PUSHMEPULLYOU: The REALITY of INTERACTION with SHARED OBJECTS in NETWORKED WALK-IN DISPLAYS

David Roberts
Centre for Virtual Environments
University of Salford
United Kingdom
d.j.roberts@salford.ac.uk

Oliver Otto
Centre for Virtual Environments
University of Salford
United Kingdom
o.otto@salford.ac.uk

Robin Wolff
Centre for Virtual Environments
University of Salford
United Kingdom
r.wolff@salford.ac.uk

Abstract

This paper investigates the reality of shared manipulation of objects between users in distributed walk-in displays and presents solutions to address the effects of constraints of network technology. Various forms of shared interaction are examined through a single structured task of building a Gazebo. Communication of tracking data can saturate the network and result in delay or loss of messages vital to the shared manipulation of an object. We report on extensive trails between three walk-in displays in the UK and Austria, linked over the Internet using a Collaborative Virtual Environment (CVE) and demonstrate such effects on a naive implementation of the Gazebo building task. We then present and evaluate application-level workarounds and conclude by suggesting solutions that may be implemented within next-generation CVE infrastructures.

Keywords

shared object manipulation, communication, IPT, immersive, CVE

1 INTRODUCTION

Advances in immersive display devices are ensuring their acceptance in industry as well as research. Within an immersive device, you can walk around an object, move your body and head to examine it from every angle and manipulate it with your hand. Walk-in displays increase this naturalness by allowing you to see your own body within the environment. They allow a user to see his body within the spatial context of the environment. A virtual object can actually be walked up to, around, reached for and manipulated in a highly intuitive way.

Many team related tasks in the real world centre around the shared manipulation of objects. A group of geographically remote users can be brought into social proximity to interactively share virtual objects within a Collaborative Virtual Environment (CVE). CVEs are extensively used to support applications as diverse as industrial design review, medical simulations, military training, online games, and social meeting places. A user's real body may be situated within a group of remote users congregated around a shared object by linking walk-in immersive displays through a CVE infrastructure. This allows each team member to use their body within the space to interact with others and virtual objects. The spoken word is supplemented by non-verbal communication in the form of pointing to, manipulating and interacting with the object as well as turning to people, gesturing and other forms of body language. This offers unprecedented naturalness of interaction and remote collaboration. The constraints of network technology, however, make the experience of shared interaction both disappointing and frustrating.

This paper investigates the reality of shared interaction of common objects between users in distributed walk-in displays and presents solutions to address the effects of the network. Various forms of shared interaction are examined through a single structured task of building a Gazebo. We report on a number of trails between three walk-in displays in the UK and Austria. These were linked over the Internet using the DIVE-Spelunk [1] immersive CVE.

The actions of a remote user must be reflected through an avatar in the local display. Driving an avatar from motion tracking of a user considerably improves non-verbal communication. Tracking data can, however, saturate the network resulting in the delay or loss of other messages describing vital interactions with shared objects. The effect on our application is measured and we describe what application level constraints were needed to allow the Gazebo to be built.

1.1 Related Work

Various forms of interaction with shared objects have also been considered. A simple ball game using prediction to overcome the effect of network delays in a football game between UK and Germany [2]. Advanced ownership transfer allows instantaneous exchange of a ball between players in competitive scenarios. IEEE 1516 defines concurrency control that allows various attributes of a given object to be affected concurrently by distinct users. [3] describe optimisations above the standard that allow control of an artefact to be passed to a remote user with little or no delay. A virtual tennis game is played

between remote sites in [4]. [5] investigate the importance of haptic interfaces for collaborative tasks in virtual environments. The authors state that finding a general solution to supporting various collaborative haptic tasks over a network may be "too hard". A distinction is made between concurrent and sequential interaction with shared objects but this is not discussed further. As with [6] a spring model is used to overcome network latencies to support concurrent manipulation of a shared object. Four classes of shared behaviour: autonomous behaviours, synchronised behaviours, independent interactions and shared interaction are introduced by [7]. The COVEN project [8] undertook network trials of large scale collaborative applications run over the DIVE [9] CVE infrastructure. This produced a detailed analysis of network induced behaviour in CVE applications [10]. DIVE was ported to cave-like display systems [1] and consequently an experiment on a non-coupled inter-action task with two users in different walk-in displays was found to be very successful [11]. It was shown that closely coupled concurrent interaction with a shared object was not possible with CVE technology in 1995 [12]. Causal surface manipulation allows two users to carry a shared object while hiding the effects of latency through gradual deformation [13]. Recent work [14], investigates carrying a stretcher by allowing the material to follow the handles. The work concludes that, although the Internet-2 has sufficient bandwidth and levels of latency to support joint manipulation of shared objects, the CVE did not adequately address the consistency issues arising from the networks characteristics.

