CESNET Technical Report 23/2008

CAVE to CAVE: Communication in Distributed Virtual Environment

Roman Berka, Zdeněk Trávníček, Vlastimil Havran, Jiří Bittner, Jiří Žára, Pavel Slavík, Jiří Navrátil

Received 30.12.2008

Abstract

Virtual environment (VE) is often used as a tool in the design process or as a part of visualisation techniques which make it possible for the user to enter space which is otherwise not accessible or even does not exist. Virtual environments can also be used for collaboration when some of the users are not present locally. The remote users communicate with local users via network which requires configuration based on more VEs (e.g., CAVE-like devices) connected together. The applications used to control such devices are usually designed to communicate with the same type of the remotely connected system. On the other hand, the communication between heterogeneous systems becomes increasingly important with the technological advances in presentation technologies. This report describes a project which aims to transmit video signals between heterogeneous VE devices over a high speed network connection.

Keywords: virtual environment, CAVE, high-speed network

1 Motivation

1.1 A Case Study

The VEs based on CAVE-like devices [3], [4] are usually used to visualize any modelled virtual environment (which does not exist, e.g., evaluation of a new interior design) or is not accessible (e.g., surface of a distant place, a planet, the bottom of the ocean). These devices can be entered simultaneously by more than one user. A configuration of the CAVE system with three projection walls is shown in Figure 1. It usually uses software producing panoramic stereoscopic views to the virtual scene in real-time. Using an equipment to track motions of users it is possible to interact with the presented scene and its parts. Two or more CAVE systems can be connected together and the immersion of the visualized data can be shared by different groups of users. Each of the groups can stay in their CAVE laboratory and just share the virtual environment using the network connection. This allows for example to perform shared walk-throughs of the VE or common exploration of visualized simulation processes.

The remote connection of VEs can be divided into several categories based on a degree of interactivity between its participants:

1. Remote presentation
2. Remote interaction
3. Interaction in a colaborative virtual environment

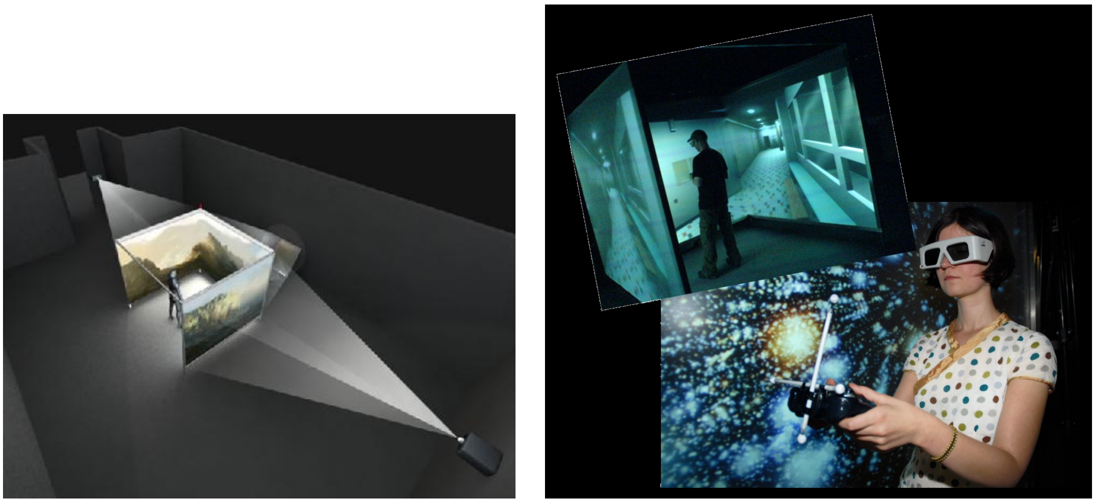

Figure 1. The CAVE system

1.1.1 Remote presentation

The remote presentation makes it possible to passively enter the remote virtual space and watch processes inside. As an example we can imagine virtual remote visits of groups to historical buildings or to „Jurassic park“-like places. The difference from a normal movie is that in this case all (or the most of) presented material is computed in real-time. This brings a new quality which could be compared with the idea of „Kinoautomat“ from 1960's, but owing to contemporary technologies we can reach a new form between classical movie and theatre. The processes inside the scene can be then influenced by various events changing actual conditions (e.g., weather simulations or by branching the story by observers' voting).

