Building Networked Immersive VR Applications Using the MVL Toolkit

Tetsuro Ogi
University of Tsukuba, 1-1-1 Tennoudai, Tsukuba, Ibaraki, 305-8577 Japan
tetsu@cc.tsukuba.ac.jp

Abstract

In this study, various functions that are required in an immersive shared virtual world were categorized into some groups such as sharing space, sharing users, sharing operations, sharing information and sharing time. Then the MVL Toolkit was developed to implement these functions. The MVL Toolkit contains several new technologies such as the stereo video avatar, the shared-type database, the cellular phone interface and the synchronization mechanism. By using the MVL Toolkit to add some functions, collaborative virtual reality applications can be easily constructed by extending existing stand-alone application programs. This library was applied to the development of several shared virtual reality applications and its effectiveness in enabling these tasks was evaluated.

Key words: Collaborative Virtual Reality Application, Immersive Projection Technology, VR Software Library, Video Avatar, Cellular Phone Interface

1. Introduction

Recently, virtual reality technology has been used not only for stand-alone applications, but also for networked applications. For example, typical scenarios for networked VR applications include collaborative design, scientific visualization, remote education, and so on [1][2][3]. In these applications, several challenging techniques such as guaranteeing the consistency of the shared virtual world, achieving natural communication between remote users, the collaborative manipulation of objects, the ability to access external resources through the network, and the synchronization between remote sites are required. Therefore, it is desirable that the user can use a software development tool that includes functions to solve these issues in order to develop networked VR applications easily.

In general, the functions that are required in the networked virtual world depend on the purposes of the applications. For example, an effective method for representing the image of a user would be different in a situation where remote users are talking to face to face in a virtual conference room than it would be if remote users are walking side by side in a virtual town [4]. Therefore, the software structures of the current collaborative VR applications are often complicated by the addition of various ad-hoc functions. On the other hand, several virtual reality software libraries that can be used to build a networked virtual world have already been developed [5][6][7]. However, it is difficult to use these libraries to develop collaborative VR applications by extending the existing stand-alone applications, because the developer is then required to redesign the software of the networked application.

We are aiming to construct a collaborative virtual reality application by extending a stand-alone application program, in order to utilize a lot of existing stand-alone contents. In this study, a software library which we called the MVL Toolkit was developed. This library can be used to extend a stand-alone application to a collaborative system without any special redesign of the software for adding several functions. In particular, the aim of this library is to construct an immersive shared virtual world by connecting high presence IPT environments such as the CAVE or CABIN [8][9]. In addition, the constructed shared virtual world must be used in practical applications, such as collaborative design, scientific visualization or remote education.

In order to meet such a demand, we first investigated various functions that are required in the immersive shared virtual world and categorized them into groups. Then, the MVL Toolkit was developed as a collection of small library sets. In the following chapters we will discuss the elemental functions required in the immersive shared virtual world and a method for implementing these functions with the MVL Toolkit. In addition, we will describe the software structure and the usage of this library. Finally, we will evaluate the effectiveness of this library by applying it to the development of several collaborative VR applications.

2. Elemental Functions

A shared virtual world that enables remote users to conduct collaborative work can be constructed by networking several virtual reality environments. This study focuses in particular on developing an immersive shared virtual world by connecting several immersive projection displays such as the C.C. at the University of Tsukuba, the CABIN at the University of Tokyo and the COSMOS at the Gifu Techno-plaza. Figure 1 shows the networked environment used in this study, in which remote users held discussions with each other while sharing various information resources.

In this paper, we investigated the functions that are required in the immersive shared virtual world, and categorized them into some elemental functions, such as sharing space, sharing users, sharing operations, sharing information and sharing time.

