A HAPTIC INTERFACE for LINKED IMMERSIVE and DESKTOP DISPLAYS: MAINTAINING SUFFICIENT FRAME RATE for HAPTIC RENDERING

Marcel Seelig, William Harwin
Computer Science Department
University of Reading
United Kingdom
marcelseelig@web.de, w.s.harwin@reading.ac.uk

David Roberts, Oliver Otto, Robin Wolff
Centre for Virtual Environments
University of Salford
United Kingdom
{d.j.roberts,o.otto,r.wolff}@salford.ac.uk

Abstract

We perceive the real world through many senses. Most Virtual Reality (VR) display systems, however, only present the illusion to our visual and sometimes audio senses. Walk-in display systems, such as a CAVE, allow users to see their own bodies as they interact with virtual objects. Collaborative virtual environments CVE allow many users to share the same virtual experience including the manipulation of common objects. Haptic display systems allow a user to feel virtual objects.

This project combines these three technologies to provide a natural interface for the shared manipulation of objects. Maintaining a sufficient frame rate, regardless of graphical complexity, is essential for feeling the texture of objects. We demonstrate that decoupling graphical and haptic rendering on to separate machines can maintain suitable frame rate, latency and jitter characteristics for visual and haptic senses, while maintaining sufficient consistency between them. We observed a relationship between the frame rate of visual representation affects the usability of the haptic interface.

Keywords

shared object manipulation, haptic, IPT, immersive, CVE

1 INTRODUCTION

Virtual reality environments are typically only presented to the visual and audio senses of users. Human perception is more complex, expecting additional coinciding input through our other senses. Integrating many senses into a single display system should increase the feeling of presence in the environment. Walk-in display systems, such as a CAVE, allow users to see their own bodies as they interact with virtual objects. By adding haptics, the feeling of touch, to a walk-in display we can allows natural interaction of objects. Users in geographically separate display systems may share the same virtual experience within a Collaborative Virtual Environment (CVE). Combining visual, audio and haptic display systems in collaborative virtual environments allow many users to share the same virtual experience including the manipulation of common objects.

A haptic device involves physical contact between the computer and the user. This is usually achieved through an input/output device, such as a joystick or data glove, that senses the body's movements. By using haptic devices, the user can not only feed information to the computer but can receive information from the computer in the form of a tactile & kinestatic sensation on some part of the body. This is referred to as a haptic interface. For example, in a virtual reality environment, a user can pick up a virtual tennis ball using a data glove. The computer senses the movement and moves the virtual ball on the display. However, because of the nature of a haptic interface, the user will feel the tennis ball in his hand through tactile sensations that the computer sends through the data glove, mimicking the feel of the tennis ball in the user's hand.

Many applications would stand to benefit from virtual objects that can be touched, pushed, lifted, moved etc in a closely analogous way to real objects. In virtual training simulations as well as in virtual construction simulations the users would have a great advantage if they were able to feel what they were doing. Virtual reality could be used much more effectively, for tasks that involve manipulation of objects.

Haptic interfaces typically take the form of a framework with multiple degrees of freedom. Motion is then constrained using high gain positional feedback, giving the user the illusion of hard contact with a surface. This work uses the PHANToM as the haptic interface although results would apply to any constrained motion device.

Virtual worlds are typically restricted to visual and audio feedback to human senses. Many real world collaborations rely on simultaneous close interaction through a number of senses. Multi-sensory distance collaboration has many potential applications from design to training across a wide range of sectors from medicine to aerospace. Balance integration of the senses in collaborative tele-immersive environments is significantly restricting the general uptake of the technology.