1.2 Principles of distribution within CVEs

A key requirement of Virtual Reality (VR) is the responsiveness of the local system. Delays in representing a perspective change following a head movement are associated with disorientation and feelings of nausea. A CVE system supports a potentially unlimited reality across a number of resource bounded computers inter-connected by a network which induces perceivable delays. Key goals of a CVE are to maximise responsive-ness and scalability while minimising latency. This is achieved through localisation and scaling. Localisation is achieved through replicating the environment, including shared information objects and avatars, on each user's machine. Sharing experience requires that replications be kept consistent. This is achieved by sending changes across the network in the form of events. Localisation goes further than simply replicating that state of the environment, it also includes the predictable behaviour of objects within it. The organisation and content of a scenegraph is optimised for the rendering of images. Although some systems [15, 16] directly link some scenegraph nodes across the network, most systems introduce a second object graph to deal with issues of distribution. Known as the replicated object model, we will from here on refer to it as the replication and its nodes as objects. Objects contain state information and may link to corresponding objects within the local scenegraph. A virtual environment is composed of objects, which may be brought to life through their behaviour and interaction. Some objects will be static and have no behaviour. Some will have behaviour driven from the real world, for example users. Alternatively, object behaviour may be procedurally defined in some computer program. In order to make a CVE attractive and productive to use it must support interaction that is sufficiently intuitive, reactive, responsive, detailed and consistent. By replicating object behaviour we reduce dependency on the network and therefore make better use of available bandwidth and increase responsiveness. Early systems replicated object states but not their behaviour. Each state change to any object was sent across the network to every replica of that object.

1.3 Road Map

This paper uses the structured task of building a Gazebo to examine various forms of shared object ma-nipulation between users in distributed walk-in devices. The Gazebo, along with lessons learnt in prototyping, is presented in section 2. Revisions to the application and CVE, along with a detailed analysis of the effect of net-work delays, are given in section 3. Section 4 concludes and suggests how a CVE could be improved to overcome our problems without resorting to application level constraints.

2 GAZEBO PROTOTYPE

We have designed the structured task of building a gazebo in order to examine distinct scenarios of sharing the manipulation of an object. This section introduces the original Gazebo, describes how it was tested between the UK and Austria and how results of these lead to a rethink.

A Gazebo (see Figure 1) is a simple structure that is often found at a vantage point or within a garden. The working environment contains materials, tools and users. Wooden beams may be inserted in metal feet and united with metal joiners. Screws fix beams in place and planks may be nailed to beams. Tools are used to drill holes, tighten screws and hammer nails. To complete the Gazebo, tools and materials must be shared in various scenarios of shared object manipulation, distinct in the method of sharing attributes (see Table 1).

Table 1. Object sharing scenarios

Scenario	Description	Method of sharing
Moving a beam (see Figure 2)	A wooden beam is too heavy to lift alone re-quiring one user to lift each end.	Concurrent manipulation of the same attributes

Fixing a beam (see Figure 3)	A horizontal beam may be united with a vertical beam with a joiner. The former must be held in place while a hole is drilled and screw fitted.	Concurrent manipulation of distinct attributes
Passing a tool (see Figure 4)	A shortage of tools requires their exchange between users.	Sequential manipulation of the same attributes

2.1 Building the Gazebo

On logging in, the user is placed in a garden strewn with building materials and tools. Avatars appear, as the rest of the team enter the garden. “Multi” stacks keep the building site tidy by creating materials on demand. A user can take material from a nearby stack and start to build the Gazebo. In the real world, constructing a Gazebo on your own is not an easy task. To simulate the task as in the real world we introduced some constrains. The simulation of gravity prohibits leaving materials in thin air and makes some materials to heavy to lift alone. The only task a single person can undertake is to drill holes and fit nails or screws. Moving, positioning and building all require teamwork. For example, one user must hold a joiner in place so that another user can fix it with a screw.

