

Overcoming Time-Zone Differences and Time Management Problems with Tele-Immersion

Tomoko IMAI <timai@mlab.t.u-tokyo.ac.jp>
University of Tokyo
Japan

Zhongwei QIU <zhwqiu@evl.uic.edu>
Sowmitri BEHARA <sowmitri@evl.uic.edu>
University of Illinois at Chicago
USA

Susumu TACHI <tachi@star.t.u-tokyo.ac.jp>
Tomonori AOYAMA <aoyama@mlab.t.u-tokyo.ac.jp>
University of Tokyo
Japan

Andrew JOHNSON <aej@evl.uic.edu>
Jason LEIGH <spiff@evl.uic.edu>
University of Illinois at Chicago
USA

Abstract

N*Vector (Networked Virtual Environment Collaboration Trans-Oceanic Research) is a project to develop innovative networked virtual reality technologies to support transoceanic collaboration [1]. The project is currently being conducted between Japan and the United States, but the results can be generalized to other long-distance collaborative situations. In this paper we present one of the goals of N*VECTOR, to develop tools to bridge time zone differences.

Contents

- [1. Introduction](#)
- [2. Tele-Immersion](#)
- [3. Asynchronous collaboration](#)
- [4. VR-mail system](#)
- [5. VR-annotator](#)
- [6. VR video cassette recorder \(VCR\)](#)
- [7. Conclusions and future work](#)
- [References](#)

1. Introduction

When time zone differences are large, participants have to work asynchronously. Asynchronous work also occurs when we have to share expensive or scarce computing or scientific resources, or if the collaboration involves individuals who are too busy to schedule face-to-face meetings. To overcome this situation, we often exchange e-mail and fax or leave messages on physical Post-it notes. However, it is often difficult to explain visual content using text even if additional illustrations and perhaps audio explanations are attached to a message. This is because these media often come in a separate, un-unified form. The viewer of the message must correlate the text with the picture and the audio explanation in order to form a full understanding of the message.

This problem is even more challenging in collaborative virtual reality (VR) because VR is not particularly well suited for the display of text, which is the basis of most traditional asynchronous collaboration tools. Our approach leverages the unique capabilities of VR to allow users to naturally annotate visual information by directly pointing at objects and speaking about them. These annotations are then transformed into either virtual Post-it notes that are left in the environment or e-mailed to collaborators. When the recipient of the message plays back the message, an avatar (a virtual representation of the sender of the message) will appear to reenact the message. This is similar in concept to the playback of a video recording; but unlike a video, which always depicts the recorded event from one point of view, a VR recording can be observed from any point of view.

This paper will describe three approaches that we are currently exploring to support annotations in VR for asynchronous collaboration. These include (1) *VR-annotator* -- an annotation tool that allows collaborators to attach 3D VR recordings to objects in a VR scene; (2) *VR-mail* -- an e-mail system built to work entirely in VR; and (3) *VR-ucr* -- a streaming recorder to record all transactions that occur in a collaborative VR session so that a full recreation of a past VR event is possible.

2. Tele-Immersion

Tele-Immersion is the union of networked VR and video in the context of significant computing and data mining, and supports collaborative VR in design, training, education, scientific visualization, and computational steering. The ultimate goal of tele-immersion is not to reproduce a real face-to-face meeting in every detail, but to provide the "next generation" interface for collaborators, worldwide, to work together in a virtual environment (VE) that is seamlessly enhanced by computation and large databases. A typical tele-immersive space will be a persistent VE maintained by a computer simulation that is constantly left running. The space exists and evolves over time. It may be the evolving design of a car, or the evolving simulation of climatological data. Users enter the space to check on the state of the simulated world, discuss the current situation with other collaborators in the space, make adjustments to the simulation, or leave messages for collaborators who are currently asleep on the far side of the planet. When participants are tele-immersed, they are able to see and interact with each other and the objects in a shared VE and can talk to each other via ambient or personal microphones. The focus of tele-immersion is supporting high-quality interaction between small groups of tele-immersed participants.

Presence in the virtual world is typically maintained using an avatar, or a

computer-generated representation of a person. These avatars may be as simple as a pointer, but having physical body representations can be very helpful in aiding conversation and understanding in the virtual space because you can see where your collaborators are, and at what they are looking or pointing. Tracking the user's head and hand position and orientation allows articulated avatars to transmit a decent amount of body language, and are very useful in task-oriented situations. Seeing high-quality live video of a person's face can improve negotiation. Video avatars -- full-motion, full-body video of a user -- allow very realistic-looking collaborators in the space, improving recognition. But they require higher network bandwidth and high-quality cameras and low-light CCD cameras.