The shared manipulation of visual objects between distributed users has been researched by many but this has mostly focused on either purely sequential sharing or concurrent sharing while ensuring consistency over either visual or haptics displays. The level of cooperation in CVEs has been categorised as [1]: level 1 - co-existence and shared-perception; level 2 – individual modification of the scene; and level 3 – simultaneous interaction with object. Another catagorisation is the sequential (level 2) and concurrent (level 3) manipulation of objects through the same and distinct attributes [2]. Prediction was used to hide network delays during competitive sharing in a visual ball game played between UK and Germany [3]. Molet et al. base their work on a virtual tennis game played between remote sites[4]. A spring model is used by Choi, Choi & Ryew [5] to overcome network latencies to support concurrent manipulation of a shared object. Broll defined four classes of shared behaviours as being autonomous behaviours, synchronized behaviours, independent interactions and shared interaction [6]. An alternative approach is to define causal surfaces so that manipulations are allowed between two users who are carrying a shared object while hiding the effects of latency through a gradual deformation [7]. Recent work [8], investigates carrying a stretcher by allowing the material to follow the handles. This work concludes that adequate bandwidth and latency criteria are met by the Internet-2, the CVE did not adequately address the consistency issues arising from the networks characteristics. Probably the widest used CVE in research is The DIVE platform, which was extensively modified in the COVEN project [9, 10]. This work produced a detailed analysis of network induced behaviour in CVE applications [11]. DIVE was also demonstrated on a cave-like display where two remote users interacted with distinct objects to solve a Rubik's cube like puzzle [12, 13]. Collaborative haptic environments may be classified according to the level of sharing. [14] introduces the categories of Static, Collaborative and Cooperative which map to Ruddle's levels 1 to 3 respectively. Collaborative haptic interfaces have been studied by Basdogan, Ho, Srinivasan, & Slater [15] who 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. Another approach of interfacing haptics to a virtual environment is presented by Bouguila, Ishi & Sato [16]. Their Scaleable-SPIDAR can provide different aspects of force feedback sensations. Several studies have looked at the effects of frame rate, latency and jitter in collaborative haptic environments [17-19].

Previous experiments showed that it is possible to greatly enhance virtual experience by introducing a haptic interface [20],[21]. Nevertheless there is still much scope for research. Mortensen et al. attempted to integrate haptics in DIVE and were partially successful, but they did not achieve adequate frame rates to support the feeling of touch [8]. We extended this work by decoupling the haptics rendering from the graphics, running it on a Real-time machine linked to the graphics work-station. Thus haptics and graphics frame rates are running independently. This allows a high update for the haptics necessary for the feeling of touch.

2 MOTIVAION

Multi-sensory concurrent interaction with shared objects in a distributed virtual world remains a significant research challenge. User studies in close collaboration around shared objects, where a distributed team constructed a Gazebo [22] (see Figure 1), highlighted a negative impact of the lack of feeling of touch while working together to move, place and fix materials. Lifting a heavy beam for example without feeling its weight is not comparable to the real world where more than one person is needed for that task. The gazebo application requires people to collaborate by simulating the effect of gravity on materials. For example, two people a required to lift a heavy beam and one must hold it in place for another to fix it. Users felt reported that it would be easier to collaboratively position materials if they could physically feel them and the pressure exerted by others. Specific examples include: being able to feel the direction someone else is pulling a beam while sharing the carrying; feeling resistance when one object is pushed against another; and feeling when someone tiers or lacks concentration, through the steadiness of hand. Although all of these can be seen, it is arguable that they can be sensed more finely and quicker through touch. By providing hi-fidelity visual and haptic feedback we hope to close the gap between the shared manipulation of objects in the real and virtual world, thus supporting a wide set of new applications. The gazebo is a good application to test this as it requires a variety of forms of closely coupled shared manipulation, including sequential and concurrent sharing of objects through the same and distinct attributes. The Gazebo was first implemented above the DIVE platform as this provides comparison to much other work. This work develops a haptic plugin for DIVE and tests this within the gazebo application. A single finger Phantom device is used to provide haptic feedback.

