Virtual Environments '98: Proceedings of the Eurographics Workshop,
Springer Computer Science Series, Springer-Verlag, Berlin, Germany, 1998, pp. 151-161.

Sharing virtual environments over a transatlantic ATM network in support of distant collaboration in vehicle design

Volodymyr Kindratenko[1] and Berthold Kirsch[2]
[1] National Center for Supercomputing Applications (NCSA), University of Illinois.
[2] GMD - German National Research Center for Information Technology, Sankt Augustin.
E-mails: kindr@ncsa.uiuc.edu, kirsch@gmd.de

Abstract

Research in Virtual Reality no longer is focused only on computer graphics, it has become an interdisciplinary, involving new fields such as teleconferencing, networking, and distributed computing. This article presents results of a research project in the Distributed Virtual Reality. The goal of the projects was to evaluate the capabilities, practicality, performance, and cost of this technology for performing collaborative product design review on an industrial showcase application. In the course of the project a dedicated network between two virtual reality visualization systems was established across the North Atlantic. Tele-conferencing applications were integrated with a virtual reality system allowing users from geographically remote locations to see and talk to each other in the shared virtual environment while performing collaborative product design review.

Keywords: distributed virtual reality, teleconferencing, asynchronous transfer mode, IP multicast, collaborative product design review.

1 Introduction

As virtual environments become a part of industrial research programs, their applicability to machine design becomes increasingly apparent. Engineering is an iterative process whereby designs are created, analyzed, tested and modified until the very best design emerges. This process is inherently time consuming and expensive. Virtual prototyping is a method of decreasing the amount of time between the design phase and the time of introduction of a new product into the marketplace, allowing simultaneously for improved quality. A virtual environment (VE) allows engineers to interact with their designs (i.e. a vehicle model) in three dimensions in real time. Extending the virtual environment to several geographically remote sites, where each site is looking at the same model, allows interactive communication of design information. The challenge is determining how complex virtual environments can be shared between geographically remote sites involving large distances, and how real-time live video and audio can become an integrated part of such environments.

The Distributed Virtual Reality (DVR) project was an experiment to address these issues. The goal of the project was to establish an Asynchronous Transfer Mode (ATM) network between two virtual reality visualization systems across the North Atlantic and evaluate the capabilities, practicality, performance and cost of Distributed Virtual Reality technology for performing collaborative product design review on an industrial showcase application. The project started in September 1995. Functional specification [1] of the DVR system was finalized in December 1995. Development of the technical specification [2] continued through February 1996. The first implementation of the system and tests over local area network at NCSA [3, 4] were finished in September 1996. Since that time several major software components were redesigned [5] and some preliminary tests of the system between NCSA and GMD were performed over the Internet. ATM network between NCSA and GMD was established in the end of August 1997 and 19 transatlantic trials of the system took place in September – December 1997 [6, 7]. Two trials were solely dedicated to the system evaluation by Caterpillar engineers from the US and Belgium. This work describes the final implementation of the DVR system, transatlantic ATM network setup, trials, and their results.

2 Requirements for the DVR system

The system needed to be designed for engineers, located at geographically remote sites, to interact with virtual vehicle models in real time and to communicate design information to each other. Therefore, the following concept, functionality, and performance requirements to the system were defined:

Concept requirements. *Portability* implies that the system can be ported to different virtual reality (VR) platforms. *Scalability* means that an unlimited number of remotely located sites can join the shared VE. *Security* addresses the concern for protection of proprietary information in a DVR application.

Functionality requirements. *Ability to interact* with vehicle models in the shared VE. Each participating site should have a *virtual pointer* enhancing collaboration with other sites by pointing to an object of interest. Although, avatars are frequently used, we chose to investigate the applicability and usability of another type of representation - *live video* coming from a video camera located in front of the participants at each site, and seen in the shared VE. *Natural audio communication* should be a part of the shared environment.

Performance requirements. Delays in object modification, animation or motion data transmittals are unavoidable due to the long distances and heavy computations involved, but they should be minimized. The network performance should be such that the object manipulation looks smooth and natural. Voice transmission delays are expected to be equal to or less than those of conventional teleconferencing. Synchronization of voice and speaker image appearance is an essential feature.