Tele-immersion will not replace e-mail, phone calls, or existing teleconferencing systems. They each have their strengths and uses. Just as word processing documents, spreadsheets, and white boards are shared across the Internet to put discussions into their appropriate context, sharing a virtual space with your collaborators as well as the 3D design being considered or the simulation being visualized, puts these discussions into their appropriate context. Tele-immersive environments are not turn-key systems yet and require a fair amount of infrastructure to be established and maintained. Scientists and engineers are interested in using tools that help them do their work more conveniently and efficiently. Current tele-immersive work is priming the pump with collaborative projects with interested domain scientists to create these tools, deploy them, evaluate them, and then generalize their effectiveness.

3. Asynchronous collaboration

In transoceanic tele-immersion, different time/different place collaboration is the most attractive. Because of time zone issues, it may be inconvenient to schedule synchronous meetings, so asynchronous work may be the most appropriate method. Asynchronous work also has the advantage that geographically distributed teams can work on the problem around the clock, by passing the work off at the end of the day to another team who is just arriving at work in their morning. E-mail is a very successful tool supporting asynchronous work. However, in international collaborations there is typically a one-day turnaround time to get responses, so collaborators can easily waste days clarifying the work to be done and making instructions clear. When working in a VE, this is even more difficult because it is hard to use text, speech, or even 2D images alone to describe work to be done, or discoveries that have been made in a dynamic 3D environment. It is important that messages among the distributed team members be clear, to reduce misunderstandings. In a VE it is important to be able to put these messages into their appropriate context -- the context of the virtual world itself.

One of the advantages of doing design or scientific visualization in an immersive environment is the ability to have geographically distributed participants sharing space with each other and the objects under discussion. This allows the participants to point at specific objects in the scene or set the parameters of the simulation to specific values to clarify what they are saying. It gives the users a common context for their discussions. Especially in international collaborations where the language barrier can be a large hurdle, being able to gesture relative to the environment (pointing at the red box, turning your head to look at the green sphere) helps to clarify the discussion. In asynchronous collaboration, the ability to hand off work quickly and accurately is of great importance. A user stepping into the VE needs to know what work has been done since

he or she was last there, and what new work may need to be done. What is the best way to transmit that information?

Lessons can be learned from the sharing of more traditional text and image information on remote computers, through such tools as Lotus Notes, Microsoft Net-meeting, Habanero, and MUDs / MOOs, and from the sharing of graphical, audio, and video data through media spaces, and also through previous work in VEs. Cspray can save both the collaboration state and a trace of a session; TeleInViVo can record and play back rendered pictures without audio. Verlinden et al. allowed a user in a VE to record a voice message and attach it to an annotation marker. Vanno extended this capability by allowing a user to record and replay voice annotations attached to places, objects, or times in the VE. Crumbs allows users to annotate parts of rendered volumes. Immersive tutoring systems such as Steve can monitor user's behavior and replay prerecorded speech and gestures to help in training situations. The virtual director allows the creation of 2D animated movies by pointing a virtual camera from within an immersive VE. Although these tools all support various aspects of collaborative work, they tend to be very application specific, with few studies performed on their effects.

4. VR-mail system

As a first prototype VR recording/playback system, we developed the Virtual Reality Mail System (VR-mail)[7]. In VR-mail, users make a recording by speaking and gesturing. The audio and gestures are captured and saved in a format that allows a synchronized playback at a later time. This recording can then be sent to another user in the VE. When the recipient of the message enters the VE, he or she will find a VR-mail message waiting for him or her. The recipient may then play back the message. As in a traditional e-mail system, the recipient is then able to respond to the original sender of the VR-mail.

VR-mail messages and the general state of the VE are maintained by a central server. The VR-mail system is designed to be used in the CAVE (figure 1) and ImmersaDesk (figure 2) VEs[10]. The CAVE VR system is a 10 foot-cubed room that is projected with stereoscopic images, creating the illusion that objects appear to coexist around the user in the room. A user dons a pair of lightweight liquid crystal shutter glasses to resolve the stereoscopic imagery, and holds a three-button wand for three-dimensional interaction with the VE. An electromagnetic tracking system attached to the glasses and the wand allows the CAVE to determine the location and orientation of the user's head and hand at any given moment in time. This information is used to instruct the Silicon Graphics Onyx that drives the CAVE, to render the images from the point of view of the viewer.

