Effective Cooperative Haptic Interaction over the Internet

Mashhuda Glencross*
The University of Manchester

Caroline Jay
The University of Manchester

Jeff Feasel
UNC-Chapel Hill

Luv Kohli
UNC-Chapel Hill

Mary Whitton
UNC-Chapel Hill

Roger Hubbold
The University of Manchester

Abstract

We present a system that enables, for the first time, effective transatlantic cooperative haptic manipulation of objects whose motion is computed using a physically-based model. We propose a technique for maintaining synchrony between simulations in a peer-to-peer system, while providing responsive direct manipulation for all users. The effectiveness of this approach is determined through extensive user trials involving concurrent haptic manipulation of a shared object.

A CAD assembly task, using physically-based motion simulation and haptic feedback, was carried out between the USA and the UK with network latencies in the order of 120ms. We compare the effects of latency on synchrony between peers over the Internet with a low latency (0.5ms) local area network. Both quantitatively and qualitatively, when using our technique, the performance achieved over the Internet is comparable to that on a LAN. As such, this technique constitutes a significant step forward for distributed haptic collaboration.

Keywords: Human Factors, Haptic Cooperation, Virtual Environments, Multi-user, Networked Applications.

Index Terms: I.3.2 [Computer Graphics]: Graphics Systems—Distributed/network graphics; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Haptic I/O;

1 Introduction

The inclusion of active or passive haptics in a virtual environment (VE) improves user experience by providing a reinforcing additional sensory cue, and in doing so overcomes the basic problem of there being 'nothing to feel,' as is the case in most VEs.

Haptic feedback in VEs has been found to contribute significantly to intuitive interaction [4], a sense of presence or co-presence [2, 19] and skills transfer [1]. Haptic feedback can benefit applications in diverse areas such as military training [7], CAD prototyping assembly and maintenance [27, 24], tele-surgery [12, 35, 11] and creative painting [3] or sculpting [9]. In the real world, tasks performed in these areas can often be collaborative, involving the cooperative efforts of small teams of individuals working towards a shared objective. Members of such design teams may be geographically dispersed, and therefore may need to work together using distributed collaborative applications connected by the Internet.

With the increasing monetary and environmental costs of business travel, intuitive mechanisms which facilitate long-distance cooperation between small but geographically dispersed design teams will become increasingly important. A number of specific training scenarios exist, in both the defense and aircraft manufacturing industries, in which cooperative manipulation is important. The CAD assembly task reported in this paper is a simplified version of a real test case (provided by a Spanish defense company) requiring cooperative manipulation.

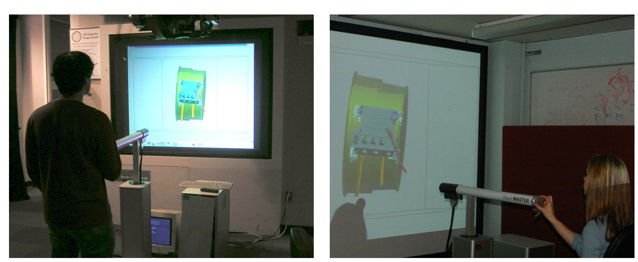
The additional sensory feedback provided by haptics can impart valuable cues regarding the actions and intentions of participants performing cooperative tasks [32, 2]. However, three major factors make haptic cooperation over the Internet challenging:

1. Maintaining both local responsiveness and consistent simulation state for distributed participants is difficult. Client/server approaches suffer round-trip communication delays, whereas peer-to-peer systems are difficult to synchronize [6].
2. Latency, especially over large distances, is unavoidable. Network latency adversely affects the stability of haptic rendering, as an object can be penetrated before its correct position is received, resulting in disconcerting rebounds [8].
3. Haptic rendering must be performed at a 1KHz update rate (or better) for correct perception of solid contact [29, 5].

We have developed what we believe to be the first system supporting haptic cooperation over the Internet that effectively addresses these issues. Furthermore, we have demonstrated its utility via experimental studies.

Our system adopts a peer-to-peer architecture to maintain local responsiveness; over the Internet conditions tested, it reduces positional and rotational divergence between objects at each peer to 0.02mm and 0.03–0.04 degrees respectively (mean values measured in the context of the large workspace of an FCS HapticMASTER device). This is achieved through the separation of objects simulated at each peer into *globally correct* and *locally perceived* representations. Synchrony between globally correct representations is maintained through event ordering using a consistent global time co-ordinate frame, while local responsiveness is achieved through an *optimistic* simulation and roll-back mechanism. The system decouples haptic display, simulation, and graphical rendering into asynchronous components so that each can run at appropriate rates.

We show, through studies of users undertaking a collaborative haptic prototyping task (shown in Figure 1) employing a realistic



(a) User in the USA (b) User in the UK

Figure 1: Two geographically distant users performing a cooperative haptic prototyping task. The user on the left is in the USA and the user on the right in the UK. These photographs were taken during an actual transatlantic trial.

*e-mail: mashhuda@manchester.ac.uk

physics simulation, that haptic cooperation **is possible** over the Internet using TCP/IP. Our methods maintain stable haptic feedback and consistent simulation state between participants.