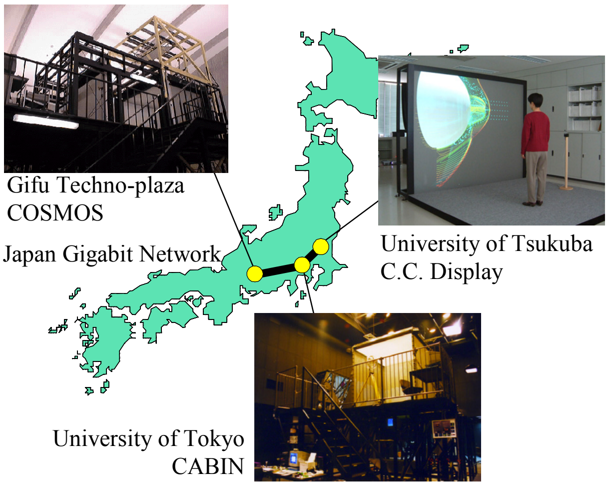


Fig. 1 Networked immersive projection environment used in this study

2.1 Sharing Space

In order to construct an immersive shared virtual world, the first requirement is that the remote users can experience the same space. This means that the geometrically same space is displayed at both sites and that consistency of the virtual space is guaranteed between each site, even when changes occur within it. Therefore, it is essential that the shared virtual world should include a function that can manage changes in the virtual space between remote locations.

2.2 Sharing Users

Another requirement is that remote users can see the other user's figures in the shared virtual world. Namely, the user's image should be visualized realistically, and it should be integrated at the correct position in the three-dimensional virtual space. By representing the positional relationship between remote users, they can communicate with each other in various situations such as meeting face to face or standing side by side in the shared virtual space.

2.3 Sharing Operations

Thirdly, in order to realize collaborative work in the shared virtual world, it is necessary that the users can operate virtual objects freely at both sites. In addition, any user's operation that impinges on local environment should be shared to guarantee the consistency of the shared virtual space. Therefore, a function is required that can transmit the operations performed by each user among all of the relevant sites without contradiction.

2.4 Sharing Information

Moreover, it is desirable for remote users to be able to share the required information to perform practical work. For example, when one user accesses information in the shared virtual space, the other users should also be able to refer to the same information. Therefore, an information management function is necessary in the shared virtual world. Of course, this function also should be able to conceal the information from remote users so that the primary user alone can refer to the information if required.

2.5 Sharing Time

Although the virtual world is generally updated according to the simulation time at each site, remote users should be able to experience the same environment at the same moment in the shared virtual world. In particular, when data is transmitted between remote sites, a time delay due to data communications is unavoidable. Therefore, an accurate clock control function that reduces the time gap between each site is required.

By realizing these functions, it would become possible for remote users to achieve common experiences and common understanding in the shared virtual world.

3. Technology for Constructing Immersive Shared Virtual World

In this study, the following techniques were developed in order to realize the above-mentioned functions.

3.1 Duplicate Program

The aim of this study is to develop a method of constructing collaborative VR applications by extending the existing stand-alone applications. The simplest method of running the same program at both sites was used as a fundamental approach to sharing the space in a networked application. In this method, updates of the shared virtual world are calculated in duplicate at each site, and these programs run while communicating with each other in order to guarantee the consistency of the shared virtual world. Both the TCP and UDP protocols were used as communication protocols for transmitting the data. When using this system, data that required a guaranteed transmission were sent using the TCP protocol, and data that required only the latest one were sent via the UDP protocol.

3.2 Video Avatar

Next, a method of representing a high presence image of the user is required for sharing with other users in the shared virtual world. In particular, in order to realize natural communication using gestures and facial expression, a representation of the user using a live video image is desired. In this study, video avatar technology that uses the live video image in the shared virtual world was developed as a communication technique. Figure 2 shows an example of the video avatar that is displayed in the immersive virtual world.