The ImmersaDesk and its successor, the ImmersaDesk2, are smaller, drafting-table-like systems also capable of projecting stereoscopic images. Applications built for the CAVE are fully compatible with the ImmersaDesk. Whereas the CAVE is well suited for providing panoramic views of a scene (particularly useful for visualizing architectural walkthroughs), the ImmersaDesk is designed for displaying images that fit on a desktop (for example, CAD models).



Figure 1. Image of a person working in the CAVE

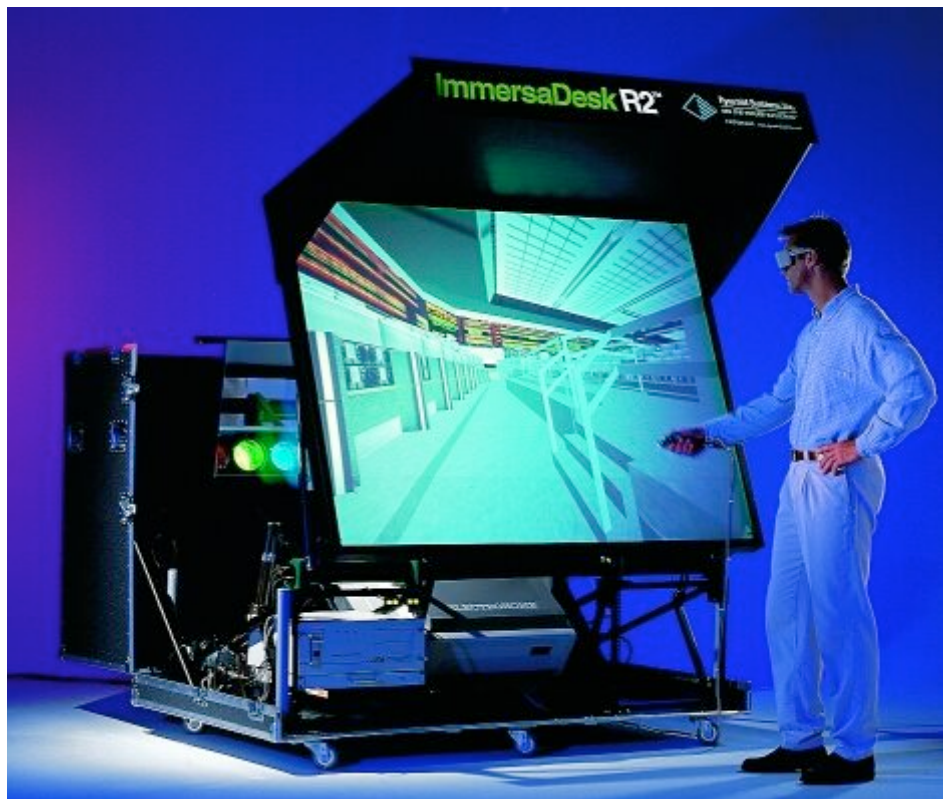


Figure 2. The ImmersaDesk 2

VR-mail is built on top of the CAVE library, a high-level graphics library called XP (eXtended Performer), and the CAVERNsoft networking library [5,6].

VR-mail's user interface is embodied in a virtual friend or pet that follows the participant as he or she interacts with the VE. Touching the pet pops up an interface for

recording, reviewing, and deleting messages as shown in figure 3. This pop-up menu may either follow the user as with the virtual pet, or it may be set to a stationary location in the environment. Allowing the pet to constantly follow a participant has the advantage that the interface for recording and stopping the recording can be reached anywhere in the VE. However, in some instances the pet may actually occlude part of the VE, and hence the option is provided to leave the pet behind. At a touch of a button the pet can be instantly recalled to the participant's side.