2.1.1 Moving a beam

A wooden beam is artificially made too heavy to lift alone requiring one user to lift each end (see Figure 2). This demonstrates concurrent manipulation of the position attribute as well as that of orientation of the beam. Ideally, when two users attempt to drag the beam in opposing directions, it should move to a mean position between them.

2.1.2 Fixing a beam

Beams can be united with a metal joiner and screws. A joiner may be attached to a beam by drilling a hole through both and fixing with a screw. A second beam can then be fitted into that joiner in a similar manner. One person must hold a beam while it is attached to prevent it falling (see Figure 3). This demonstrates that one user is able affect the attribute for fixing while another affects those of position and orientation; in other words, the concurrent sharing of distinct attributes.

2.1.3 Passing a tool

A hand held “multi” tool can be fitted with the necessary attachments for construction. A drill makes holes in wood and metal, a screwdriver tightens screws and a hammer hammers nails. The garden only contains a single tool so that users will need to pass it between each other. Ideally, as a tool must be held in order to move it or to change attachments, only one user can hold the tool at a time but can pass it smoothly to another user. Passing the tool (see Figure 5), demonstrates sequential manipulation of the position attribute as well as that of orientation.

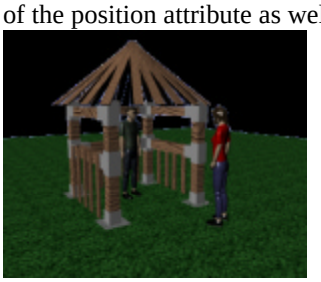

Figure 1. Gazebo

Figure 2. Moving a beam

Figure 3. Fixing a beam

Figure 4. Passing a tool

2.2 Application Design

The gazebo application was developed to work over the DIVE CVE [9]. DIVE was chosen firstly because of its wide uptake in research, secondly because of ease of application development and thirdly its advanced and modularised architecture. The behaviour of materials, tools and avatars and how the users can control this is defined in DIVE/Tcl scripts.

In DIVE, all objects are structured hierarchically in a distributed database. Their current state is represented by attributes, which may be modified by DIVE events or user defined scripts (object behaviour). A better understanding of the Gazebo application may be gained through table 2, which details the various object attributes that define the shared behaviour of objects. These are added to the default attributes such as position, rotation, parenting and graspable.

In DIVE, scripts are distributed and executed once on each remote node ensuring an identical initial state. However, when DIVE events occur, an event notification will run on that node only where it occurred. The distribution layer of DIVE is responsible for delivering this event to other nodes so that they update their simulation. Distribution of a shared environment introduces the possibility of inconsistency, caused by latency and message loss. Inconsistencies between remote replications of attributes may lead to divergent behaviour of a shared object, creating confusion between users.

Table 2. Shared attributes and the effect of divergence

Attribute	Purpose	Effect of critical divergence
All objects		
"falling"	Applies gravity to an object.	Would stay in the air when released.
"users"	Count number of acting users to simulate mass.	Would/would not be movable.
All materials		
"fixed"	Fix two materials together by disabling further manipulation and gravity	Would still fall down or be still graspable although fixed.
Beam, plank and all joiners		
"holes"	Count number of holes.	Screws could not be inserted if hole did not appear.
Screw and nail		
"coll_list"	Remember collided objects for intersection testing when fixing parts.	Would not fix some parts.
"sticked"	Signalise state for fixing parts and impact of gravity.	Would fall down and not stay in or allow to fix a part.
Power tool		
"curr_id"	Handle to currently active attachment.	Would apply incorrect attachment.

2.3 Experimentation

The prototype Gazebo application has been tested between walk-in display devices in Europe. The majority of trials were undertaken between walk-in displays at the University of Reading (UK) and Joh. Kepler University Linz (Austria). On two occasions, these were joined by another at University College London (UK). Further desktop users often joined from Reading and Linz. The Spelunk CVE [1] was used to link the walk-in displays. Spelunk is an immersive extension to the DIVE CVE. Here we present findings of the first application prototype which was regularly tested between sites over a three week period.

DIVE uses multicast messages, which is used extensively by many CVE systems to increase scalability of group communication. Although multicast works within a local area network, it is usually necessary to tunnel multicast packets between local area networks, particularly when they are separated across the Internet. DIVE proxy servers [17] were used to tunnel packets between local area networks at each site. Audio communication was supported through the UCL Robust Audio Tool (RAT) [18].