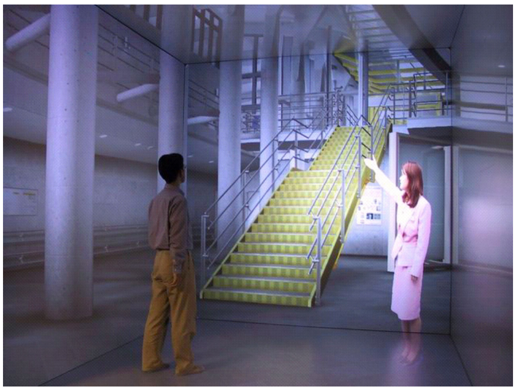


Fig. 2 Video avatar displayed in the immersive virtual world

Video avatar is a technology that can be used for communication between remote places. This method transmits the user's live video image mutually and superimposes it in the shared virtual world [10]. In this study, stereo video avatar technique was specifically developed in order to realize high presence communication using a stereo video image [11]. Figure 3 illustrates the method of generating the stereo video avatar. In this method, the user's image is captured by the stereo video camera and the depth value for each pixel of the captured image is calculated using a stereo matching algorithm. From the depth value for each pixel of the captured image, the user's geometric surface model can be created and their image can also be segmented from the background. By texture mapping the segmented video image of the user on the geometric surface model, a video avatar of a 2.5-dimensional figure is generated. This video avatar can be used for the communication by transmitting it mutually with the positional information through the UDP socket.

In this method, the positional relationship between remote users in the shared virtual world can be represented by placing the video avatar image at the user's existing position. Moreover, since the stereo video avatar incorporates the user's geometric model, it can be used effectively for the communication using gesture, such as pointing at an object. In addition, the video avatar data can also be used repeatedly in the offline mode to represent the high presence image of the human in the immersive virtual world, by storing them in the database.

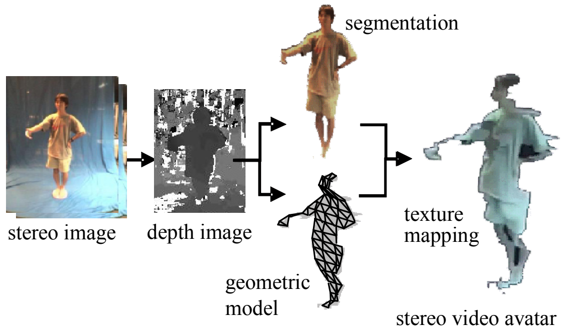


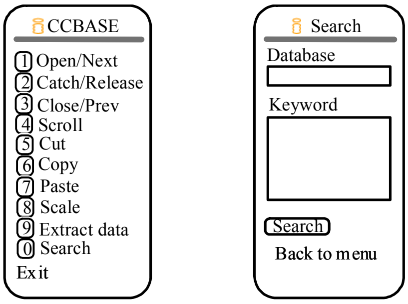
Fig. 3 Making method of 2.5 dimensional stereo video avatar

3.3 Interaction Server

In order to share operations carried out by the user in the shared virtual world, a method of transmitting data about the operation to the other sites is needed. In this case, when different operations are carried out at each site, exclusive control is generally required to maintain the consistency of the virtual world between the remote sites. In this study, an interaction server was prepared at one site and the interaction method using the cellular phone or the wireless PDA was developed to solve this problem [12].

Since the interaction server receives operation commands that were input from every site on the system, both users can access the server computer directly. The interaction server transmits the input commands to the other client sites using the TCP protocol immediately after receiving them, and then the same commands can be executed at all relevant sites. In this method, the consistency of the shared virtual world can be guaranteed between remote sites without any exclusive control process by simply performing the input commands in the order that they are received at each site, even when different commands are input.

In this system, the user inputs the interaction command using the i-mode function of a cellular phone. The i-mode is an Internet access service provided by the NTT DoCoMo, by which the user can access a computer connected to the Internet through the i-mode center. Figure 4 shows the example of the i-mode menu displayed on the cellular phone to interact with the virtual world. Since a feature of the cellular phone interface is the capability for communication with the interaction server using a wireless device, other devices such as a PDA equipped with a wireless LAN can also be used for inputting operation commands. In addition to this, the cellular phone device can also be effectively used to input characters in the virtual world, by pushing the dial buttons assigned to alphabet letters. Figure 5 illustrates the method of the interaction using the cellular phone interface in the shared virtual world.