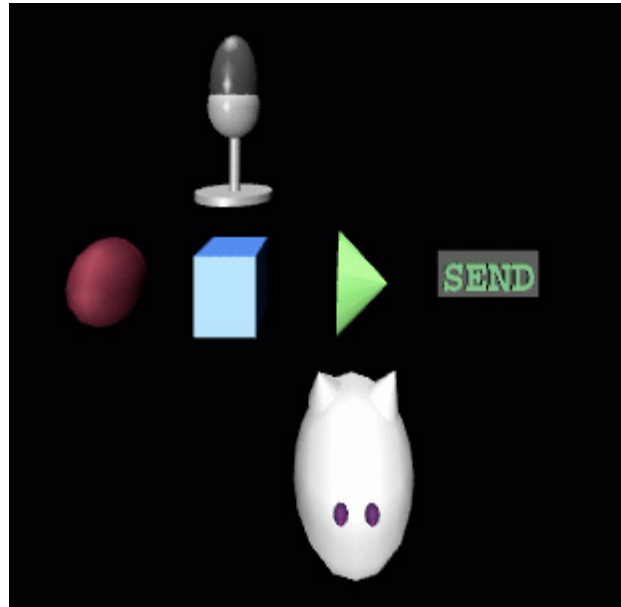


Figure 3. The 3D icons. A can at the bottom is a virtual friend, and other icons are controlling buttons for audio recording and playback.

To record a message, a user begins by selecting the virtual pet. In response, a microphone, mailbox, and DELETE icon appear. If the user selects the microphone, the recording, stop, and playback icons appear. The user's voice is picked up by a wireless microphone worn by the user and digitized by the Onyx2. A sound server stores the digitized data in an audio file. At the same time, tracked head and wand data, consisting of position and orientation information, coupled with time stamp data, are stored in a gesture file. This time stamp information is used during playback to ensure that the audio is synchronized with the reanimation of the gestures.

The recording of both voice and gesture stops when the user selects the STOP button. Selecting the SEND button pops up a list of the virtual pets belonging to the other participants. VR-mail can be sent to the owner of any of the pets by selecting the pet under the SEND icon. The sent VR-mail is stored at the central server to await download from its intended recipient.

When we conducted a user test with this VR-mail system, participants seemed to be able to effectively use voice and gesture to asynchronously communicate ideas to one another.

5. VR-annotator

Note-taking serves to extend human memory [8]. Researchers take notes so they can review their experiments later. Students take notes to memorize new materials taught in lectures. The notes are also used to leave messages for friends and colleagues. They can be written on notebooks, on adhesive pieces of paper or Post-its, or in textbooks as marginal

notes. Voice messages can also be left on telephone answering machines. In a three-dimensional interactive virtual world, it is necessary to take notes in a three-dimensional and interactive format. Within VR-annotator, the messages can be attached to virtual objects much like how we attach Post-it notes to objects in the real world. Hence, when the object is picked up and moved, the annotation remains attached to the object. The replay of the message will occur relative to the object.

In the VR-annotator system, each client has an annotation controller consisting of a network component and a recording engine component, as shown in figure 4. The recording engine is the same as that in the VR-mail system. The annotation controller maintains a list of annotations and allows the user to play, record, and stop an annotation. The functions can be accessed through a 3D icon shown in figure 5. A user sends a request to play or record a specific annotation to the annotation controller. The annotation controller asks the recording engine to perform the action. The engine records the user's gesture and voice into separate files with headers that include the author's name, the annotation's name, and the location of the annotation in the VE. Finally, the network component will send all the updated annotations to the server. The server stores all the annotations and distributes them to the clients as they are accessed.