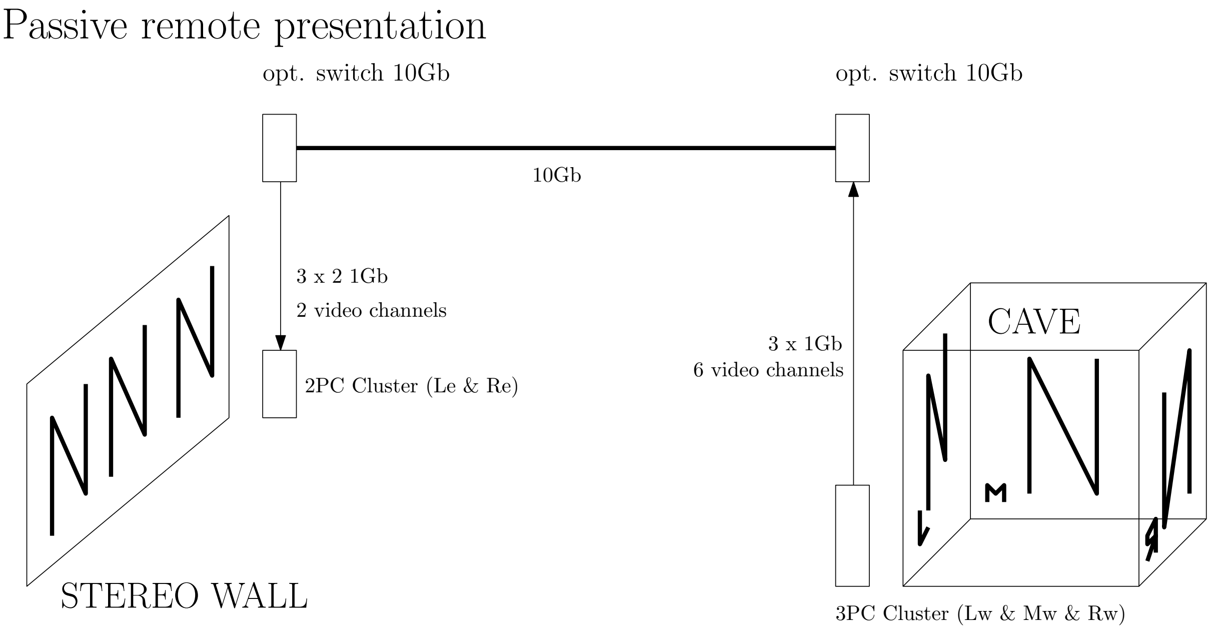

Figure 2. Passive remote presentation scheme CAVE stereo wall

The technical solution using transport of the multimedia data over the network allows a parallel presentation to more groups of observers located at different places around the world. The scheme of such application based on equipment developed by our team is presented in Figure 2. Here, the visualisations are computed by the computer cluster of the CAVE system and are presented to the users inside the CAVE where an interaction with the scene is also possible. In addition, the visualised output is also transmitted to the remote stereoscopic wall, where another group of users can passively watch the processes and the interactions taking place in the CAVE.

1.1.2 Remote interaction

The remote interactive application allows the user to walk through a remote virtual scene and to interact with objects inside. The virtual environment can represent a real scene built from pre-measured data. This configuration expects that the users are equiped by a tracking system which allows to interact with the scene. There are several applications where a group of users can share a scene with another distant group [6]. In addition, the motion tracking gives a possibility of interaction. The typical such application then enables different groups of users to edit some common data (architecture, scientific visualisations, etc.).