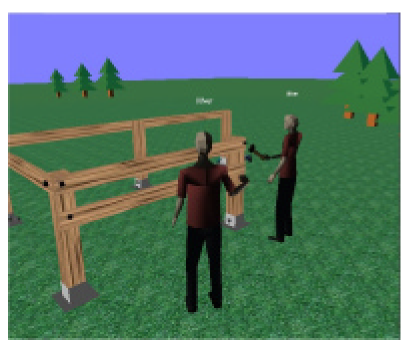

Figure 1. Virtual Gazebo

3 REQUIREMENTS

The requirements for the limits and tolerances in perceivable force are well known and devices are built to common haptic devices. The real time requirements of haptic rendering are also well known for single user interaction. Frame rates must be maintained above 1KHz [18], some 20 times above that of visual rendering. Jitter must be kept to within ms/sec, again far less than visual. Responsive, that is the latency of force feedback, must be within 200msecs [19] to allow reasonable control of an object through touch. Expectable delay between two users, while implementing spring damping is 200ms [18]. Finally reliability of event transfer should be kept above 99.9% [18].

4 CONCEPT

Other attempts to integrate haptics within CVEs have partially failed because they have not met the stringent hard real time requirements of haptic rendering. The reason for this is that they attempted to do both graphics and haptics rendering on the same machine and, to make matters worse, not using a real time operating system. Graphics rendering tends to suffer considerable jitter caused by changes in complexity of the viewed scene as the view is moved. Visual perception can tolerate this but haptics can not. This is because frequency of movement is used to describe the texture of a surface. Jitter makes the surface texture appear to change. Our approach differs as it decouples haptic and graphics rendering, running each on a separate machine.

Responsiveness to touch will be maximized through a process of replication, communally used to increase that of interaction with visually perceived scenes. However, this will be taken one step further with the haptics model being replicated at machines directly connected to haptics devices with the graphical model replicated on machines connected to the visual display.

Each haptic model will be coupled to a graphics model on a different machine on the same local area network. The two machines supporting these models and their renderers, along with another running an audio client, all connected to the same local area network, will run a single multi-sensory display environment. Collaboration between distributed users is then supported by linking these display environments via their supporting computers. The CVE is used to maintain the graphics representations and the consistency between them. In our first prototype, distributed haptic models are linked via the distributed graphics model but it intended to supplement a direct connection to perform comparative experimentation at a later date. The implemented architecture can be seen in Figure 2.

4.1 Haptic Rendering and Physics

The performance of haptic rendering is optimized by detecting collisions between objects and using such collisions as a perquisite for calculating appropriate haptic feedback. Responsiveness is therefore improved by calculating collisions as well as response at the haptic renderer. Awareness management [23] can be used to manage the appropriate set of objects that need to be replicated. To support complex worlds, the CVE must implement awareness management and it is reasonable to use this to also limit the set of objects known to the haptic renderer. However, a further optimization might remove the requirement to replicate objects that can not exhibit or repel any physical force. Object physics is an integral part of haptic rendering as we want to be able to manipulate the virtual objects in a natural way. Because the haptic renderer already has a set of physical rules for dealing with haptic objects, it is appropriate that the haptic renderer should be responsible for physics simulations in the whole environment. This overcomes two of the limitations of many CVEs: surface level collision detection; and application wide physics.

4.2 Data Representation

Objects held by the haptics render have visual counterparts in the CVE visual render. These counterparts have some aspects in common, for example, unique identifier, position and orientation and physical extent. However, they each also contains information specific to that sense. For example, a graphics object might have colour, whereas the haptics object might have stiffness. The design of the system is simplified by allowing one to be a master object that contains all information.

5 IMPLEMENTATION

The final architecture is shown for a two user system in Figure 2. Further users can be added routinely through the DIVE CVE. A network connects a DIVE plugin with the haptic PC, which is directly connected to the haptic devices (the PHANToMs). The figure also shows direct connection to the remote haptic computer. This feature is not yet part of the project but it is planned for future enhancements to improve haptic collaboration. The physics models that are used in the haptic renderer are based upon work done by Melder and Harwin [24, 25].

The implementation is divided into three parts. Each part is described in the following chapters:

1. A messaging system to the haptic renderer
2. A proxy on the haptic PC
3. A DIVE plugin on the system running DIVE and the CAVE which is a SGI.

