

大画面ディスプレイアプリケーション開発のための OpenCABIN ライブラリ

OpenCABIN Library for Developing Applications on Large Display Systems

立山義祐／慶應義塾大学大学院システムデザイン・マネジメント研究科, 小木哲朗／慶應義塾大学大学院システムデザイン・マネジメント研究科

Yoshisuke TATEYAMA¹/Graduate School of System Design and Management, Keio University, Tetsuro Ogi²/ Graduate School of System Design and Management, Keio University

*¹tateyama@sdm.keio.ac.jp, *²ogi@sdm.keio.ac.jp

Abstract: Personal computers become fast enough to render an interactive virtual reality world and gigabit local area network devices become available easily. We can construct a large display system using them. OpenCABIN library is developed to support constructing a large display system using multiple PCs and gigabit LAN devices. OpenCABIN library is a fundamental software library for developing virtual reality applications. We show design concepts of OpenCABIN library. Using OpenCABIN library, a tele-immersive application was implemented successfully. This result showed the effectiveness of design concepts of the OpenCABIN library.

Keywords: CAVE, virtual reality application development, large display systems, software information sharing mechanism.

1. Introduction

PCs, Projectors, LCDs and network devices become cheaper and we can easily construct large display system hardware by combining them. To make large display systems work, we may combine multiple screens to show a world and must manage multiple rendering software processes that are designed to cooperate to draw a single world. To coordinate them, you must synchronize information and can use shared memory technique. Shared memory technique is easy method to make multiple rendering processes to cooperate to draw a single world. You can implement this easily to introduce hardware shared memory systems but they are very expensive. In these days, network devices become cheaper and faster, so we can choose an information sharing mechanism via a local area network. But it is difficult to implement it.

We are developing the OpenCABIN library. It is designed for CAVE type VR display system rendered by multiple PCs which are connected via a local area network.

2. OpenCABIN library

OpenCABIN library is a fundamental software library for developing virtual reality applications. After CABIN library was developed at the university of Tokyo[Hirose 99], its successor, called OpenCABIN library, was developed from scratch as opensource software. It absorbs differences kinds of display systems, so we can easily develop applications for a multi-screen stereoscopic display. Figure 1 shows a view that its application is running at the K-Cave system at Keio University. When designing OpenCABIN library we think about more than one PC display system. Recently multi-screen display systems like CAVEs are consisted of multiple PCs instead of a high-end graphics workstation. In addition to this basic nature as a VR library, it has two special features that enable application programmers to develop VR applications easily: plug-in mechanism and master/renderer programming paradigm.

3. 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. 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.



Figure 1: The OpenCABIN library supports CAVE application development. An OpenCABIN library application is running at the K-Cave system at Keio University.

4. Master/Renderer Programming Paradigm

A multiple 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. 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

callbacks for renderer part. We call this OpenCABIN library philosophy: a master/renderer 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. This feature works fine not only at tele-immersive environment but also accessing some servers, for example database servers, license servers, dynamic web pages, web application services and so on.

5. OpenCABIN library for tele-immersive environment

OpenCABIN library is designed to abstract variations of display devices configurations. If you want to implement virtual conference system using multiple large display systems like CAVEs or tiled displays at remote places, you must think about variations of different display system configurations. But OpenCABIN library's display device abstraction mechanism reduces application development cost. Because of master/renderer programming paradigm, it is easy to develop some applications that communicate other display systems.

Oonuki et al. developed a seismic data visualization system in tele-immersive environment using OpenCABIN library [Oonuki 08].

6. Conclusions

We implemented a version of the OpenCABIN library and released it as an open source software. It can help to develop VR applications in large display systems that consist of multiple screens and PCs. Using OpenCABIN library, a tele-immersive application was implemented successfully. This result showed the effectiveness of design concepts of the OpenCABIN library.

References

- Hirose, M., Ogi, T., Ishiwata, S., Yamada, T., 1999. Development and Evaluation of Immersive Multiscreen Display "CABIN". Systems and Computers in Japan, Scripta Technica, Vol.30, No.1, pp.13-22.
- Oonuki, S., Tateyama, Y., Ogi, T. 2008. Seismic Data Visualization in Tele-immersion Environment. Asiagraph 2008 in Tokyo Proceedings, Vol. 2, No. 2, pp. 186-189.

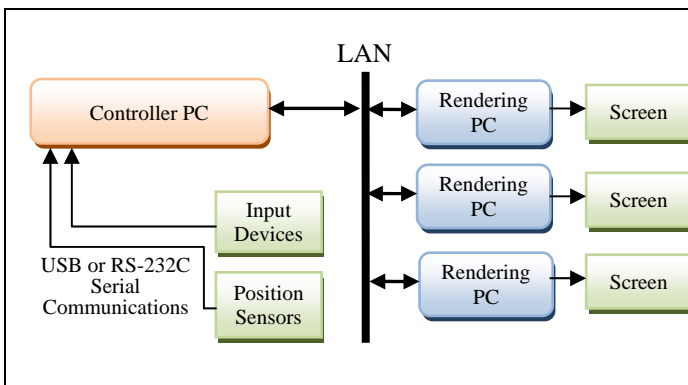


Figure 2: A typical large display hardware using multiple PCs.

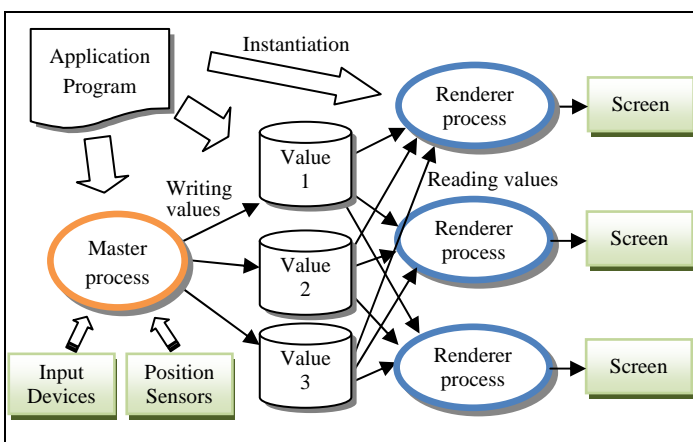


Figure 3: A master part and renderer parts are connected via shared values pointed by shared variables in a VR application.

mechanism and there are callbacks for master part and