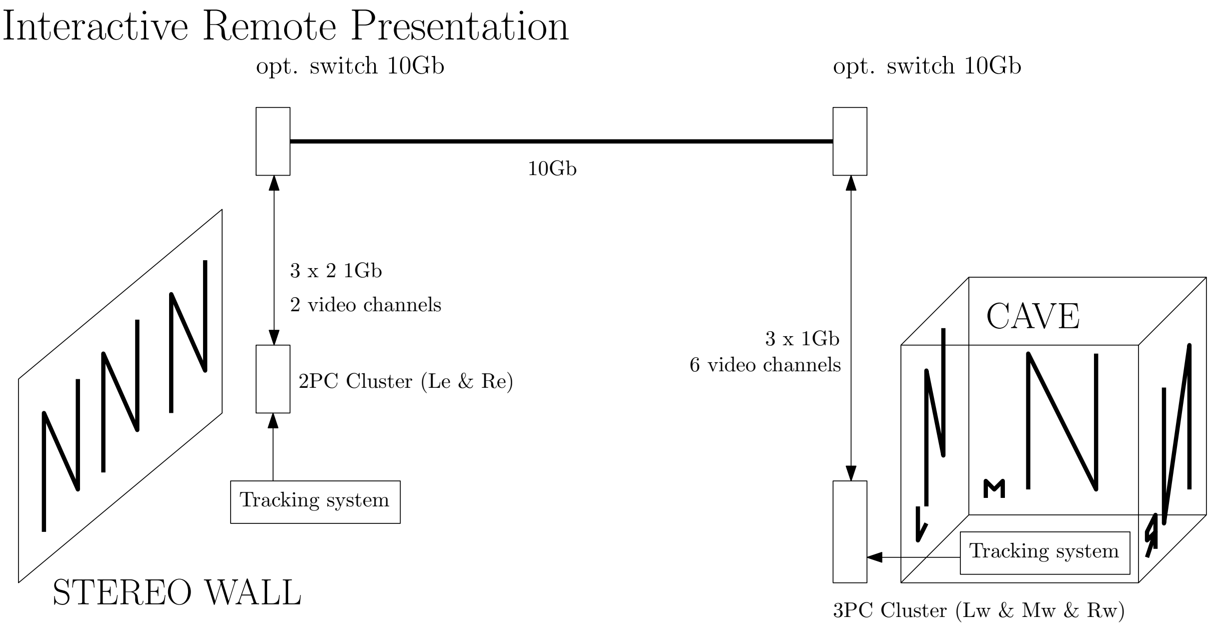

Figure 3. Interactive remote presentation CAVE stereo wall

The remote interaction can use various resources to influence the input data e.g., a remote control of a mobile robot which is able to obtain geometry representation of the screen. Then it is possible that the user can walk through a virtual scene sent by the remotely controlled robot (e.g. underwater scenery, dangerous environment, such as house on fire).

The scene representations sent by a distant source of data can be then rendered by the CAVE computer cluster and distributed to other VE devices. Owing to possibility of interaction all participating groups of users can remotely control the process (e.g., motions of the robot).

1.1.3 Interaction in Collaborative Environment

In this case, the rendered scene and the real video of users are combined together with a help of motion tracked data. This allows the users to see not only the shared scene and interact with it but they can also see their remote colleagues in the form of a video rendered and mixed directly to the virtual environment [1].

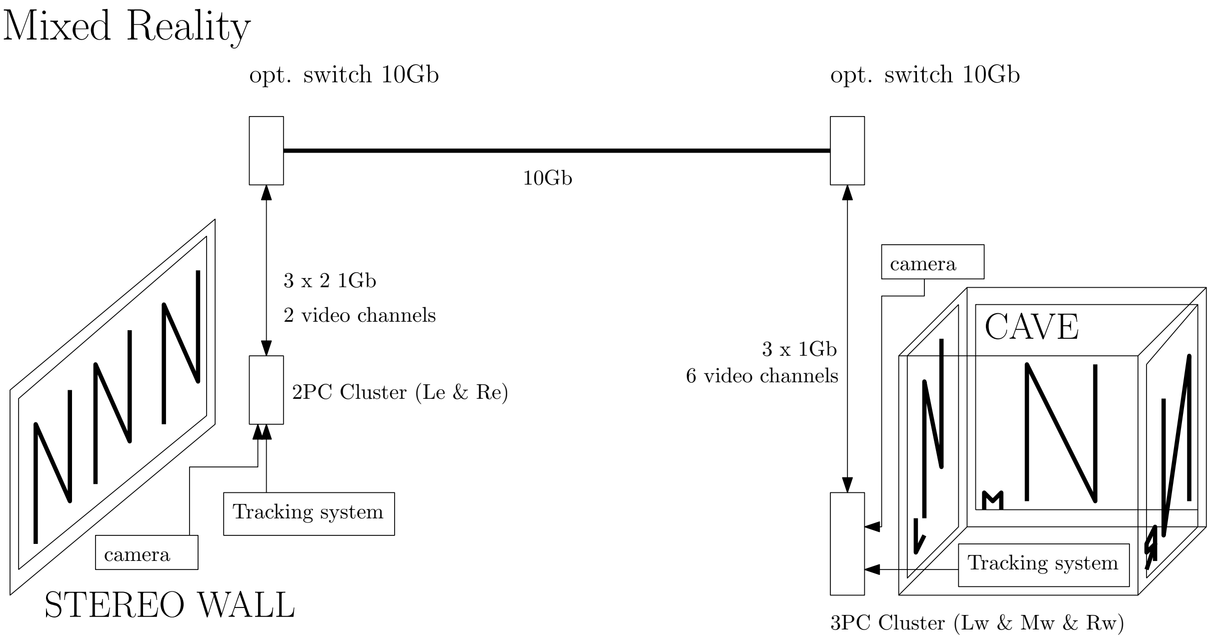

Figure 4. Mixed reality CAVE stereo wall

The technical concept is then based on similar idea like TV virtual studio where live video is composed by keying virtual scenes. The application of such an approach then leads to a distributed collaborative environment based on the video signal distribution and on the real-time composition.

1.2 Goal of the Project

To connect the CAVE-like devices we need a protocol which enables the communication and the interaction of all participating users. There already exists a possibility of communication between two CAVE-like systems based on transmission of vector-data representing the changes in the virtual world. This approach usually expects homogeneous distributed environment with the same clients on both sites. The implementations of this approach often use the CAVElib library, which also have to be used by the distant distant client so that it can receive the transmitted data.

In contrast we propose to use a standard data format for communication, which makes the presented data more accessible for remote users using standard visualisation tools as e.g., common video-player. Specifically, we propose to grab and stream real-time stereo video signal that is used to share the views to the VE.