3 Development of the DVR system

The model, selected for implementation, is a real-time vehicle simulation computation linked with a VR rendering process allowing one participant to control and drive a virtual model of a vehicle in a shared virtual environment whereas participants at geographically remote sites evaluate it communicating with each other using video and audio integrated with the shared VE.

3.1 VR visualization hardware

The display of the system at GMD is on the Responsive Workbench^{TM} [8], a tracked, stereo, table-like display (Fig. 1a). The Responsive Workbench operates by projecting a computer-generated stereoscopic image off a mirror and through a table surface. Users observe a 3D image displayed above the tabletop by using stereoscopic shutter glasses. The group leader's head is tracked allowing to change the view angle according to his movement. Other group members observe the scene as the group leader manipulates it by using a fastrak stylus.

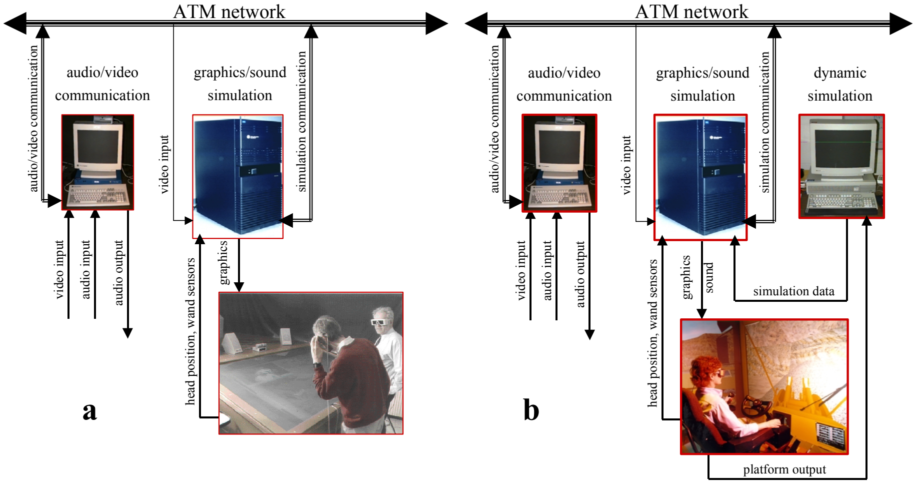

Fig. 1. System setup at **a**) GMD and **b**) NCSA. The display of the system at GMD is on the Responsive Workbench^{TM}; the display of the system at NCSA is in the CAVE^{TM}.

The display of the system at NCSA is in the Cave Automated Virtual Environment (CAVE^{TM}) [9] (Fig. 1b). The CAVE is a projection-based VR system that surrounds the viewer with 4 screens in a 10-foot by 10-foot by 9-foot surround-screen surround-sound cube with three rear-projection screens for walls and a down-projection screen for the floor. A viewer wears stereo shutter glasses and a six-degree-of-freedom head-tracking device, just like in the case of the Responsive Workbench. As the viewer moves inside of the CAVE, the corresponding stereoscopic perspective projections are calculated for each wall. A second sensor and buttons in a wand, held by the viewer, provide interaction with the virtual environment.

At each site SGI Onyxes with Infinite Reality graphics pipelines generate the projected images and SGI Indys with built-in video/audio board take microphone and video camera inputs and give speaker outputs. Wide-angle, low-light cameras and hands-free tie-clip omnidirectional microphones are used.

3.2 Vehicle simulation

The vehicle model implemented is a Caterpillar 990 wheel loader. Virtual prototype of the vehicle was created from the same Pro/Engineering CAD designs used throughout Caterpillar's design process. Caterpillar's proprietary dynamic simulation package Dynasty is used to perform real-time vehicle simulation. A vehicle mock-up platform, installed in the CAVE, is the main user interface in the simulation. It takes real-time operator inputs to the system including steering, throttle, gears, brake, lift, and tilt and communicates it to Dynasty which performs computations and sends results to the VR rendering process to update the position and orientation of the vehicle body in the shared virtual environment.