(a) Command selection (b) Keyword search

Fig. 4 Examples of the i-mode interaction menu

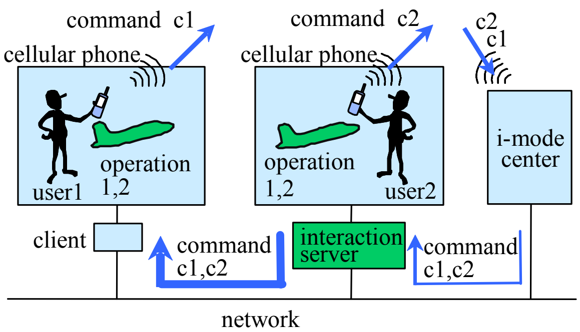


Fig. 5 Interaction mechanism using the interaction server and cellular phone interface

3.4 Database Access

In this study, the shared-type database interface named CCBASE (Cyber Communication data BASE) was developed as a framework for sharing information in the shared virtual world [13]. CCBASE is a database interface system in which the user can access a multimedia database through a network and then share information in the shared virtual world. Various kinds of data can be manipulated using this system, such as images, sound, movie-clips, three-dimensional models and PowerPoint files. Therefore, the user can manipulate required data in the shared virtual world, by storing them in the database system and taking them into the virtual world using this system.

The data taken into the shared virtual space is visualized using the book metaphor, as shown in Figure 6, and it can then be extracted from the book to be manipulated in the virtual world. In this system, when the user opens the book or inputs a keyword using the cellular phone interface device, an SQL command is generated and is sent to the database system. Thus, the user can retrieve the data through the network and refer to it in the virtual world.

When CCBASE is used in the shared virtual world, the operation command received by the interaction server is transmitted to the client using the above-mentioned sharing operation function. The operation commands for the database system are divided into two different groups, those that affect the contents of the database and those that do not affect the contents of the database. Although commands that affect the contents, such as data registration or data deletion, are executed only at the server site, commands that do not affect the contents, such as data retrieval, are executed at both sites in order to share the data between remote sites. In addition, an optional concealment function can be used so that only a single user at one site can access the database and refer to the information while concealing it from the other users. Figure 7 shows the data sharing mechanism used in the CCBASE system.

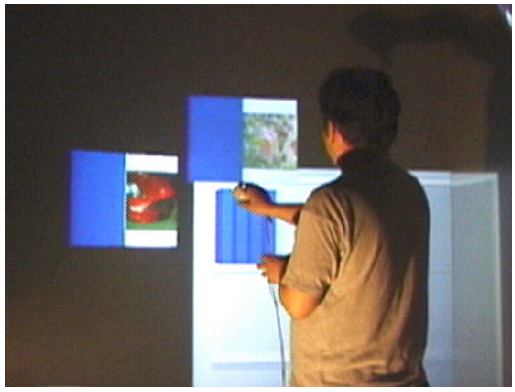


Fig.6 Visualized Data using book metaphor in the virtual space

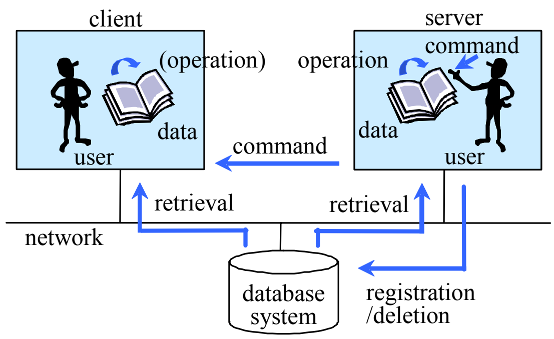


Fig.7 Data sharing mechanism used in the CCBASE system

3.5 Time Control

In order to share the correct time in the shared virtual world, it is a requirement that clocks are adjusted at all of the sites and that time passes correctly at each site. In this study, in order to meet such a requirement, extra functions governing time passage control and clock adjustment were implemented.

In ordinary virtual reality applications, the calculation for updating the virtual world for the next time step is iterated by advancing the clock. However, since the time required for calculating the movement of an object or for rendering an image is variable, time often passes irregularly at each site of the shared virtual world. Therefore, in this study, a time passage control function was introduced. In this method, the time taken to calculate the movement of the objects and to render the image is measured in every simulation loop, and the next time-step is determined according to the elapsed time. By performing this process repeatedly, accurate time passage control can be realized at each site.