The goal of the project is to develop methods, based on contemporary technologies, enabling VE to be shared by users independently on various VR devices and making it possible to collaborate inside the VE. To satisfy this goal we will try to find a way how to take the necessary data from a VE device computer controling system and transmit them to the remote system. The first approach will be based on a video data transmission and the remote system is not necessarily the same type as the local one.

2 The Project Progress in 2008

The project CAVE2CAVE (C2C) is an activity of three cooperating partners: Institute of Intermedia, Faculty of Electrical Engineering, Czech Technical University in Prague (IIM), Department of Computer Graphics and Interaction, Faculty of Electrical Engineering, Czech Technical University in Prague (VRLab), and CESNET, association of legal entities. The first two partners have laboratories working in the area of computer graphics, multimedia and virtual reality, CESNET plays the role of the connectivity provider. The goals for year 2008 were characterised by preparation of an equipment and concepts which were then evaluated. First of all, the method of graphical information acquisition from virtual devices has been designed and tested (see the following text). Each of the three cooperating partners had to setup the technology and equipment associated with the particular role of the partner in the project.

2.1 Institute of Intermedia

IIM, as laboratory equiped by virtual reality system - CAVE, prepared the device for connection to the high bandwidth network technology and prepared the implementation of the video grabing module. The development and testing proces of this method is described further in this section.

IIM also initiated the preparation of tools enabling acquisition of stereo-video signal using special stereo-camera system. This tool was prepared as a part of a student project and it will be used in the next phase of the C2C project during 2009. This constructed system represents stereo video production chain which can be used to capture real video and transport it to the remote projection system. The configuration of the system is based on two-fixed-camera system (Figure 5) connected to the controlling computer by two ISO1394 cables or directly connected to a network using ethernet interface (Figure 6). The control computer then processes two video signals and sends them to the remote presentation system. The two-channel video then can be presented on stereoscopic projection systems or auto-stereoscopic system.

Two industry level cameras with 1Gbit ethernet interface and auto-stereoscopic display were acquired during 2008 and will be incorporated to the project equipment in next project part. Another tool prepared during students' participation in the project is the motion tracking system VICON which was tested to be used in the CAVE system to track the user and his interactions. Both the presented tools are parts of the distributed VR collaboration system described above in Remote Interaction and Interactive Collaborative Environment case studies.