3.3 VR rendering

VR rendering software is written in C using Sense8's WorldToolKit® (WTK) [10] as the underlying graphics library. The software and all the virtual objects reside locally at each site. During initial set-up time the geometric description of the vehicle and the graphical environment are loaded from files. In a simulation loop several tasks are performed to update the VE. Inputs from the head and wand trackers are detected and used to calculate new head and hand position and orientation. If Dynasty is running, the simulation at that site gets updated vehicle position. Collision detection code is executed in order to insure that the vehicle is always located on top of the virtual terrain in the VE. Networking code is executed to communicate data between other participating sites. Finally, the rendering procedure is called to render the updated scene from the headtracked viewpoint. The simulation runs at a target frame rate of 15 times per second, achieving a relatively high degree of realism.

3.4 Communication with remote sites

The distributed architecture, that has been selected, is peer-to-peer rather than client-to-server. In the implemented model any site can begin first, and as long as there is more than one site participating, a shared environment is possible. Any site can join or leave the shared VE at any time. A new site that joins is immediately synchronized with the other participating sites. Each site maintains its own viewpoint and a virtual pointer. The communication with other sites is built using IP multicast protocol.

Communication with remote sites is a part of the main simulation loop. It is built using WTK network communication API. Network communications in WTK are established using the concept of individual *message items* that are assembled into IP/UDP packets that are than sent to a specific port number on the network. Only simulations that are "listening" to the particular port will be able to receive the packet. During the initial setup time the simulation sends out a packet containing a message having a unique integer identifier (ID) and the name of the computer running the site's simulation. Other sites listen on the same multicast address and port and respond by sending similar packets with their unique IDs and names. When a site leaves, it sends a message communicating this to the other sites. During each simulation loop, just before the actual rendering procedure is called, each site communicates all the pieces of information required to maintain the synchronization of the shared VE. These include the site's viewpoint and virtual pointer coordinates and orientation, and vehicle body coordinates and orientation. Within the time period of the network latency, changes in the vehicle position at one site are visible to all other participating sites. All positions and orientations are absolute, so any lost packets do not destroy the synchronization.

3.5 Integrated video and audio

Video conferencing application VIC [11], developed by the Network Research Group at the Lawrence Berkeley National Laboratory, is used for video communication. Source code of VIC is available making any particular customization possible. Each site runs two copies of the VIC tool, one has been modified the other not. Video input from a camera, located in front of participants, is digitized on the Indy workstation and sent by the unmodified copy of VIC over the network. At the other site the modified version of VIC, running on the same workstation with the VR rendering software, picks up the video from the network. This video is written frame-by-frame from the VIC process, via shared memory, to the graphics rendering process where it is converted to RGB format and placed into texture memory. The texture then is applied to a polygon located at the viewpoint of the corresponding remote site. This results in a dynamic texture displaying the incoming video in the VE (Fig. 2).

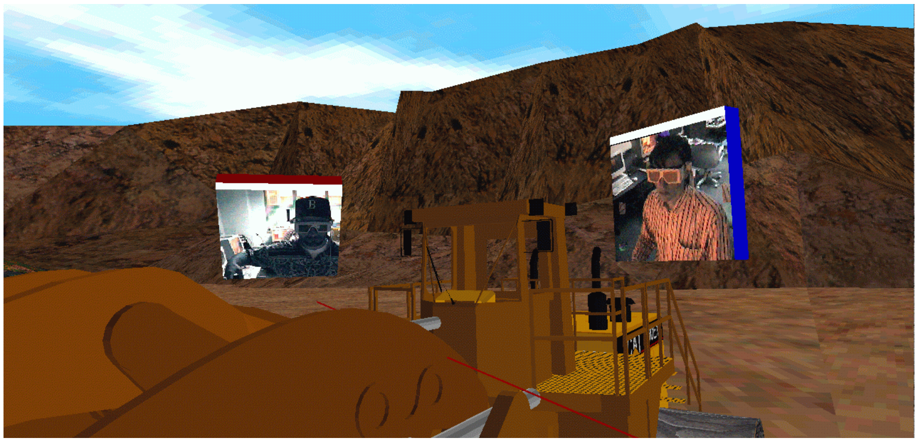

Fig. 2. Two video avatars inside of the shared virtual environment.