2.4 Results

Using the prototype Gazebo, each user was able to interact with objects successfully and it was generally easy to interpret what remote users where doing, especially with the support of audio communication. This reinforces the findings of other work, like [19]. The actions and gestures of tracked users were much easer to understand than those of desktop counterparts.

Two problems, however, severely hampered collaboration around shared objects. Firstly, the (although loss) ownership mechanism in DIVE made it difficult for two users to carry a beam concurrently. Secondly, many important interactions with shared objects were not being reflected remotely, such as creating objects or grasping parts. With these problems it was very difficult to build the gazebo. We lightened the beam so that one user could lift it and undertook user trials to see what could be achieved. Users in a link-up between the three walk-in displays achieved what resembled a sloppily constructed corral or sheep pen. A series of later test between Reading and Linz with users of various experiences did not improve upon this.

An investigation was undertaken into the loss of remote representation of interactions with objects. The effects of a remote user's interaction with an object were seldom presented. This was most apparent with the following interactions: creating a material from a "multi" stack; picking; passing; drilling holes; inserting nails or screws and switching tool attachments. Unlike the above user-to-object interactions, movement was always represented remotely. The primary difference between the two is the frequency and importance of updates. Our avatar's movement was represented by a continuous stream of position and orientation events. The effect of losing movement events in transit is an increase in the jerkiness of avatar movement, something an observer can cope with. In contrast, the effect on an object of user interactions is communicated by a short burst of events, that if lost will result in a lack of remote representation.

We undertook extensive tests to verify a hypothesis of event loss and why this should be a particular problem for shared manipulation between walk-in displays [20]. The movement of avatars, materials and tools all increased during shared manipulations, causing bursts of events at exactly the time when reliability and low latency were needed. These bursts were evident in latencies rising to several seconds for scenarios such as fixing beams with a joiner. We found that the problem did not arise when representing a desktop user interacting with an object. The avatar used to represent a desktop user is simpler than that used for the user of a walk-in display. We tried a simpler avatar to represent the walk in display user and found this to solve the problem. The new avatar had less moving parts and thus produced less network traffic to update. Although this avatar solved one

problem, its simplicity made human-like, non-verbal communication much harder.

DIVE incorporates an optional reliable message service, Scalable Reliable Multicast (SRM) [21]. When enabled, SRM ensures all messages from that user's device are delivered. Enabling SRM, while using the more complex avatar, ensured the representation of the effect of remote interactions with an object. The drawback of using SRM was a lag of greater than a second in the representation of the actions of a remote user, including movement and interaction.

3 IMPROVED GAZEBO

The earlier trials showed that the implementation of the Gazebo prototype application lacks heterogeneous mechanisms for concurrent sharing of objects. The reason for this can be found in the ownership transfer when grasping an object. This is to avoid consistency problems in interactive environments. To overcome this problem, intermediate carrying objects have been implemented to support cooperative manipulation of the beam. Now, two carrying tools attract the beam to align its position and orientation between the two. With this solution, hierarchy changes affect the tool instead the beam, enabling concurrent manipulation.

A second fundamental problem we experienced, was loss of critical messages that disturb the consistent state of the environment. These occurred mostly at hierarchy changes, e.g. when creating new objects at run-time or switching the attachments of a multi tool. To overcome problems of critical event loss the application was constrained to avoid vital, infrequent, events. Multi stacks were replaced by stocked material stores. The universal multi tool was replaced by distinct tools to avoid dynamic switching of state.

Table 3. Frequency of attribute modifications

Attribute	Trigger event	Typical occurrences per scenario
Prototype		
"falling"	move/release	>100
"users"	grasp/release	1-5
"fixed"	collision	1
"holes"	collision	1-10
"coll_list"	collision	1-5
"sticked"	released	1
"curr_id"	select	2-3
Revised		
"falling"	grasp/release	1-5
"users"	-	-
"fixed"	collision	1
"holes"	collision	1-10
"coll_list"	collision	1-5
"sticked"	released	1
"curr_id"	-	-

To overcome the problem of a high amount of events, introduces by the tracking system, we enhanced our DIVE version with an event filter. This reduced the frequency of events and allowed us to use our more human-like avatar.

Finally, the complexity of application level scripts was reduced to minimize the frequency of events. Table 3, compares per scenario expected event occurrences between the original and revised gazebo for various attributes detailed in the previous Table 2.