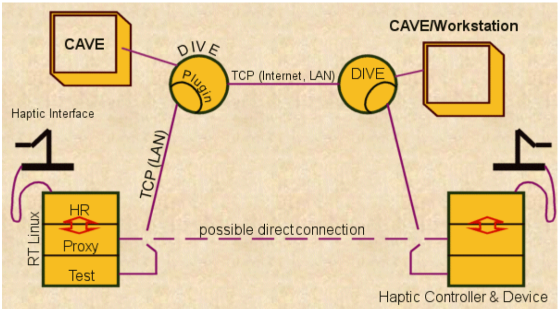

Figure 2. Overall structure of whole system

5.1 Messaging

The messaging system uses two FIFOs for communication between the Proxy and haptic renderer. It consists of functions to open a FIFO and to read and write to and from a FIFO. The messaging system also defines the structure of a message. A detailed listing of a message is given in [26]. There are currently eleven types of messages each identified by a unique id. Additionally each message contains the id of the objects that it refers to as well as other message specific information such as object coordinates.

The message number 3 (MOVE_OBJECT) is the key message for all interactions. It allows the haptic interface to move objects in DIVE based on user interactions. This message is also used to update position in the haptic renderer if any objects are moved by other users in DIVE.

5.2 Proxy

The Proxy is responsible for a reliable connection between both computers and runs as a concurrent thread on the real time machine. It converts all data into a machine independent format and forwards data from the FIFO to the network and vice versa. The Proxy is needed to have a connection to the haptic program without interfering in its computation. It is also possible to watch all traffic and to manipulate passing data. Later versions will support direct connections between proxies. This allows a faster and more reliable data transfer for haptic collaboration.

5.3 DIVE Plugin

The DIVE haptics plugin is the key part of this project. It extends the CVE system DIVE with a haptic interface. The basic principle of the plugin is to move objects according to the data received from the haptic renderer. It is also responsible for updating the haptics renderer about all touchable objects in the current DIVE world. The plugin must therefore watch all events occurring in DIVE by registering callbacks. Each time a new object is created it is checked to determine if it should be registered with the haptics renderer. For example, it is not necessary to send information about ghost objects (objects which do not have a dimension or a shape) or objects that are impossible to be rendered in the haptic space. Information about an object's haptic attributes is stored in the DIVE object properties database, which must be set in the source file of an object. If an object has been transmitted to the haptics it is then watched continuously. Any changes to an object that has been copied to the haptic renderer are sent to it. This includes movements such as translation or rotation of an object as well as its complete removal.

5.4 Interface Definitions

All data streams must pass several interfaces to go from one side to the other. The connection between DIVE and a plugin is relatively simple, because a plugin has access to all available data. The common data interface of DIVE itself is used. On the other side a similar interface to the haptic space is required. For implementation reasons and also for a better structure it is desired to have a separate interface to the haptic renderer which does not interfere with its processing. For that reason two FIFOs are implemented. It allows a fast communication between two programs without knowing the interior of the other program. All that is needed is the filename of the FIFO and the structure of the transferred data. A network is used to connect the two machines using the same FIFO data structure. Thus all data to and from the FIFO must be forwarded through the network, Figure 3.

