ratio of more than 65 years old persons exceeded 21.5% in 2007. Older motor drivers' ratio reaches 14% among all motor drivers and some of them are actually driving for their daily life[1]. Unfortunately older person cannot drive as well as her/his younger era. Expected response time against accidental crises becomes longer. Worse, they may not realize this fact actually. They have very rich driving experience and have self confidence, but traffic accidents sometimes occur where they realize as safe.

We try to find how older person can drive more safely rather than being confiscated their licenses. To achieve this, we investigate how we train those persons or younger 40 to 65 years old persons to drive more carefully and to change their mind to realize accurately what they can and cannot currently.

Our training system consists of a fine grained driving simulation system, K-Cave system and software. Our driving simulation system has fine grained force feedback system. For example, it can transmit computer generated road vibration precisely to the driver. The driver can easily realize the driving to the real one.

Combination of the driving simulator and the K-Cave produces us completely virtual but realizable to the real driving environment.

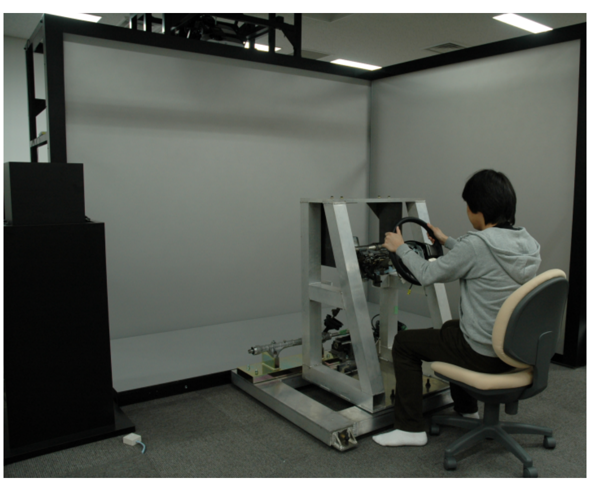

Figure 1. Our virtual driving environment hardware exterior. It consists of a driving simulation system and K-Cave.

II. K-Cave: a CAVE at Keio University

Fig. 2 shows an exterior that a user is operating some VR applications on the K-Cave. The K-Cave is a CAVE-clone[2] VR display system at Keio University. It consists of 3 screens, 6 projectors, 3 rendering PCs and a control PC. Binocular parallax is achieved by circular polarization filters in front of each projector and in a pair of glasses. Passive stereo view feature does not require special high-end graphic rendering functions like genlock nor framelock fuctions. Dimensions of front and floor screens are 2.6 meters width and 2.1 meters height, and right screen is a square whose sides have length 2.1 meters. Each screen's pixel resolution is XGA (1024 x 768 pixels). The viewer's position is tracked by a magnetic position sensor system: Ascension technology corporation's Flock of Birds with an Extended Range Transmitter. 4 PCs are connected via 1000 base TX ethernet and can communicate using TCP/IP protocols. Each PC runs on the Linux operating system.

Figure 2. The K-Cave system.

978-0-7695-3639-2/09 $25.00 © 2009 IEEE
DOI 10.1109/WAINA.2009.208

III. . OUR DRIVING SIMULATION SYSTEM

Our driving simulation system provides a sensation modality to the driver. Mechanical characteristics of the steering system play an important role at accidental situations. We apply a steering system of a real car to our driving simulation system. It has an EPS (Electrical Power Steering) system. Also this system is carefully designed to show precise change of self aligning torque to the driver. Displayed force is calculated precisely according to dynamics model simulation. For example, it can display road vibration to the driver as reactive torque of the steering wheel. When the subject driver can predict an accident, she/he starts risk-avoidance behavior. If the steering wheel is too heavy for her/him, the behavior may be late and the car cannot avoid accident. Or if she/he rotate a steering wheel too quickly, the car starts slip and it may be out of control. So the characteristics of the steering system are important at accidental situations.

Figure 3. An overview of our driving simulation system.

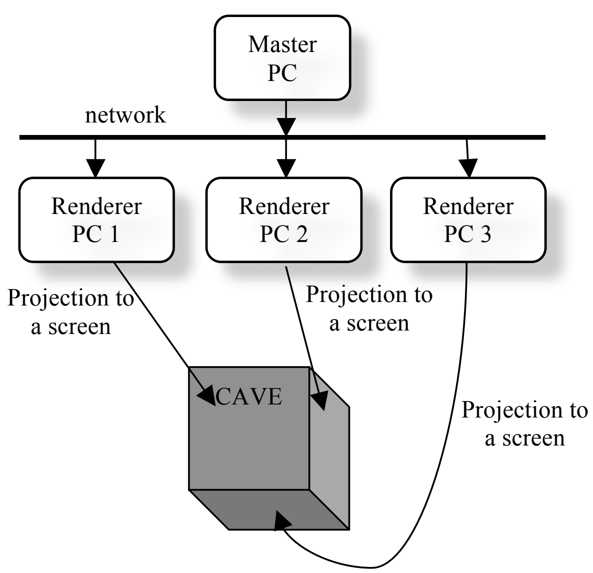

Figure 4. Information of the application flows from the master PC to renderer PCs at 3 screens Cave configuration for multi-PCs.