Next, in order to realize accurate synchronization, the clocks must be adjusted between the remote sites. In this system, a clock adjustment method was also introduced to realize the synchronization. In this method, when the operation data is sent from the interaction server to the clients, the time data is also sent with it, and the clock at the client site is adjusted to the server site.

Thus, it has become possible to share the correct time between remote places by adjusting the clocks and controlling the passage of time. Figure 8 shows the process of the time passage control and the clock adjustment method that we used in the shared virtual world. In this study, the communication experiment was conducted between the University of Tokyo and the Gifu Techno-Plaza through the JGN (Japan Gigabit Network). These two sites are about 300km away, and the time gap was measured when the animation data was visualized at both sites. The results showed that the time gap between these sites was less than 0.06 sec when the time control method was used.

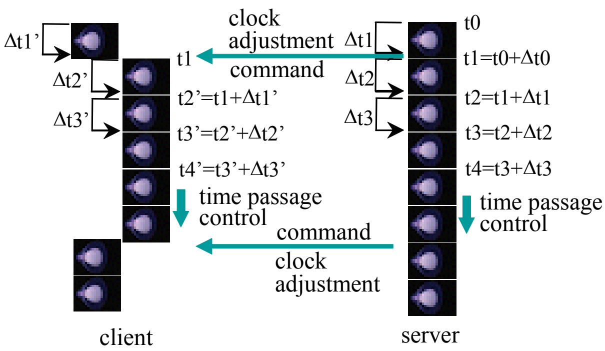


Fig.8 Process of time passage control and clock adjustment method

4. MVL Toolkit

In this study, the software library named MVL Toolkit was developed in order to implement the above-mentioned functions. Figure 9 shows the software structure of the MVL Toolkit. The MVL Toolkit consists of several library function sets, such as the *mlSvw* library, the *mlAva* library, the *mlCel* library, the *mlCcb* library and the *mlTim* library, which were developed corresponding to each of the elemental functions. Therefore, the user can selectively use the required functions to develop the appropriate collaborative VR application.

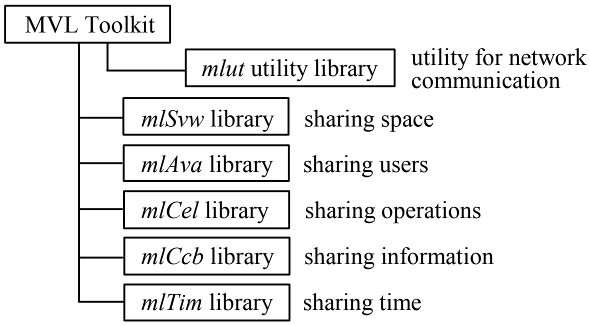


Fig.9 Software structure corresponding to the categorized functions of the MVL Toolkit

The *mlSvw* library is an API set for "sharing space". By using the *mlSvw* library, the basic functions of sharing the geometric virtual space are added and the stand-alone application program can be extended to the collaborative system. Typical functions of the *mlSvw* library include making the TCP and the UDP connections between remote sites, transmitting the changed data, setting the users' initial positions in the shared virtual space, etc.

The *mlAva* library provides the APIs to use the video avatar communication functions for "sharing users". Typical functions of the *mlAva* library include generating and updating the video avatar data from the live video image, retrieving the required video avatar from the recorded data, integrating the video avatar image in the virtual space, etc. In this method, a two-dimensional plane model and a 2.5-dimensional surface model can be used as the video avatar data.

The *mlCel* library provides the interaction functions using the cellular phone and PDA devices for "sharing operations". This library includes functions such as starting the interaction server, receiving the command from the cellular phone or the PDA devices, transmitting the input command from the interaction server to the client sites, etc.

The *mlCcb* library is an API to use the CCBASE functions for "sharing information". By using the *mlCcb* library, the functions of accessing the database are added to the virtual reality applications. Typical functions of the *mlCcb* library include generating the SQL command from the user's action, accessing the database through the network, visualizing the retrieved data, managing the data taken into the shared virtual world, etc.

The *mlTim* library can be used for “sharing time” between remote sites. This library provides functions such as transmitting and receiving the clock data, adjusting the clock, measuring the elapsed time, setting the next time step in the simulation loop, etc.