3.1 Experimentation

Again we attempted to build the Gazebo in a number of linkups between the three sites, tuning event communication to gain acceptable levels of latency and reliability. In earlier experiments we compared the usability of the application while enabling and disabling reliable multicast for all events. Investigation of the DIVE source code unearthed a way to map reliability to three categories of events: movement, geometry and general (everything else).

3.2 Results

In order to obtain a workable level of reliability, while three users shared the manipulation of objects, it was necessary to reduce the rate of sending of avatar movement to the network to 5Hz while maintaining 10Hz for the shared objects. For example, this was found sufficient when three users fixed a beam, one holding the beam, another drilling the hole and a third inserting a screw. In order to gauge latency between displays we undertook a wave test. A user at UCL moved his hand up, down, left then right, speaking the movements as he did them. At a 5Hz update rate, these movements were reflected in Reading before the spoken word, suggesting that the CVE had less latency than the audio tool. Network latencies between Reading and UCL typically vary between 15 and 25ms. The reduced update rate of avatars resulted in less natural movement making it harder to interpret their actions.

Objects still became unpickable after they had been picked by another user but far less frequently than in the original gazebo. The reliability of infrequent state changes such as drilling holes and inserting screws was also increased. Infrequent loss of such changes was often be overcome through team work. For example, if the creation of a hole is not reproduced at all sides, users can report this and ask for another hole to be drilled. The introduction of a carrying tools enabled joint manipulation of beams. Human communication helped to synchronise lifting of the beam and choosing a direction in which to carry it. Although latency was not apparent in avatar movement, remote manipulation of the beam was often delayed by up to one seconds. This resulted in wild beam movement not unlike that of a rodeo horse. The fact that

only one object was affected suggests a backlog of interpreted script events as opposed to filling of a receive buffer, which would have effected all.

Enabling reliable multicast for all events solved the problem of event loss but brought latency up into the order of seconds, even with avatar movement update was set to 5Hz. This suggests that SRM is not appropriate to our test conditions. In the current DIVE version SRM cannot be applied to selected event types only. For example, movement events could be send unreliable since they will be updated frequently. Enabling reliability for only vital messages could solve problems of critical event loss without incurring excessive latency.

4 DISCUSSION

We have found that multiple tracked users sharing the manipulation of common objects through dynamic interpreted scripts can lead to unacceptable latency and level of reliability but have shown how this may be overcome through careful tailoring of the application and configuration of event communication within the CVE. The Gazebo application was constrained to avoid vital changes to shared objects that might easily be lost, for example, minimising the dynamic creation of objects and tools with various attachments were replaced with separate tools for different jobs. Carrying objects addressed problems of joint ownership in terms of hierarchy and movement. Reducing the event communication rate from 10Hz to 5Hz reduced latency close to the level of perception and provided acceptable reliability. Although problems of remoteness can be tackled in the application, it would preferable to solve problems within the CVE itself, setting the application programmer free of concerns of the network.

The latency we have experienced is orders of magnitude grater than that of the network and comes from events being received faster than they can be processed. The problem of event loss is may be related to this, arising from overflow of receive buffers. Movement events generated from tracking are highly frequent whereas those describing vital object manipulations, such as a pick, come in short bursts. It appears that the latter are being lost by being overwritten by the former. This problem is exacerbated by a bucket algorithm within DIVE that throws away events when too many are received. Although these problems arise from receiving tracking generated events from many users, they can not be addressed with traditional scalability mechanism such as awareness management, level of detail and augmentation. These address scalability of large groups of users with subjective views of the environment. Here we have a small group sharing the same view. Mechanisms within the network level offer more appropriate answers. Categories of events, for example, may be mapped to various qualities of service for delivery. CAVERNSoft [16], a network framework for walk-in displays, and PING {Ou, 2002 #69} organise these mappings in channels, each of which has its own send and receive buffer. CAVERNSoft, unlike DIVE, relies on the application programmer determining event categories. Typically one channel is dedicated for movement and another for everything else. Restricting a channel to low frequency events thus decreases their latency.