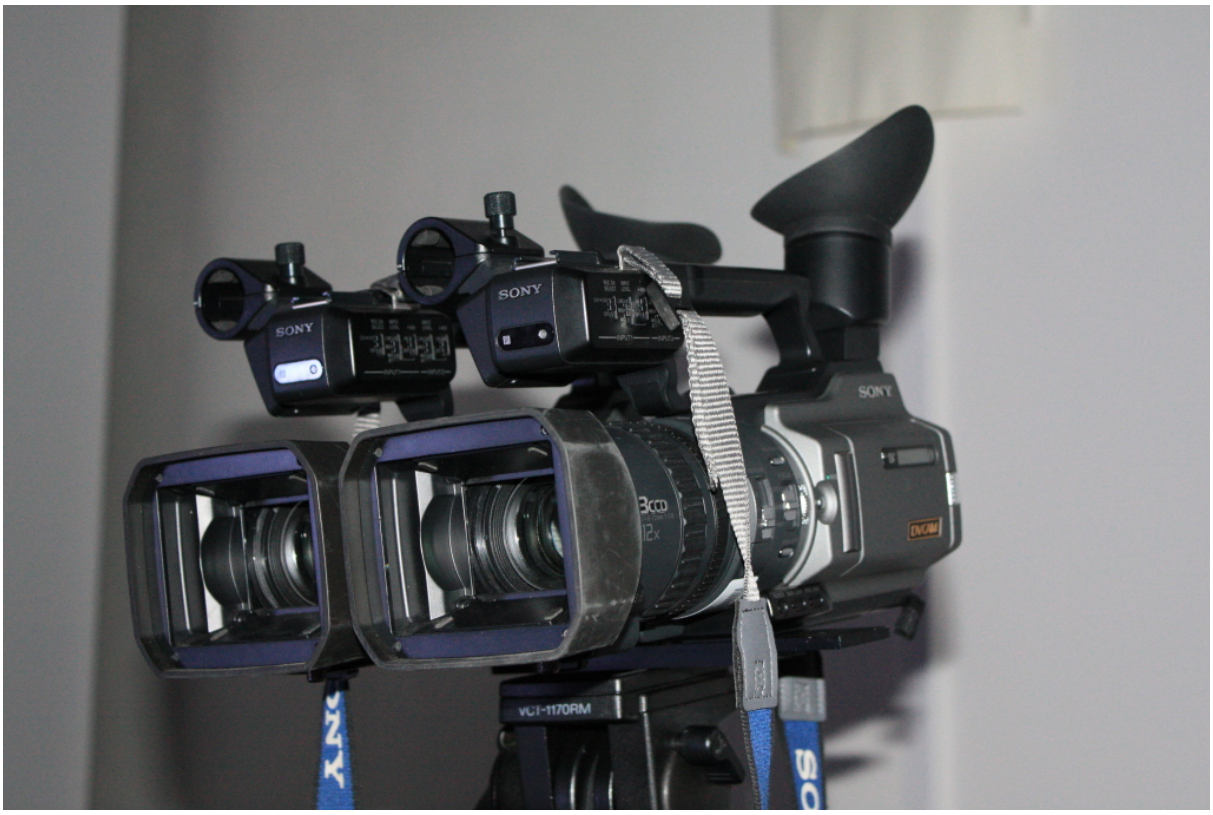

Figure 5. Stereo camera setup

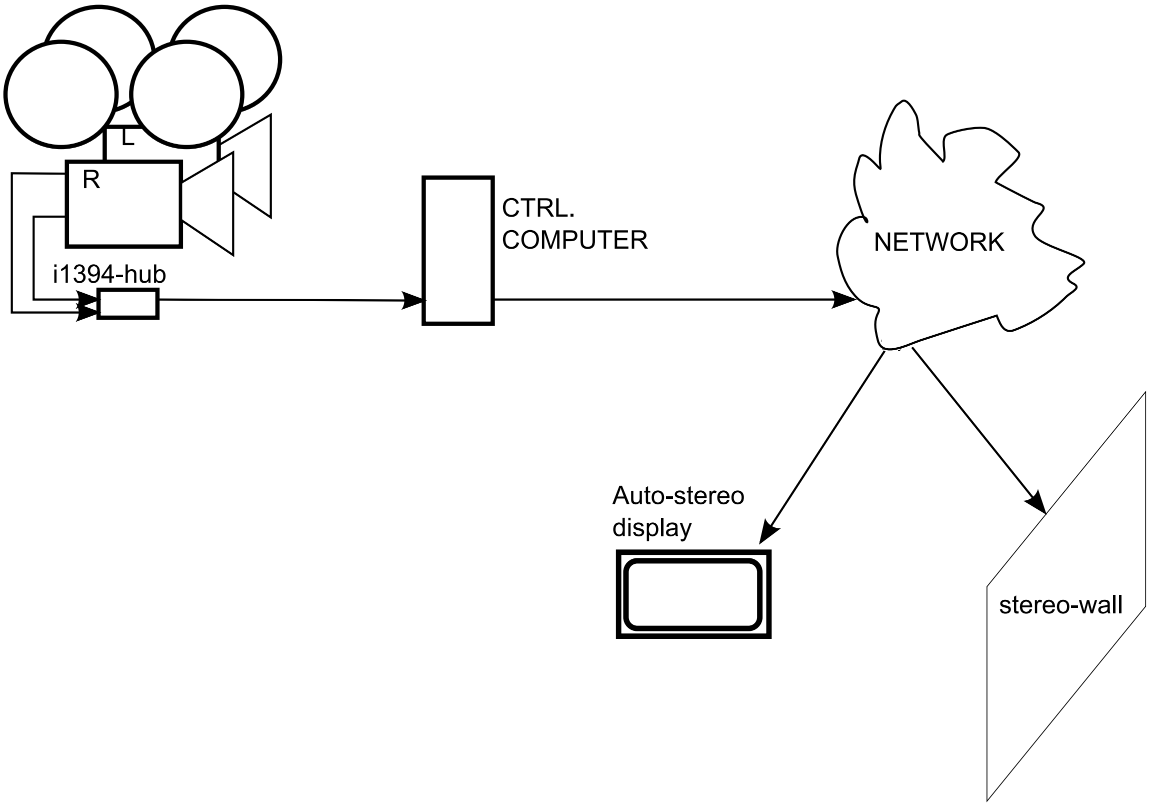

Figure 6. A stereoscopic chain

2.2 Laboratory of Virtual Reality (VRLab), DCGI

The second side of the communication chain has been prepared in the VRLab. The most important was to setup the stereoscopical wall (supplied by EON Reality) and to acquire the computer equipment consisting of two control computers connected directly to IIM using an optical 10Gbps link and the computational server. Next, the VRLab also tested the motion tracking system (A.R.T.) newly acquired to its equipment in 2008. Using all these devices with the already mentioned special 1Gb cameras will enable key experiments with collaborative distributed environment planed for 2009.

2.3 CESNET

The role of CESNET was to setup the connection between IIM and VRLab using 10Gbps optical link. This was succesfully realized and the link has been tested in the third quarter of 2008. In addition, CESNET supported acquisition of the necessary technical devices (control computers for stereoscopic wall in VRLab, special 1Gb cameras, and auto-stereoscopic display as an alternative to the stereoscopic projection systems).