The audio conferencing application VAT [12], also available from the Lawrence Berkeley National Laboratory, is used for audio communication. The audio input from a microphone, attached to a participant, is digitized on the Indy workstation and sent by VAT over the network. Both VIC and VAT use Real-time Transport Protocol (RTP) for data communication and they can both send and receive transmissions from any number of remote sites.

4 Transatlantic ATM network

For real-time interaction and digital communication remote sites ideally should be connected by a network possessing the following properties: bandwidth of the order of a few megabits per site, low latency, ability to run IP multicast, and be available and affordable. ATM network is a good choice. With ISDN network we eventually run in some bandwidth-related problems since it is commonly referred to as a technology used to handle connections below 2 Mbps.

Connectivity between NCSA and GMD was established across several ATM-based networks (Fig. 3) including the very-high Backbone Network Service (vBNS), CA*net II, ATM services provided by Teleglobe Canada under the Multimedia Applications on Intercontinental Highways (MAY) project, and Deutsche Telecom's ATM network in Germany. The vBNS and CA*net II interconnect in Chicago at the STAR TAP, an interconnection point for international high-performance research and education networks. To complete the connectivity, a permanent virtual circuit (PVC) was established between vBNS router in Chicago and router at GMD. This PVC traversed CA*net II to Nova Scotia where it was switched to the Teleglobe ATM network to cross the Atlantic (via CANTAT-3) into Sylt, Germany. From Sylt ATM cells were forwarded to GMD in Sankt Augustin via Deutsche Telecom's ATM network. Since all communications in the application are built on top of IP multicast, a multicasting tunnel was installed over the ATM network.

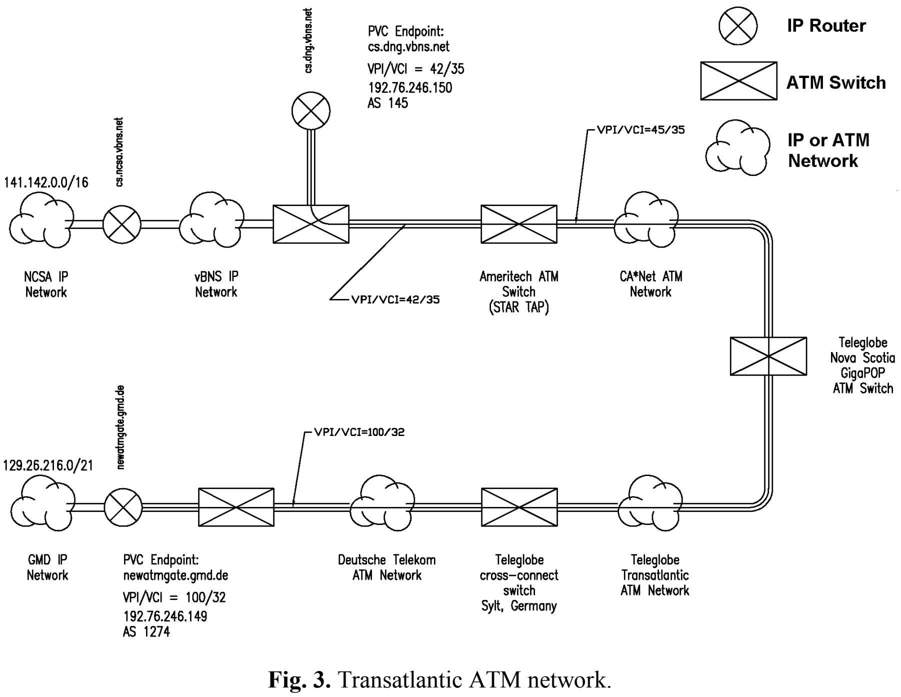

Fig. 3. Transatlantic ATM network.