2 Design Considerations

Collaboration in shared VEs over the Internet usually breaks down due to network latency (the time taken between packet sending and its receipt) and jitter (variation in latency) causing errors in user perception of causality [28, 21, 30]. Studies of collaboration systems show that latency and jitter adversely affect users' abilities to correctly predict the actions of others and hence their success at performing coordinated actions [13].

To illustrate these issues, consider the situation in Figure 2. Here, we see a two-dimensional depiction of two users, referred to as *A* and *B*, manipulating a box. *A*'s end-effector - the cursor they use to grab the box - is represented by the larger circle with the hatch pattern, while *B*'s is coloured green. In the situation shown on the left, user *A* perceives the box to be where the blue (dark) box is shown, whereas user *B*, whose simulation is running ahead, perceives it to be where the red (lighter semi-transparent) box is shown. Note that we have exaggerated the box positions for clarity. User *B* establishes contact with the box and pulls it. The box rotates, as seen on the right, but since user *A*'s view is delayed, he or she still sees the box in the blue position and proceeds to make contact with it. At this point, each user is seeing something different, and as they push or pull the box their simulations diverge.

If purely linear motion computation is used and no discrete events which affect the motion can occur, then separate simulations can be largely independent. Latency in the exchange of state information can be tolerated because the order in which events are applied at each peer does not matter in such simple simulations. On the other hand, when torque is introduced into state equations then the order in which interaction events affecting motion are applied becomes critical. Furthermore, discrete events such as holding or releasing the box in Figure 2 require all cooperating users to share a common and consistent perception of state to ensure the same outcome. In a peer-to-peer system without suitable concurrency control, and reliable simulation event delivery, divergent situations can never again agree [25].

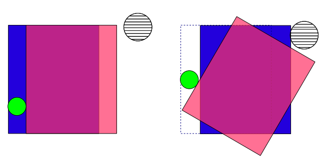


Figure 2: Potential inconsistency between peers

Applications that demand consistency can adopt a client/server approach in which state is maintained at a single locus of control. In this scenario, all inputs are routed through a server which provides state information to all connected clients, and all participants suffer a round-trip delay equivalent to that of the slowest connected client [6, 31]. Alternatively, peer-to-peer approaches replicate the system at each client, and no single participant has complete knowledge of global state. Here, each participant enjoys responsive local interaction, but techniques must be incorporated to ensure consistency [25, 26]. For a detailed discussion of the implications of distribution architectural choices see 'Networked Virtual Environments Design and Implementation' [36]. We compare our system with others supporting haptic collaboration in Section 6.

Four requirements drove our implementation choices. First, local interaction should be highly responsive, to provide a compelling user experience. Second, realistic physics simulation of translational and rotational motion of the shared object must be supported for carrying out real tasks. Third, consistent global state should be maintained to prevent divergent simulations at each client. Fourth, the solution should support concurrent manipulation of shared objects over the Internet, which has typical latencies of 50–200ms.

3 Enabling Haptic Cooperation

In order to maintain a highly responsive interaction, we adopt a peer-to-peer distribution architecture, with an identical software and hardware setup replicated at each peer as shown in Figure 3. Note that each peer has an associated local haptic daemon (HD) which is responsible for ensuring high-quality haptic display.

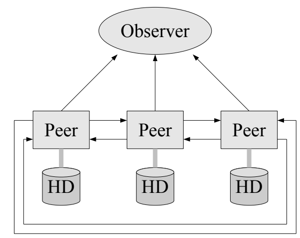


Figure 3: System architecture

As a consequence of multiple loci of control in fully peer-to-peer systems, there is no definitive global knowledge of state maintained at any instant in time. This poses a problem for managing state information when participants join or leave an active simulation. A common solution is to use a broker which manages and supplies information as peers connect or disconnect [36]. We adopt this solution through an 'observer' which records definitive snapshots of state (termed *objective state*) and supplies this when required. Further, it informs all participants when a peer disconnects from an ongoing simulation. For our purposes, the observer is similar in functionality to a full peer but does not execute haptic and graphical display as shown in Figure 4.

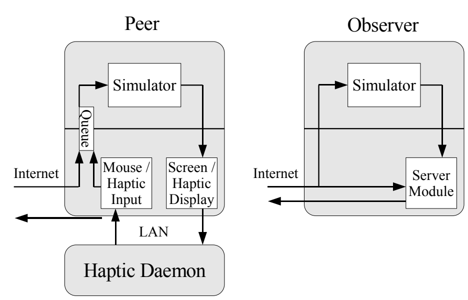


Figure 4: Comparison of peer and observer functionality

Each peer has two separate processes shown in Figure 5, one responsible for haptic rendering and the other for simulation and communication. The main process shown in Figure 5 executes a physics-based simulation in one thread, and handles network communication and graphical display (I/O) in the other. Simulation, communication and graphical/haptic display all run asynchronously. Only the end-effector position of the haptic device and force readings measured by the device are broadcast to all peers. These are inserted into each independently executing simulation. We compute and integrate over time linear forces and rotational torques, applied through user interaction with a rigid body, at each peer using standard textbook Newtonian forward dynamics [10].