In addition, the *mlut* utility library is prepared to provide the basic functions for the network communication. The functions of this utility library includes allocating the shared memory, opening and closing the network socket, sending and receiving the data, etc, which are then called in the above-mentioned library functions.

Since the MVL Toolkit is constructed as a collection of these library sets and is capable of being used independently of the basic VR software, such as the CABIN lib., the user can add the required functions selectively to the existing program by using each library set. Therefore, a collaborative VR application can be developed easily by extending the stand-alone application program. Figure 10 shows the software construction of the virtual reality application program that uses the MVL Toolkit, and Figure 11 shows an example of the program source code. In this example, the MVL Toolkit is used to extend the single-user application that uses the OpenGL and the CABIN lib.

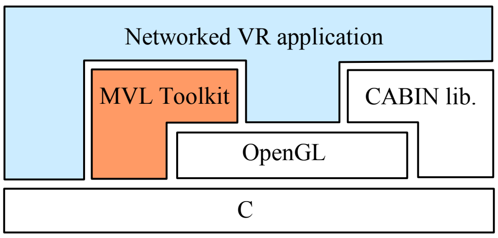


Fig.10 Software structure of the networked VR application using the MVL Toolkit

```
main(int argc, char **argv)
{
  glCABINinit(argc,argv);
  mlSvwInit(&mltest->svw, svw_site, index, 0, NUM_SCREENS);
  mlAvaInit(NULL, GLCABINindex, FRONT);
  mltCcbInit(NULL,NULL,NULL,NULL,NULL,NULL, &mltest->ccbt_info,
     "test", ccbt_site, index, 0, NUM_SCREENS, execute, quit, TRUE);
  myinit();

  glCABINdisplayFunc(display);
  glCABINidleFunc(idle);
  glCABINultraFunc(ultra);

  glCABINmainLoop();
}

void display(void)
{
  mlSvwCoordinate(&mltest->svw, SVW_WORLD);
…..
  mltCcbDisplay(&mltest->ccbt_info, (mlCcbUltraData *)&ultra_hand);
  mlAvaDisp();
…..
}
```

Fig.11 Example of program source code using the MVL Toolkit

Since the MVL Toolkit is designed so that the functions are divided into five library sets corresponding to each elemental function, it is not necessary to use all of the library functions simultaneously. For example, when only the *mlAva* library is used, a stand-alone application using the video avatar can be built, and when only the *mlCcb* library is used, a single user application in which the user can access the database can be developed. However, when the *mlAva* library is used together with the *mlSvw* library, the video avatars can be used for mutual communication in the shared virtual world.

5. Applications

In order to evaluate the effectiveness of the MVL Toolkit, several kinds of networked immersive VR applications such as the flow field visualization system and the psychological experiment system were developed using this library.

Figure 12 shows the application to the flow field visualization system [14]. This system is aimed at realizing a situation in which remote researchers can discuss with each other while looking at the same visualization data in the immersive shared virtual world. In this particular example, analysis data regarding supersonic opposing jet flow were visualized as a three-dimensional animation using several visualization methods, such as streamlining, particles, contours, shading and volume rendering.