Bandwidth use per sending site using H.261 encoding technique and sending 30 fps of a good quality low-motion color video is approximately 300 Kbps. For a higher quality or a high-motion video it might require a bandwidth over 500 Kbps. For the same video using NV encoding technique the bandwidth is of the order of 900 Kbps. The frequency used for audio communication is 8 kHz, resulting in a per-site bandwidth of 64 Kbps. Bandwidth, required to exchange position information in the shared VE, is 40 Kbps. In total, a bandwidth up to 1 Mbps per sending site is required. Therefore, the transatlantic ATM circuit was configured for 5000 cells (2 Mbps).

5 Trials

Nineteen transatlantic trials of the DVR system were performed from September to December 1997. Two trials were solely dedicated to the system evaluation by Caterpillar engineers from the factories in Aurora (US) and Gosselies (Belgium). Participants shared a virtual environment and communicated using digitized audio and video. They saw the real-time video of the other participants integrated with the virtual environment. Participants at GMD could see the vehicle move as the engineer drove it at NCSA.

5.1 Measurements

Numerous performance parameters were measured during the trials. *Round-trip time* was measured both by *ping* and *traceroute* utilities. For example, a typical output of *traceroute*, tracing network route from NCSA to GMD, was

```
traceroute to geier-a1.gmd.de (129.26.216.86), 30 hops max, 40 byte packets
1 pigpen.ncsa.uiuc.edu (141.142.223.120) 2 ms 2 ms 2 ms
2 141.142.11.150 (141.142.11.150) 2 ms 3 ms 3 ms
3 cs-atm0-0-12.ncsa.vbns.net (141.142.11.1) 3 ms 4 ms 2 ms
4 cs-atm0-0-6.dng.vbns.net (204.147.129.246) 5 ms 5 ms 5 ms
5 192.76.246.149 (192.76.246.149) 125 ms 125 ms 125 ms
6 geier.gmd.de (129.26.216.86) 125 ms 125 ms 131 ms
```

Results show the round-trip time in the range of 125 to 130 ms.

ATM cells measurements were performed at GMD's ATM switch. Measurements, performed on December 12th, 1997 during the evaluation of the DVR system by engineers from Caterpillar, are shown in Fig. 4. During the test we used NV video encoding sending 8 fps by each site and PCM2 audio encoding. Results show that the actual load of NCSA-GMD ATM network was below 1 Mbps.

Both VIC and VAT tools have built-in measurement capabilities based on analysis of data from Sender Report Packets and Receiver Report Packets defined in RTP. The following parameters were measured during the trials: 1) Amount of data sent/received (Kbps). This parameter is both a quantitative (defines the bandwidth use per sending site) and qualitative (comparison of amount of data sent at one site with amount of data received at another site gives a delivery quality) measure. 2) Number of packets sent/received (packets/sec). This parameter defines the number of RTP packets containing payload data sent/received at one site. 3) Number of missing packets (packets/sec) and cumulative number of packets lost. 4) In video conferencing the number of video frames per second sent/received was counted. 5) In audio conferencing playout time (ms) was measured. This measure corresponds to the time difference between when a sound was digitized at the sending site and played back at the receiving site.