We use a second dedicated computer (a haptic daemon), connected to the local network to control the haptic device. This provides high-quality haptic feedback by maintaining the required 1KHz haptic update rate [22, 37]. Flow of control within the I/O and simulation threads is shown by the blue (dashed) arrows, and

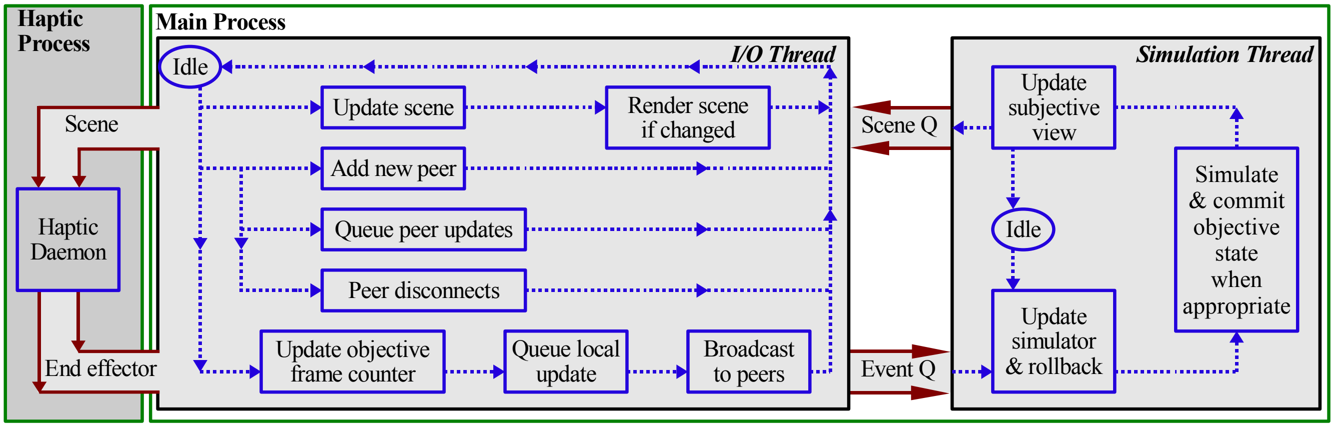


Figure 5: Flow of control and data at each peer in our system

data transfer between them occurs through the scene and event queues. The haptic process, shown in dark grey at the left side of the figure, exchanges scene data and end-effector positions with the I/O thread, so that the haptic representation always corresponds to the graphical display.

The main challenge is to maintain consistency between peers communicating over a high-latency connection. To ensure consistent state across all connected peers, we use the concept of *virtual time* together with simulation and rollback [17]. Virtual time is an abstraction of real time and is used to enforce a consistent temporal coordinate system in distributed simulations. To implement virtual time, we use a system of logical clocks [20] (one at each peer) to order events for correct insertion into each independently executing simulation, and to determine how well synchronized they are with respect to other simulations in the system. These logical clocks are simply integer counters representing a frame number (or clock-tick) for the simulation. The interval itself between clock-ticks at each peer in each separate simulation's time-base is adaptively adjusted, based on network conditions, to maintain overall synchrony between peers. We determine network conditions on-the-fly through ping-like requests interspersed in our communication protocol. This information is used to constantly monitor latency and re-calibrate the simulation time frame.

Recall that motion of the shared object is computed by integrating calculated forces and torques over time using forward dynamics. Due to network latency, at any given instant in time peers will not have received all inputs necessary for computing a correct position for the shared object. As shown in Figure 6, in the absence of remote updates, we maintain responsive local feedback by performing an *optimistic* simulation for the current and future times (t_n and t_{n+1}) using partial knowledge of state. Given the available knowledge, a possible outcome is computed and the simulation continued. However when remote updates arrive late, we re-write history by rolling-back the simulation, inserting the update into the correct position and invalidating the subsequent optimistically computed states. In the interests of performance, we lazily re-calculate invalidated states when the simulation actually needs to be graphically or haptically displayed. This may potentially cause brief glitches if a very late event is undone as a consequence of a rollback, and our optimistic prediction is also wrong. However with our approach consistent global state is always recovered. Incorrect predictions are rare, so we prefer to accept an infrequent glitch rather than introducing the additional latency required to explicitly interpolate and smooth over this glitch.

We refer to a snapshot where all inputs for a given logical clock time are known to be received as the *objective* state of the system (shown at time t_{n-1} in Figure 6). It is critical to the system synchrony that these objective snapshots agree at each peer for a given logical clock tick. However, peers may compute objective snapshots at different wall-clock times due to network latency. Since the number of clients connected is stored in the snapshot data-structure and their inputs are time-stamped with both the logical clock time and the wall-clock time we can (at each peer) be certain when all inputs have been received. This objective state is entirely serializable and recorded by all of the connected peers, including the observer. When new participants join, the observer provides an objective snapshot at a given logical clock time and the newly connected peer simulates from that time forwards.