3 Technical Details of Problems Solved in 2008

3.1 Connecting VR Devices

In many applications more VR devices communicate in order to establish collaborative environment [2]. Here, the vector-data and video signals are used allowing users to exchange the audio-visual information. The communication of such devices requires a suitable protocol. As there exist more solutions of how to acquire the data from the source, we have to recall the whole visualization pipeline. Figure 7 represents the pipeline where a data describing a scene are loaded by an application which stores them in its internal form. The data are then sent to GPU, then rendered and sent to the output in the form of VGA/DVI signal.

The first possibility lies in a simple data exchange at the level of the shared file system. This approach expects that local and remote applications have access to the same file and requires synchronization of the appropriate processes.

The second approach is based on the communication at the level of the vector-data exchange between applications [7]. This is the most typical solution often existing in a form of a run-time library. One example could be also the library CAVElib which is used by many CAVE systems over the world including the one at Institute of intermedia at the CTU in Prague.

The next possibility is based on the idea to fork the flow of OpenGL commands before they are sent to the GPU and transmit them also to the remote station where these commands are then executed. This solution requires remote cluster with enough power to perform rendering process and some modifications in the remote execution pipeline, for example remapping of IDs for display lists. On the other hand this method have also advantages. One of them is the possibility of simple conversion from mono-scopic channel to stereoscopic reproduction on the remote system.

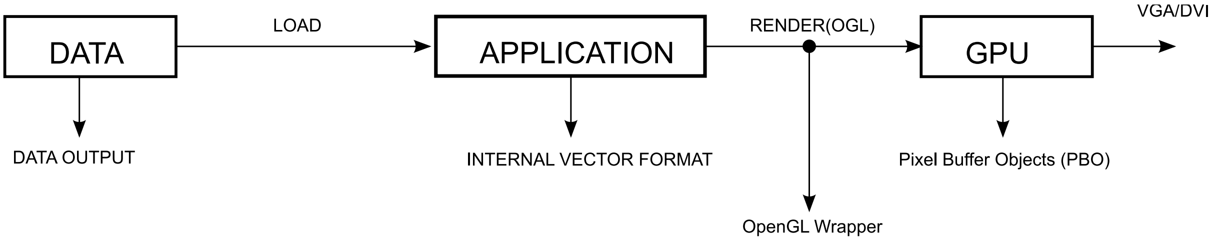

Figure 7. Data flow scheme

The fourth method is based on a process reading rendered data directly from the GPU buffers and sending them to the remote VR system. This case is related to the video-signal transmission. The weakness of this method lies in a bottleneck represented by the process reading the video-buffer. We will show our solution of this problem next in this report.

Finally, the fifth method uses hardware solution when the video-signal is captured directly from the VGA/DVI output of the graphical subsystem converted to video-stream and then sent through the network. This approach is also based on the video/signal transmission. Its advantage is the rate of data process but the disadvantage could be impossibility to control the render process. There is just only one resolution and frame rate on a graphic card output. Thus this approach has lower degree of usability when synchronization control of the multiple channel signal (e.g., stereo video) is required.

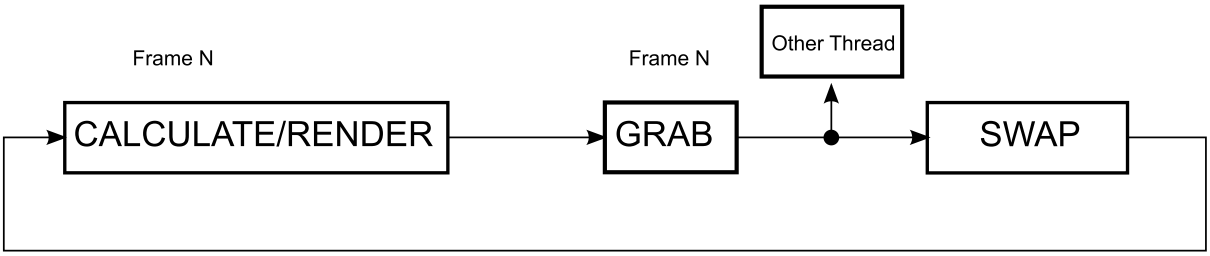

Figure 8. Direct frame buffer access method

4 The Video Transmission Based Methods

The communication of VR devices using video signal transmission is not a typical problem. We require a fast and well controlable data source and high bandwidth according to resolution and frame-rate. On the other side, the video signal transmission allows the use of video transmission standards (e.g., for streaming) and this allows to connect VR devices independently of system of remote devices. In addition, many video-processing algorithms can be used during the transmission.

As the main problem of video-based communication was to get the signal from the graphics card, we tested three solutions how to achieve it. The first approach is naive and its idea is to directly read the content of video-buffer. The second approach is based on OpenGL command transmission and the last approach refines the first one using Pixel Buffer Object extension of OpenGL.