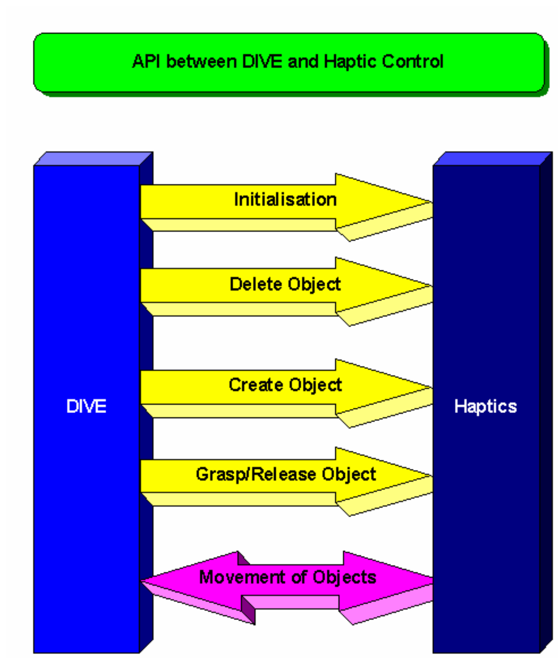

Figure 3. Interface between Haptic Control and CVE

The implementation is not dependent on the type of attached haptic interface as long as it is using the defined interface. The DIVE plugin, acts as an additional module to the DIVE CVE. When it is loaded it establishes a network connection to the proxy on the haptic renderer. After successfully connecting, the plugin splits into two threads: One to handle all incoming data from the network, the other to monitor DIVE.

When the haptic renderer is first started there is a delay before it is ready to be used. This is because the Phantoms need to be initialized internally and they need to know their position in the world. The number of phantoms is already known to the haptic renderer so the purpose of the NEW_PHANTOM message is to assign it an ID. If more NEW_PHANTOM messages are received then there are Phantoms, then these messages are ignored.

The Phantoms are stored internally as a hierarchy where the first Phantom is the parent of all other Phantoms created. The relative positions and orientations of these Phantoms to the master Phantom are internal to the haptic renderer. The SET_TOP message is used to set the position and orientation of the master phantom. It is also used to tell the haptic renderer that no more phantoms will be created. When all the Phantoms have been initialized a READY message is sent to DIVE.

An object that is moved inside the virtual world generates an event for each movement. These events are caught by the plugin and transmitted to the haptic renderer. All movements there are sent back and realized inside the DIVE world. That means that all haptic movements generate many events which would cause a loop-back. For that reason a *phantom-actor* is introduced by the plugin. That actor is not visually present in the virtual world. It is only used for being “responsible” for all movements of the haptic interface. The plugin is now able to distinguish between its own movements (originated by the haptics) and normal ones by comparing the originator of an event with this phantom-actor.

When objects are moved by other DIVE participants who do not have haptic devices, the haptic renderer must not apply forces on these objects. The reason for this is that someone without haptics cannot deal with forces. To avoid difficulties when handling haptic objects, an object is locked for the haptic renderer and all forces are switched off as long as an object is grasped by a normal DIVE user to whom, haptics are not presented. When the haptic connection is established the haptic devices are configured by sending its position of the user’s head within the visual display. Because the virtual representation of each Phantom is attached to the avatar’s body its position is updated each time the avatar moves.

6 RESULTS

Initial tests with different users have shown that the attitude towards objects in DIVE has changed. Without touch, an object was just something to click on, whereas users are now much more careful when manipulation objects with the phantom. Because of the early stage of the project all candidates had some experience with DIVE. These users all reported a significant improvement in realism. Lifting an object for example now requires more than a mouse click. The test subjects confirmed that they now needed to consider the physical attributes of an object like size, shape and weight.

Although there are delays due to networking issues and program computation time, thanks to decoupled rendering, this latency does not seem to interfere with the realism of the feeling of touch. An interesting observation is that people relied more on touch when the graphics rendering was kept above 50Hz. We have not undertaken sufficient multi-user trails to publish results on.

7 CONCLUSION

This paper outlines the methodology used to connect a general haptic interface (in this case a PHANToM) to a walk-in-display (REACTOR) via a CVE (DIVE). Its primary contribution is to demonstrate that decoupling graphical and haptic rendering on to separate machines can maintain suitable frame rate, latency and jitter characteristics for visual and haptic senses, while maintaining sufficient consistency between them. Probably the most interesting observation is that the frame rate of visual representation affects the usability of the haptic interface. This work used only loose consistency control implemented within DIVE which does incorporate advanced mechanisms for overcoming the effect of latency between cooperating users, such as [14],[27] and haptic [28]. We hope to address this in future work. Another improvement would be to use a haptic device with a working area more appropriate to a CAVE-like display, as our configuration prevented the user from walking around this area.