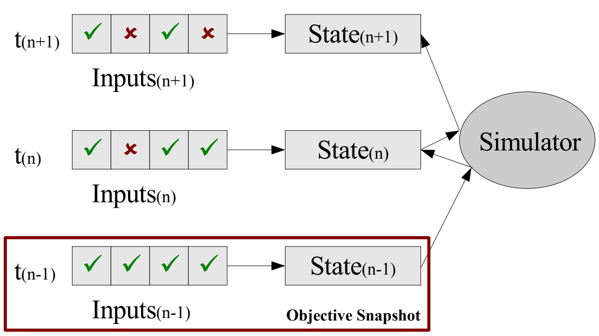


Figure 6: Simulation snapshots, the checks represent known inputs while the crosses indicate missing inputs

In contrast, the *subjective* state shown at t_n and t_{n+1} perceived by each user may vary slightly from this objective state. This variation is quantified in Section 5. We have borrowed the terms *objective* and *subjective* from the Deva collaborative virtual reality system [31]. However, Deva is a client-server system, and this allows the server to maintain the true objective state [23]. In contrast, all peers in our system (including the observer) know the true objective state only at particular, prior intervals in time, but not necessarily at the current time. The objective state is updated each time that all updates for a specific time interval have been received, while the subjective state is continually updated. If communication with any client disconnects unexpectedly, other clients will continue to rollback and synchronize correctly, using the data from remaining clients.

4 EXPERIMENTS

We performed experimental studies to verify that our simulation-state synchronization facilitates cooperative haptic activity across the Internet. In particular, we demonstrate that minor differences in the subjective view between participants is not detrimental to performing a shared cooperative haptic task. We additionally investigate the use of Kalman filtering [18] as a predictive method to reduce variation in subjective views.

Our experimental task was chosen from the mechanical CAD prototyping domain: cooperative mounting of a fuel control box (hereafter, *box*) onto an aircraft engine housing (hereafter, *target*). This is a real industrial case study of assembly procedures from a collaborative design project. The use of human subjects in this study was approved by both the University of North Carolina at Chapel Hill, and the University of Manchester ethics committees.

4.1 Experimental Design

The experimental study was carried out in two parts; both were within-subjects repeated-measures designs. *Part 1* was a 2×2 design that tested cooperative haptic performance in local area network (LAN) and wide area network (WAN) conditions, both with and without our synchronization techniques enabled. The second part tested only two conditions: cooperative haptic performance in the WAN condition, with and without prediction enabled. To minimize learning effects, in both parts of the study each participant performed two randomized trials per condition, consequently in *Part 1* eight random order trials and in *Part 2* four random order trials were performed. Identical hardware and software configurations were used in both parts with the common condition between the two being the synchronized case.

Metrics

During each trial, we recorded the position and orientation of each participant's box at each clock-tick of the physics simulation. From this data we generated two time-series showing the positional and rotational difference between each user's subjective view of the box, as our quantitative indicators of overall system synchrony. We recorded the elapsed time and time-to-successful-completion during each trial. User perception of each condition was assessed through questionnaires administered after each trial. Participants were asked to rate on a scale of 1 to 10 how difficult they found the task and how disrupted they found the visual and haptic feedback, where 1 = the practice condition (LAN, synchronized) and 10 = very difficult/disrupted.

We tested the statistical significance of the divergence and time-to-completion data from *Part 1* using an Analysis of Variance (ANOVA) and the data from *Part 2* using a t-test.

Hypothesis

Our hypothesis is that there will be significantly less divergence of the mean position and orientation in the synchronized than in the unsynchronized case, and that the synchronized-predicted case will reduce divergence further.

4.2 Method

An expert user in the USA (site 1) cooperated with study participants in the UK (site 2). Volunteers were recruited from the UK site's graduate student pool. Data was collected for six participants in *Part 1*, and 15 participants in *Part 2*. Since we collected quantitative measures of system synchrony, and task performance, the number of participants in the study was considered to be sufficient. The level of experience with the haptic device varied from none to moderate.

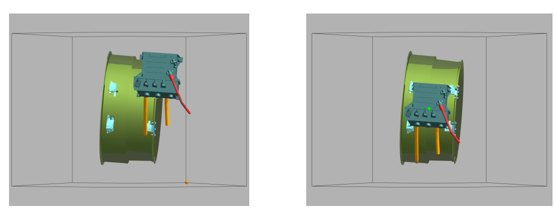
For interaction with the box (and each other), each user was equipped with an FCS Robotics HapticMASTER force feedback device. Cursors, visible to both participants, indicated where the two HapticMASTER end effectors were in the environment. Each participant viewed his or her simulation of the collaborative VE on a projection screen (Figure 1), and communicated by speaker-phone to discuss an appropriate strategy while performing the task. At site 1, the application ran under Red Hat Enterprise Linux on a 2.4GHz PC with 4GB of memory and NVidia GeForce4 Ti 4600. Site 2 used Gentoo Linux on a 3.2GHz PC with 1GB of memory and an NVidia 6800 Ultra graphics card. Network latency between study machines at site 1 and site 2 was measured using ping before and after each scheduled participant study. The round-trip-time latency ranged from 105 to 120 ms, and averaged 117 ms, while on the local area network this averages 0.5ms.