The problem, however, does not stop there. Many events are causally dependent on others sent down a separate channel. For example, many CVEs communicate movement relative to an object's parent, such events become invalid when the object is picked up and those generated after the pick must not be delivered until after the pick event. For example, in a networked ball game [23] a delayed ball movement coming after the ball had been caught by a player resulted in the ball appearing under the ground and becoming stuck there. Neither DIVE nor CAVERNsoft can cope with this type of divergence. This problem was addressed in PaRADE [24], which made a distinction between causally supportive and relative events, sending the former reliably and ensuring any event was only enacted if causally supported. In addition to a causal time stamp, all events were stamped with natural (wall clock) time, allowing superseded events to be discarded, thus reducing latency.

We postulate that extending the PaRADE approach to time management through the use of channels would provide a level of CVE adaptation capable of supporting applications like Gazebo without the need for the application programmer to worry about delay and event loss. Such a system could be improved through an application interface and language that supports hints and reflection. Hints may allow the application or application programmer to suggest event categories, defining acceptable latency and reliability of each as well as causal relationships between them. Reflection allows the application's behaviour to adapt to available qualities of service in terms of reliability, ordering and latency. A similar philosophy was taken in the design of PING, which unfortunately was not completed.

5 CONCLUSION

The gazebo experiment has demonstrated that users, sharing the manipulation of objects, can adapt to the limited effects of remoteness between networked walk-in displays. Limiting these effects, however, required considerable effort in application development and deployment. Although many CVEs provide mechanisms for dealing with the effects of remoteness, these are barely sufficient for such linkups and require a combination of application constraints and workarounds as well as fine-tuning of event communication. CVEs have been routinely used for linking desktop display systems for some years. Walk-in and other immersive displays are different because the users are tracked. We concur with earlier work [25] that it is easier to

collaborate with a remote user when their avatar is driven by tracking data.

We conclude that a CVE does not yet exist which is capable of supporting applications like the Gazebo across walk-in displays, without unnaturally constraining the application and laboriously tuning event passing. Combining best practice from CVEs such as DIVE, PaRADE and PING could overcome the critical problems of remoteness, such as minimising latency while providing sufficient levels of reliability and causality. Network latencies will always be perceptible for some forms of shared manipulation. Application workarounds such as intermediate objects, prediction [24], or causal surface manipulation [13] can help to reduce the effect of latencies on shared manipulation.

6 ACKNOLGEMENT

We wish to thank Christoph Anthes for helping with developmental trials between the UK and Austria and Dieter Kranzlmüller and Vassil Alexandrov for facilitating collaboration between the University of Reading and the Johannes Kepler Universität Linz. We also wish to thank Anthony Steed for facilitating collaboration to UCL London.

7 REFERENCES

[1] A. Steed, J. Mortensen, and F. E., "Spelunking: Experiences using the DIVE System on CAVE-like Platforms," in *Immersive Projection Technologies and Virtual Environments*, B. Frohlicj, J. Deisinger, and H.-J. Bullinger, Eds. Wien: Springer-Verlag, 2001, pp. 153-164.

[2] D. Roberts, J. Strassner, B. G. Worthington, and P. Sharkey, "Influence of the Supporting Protocol on the Latencies Induced by Concurrency Control within a Large Scale Multi User Distributed Virtual Reality System," presented at Interanational Conference on Virtual Worlds and Simulation (VWSIM), SCS Western Multi-conference '99, San Francisco, CA,, 1999.

[3] P. M. Sharkey, M. D. Ryan, and D. J. Roberts, "Theoretical and Practical Constraints on Shared Virtual Environments due to Laten," presented at Systems Aspects of Sharing a Virtual Reality Workshop, Collaborative Virtual Environments (CVE 98), Manchester, UK, 1998.

[4] T. Molet, A. Aubel, T. Çapin, S. Carion, E. Lee, N. Magnenat-Thalmann, H. Noser, I. Pandzic, G. Sannier, and D. Thalmann, "Anyone for Tennis?," *Presence: Teleoperators & Virtual Environments*, vol. 8 (2), pp. 140 - 156, 1999.

[5] C. Basdogan, C.-H. Ho, M. Srinivasan, and M. Slater, "An Experimental Study on the Role of Touch in Shared Virtual Environments," *ACM Transactions on Computer Human Interaction*, vol. 7 (4), pp. 443-460, 2000.

[6] H. Choi, B. Choi, and S. Ryew, "HapticDisplay in Virtual Collaborative Shared by Multiple Users," *Proceedings of IEEE International workshop on Robot and Human Communication*, pp. 478-483, 1997.