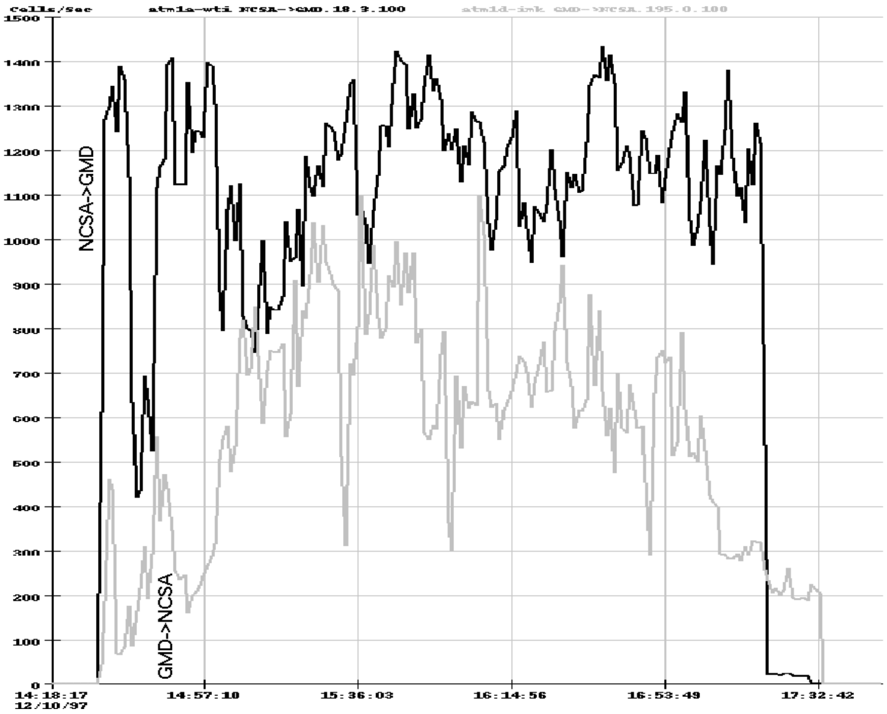

Fig. 4. ATM cells measured on December 12th, 1997 at GMD's ATM switch during one of the transatlantic trials.

The following results were obtained for VAT (Fig. 5). Typical bandwidth per sending site using PCM2 encoding at 8 kHz was 60-70 Kbps. Number of packets per sending site varied from 7 packets/sec for low intensity talk to 16-20 packets/second for high intensity talk. Usually there were no packets missing using unicast, but a considerable amount of packets was missing using multicast. Playout time typically varied between 80 and 150 ms, comparing to the one-way trip time of 62-65 ms.

The following results were obtained for VIC (Fig. 6) using NV encoding. Bandwidth per sending site using NV encoding and sending 8 fps varied between 10 - 20 Kbps for low-motion, almost still video, and up to 400 – 600 Kbps for high-motion video. Typically it was between 150 and 250 Kbps during system evaluation by Caterpillar engineers. Amount of sent/received packets varied proportionally to the bandwidth used and was in the range of 20 to 40 packets/second during system evaluation by Caterpillar engineers. From time to time some packets were missed introducing distortions in received video. We did not succeed to correlate missing packets with any of the set-up parameters that we could control.

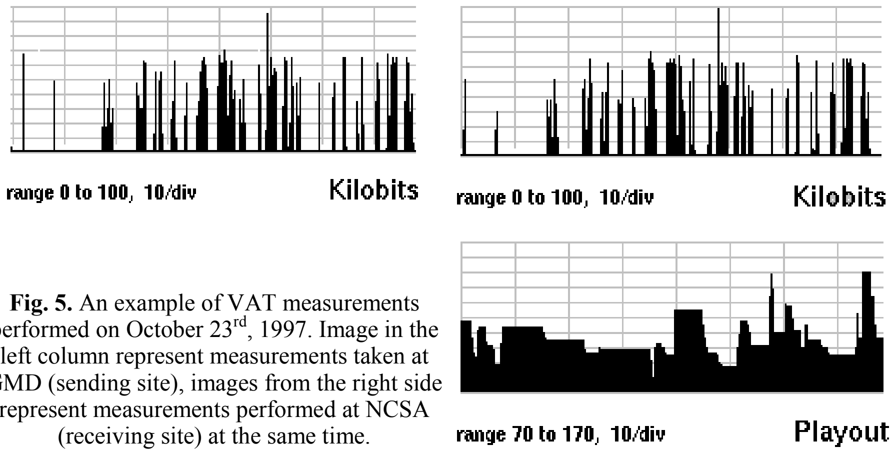

Fig. 5. An example of VAT measurements performed on October 23rd, 1997. Image in the left column represent measurements taken at GMD (sending site), images from the right side represent measurements performed at NCSA (receiving site) at the same time.

5.2 Lessons learned

The major lessons learned involved the use of IP multicast. It was found that multicasting packets were periodically missed resulting in a distortion of data communication. We found it more appropriate to use unicast instead of multicast for the audio transmissions. A problem also was discovered in our DVR software. Time, needed to copy and convert video from VIC process to graphics rendering process, was relatively long resulting in skipping a considerable amount of incoming video frames. There were some difficulties with handling more than 8 fps of video integrated into the VE. We see a solution of this problem in optimization of the video updating procedure.