4.1 Direct Real-Time Video Buffer Access

The direct video-buffer access method is presented here by scheme in Figure 8. The figure shows the pipeline consisting of the rendering block, the grabing block and the swap buffer call. After the frame *N* is rendered, the content of actually filled video memory buffer is read and the data are sent to the next processing stage to be transmitted. The grabed data are usually processed by another thread. Then the `glxSwapBuffers()` function is called and frame *N+1* can be rendered.

This method conceptually meets our requirements because we have full control over the image quality and synchronization. On the other hand the method does not work succesfully in a real situation due to low bandwidth between the graphics card and the main memory. We used this method just to test our concept.

4.2 OpenGL Command Transmission

The OpenGL command transmission method is based on interesting idea of sending just only commands normally sent to GPU through OpenGL API. Each OpenGL call is here also sent to the remote system which executes it after necessary modifications. This method does not correspond to our original goal and thus we partially leaved it as a minor task.

4.3 Parallel Video Buffer Access

The parallel video buffer access method is based on the use of OpenGL extension called Pixel Buffer Object (PBO), see Figure 9. Here, after the frame *N-1* is rendered the wrapper of function `glxSwapBuffers()` initiates grabing proces of this frame. This does not block next operations and the original function `glxSwapBuffers()` is called and next frame *N* can be rendered. As the rendering proces is controlled by the GPU and grabing proces by DMA, both can be running simultaneously. The data of frame *N-1* are transfered to the main memory while the frame *N* is rendered to the second buffer.

This method we finally tested as core of our video based communication between two VR systems. The implementation based on the use of PBO extension works as OpenGL wrapper which is applicable on any OpenGL program including applications implemented using the SDL library. The content of the graphics window is sent to the remote system where it can be presented. Such solution has relatively low computational requirements.

The data are transported through the standard RTSP/RTP protocol as an uncompressed video meeting standard RFC 4175 [5]. The implementation is build on

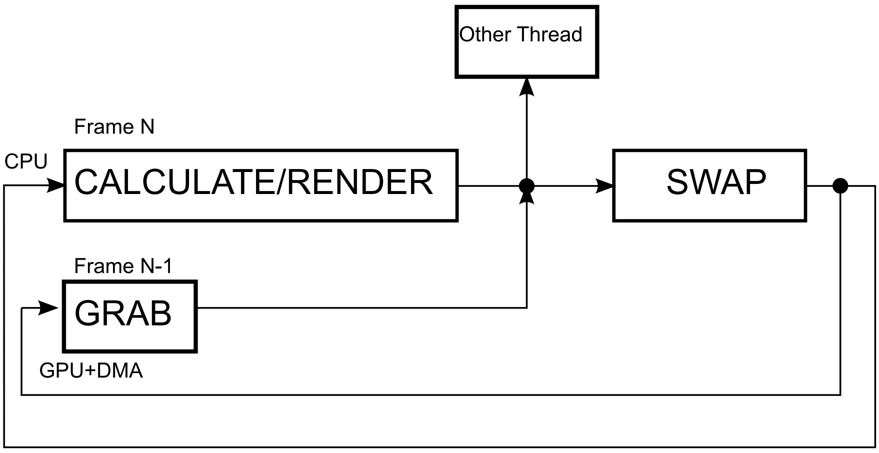

Figure 9. Parallel frame buffer access method

libraries *live555*[1] and *libavutil* (part of FFmpeg[2] project) under the Linux operating system. The method has been tested in an OpenGL application running on local system with four different resolutions simulating the different data bit-rate requirements. The real-time computation produced sequence of images which has been sent to the network using the RTP protocol.

The configuration of the laboratory experiment is illustrated by the scheme in Figure 10. One computer node of our CAVE cluster has been used as a source of testing video signal. The rendered data captured from the graphics card have been scaled using *libswscale* library, which is also part of FFmpeg[3] project, and encoded by our proprietary application implemented along the standard RFC 4175 [5]. Next, the resulting data have been sent to a client computer as a video stream (using *live555* library) via a dedicated 1Gb network connection containing one 1Gb switch. Finaly, the video stream received on the client side has been processed in reverse order to give the final video presentation.

The complete transmission chain with basic functionality of video capturing wrapper, transmission mechanism using standard protocols and simple presentation component gave us a proof of concept which is to be developed during next project phases.

5 Features and Results

The method of parallel video buffer access has been tested on various data bandwidth demands. The measured results are presented in Table 1.

The testing video signal was transmitted in the RGBA format (32bpp) and with framerate 60Hz which corresponds to two streams carrying two video channels each with framerate 25-30Hz.