[7] W. Broll, "Populating the Internet: supporting multiple users and shared applications with VRML," presented at Second Symposium on Virtual Reality Modelling Language (VRML'97), Monterey, California, 1997.

[8] E. Frécon, G. Smith, A. Steed, M. Stenius, and O. Stahl, "An Overview of the COVEN Platform," *Presence: Teleoperators & Virtual Environments*, vol. 10 (1), pp. 109 - 127, 2001.

[9] E. Frécon and M. Stenius, "DIVE: A Scalable network architecture for distributed virtual environments," *Distributed Systems Engineering Journal (special issue on Distributed Virtual Environments)*, vol. 5 (3), pp. 91-100, 1998.

[10] C. M. Greenhalgh, A. Bullock, E. Frécon, D. Lloyd, and A. Steed, "Making Networked Virtual Environments Work," *Presence: Teleoperators and Virtual Environments*, vol. 10 (2), pp. 142-159, 2001.

[11] R. Schroeder, A. Steed, A. Axelsson, I. Heldal, A. Abelin, J. Widestom, A. Nilson, and M. Slater, "Collaborating in networked immersive spaces: as good as being there together?," *Computers and Graphics*, vol. 25, pp. 781-788, 2001.

[12] W. Broll, "Interacting in distributed collaborative virtual environments," presented at IEEE Virtual Reality Annual International Symposium (VRAIS'95), Research Triangle Park, North Carolina, 1995.

[13] M. D. Ryan and P. M. Sharkey, "Distortion in Distributed Virtual Reality," presented at First International Conference on Virtual Worlds, Paris, 1998.

[14] J. Mortensen, V. Vinagayamoorthy, M. Slater, A. Steed, B. Lok, and M. C. Whitton, "Collaboration in Tele-Immersive Environments," presented at Eighth Eurographics Workshop on Virtual Environments, Barcelona, 2002.

[15] H. Tramberend, "Avocado: A Distributed Virtual Reality Framework," presented at IEEE Virtual Reality Conference (VR'99), Houston, Texas, 1999.

[16] J. Leigh, A. E. Johnson, K. S. Park, Y. J. Cho, C. Scharver, N. K. Krishnaprasad, and M. J. Lewis, "CAVERNsoft G2: A Toolkit for High Performance Tele-Immersive Collaboration," presented at Symposium on Virtual Reality Software and Technology, Seoul, Korea, 2000.

[17] E. Frécon, C. Greenhalgh, and M. Stenius, "The DiveBone - An Application-Level Network Architecture for Internet-Based CVEs.," presented at VRST'99 - Symposium on Virtual Reality Software and Technology 1999, University College London, UK, 1999.

[18] V. Hardman, A. Sasse, M. Handley, and A. Watson, "Reliable Audio for Use over the Internet," presented at INET'95, Honolulu, Hawaii, 1995.

[19] C. M. Greenhalgh, A. Bullock, L. E. Fahlén, D. Lloyd, and A. Steed, "Making Networked Virtual Environments Work,"

Presence: Teleoperators and Virtual Environments, vol. 10 (2), pp. 142-159, 2001.

[20] R. Wolff, D. Roberts, and O. Otto, "A study of event traffic during the shared manipulation of objects within collaborative virtual environments," *Presence: Teleoperators & Virtual Environments*, vol. 13 (3), pp. 251-262, 2004.

[21] S. Floyd, V. Jacobson, S. McCanne, C.-G. Liu, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," presented at Applications, Technologies, Architectures,and Protocols for Computer Communication (ACM SIGCOMM'95), Cambridge, Massachusetts, 1995.

[22] "http://www.pingproject.org/."

[23] D. J. Roberts, J. Strassner, B. G. Worthington, and P. Sharkey, "Influence of the Supporting Protocol on the Latencies Induced by Concurrency Control within a Large Scale Multi User Distributed Virtual Reality System," presented at International Conference on Virtual Worlds and Simulation (VWSIM), SCS Western Multi-conference '99, San Francisco, CA, 1999.

[24] D. J. Roberts, "A Predictive Real Time Architecture for Multi-User, Distributed, Virtual Reality." Reading, UK: University of Reading, 1996.

[25] M. Slater, A. Sadagic, M. Usoh, and R. Schroeder, "Small Group Behaviour in a Virtual and Real Environment: A Comparative Study," *Presence: Teleoperators and Virtual Environments*, vol. 9 (1), pp. 37-51, 2000.