8 REFERENCES

[1] R. A. Ruddle, J. C. Savage, and D. M. Jones, "Symmetric and asymmetric action integration during cooperative object manipulation in virtual environments," *ACM Transactions on Computer-Human Interaction (ToCHI)*, vol. 9 (4), pp. 285-308, 2002.

[2] D. J. Roberts, R. Wolff, and O. Otto, "Constructing a Gazebo: Supporting team work in a tightly coupled, distributed task in virtual reality," *Presence: Teleoperators & Virtual Environments*, vol. 12 (6), pp. 644-657, 2003.

[3] 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.

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

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

[6] W. Broll, "Populating the Internet: Supporting Multiple Users and Shared Applications with VRML," presented at Proceedings of the VRML'97 Symposium, Monterey, Ca, 1997.

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

[8] 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.

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

[10] C. Carlsson and O. Hagsand, "DIVE - A platform for multi-user virtual environments," *Computers & Graphics*, vol. 17 (6), pp. 663-669, 1993.

[11] 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.

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

[13] 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.

[14] P. Buttolo, R. Oboe, and B. Hannaford, "Architectures for shared haptic virtual environments," *Computers and Graphics*, vol. 21 (4), pp. 421-429, 1997.

[15] 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.

[16] L. Bouguila, M. Ishii, and M. Sato, "Scaleable SPIDAR: A Haptic Interface for Human-Scale Virtual Environments," P&I, Tokyo Institute of Technology, Tokyo.

[17] D. Wang, K. Tuer, M. Rossi, L. Ni, and J. Shu, "The effect of time delays on tele-haptics," presented at The 2nd IEEE International Workshop on Audio and Visual Environments and the applications, 2003.

[18] E. Ou and C. Basdogan, "Network Considerations for a dynamic Shared Haptic Environment," presented at the national conference on undergraduate research, Whitewater, Wisconsin, 2002.

[19] R. T. Souayed, D. Gaiti, G. Pujolle, W. Y. Qang, and M. A., "Haptic Virtual Environment Performance Over IP Networks: A case study," presented at the 7th IEEE International Symposium on Distributed Simulation and Real-time Applications, Delft, The Netherlands, 2003.

[20] M. Slater, C. Basdogan, C.-H. Ho, and M. A. Srinivasan, "An Experimental Study on the Role of Touch in Shared Virtual Environments."

[21] J. K. Salisbury and M. A. Srinivasan, "Phantom-Based Haptic Interaction with Virtual Objects," MIT 1997.

[22] D. J. Roberts, R. Wolff, and O. Otto, "Constructing a Gazebo: Supporting team work in a tightly coupled, distributed task in virtual reality," *submitted to Presence*, 2002.

[23] C. Greenhalgh, "Analysing Awareness Management in Distributed Virtual Environments," presented at 2nd Annual Workshop on System Aspects of Sharing a Virtual Reality, part of CVE’98, Manchester, UK, 1999.

[24] W. S. Harwin and N. Melder, "Improved Haptic Rendering for Multi-Finger Manipulation Using Friction Cone based God-Objects," presented at Proceedings of Eurohaptics Conference, 2002.

[25] N. Melder, W. S. Harwin, and P. M. Sharkey, "Translation and Rotation of Multi-Point Contacted Virtual Objects," *submitted to VR 2003, Haptic Symposium*, unpublished 2003.

[26] M. Seelig, "Implementation Of Mechanisms To Enhance Collaboration In Multi-User Virtual Environments - Haptic Interfaces," in *M.Sc. in Computer Science*. Reading: University Of Reading, unpublished 2002.

[27] J. Jordan, J. Mortensen, M. Oliveira, M. Slater, B. K. Tay, J. Kim, and M. A. Srinivasan, "Collaboration in a Mediated Haptic Environment," presented at submitted to PRESENCE 2002: The 5th Annual International Workshop on Presence, University Fernando Pessoa, 2002.

[28] K. Hikichi, I. Arimoto, H. Morino, K. Sezaki, and Y. Yasuda, "Evaluation of adaptation control for haptics collaboration over the Internet," presented at IEEE int workshop, Communications Quality and Reliability, 2002.