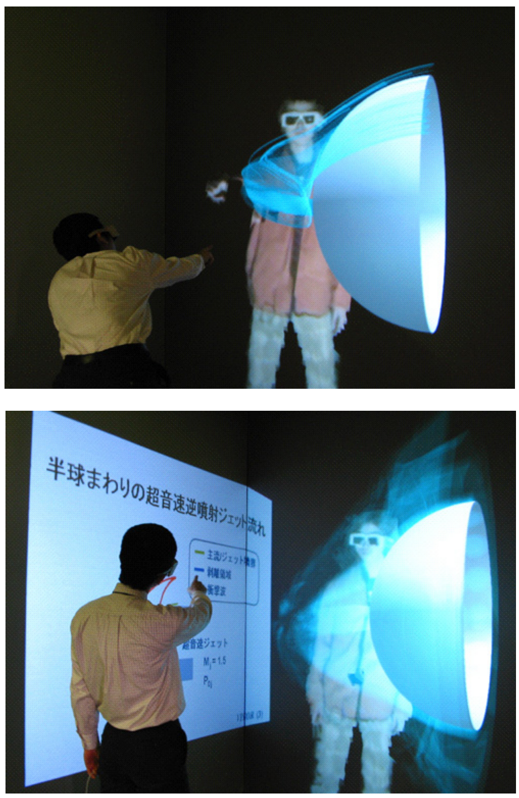


Fig.12 Collaborative flow field visualization system using the MVL Toolkit

This system was constructed by extending a single user application. In this application, the user's figure was visualized using a 2.5-dimensional stereo video image by adding the video avatar function. The users were able to interact with the visualized data from each site, such as changing the visualization method or changing the visualization data, by using the cellular phone interface. Moreover, by using the database interface function, the user was able to explain the calculation method to the other user via a PowerPoint presentation in the shared virtual world. In addition, by adding the time control function, the three-dimensional animation data could be visualized synchronously between both sites.

In this study, we asked researchers in the field of fluid dynamics analysis to use this system in the networked immersive virtual environment. The results show that the remote researchers were able to discuss effectively with each other by using the video avatar, by PowerPoint presentation and by three-dimensional gestures in the shared virtual world.

In another example, the MVL Toolkit was used to develop a psychological experiment system in the immersive shared virtual world [15]. This system was used to investigate how accurately a person's pointing gesture could be recognized by other people. This application was also developed by extending a stand-alone program to the networked system, and it was used for the psychological experiment in the networked immersive virtual environment. In this experiment, a psychologist gave instructions for the experiment to remote subjects, and the psychologist observed the subjects' behavior and recorded the experimental data in the shared virtual world.

The psychologist and the subjects were visualized using video avatars, and they performed the experiment while looking at the other users' images. In particular, the psychologist was able to observe the subjects' behavior from a close position in the shared virtual world without being observed by the subjects if only the subject's video avatar was transmitted one way. Moreover, the three-dimensional model data of the person who was pointing in the various directions was stored in the database system, and it was retrieved and displayed in the virtual world using the CCBASE function. The subjects input their answers using a PDA device equipped with a wireless LAN connection, and the psychologist was able to analyze the experimental data at the remote location through the network.

This psychological experiment was actually carried out for more than ten subjects using the networked environment, and almost the same results were obtained as were obtained for an experiment using a stand-alone system. Figure 13 shows that the subjects are experiencing the psychological experiment system in the immersive shared virtual world.

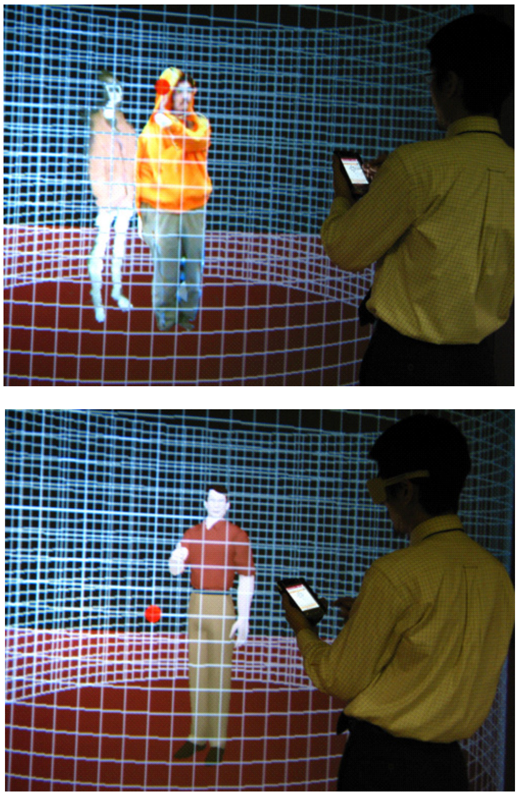


Fig.13 Networked psychological experiment system using the MVL Toolkit

From these examples, we can conclude that the MVL Toolkit was effectively used to add the elemental collaboration functions and then to develop the networked VR applications.