Evaluating the presented numbers we can summarize them so that resolutions 800x600 and higher, which are typical for VR systems used in our equipment, gen-

[1] `http://www.live555.com/`

[2] `http://www.ffmpeg.org/`

[3] `http://www.ffmpeg.org/`

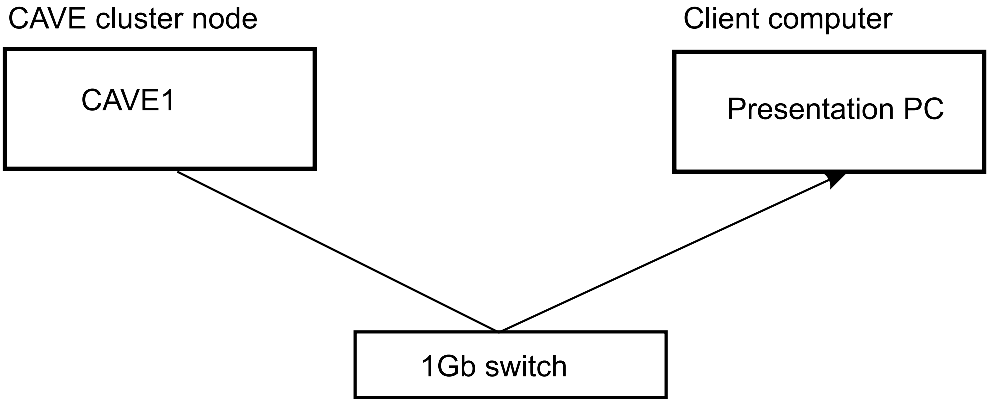

Figure 10. Experiment configuration

Table 1. Bandwidth demands for varying resolutions

Resolution **[Mpix]**	**Theoretical BW** **[Mbps]**	**Real BW** **[Mbps]**	**Note**
0.01 (100×100)	18.3	21	lossless transmission
0.09 (300×300)	164.8	174	lossless transmission
0.3 (640×480)	562.5	590	small data loss
0.48 (800×600)	879.9	810	significant data loss

erate bit-rates over 1Gbps per one video-channel. The case of the 6 video channels, which corresponds to 3-wall CAVE system, represents bit-rates requiring connection with 10Gbps capacity.

The results also show that our parallel video buffer access method is able to produce data rate which is enough for the further scheduled project experiments. The method also offers several interesting features.

Using the video stream based communication between VE devices we can remotely present the VE to a distant user which is not expected to use the same device as the local one. In addition, the stream can be presented on more than one remote device using a multicast communication.

This idea extends our case studies to situations when interactions between two virtual systems can be observed by more users using standard computer equipment.

6 The Future Work

The near future of the project will be dedicated to activities leading not only to stereo video transmission but also exchanges of other types of data including vector descriptions of 3D objects and tracking. The tracked data are typically used when interacting with objects inside the virtual scene. Exchanging video data mixed with other typed of data we can extend the form of distributed VE to more realistic collaborative tool profiting from wide bandwidth networks.

7 Conclusion

Based on the proposed goals we can summarize the achieved results as the first successful stage where the core technique has been implemented and evaluated. This means that we have a basic tool which can be now incorporated into a set of applications corresponding to our case studies presented above. The technique implemented and presented here does not represent any new technology per se, but it represents a new approach in the context of contemporary technologies and the way of their application.

References

[1] BILLINGHURST, M.; KATO, H. Collaborative mixed reality. In *Proc. of International Symposium on Mixed Reality* (Yokohama, Japan), March 1999, p. 261–284.

[2] CRUZ-NEIRA, C.; LEIGH, J.; BARNES, C.; COHEN, S. M.; DAS, S.; ENGELMANN, R.; HUDSON, R.; PAPKA, M.; ROY, T.; SIEGEL, L.; VASILAKIS, C.; DEFANTI, T.; SANDIN, D. J. Scientists in Wonderland: A Report on Visualization Applications in the CAVE Virtual Reality Environment. In *IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, October 1993, p. 59–66.

[3] CRUZ-NEIRA, C.; SANDIN, D. J.; DEFANTI, T. A. Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. In *Computer Graphics (Proceedings of SIGGRAPH '93)*, August 1993. ACM SIGGRAPH, p. 135–142.

[4] CRUZ-NEIRA, C.; SANDIN, D. J.; DEFANTI, T. A.; KENYON, R. V.; HART, J. C. The CAVE: Audio Visual Experience Automatic Virtual Environment. *Communications of the ACM* vol. 35, number 6, 1992, p. 65–72.

[5] GHARAI, L.; PERKINS, C. *RTP Payload Format for Uncompressed Video.* RFC 4175[4], IETF, September 2005.

[6] LEHNER, V. D.; DEFANTI, T. A. Distributed Virtual Reality: Supporting Remote Collaboration in Vehicle Design. *IEEE Computer Graphics and Applications*, vol. 17, number 2, 1997.

[7] MÁŠA, M.; ŽÁRA, J. Interest Management for Collaborative Environments Through Dividing Their Shared State. In *Cooperative Design, Visualization, and Engineering* Berlin: Springer, LNCS 3190, 2004, p. 156–163. ISBN 978-3-540-23149-3.

[4] `http://tools.ietf.org/html/rfc4175`