Haptic state is updated by the physics simulator (through the daemon) over a LAN with negligible local communication delay, at a configurable update rate of 20 frames per second. While the haptic daemon awaits new simulation updates, it maintains stable state based on the most current knowledge and executes haptic rendering at 1KHz to ensure stability. Graphical rendering consumes approximately a quarter of the main application processing time each frame, while simulation and roll-back consumes the majority. However, since these are undertaken on separate threads the effect of rendering and simulation delay on haptic feedback is negligible compared to the transmission delay over the Internet.

The recorded simulation time-stamp, used to properly order events during synchronization, and wall-clock time-stamp, is used to correctly match data logged at each peer on a uniform time-line.

Experimental Session Details

After completing consent forms and reading an information sheet detailing the study, the user was introduced to the expert collaborator. The pair first completed one or two practice trials to familiarize the participant with the equipment and task. Most participants were novice users of the application, and had to be acquainted with both the feel of the haptic device and how to attach/detach from the box using a mouse button. The application allows users to attach to the box at arbitrary contact points and pull or push it in any direction. At the start of a trial, the orientation of the box was correct for docking, but the position was offset (as shown in Figure 7(a)). Minimizing rotation - a key to successful completion of the task - required cooperation and communication between the users. Once the box met a threshold orientation (within 9.5 degrees in all axes) and distance (within 4cm from the centre of the box to the centre of the target) relative to the docked position (Figure 7(b)) it automatically snapped into position.



(a) Start position of the box (b) Target position of the box

Figure 7: Start and target position in the cooperative CAD prototyping study

In order to reduce the number of variables in the experiment and to meet ethical requirements for experimental studies involving human subjects, the task had to be simple enough to be repeatedly performed under the conditions tested. The shared box was easy to manoeuvre: it was only slightly damped, and possessed both mass and inertia, however no gravity or friction forces were applied. This was not a limitation of the physics engine, rather a considered choice to strike a sensible balance between task difficulty and not unduly masking the effects of latency.

During each trial, the expert user in the USA guided the novice in strategy, as well as providing specific help on how and where to push and pull on the box in order to dock it. Typically each participant in the study took around 45 minutes to carry out the entire experiment and fill out necessary forms and questionnaires. Data were collected from approximately 20 hours of validation studies.

5 Results and Data Analysis

5.1 Effect of Divergent State

In the cooperative prototyping case study, simulation of motion for the box is computed using forward dynamics. The implication of this is that state components such as current forces, velocities, accelerations and positions are functions of the previous state and subsequent inputs. A cumulative error in any of these between peers leads to different outcomes. Network latency resulting in a small difference in each peer's perception of states, compounds this problem.

In order to provide a base case with which to compare our technique for maintaining consistency, we consider the naive case with no form of concurrency control. It is not possible to show graphs for all the data collected during the experimental trials in this paper; however mean values from all the data collected are presented in Figure 13. The graphs in Figures 8 and 9 illustrate this naive scenario with typical data gathered in the unsynchronized condition between the USA and UK. In Figure 8, the difference in position of the box between simulations running at two peers is shown. The second graph shown in Figure 9, plots the orientation difference in degrees of the fuel box in each of the x (green line, # symbol), y (red line, @ symbol), and z (blue line, * symbol) axes between the simulations. The catastrophic accumulation in error is very apparent. A potential cause of this was found to be a discrete event which occurs for one peer but not the other; for example, when peers disagree over whether or not the box collided with the target.

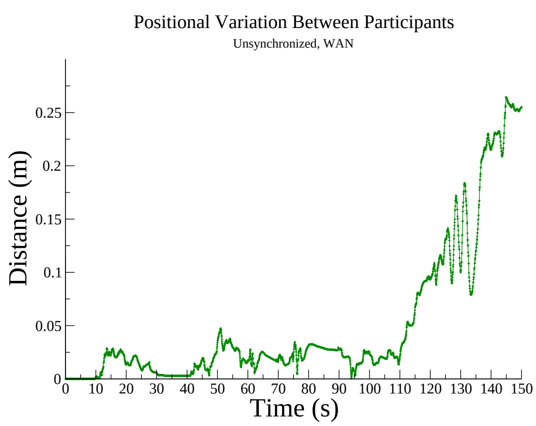


Figure 8: Positional divergence between peers, unsynchronized WAN

In contrast, the graphs in Figures 10 and 11 show the typical positional and rotational divergence between two peers, one in the USA and the other in the UK, in the synchronized case. Note the magnification in scale of the vertical axis in the synchronized conditions which exhibit significantly lower divergence. The Time axis is shorter for the synchronized conditions compared to the unsynchronized ones due to reduced task completion times in the former. Whereas the subjective views diverge by a small amount for short durations (see Figure 13), they are periodically unified when more up-to-date information becomes available.