IV. OPENCABIN LIBRARY

OpenCABIN library is a fundamental software library for developing virtual reality applications. After CABIN library was developed at the university of Tokyo[2], its successor, called OpenCABIN library, was developed from scratch as open source software. It absorbs differences kinds of display systems, so VR application developer can easily develop applications for a multi-screen stereoscopic display.

When designing OpenCABIN library we think about more than one PC display system. Recently multi-screen display systems like CAVEs are consisted of multi-PCs instead of a high-end graphics workstation (Fig. 4). A multi-PCs display system costs cheaper than a high-end graphics workstation, but it is difficult to coordinate PCs to work as a single system. To clear this situation, we introduce a local shared variable mechanism (Fig. 5). A local shared variable contains an application's value and is shared among PCs. To implement network-wide shared values, very complex exclusive control is needed and processing speed essentially becomes slow, but if a writer node is limited only one node and remaining nodes only read it, the situation becomes simple, no exclusive control is needed and processing speed becomes fast. Display systems are naturally separated by a master part and renderer parts. The master part of the application determines the application's behaviors. When the application determines representation to the application operator concretely, information of the display content is transferred from the master part to the renderer parts. The reverse direction of information flow cannot exist except for initialization stage of the display system. OpenCABIN library provides programmer written callbacks invoking mechanism and there are callbacks for master part and callbacks for renderer part. We call this OpenCABIN library philosophy a master/renderer

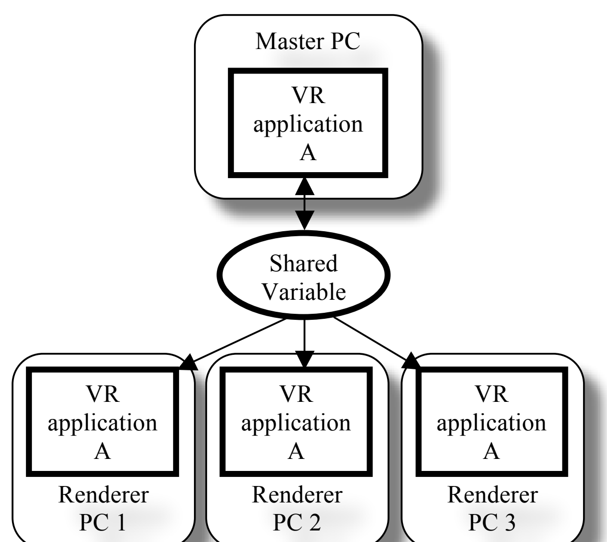

Figure 5. An example of a local shared variable. It shares an application value among multi-PCs.

programming paradigm. An OpenCABIN library application consists of two parts: a master part and a renderer part. A master part is executed in a master process on a master computer, and controls the application's behavior by producing the application context. A renderer part is executed by renderer processes on renderer computers, and those processes render an application world according to reading the application context. A master part is guaranteed that it is always executed by a master process, so it is easy to develop applications which access outside servers via networks, and applications that share virtual space among CAVEs in remote places.

In addition to the basic nature as a VR library, it has another special feature that enables application programmers to develop VR applications easily: plug-in mechanism. From a software engineering viewpoint such as implementation, testing, debugging, reusability, flexibility, and quality control, it is desirable to construct a system as several independent parts rather than as a big monolithic part. Because of limits of almost all OpenGL implementations, two or more processes cannot access to an OpenGL window. So an OpenCABIN library application is formed as plug-in software and it is loaded and executed by an OpenCABIN library execution environment at runtime. An execution environment can execute one or more plug-in applications simultaneously.

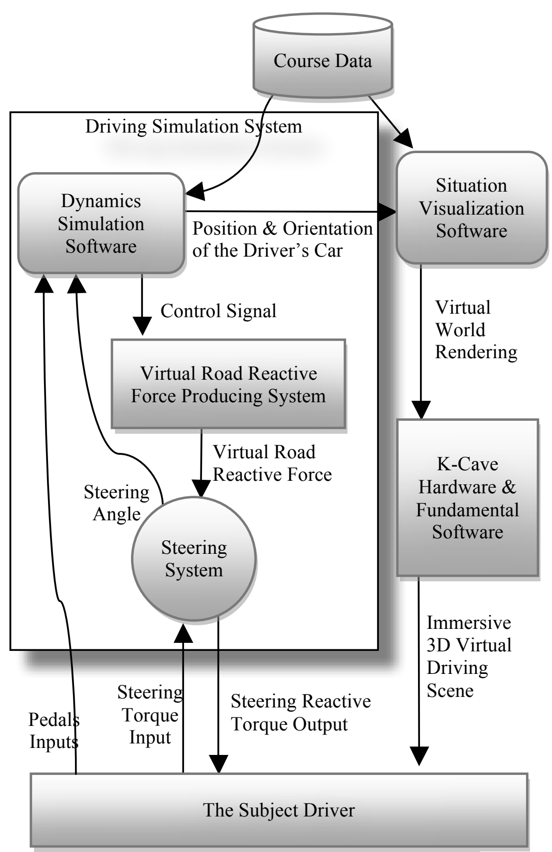

Figure 6. Overall of our training environment for the driver.