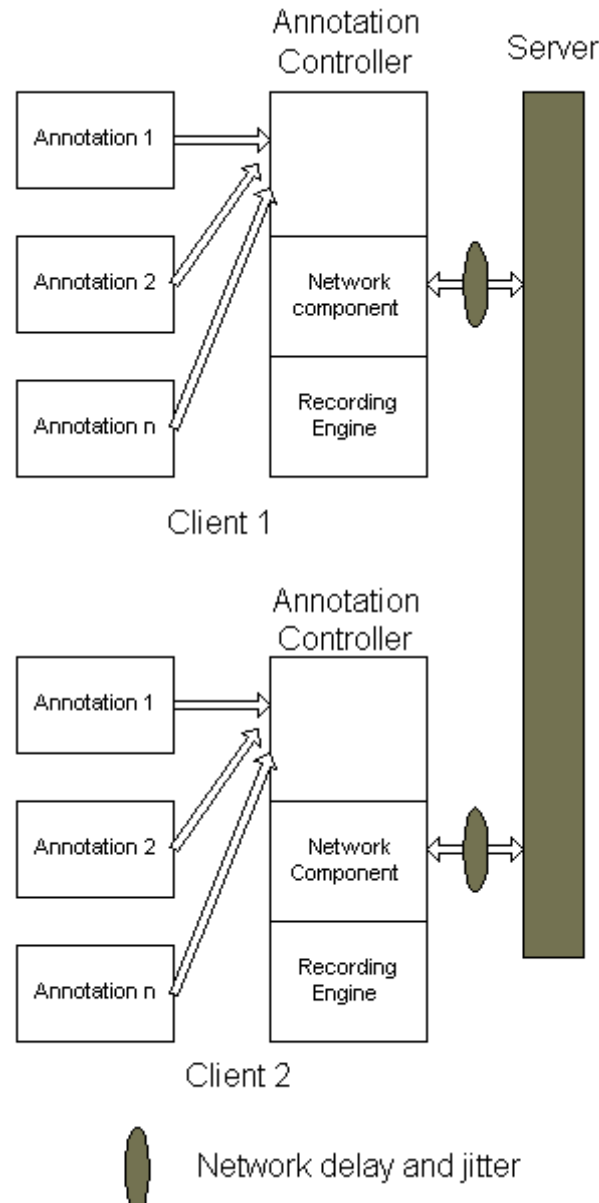


Figure 4. System diagram for VR-annotator



Figure 5. A flag at the middle shows an annotated object. An avatar represents a collaborator in the VE.

Our current application uses a flag (figure 5) to show the location of the annotation. A user can listen to the message by placing his or her virtual hand on the flag, and pressing a button on the wand, which pops up the 3D icon shown in figure 6.

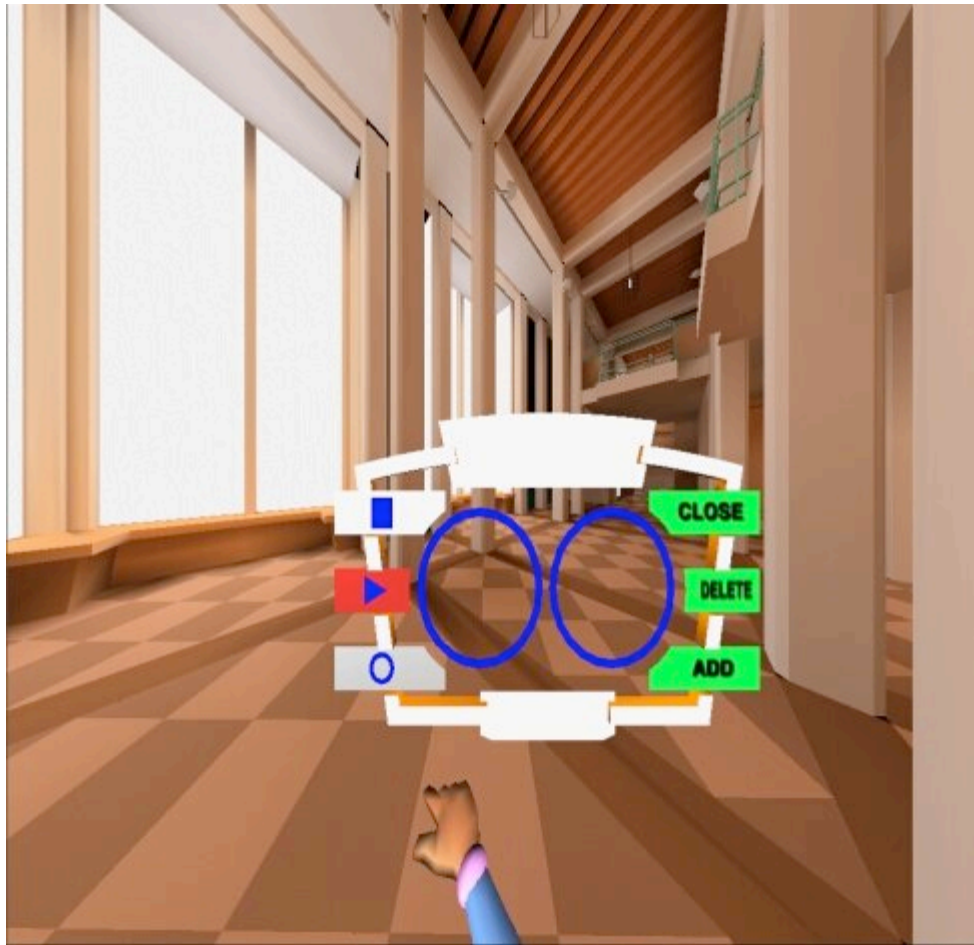


Figure 6. 3D icon menu to control add, delete, stop, play, and record annotation.