Note that the mean positional and rotational divergences (measured in meters and degrees respectively) in the synchronized con-

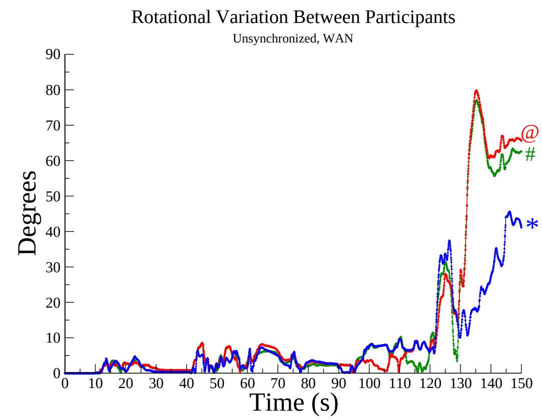


Figure 9: Rotational divergence between peers, unsynchronized WAN

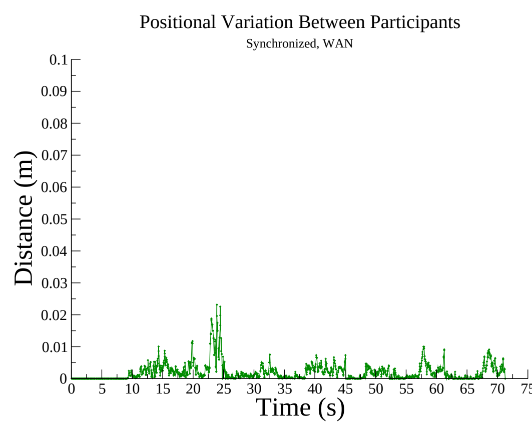


Figure 10: Positional difference between peers, synchronized WAN

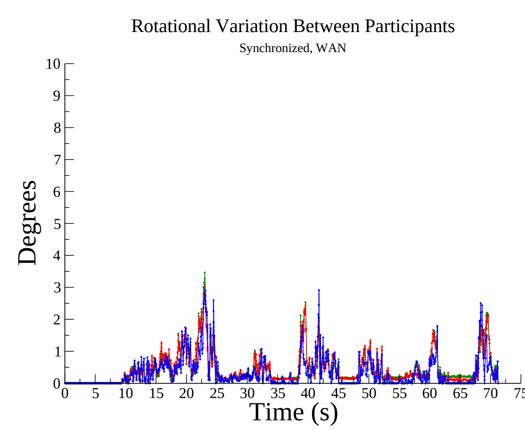


Figure 11: Rotational difference between peers, synchronized WAN

dition are very small compared to the large workspace (a swept volume of 400mm in height) of the HapticMASTER device. Our mean divergence values cannot be considered to be noise in the end-effector position of the haptic device, which is an order of magnitude lower (reported by FCS control systems to be 0.004mm). We further confirmed that the inherent divergence in our system is imperceptible by a test in which latency, equivalent to the upper level of that in the main experiments (120ms), was simulated at the UK site, which was equipped with two HapticMASTER devices. Figure 12 shows that this amount of divergence cannot be perceived even when two users are situated side by side in the same laboratory. After correction for perspective and lens distortion, the Photoshop 'Difference' operator was used to compare the two insets.

Maximum reported divergence values correspond to the completion state, where the box snaps to the correct position and orientation once it is within task completion tolerances. Task completion times in the unsynchronized cases were measured for the first person to successfully complete. Due to the level of divergence in the unsynchronized conditions, in half the trials, only one participant was able to complete the task. The best completion time occurred for the unsynchronized LAN condition. However this was not found to be statistically significant, and furthermore it masks the fact that the simulations still diverged. The difficulty, visual and haptic disruption measures are subjective measures obtained from the questionnaires. Participants did not rate the conditions as significantly different. User perception of difficulty and visual/haptic disruption may have been confounded by the fact that users were frustrated while carrying out the unsynchronized WAN conditions which were clearly difficult as reflected by the significantly higher task completion times. It was observed that in the unsynchronized case catastrophic divergence often took place before participants realised it had occurred. This could explain only the small difference in difficulty rating between the WAN conditions.

Significant Analysis of Variance (ANOVA) results are shown in Figure 14. From our logged data we have determined that there is a main effect of synchronization, indicating significant differences in the amount of divergence between conditions. In the unsynchronized condition, the greater latency introduced through the transatlantic connection causes rapidly increasing divergence. However, when the systems are synchronized, this divergence is reduced to an equivalent value to that achieved on a LAN. These results show that the synchronization technique results in a cooperative haptic experience similar to that achieved over a LAN. Task completion times could not be fairly analyzed for statistical significance as in the unsynchronized WAN condition the task was often either only completed for one participant or aborted as it became too difficult.