An interesting synchronization problem due to the network latency was observed for events depending on other events. For example, participants at GMD could attach their viewpoint to the cab of the virtual vehicle driven at NCSA. The position and orientation of their viewpoint in this case dependent on the position of the vehicle calculated at NCSA. The system, running at GMD, updated it each time a new data arrived from NCSA, thus, keeping it a few frames behind the frames currently rendered at NCSA, but in sync with GMD's rendering. In its turn, the new GMD's viewpoint position was immediately communicated back to NCSA and updated at NCSA as soon as it arrived, a few frames, relatively to GMD's simulation, after it was sent. So, the new position of GMD's viewpoint in NCSA's VE was the position expected at the time when NCSA's simulation sent cab's position to GMD, or a few frames ago at NCSA. In practice it resulted in the GMD's viewpoint to be visible behind the cab of the forward driving vehicle and in the front of the cab of the backward driving vehicle as seen at NCSA. We see a solution for this problem in illuminating the need for the second-order event dependencies.

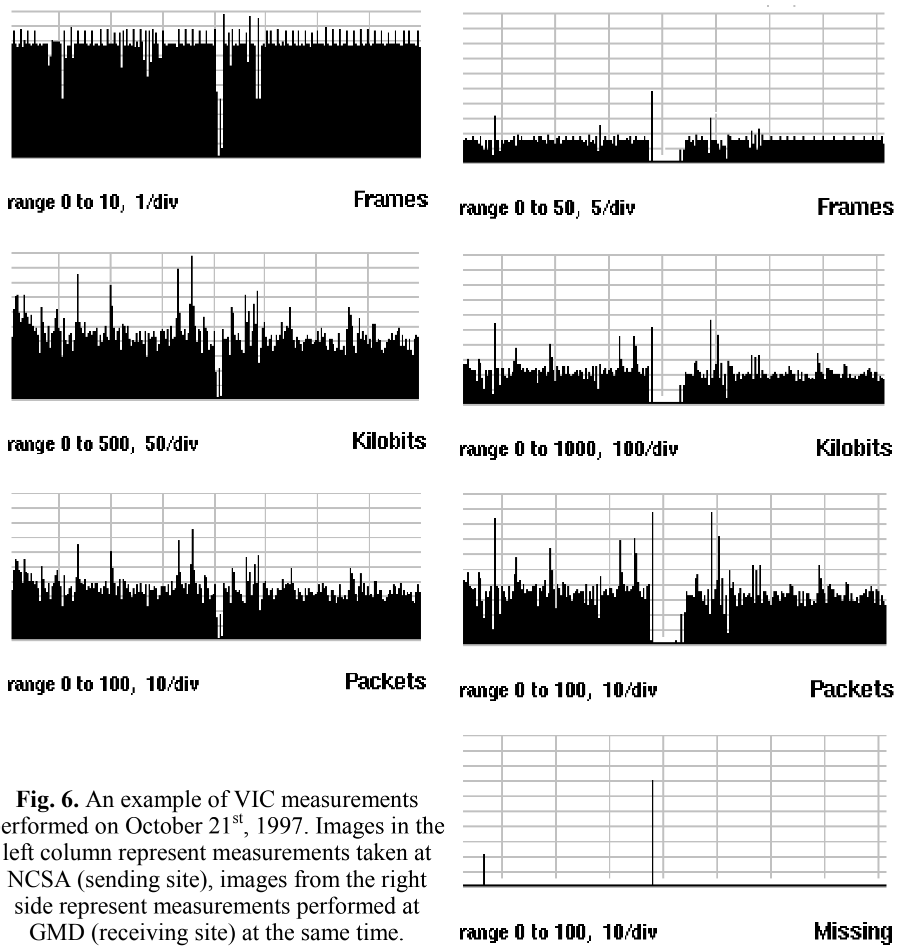

Fig. 6. An example of VIC measurements performed on October 21st, 1997. Images in the left column represent measurements taken at NCSA (sending site), images from the right side represent measurements performed at GMD (receiving site) at the same time.