6. VR video cassette recorder (VCR)

While VR-mail and VR-annotator were primarily designed to annotate static objects, VR-vcr was designed to record dynamic events in the VE. VR-vcr is able to capture local events in a VE, or global events that are been generated by other remote participants. These global events are typically streamed to all the participating clients so that each of their VEs are properly synchronized. It is this event stream that VR-vcr records. Once the recording is made, it can be played back; if other participants are in the VE at the same time, all of them are able to view the playback together as the same event stream is delivered to each of them.

Figure 7 shows the VR-vcr clients and central server. VR-vcr is designed so that a dedicated server may be used for performing the recording, or each client can perform its own recordings. In either case, the recorder receives incoming data through the various network connections and decides whether to record the data or not based on the information given by the track manager. Each stream is recorded on a separate "track." Each track is stored in a separate data file. This allows tracks to be viewed together or independently. The track manager maintains a list of tracks that will be passed to the main recorder for recording. The tracks can be added or removed at runtime through a separate interface application that maintains information on which tracks are in use. Each track is identified by a unique name and ID, which is given by CAVERNsoft.

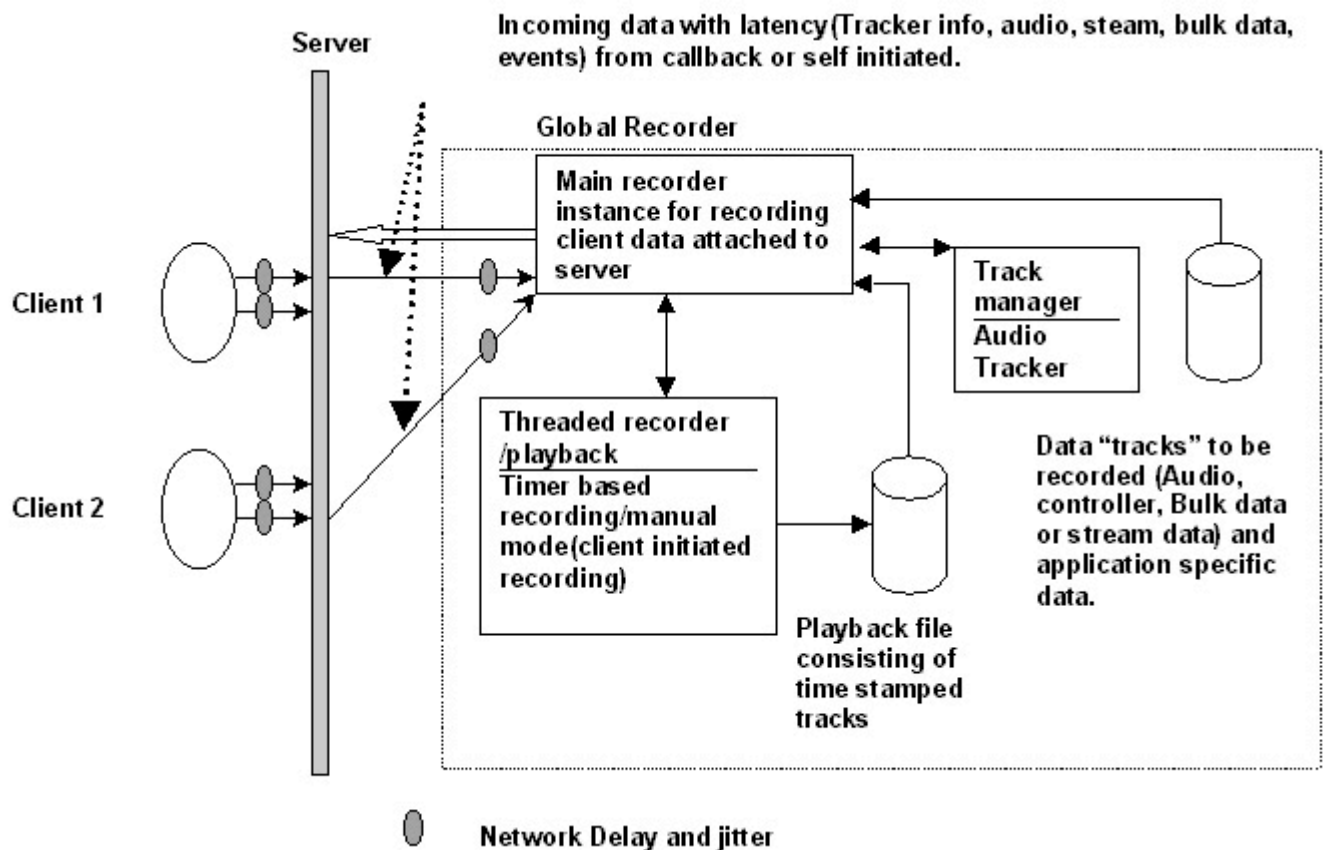


Figure 7. VR-recr created on a separate recording server. The main recorder controls the recording session, the threaded recorder/playback module records different kinds of streams into separate files, and the track manager polls tracks to be recorded.

7. Conclusions and future work

In each of the aforementioned recording systems a separate recording and playback engine was created. It became clear from the exercise that a common engine could be designed to serve the needs of all three systems. In order to achieve this, the engine would have to be decoupled from any graphical capabilities. The user-interface, networking, and associated graphics rendering would be built separately as needed by the recording application. The engine's only task is to capture events on multiple data streams, store them on disk, and allow playback and querying of the data. This is a nontrivial task because, not only must the engine be able to store and retrieve these data streams at a rapid rate, they must be able to fast-forward and rewind through the stream -- hence, requiring that all the accumulated events