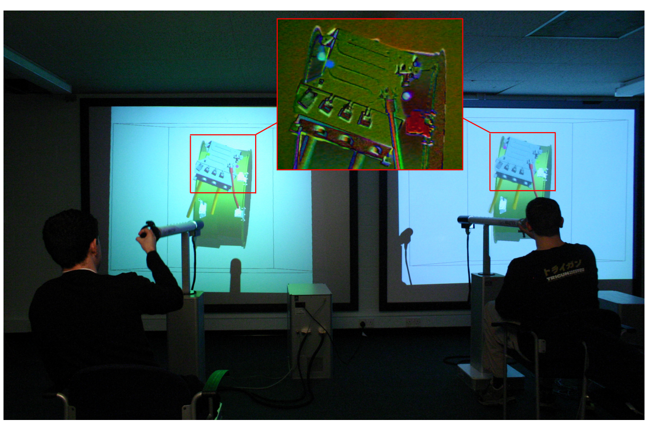


Figure 12: Two users performing the task with simulated latency equivalent to the WAN condition (120ms) and synchronization/prediction

5.2 Effect of Prediction

The prediction experiments (*Part 2* of the study) were conducted with 15 participants, with each completing 4 trials (2 per condition). These 60 trials reinforced the synchronization results obtained in *Part 1*.

Measure	*LAN, Synchronized*	*WAN, Synchronized*	*LAN, Unsynchronized*	*WAN, Unsynchronized*
Position max (m)	0.046611	0.037485	0.114211	0.173613
Position mean (m)	*0.000197*	*0.000197*	*0.003789*	*0.005506*
X orientation max (Deg)	5.823782	4.334057	20.79167	45.24178
X orientation mean (Deg)	*0.040414*	*0.043591*	*0.669967*	*1.057691*
Y orientation max (Deg)	7.02106	5.443782	20.82299	48.90432
Y orientation mean (Deg)	*0.038977*	*0.040279*	*0.681335*	*1.129944*
Z orientation max (Deg)	7.580028	6.231542	17.01083	52.79422
Z orientation mean (Deg)	*0.036494*	*0.038253*	*0.599121*	*1.036472*
Completion time (s)	74.99655	72.2705	68.76667	156.76
Difficulty	3.375	4.794118	3.205882	4.823529
Visual disruption	2.5	2	2.5	4.5
Haptic disruption	6	4	6	6.5

Figure 13: Table of mean values comparing the Synchronized and Unsynchronized conditions on the LAN and WAN

Measure	*Variable*	*ANOVA results*
Position max	Synchronization	$F_{1,5} = 98$, $p < 0.001$
Position mean	Synchronization	$F_{1,5} = 781.9$, $p < 0.001$
X orientation max	Synchronization	$F_{1,5} = 10.37$, $p < 0.05$
X orientation mean	Synchronization	$F_{1,5} = 64.1$, $p < 0.001$
Y orientation max	Synchronization	$F_{1,5} = 14.3$, $p < 0.05$
Y orientation mean	Synchronization	$F_{1,5} = 72.83$, $p < 0.001$
Z orientation max	Synchronization	$F_{1,5} = 12.57$, $p < 0.05$
Z orientation mean	Synchronization	$F_{1,5} = 55.17$, $p < 0.001$
X orientation max	Latency	$F_{1,5} = 7$, $p < 0.05$
X orientation mean	Latency	$F_{1,5} = 6.81$, $p < 0.05$
Z orientation max	Latency	$F_{1,5} = 7.31$, $p < 0.05$
Position max	Latency * Synchronization	$F_{1,5} = 7.65$, $p < 0.05$
X orientation max	Latency * Synchronization	$F_{1,5} = 8.51$, $p < 0.05$
Z orientation max	Latency * Synchronization	$F_{1,5} = 8.79$, $p < 0.05$

Figure 14: Results of ANOVA from experiment comparing Synchronized and Unsynchronized conditions on the LAN and WAN (only significant conditions are listed)

Our expectation was that the addition of prediction would reduce minor fluctuations introduced by the rollback. In practice, no significant main effects were found between the synchronized and predicted cases. However, the mean positional divergence in the predicted case was 0.01mm larger than in the synchronized case. We attribute this to the fact that at various times during a session, users attach to and detach from the box, and make sudden changes in direction, and rapid movements. These cause the Kalman filter to over-predict. It may be that some fine tuning of the filter would improve this result, but the magnitude of the divergence is already so small that any improvement would also be very small. A filter that adapts to network characteristics could offer a greater benefit over a wide range of network characteristics.