6. Conclusions

In this study, the functions that are required to construct an networked immersive virtual world were categorized, and they were implemented as the MVL Toolkit. The MVL Toolkit consists of the several library functions, such as sharing space, sharing users, sharing operations, sharing information and sharing time. By using the MVL Toolkit, the user can easily develop networked immersive VR applications by selectively adding the required functions to a stand-alone application program. In this study, the MVL Toolkit was applied to the development of several applications, and the effectiveness of this library was evaluated.

7. Acknowledgement

This research was supported by Telecommunications Advancement Organization of Japan, and the authors thank Toshio Yamada, Shigenori Toyota, Masahiro Fujita, Kazuto Kuzuu, Eiji Shima, Akio Ochi, Takuro Kayahara and Michiteru Kitazaki for their cooperation in the evaluation experiments of this library.

References

1. T. Korpippa, K. Minami, T. Kuroda, Y. Manabe and K. Chihara, "Shared Virtual Reality Interior Design System", ICAT2000, pp.124-131, 2000

2. N. Jensen, S. Seipel, W. Nejdl, S. Olbrich, "COVASE: Collaborative Visualization for Constructive Learning, CSCL Conference 2003, pp.249-253, 2003

3. K. Park, J. Leigh, A. Johnson, B. Carter, J. Brody and J. Sosnoski, "Distance Learning Classroom Using Virtual Harlem", VSMM 2001, pp.489-498, 2001

4. T. Ogi, T. Yamada, Y. Kurita, Y. Hattori and M. Hirose, "Usage of Video Avatar Technology for Immersive Communication", Proceedings of First International Workshop on Language Understanding and Agents for Real World Interaction, pp.24-31, 2003

5. E Frécon, M. Stenius, "DIVE: A Scaleable Network Architecture for Distributed Virtual Environments", Distributed Systems Engineering Journal, Vol.5, No.3, pp 91-100, 1998

6. K. Park, Y. Cho, N. Krishnaprasad, C. Scharver, M. Lewis, J. Leigh and A. Johnson, "CAVERNsoft G2: A Toolkit for High Performance Tele-Immersive Collaboration", ACM VRST 2000, pp. 8-15, 2000

7. Jeremie Allard, Valerie Gouranton, Loick Lecointre, Emmanuel Melin and Bruno Raffin, "Net Juggler: Running VR Juggler with Multiple Displays on a Commodity Component Cluster", IEEE VR 2002, pp.273-274, 2002

8. C. Cruz-Neira, D.J. Sandin, T.A. DeFanti, "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE", SIGGRAPH'93, pp.135-142, 1993

9. M. Hirose, T. Ogi, S. Ishiwata and T. Yamada, "Development and Evaluation of Immersive Multiscreen Display CABIN", Systems and Computers in Japan, Vol.30, No.1, pp.13-22, 1999

10. J. Insley, D. Sandin, T. DeFanti, Using Video to Create Avatars in Virtual Reality, Visual Proceedings of 1997 SIGGRAPH, pp.128, 1997

11. T. Ogi, T. Yamada, M. Kano and M. Hirose, "Immersive Telecommunication Using Stereo Video Avatar", IEEE VR 2001, pp.45-51, 2001

12. T. Ogi, K. Yamamoto, T. Yamada, M. Hirose, Experience of Virtual World Using Cellular Phone Interface, IEEE PCM 2001, pp.32-39, 2001

13. T. Ogi, K. Yamamoto, T. Yamanouchi, M. Hirose, Information Access Interface in the Immersive Virtual World, The Transactions of Human Interface Society, Vol.4, No.4, pp.197-205, 2002 (in Japanese)

14. K. Kusuu, M. Fujita, T. Ogi, Y. Matsumoto: Three-dimensional Flow Visualization in the Shared Immersive Virtual Space, Proceedings of 4th Pacific Symposium on Flow Visualization and Image Processing, 2003

15. T. Kayahara, M. Kitazaki, T. Yamada, T. Ogi, M. Hirose, Effect of Observer's Self-motion Sensation on Scene Recognition in Immersive Virtual-Reality Environment, ECVP 2003, 2003