As a result, even though each application shows a simple 3D object, virtual space becomes sufficient with a lot of 3D objects. An application user can freely select which object is appeared in the virtual space at runtime.

V. Virtual Driving Simulation Envionment

Fig. 6 shows overall software and hardware structure diagram of our virtual driving simulation environment. Its hardware consists of two hardware subsystems. One subsystem is driving simulation system and the other is K-Cave system. They refer the same course database. Driving simulation system calculates the position and the orientation of the driver's virtual car and sends them to the driving situation visualization software on K-Cave. The driving situation visualization software renders the virtual world of the situation. A CAVE system as a display system provides very high quality virtual space compared with a simple large field of view display like wall screen display. The driver can immerse herself/himself into the driving situation virtual world, and can feel what happens there as real.

VI. Situation Management Software

OpenCABIN library contains runtime software system that can manage multiple VR applications simultaneously. We can implement a traffic accident scenario as a combination of multiple VR programs. When the scenario varies, another combination can be applied. Combinations actualize traffic accident scenarios. For example, when a driver is driving a car running into an intersection, another car is running into the same intersection. Whether the driver can avoid accident? What's happen if the car is replaced to a bicycle, a bike, a child or some small objects?

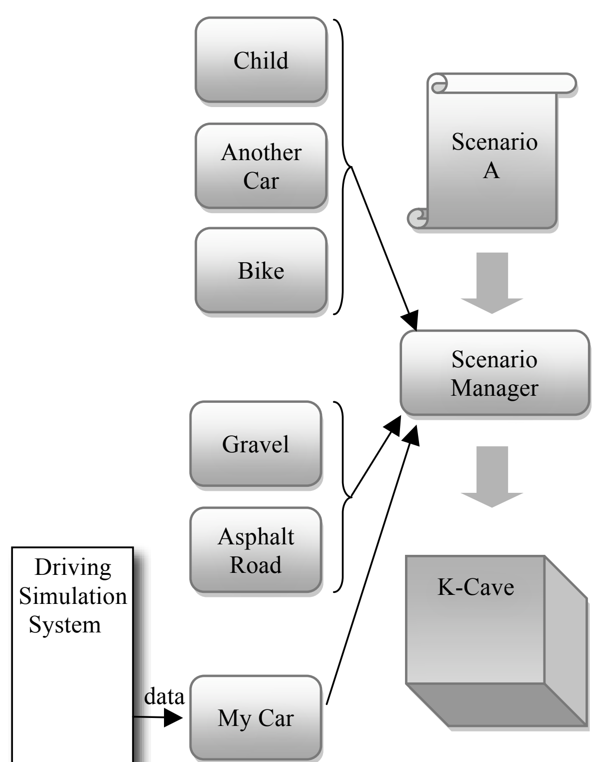

Figure 7. Scenario manager actualize a traffic accident scene.

Using OpenCABIN library, we can easily test a number of slightly different situations (Fig. 7).

VII. SUMMARY

We are developing driving simulation environment. Our system has very precise driving simulation hardware and software. Also it has a CAVE system, so the driver can immerse into the virtual driving situations. This system is suitable for training older person's driving skill.

ACKNOWLEDGMENT

This research is funded by Tokyo Marine & Nichido Risk Consulting Co., Ltd. Constructing K-Cave was supported by G-COE (Center of Education and Research of Symbiotic, Safe and Secure system Design) program at Keio University.

REFERENCES

[1] Noriyasu Kitamura, "Life Span of Safety Driving," Seiunsya, 2009. (In Japanese)

[2] Cruz-Neira, C., Sandin, D. J., & DeFanti, T. A. "Surround-screen projection-based virtual reality: the design and implementation of the CAVE," In Proceedings of the Computer Graphics Proceedings, pp. 135-142, 1993.

[3] Michitaka Hirose, Tetsuro Ogi, Shohei Ishiwata and Toshio Yamada, "Development and Evaluation of CABIN Immersive Multiscreen Display," Systems and Computers in Japan,scripta Technica,Vol.30, No.1, pp.13-22, 1999.

[4] Atsushi Arai, Hidekazu Hishimura, Hiroshi Mouri and Masahiro Kunota, "Gain-Scheduled Control of Electric Power Steering," In Proceedings of MoViC 2007, Symposium on Motion Vibration Control, Dynamics, Measurement and Control Division, the Japan Society of Mechanical Engineers, 2007. (In Japanese)