be carried forward or rolled back. To address such an issue one either has to store the state of all events all the time or create intermittent snapshots. The latter is a common technique used in other streaming media. The challenge in VR is that the recorded data is not limited to simply audio or video; it is composed of a diverse collection of data streams, including state updates of 3D models and 3D tracker information for each of the participants in the space.

Hence, research is progressing along the lines of developing the unified VR recording server (called serVR) as well as the continued improvement of the client front-ends. In the latter case it would be interesting to be able to record a number of annotations and stitch them together intelligently to create a virtual tour guide or personality that is able

to respond autonomously to the user. Such a guide could have powerful applications for training in VR.

References

1. Networked Virtual Environment Collaboration Trans-Oceanic Research Home page, <http://www-grd.rdh.ecl.ntt.co.jp/GEMnet/e/4/e41.html>
 2. Jason Leigh, Andrew Johnson, Thomas A. DeFanti and Maxine Brown. A Review of Tele-Immersive Applications in the CAVE Research Network. In Proc. *IEEE Virtual Reality Conference*, pp. 180-187, March 1999.
 3. Felicio Vanderlei Deriggi Jr. Mario Massakuni Kubo, Antonio Carlos Sementille, Jose Remo Ferreira Brega, Simone Garcon dos Santos and Claudio Kirner. CORBA Platform as Support for Distributed Virtual Environments. In Proc. *IEEE Virtual Reality Conference*, pages 8-13, March 1999.
 4. Reid Harmon, Walter Patterson, William Ribarsky and Jay Bolter, "The virtual annotation system", In proc. *IEEE VRAIS'96 Conference*, pages 239-245, March 1996.
 5. David E. Pape, Tomoko Imai, Josephine Anstey and Maria Roussos and Tom DeFanti, "XP: An Authoring System for Immersive Art Exhibitions", In proc. *Virtual Systems and MultiMedia*, pages 528-533, November 1998.
 6. Jason Leigh, CAVERNsoft home page, <http://www.evl.uic.edu/cavern/>
 7. Tomko Imai, Andrew E. Johnson, Jason Leigh, David E. Pape, Thomas A. DeFanti, "The Virtual Mail System", In proc. *IEEE Virtual Reality Conference*, page 78, March 1999.
 8. Vannevar Bush, "As We May Think", *The Atlantic Monthly*, pages 101-108, Vol. 176, No. 1, July 1945.
 9. Resource Description Framework. Working Group Webpage, April 1999, <http://www.w3.org.RDF/>.
 10. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R.V., and Hart, J.C., The CAVE: Audio Visual Experience Automatic Virtual Environment, *Communications of the ACM*, Vol. 35, No. 6, pages 65-71, June 1992.
 11. CABIN home page, <http://www.ihl.t.u-tokyo.ac.jp/Projects/CABIN>
-