6 Conclusions and future research

The system was designed for engineers located at geographically remote sites to interact with their models in the three dimensions in real time and to communicate design information to each other. It supports collaborative design review. Integrated real-time video transmissions let engineers see each other in a shared virtual environment. Audio transmissions provide natural voice communication. The current version of the DVR system may be considered as a prototype for future engineering development tools. Transatlantic trials of the system have shown a number of problems pointing network reliability and performance issues as well as some drawbacks in the developed software. Therefore, current work is focused on eliminating found problems and redesigning some components of the system.

7 Acknowledgments

The project is supported by Caterpillar Belgium S.A. and is a part of the Distributed Video Production (DVP) project sponsored by The European Union under the Advanced Communications & Technologies research and technological development program as project AC089. The partners involved into DVR project are GMD - German National Research Center for Information Technology and the National Center for Supercomputing Applications, University of Illinois, in the USA.

We are grateful for everyone who contributed to this project, in particular, Robert Fenwick (Caterpillar, Inc.), Jean-Paul Emond (Caterpillar Belgium S.A.), Dr. Manfred Kaul (GMD), Dr. Wolfgang Heiden (GMD), and Dr. Thomas DeFanty (NCSA). For Lance Arsenault and Jason Wessel (both from NCSA) for their contribution in software development. We are thankful for Paul Zawada, Randy Butler, David Mitchel, and Tony Rimovsky (networking group from NCSA), Wolfgang Ziegler, Lothar Zier, and Ferdinand Hommes (networking group from GMD), Randolph Nicklas (vBNS), Bill Arnaud (CANARIE), Alain Lechasseur (Teleglobe), and Peter Feil (DeTeBerkom) for their contribution in establishment of the transatlantic ATM link.

References

[1] J.-P. Emond, *CEC Deliverable A089.CAT.WLP.DS.L.441.b1: Distributed Virtual Reality - Functional Specification*, December 1995.

[2] V. Lehner, *CEC Deliverable A089.CAT.WLP.DS.L.442.b1: Distributed Virtual Reality - Technical Specification*, March 1996.

[3] V. Lehner, *CEC Deliverable A089.CAT.WLP.DS.I.443.b1: Distributed Virtual Reality – First Implementation*, September 1996.

[4]. V. Lehner, and T. DeFanti, *Distributed virtual reality: supporting remote collaboration in vehicle design*, IEEE Computer Graphics and Applications, 1997, vol. 17, no. 2, pp. 13-17.

[5] V. Kindratenko, and B. Kirsch, *CEC Deliverable A089.CAT.CAT.DS.I.444.b0: Distributed Virtual Reality – Implementation*, March 1998.

[6] V. Kindratenko, and B. Kirsch, *CEC Deliverable A089.CAT.CAT.DS.P.445.b0: Distributed Virtual Reality – Evaluation Report*, March 1998.

[7] V. Kindratenko, and B. Kirsch, *CEC Deliverable A089.CAT.CAT.DS.P.54.b0: Distributed Virtual Reality – Evaluation Report*, March 1998.

[8] W. Krueger and B. Froehlich, *The Responsive Workbench: A Virtual Work Environment*, IEEE Computer Graphics and Applications, 1994, vol. 14, no. 3, pp. 12-15.

[9] C. Cruz-Neira, D. Sandin, and T. DeFanty, *Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE*, in Proc. Siggraph 93, ACM Press, New York, pp. 135-142.

[10] *WorldToolKit®, the Industry's Chosen 3D Real-Time Software ToolKit*, SENSE8 Corporation, Mill Valley, CA. http://www.sense8.com/products/worldtoolkit.html.

[11] S. McCanne and V. Jacobson, *vic: A Flexible Framework for Packet Video*, in Proc. ACM Multimedia 95, ACM Press, New York, pp. 511-522.

[12] V. Jacobson and S. McCanne, *vat - LBNL Audio Conferencing Tool*, Lawrence Berkeley National Laboratory, Berkeley, CA. http://www-nrg.ee.lbl.gov/vat/.