6 Comparison With Other Systems

The importance of maintaining consistent state between peers, especially when performing cooperative haptic tasks, is apparent from our study of divergent state. This requirement however conflicts with the need for fast updates to provide high quality haptic feedback. Consequently, the few systems that currently exist simplify the problem considerably by either: (i) only enabling haptic cooperation under low latency conditions, distributing only positional information of the shared object and computing an ideal position based on these, computing only linear motions (simple physics simulation); (ii) using heavily damped motion to counteract high la-

System	*Distribution architecture*	*Haptic update rate control*	*Co-operation on high latency network*	*Locus of control*	*Physics simulation*
Transatlantic Touch [19]	Peer-to-peer	✗	✓ Internet 2	Multiple	✓ Simple
Adaptation Control [15]	Peer-to-peer	✗	✗ Simulated network	Multiple	✓ Simple
Virtual Haptic World [14]	Peer-to-peer	✗	✗ Fast link	Multiple	✓ Unclear
The Nanomanipulator [16]	Peer-to-peer	✓	✓ Internet	Multiple	✗ None
C-HAVE [34]	Peer-to-peer object management	✗	✗ Fast link	Single	Insufficient detail
Roaming Server [24]	Client-server	✓	✓ Internet, round-trip for remote participants	Single movable	✓ Simple
CSIRO surgical simulator [11]	Client-server	✗	✓ Internet	Single static	✓ Simple
Virtual Coupling Networks [33]	Peer-to-peer and Client-server	✗	✗ Simulated network with fixed latency	Multiple and Single	✓ Simple
Our system	Peer-to-peer	✓	✓ Internet	Multiple	✓ Complex

Figure 15: Summary of cooperative haptic systems

tency. We have deferred discussion of these systems until now in order to provide a contrast with our approach. Our system has the following properties:

- It is peer-to-peer
- Has haptic update rate control
- Effectively allows haptic cooperation over the Internet
- Has multiple loci of control
- Uses a physically-based simulation engine to compute both linear and rotational motion (complex physics simulation) of rigid bodies

Systems described in other prior works have some of these properties but to our knowledge none has all of them, therefore a direct like-for-like comparison is difficult. Furthermore, none of the systems supporting shared haptic environments are publicly available, thus it is only possible to make a comparison based on reported functionality. Each of the systems summarized in Figure 15 are specifically aimed at facilitating shared haptic manipulation over the Internet and therefore represent key prior work.

In an important cooperative haptic study, Kim et al. employed UDP over Internet 2 to minimise network communication latency [19]. They successfully demonstrated haptic cooperation by using heavy damping. Furthermore, they stated that the event delivery characteristics of TCP/IP were unsuitable for haptic cooperation. In an earlier work Hikichi et al. studied a target acquisition task to evaluate the effect of jitter over simulated networks [15]. An interesting aspect of their study was an analysis of how jitter adversely affects systems employing dead-reckoning. Hespanha et al. and Shen et al. both adopted solutions in which haptic cooperation was limited to participants sharing low latency network connections [14, 34]. The nanomanipulator project [16] is the most similar to our approach but with an important and fundamental difference: Hudson et al.'s system did not support a physics-based simulation, thus limiting its application. In a more recent study, Sankaranarayanan and Hannaford use virtual coupling networks (spring-damper systems) to maintain position coherency in distributed haptic environments [33]. Traditionally such schemes are used to increase haptic stability between haptic devices but at the cost of lower quality solid contact. The authors apply the technique to stable interaction between each user's representation of a shared object. The target acquisition task they studied involves simple physics, with motion of a cube constrained to one degree of freedom only. Both Marsh et al. and Gunn et al. adopt a client-server approach, so that for cooperative manipulation a round trip to the server is unavoidable for at least one of the users [24, 11].

7 Conclusions

Cooperative haptic interaction over the Internet is especially challenging. Latencies over large distances cannot be avoided, and are typically much higher than levels at which human task performance is adversely affected. A client-server approach incurs further latency, making it unsuitable for many interactive applications. Whilst peer-to-peer architectures avoid this issue, they can result in large divergences between simulations at different sites.

We have described a method for maintaining synchronization between users in a peer-to-peer system. Our experimental studies confirm the validity of our approach for distributed cooperative manipulation of shared objects. The monitored divergence between the views of the two participants is very small: typically the mean positional variation of the box is around 0.02 millimetres in a swept-volume workspace of 40 centimetres depth, while the mean rotational divergence around the x, y and z axes is 0.03–0.04 degrees.

To determine the scalability of the technique, it is important to explore some further issues. For the task we described here, rollback was extremely effective; however, as it can be computationally intensive, it remains to be seen whether its effectiveness is compromised in a more complex simulation. Its application might also be limited by the number of users engaged in a concurrent manipulation. In principle, it supports many, although so far its efficacy has only been thoroughly tested with two. There is also the question of network stability. Our present experiments were conducted over a relatively stable network with low jitter. However, the system was thoroughly evaluated in 20 hours of experiments with participants over the LAN and Internet. Further experiments are required to characterise performance on a wider variety of networks with higher latencies, or where jitter is more prevalent. It is envisaged that in these situations, predictive techniques will be far more important for minimising divergence and improving user experience.

Although there are avenues for future research, the technique in its current form enables very effective haptic cooperation over the Internet. The major goal of the research was to provide users with a compelling collaborative experience not compromised by the latencies inherent to the Internet. Both quantitatively and qualitatively, the performance levels achieved when using this method over the Internet are comparable to those on a local area network with minimal latency. As such, this technique constitutes a significant step forward for distributed haptic collaboration.

ACKNOWLEDGEMENTS

We would like to thank Professor Fred Brooks for his valuable comments while testing the system, and proof reading this paper. We would also like to thank the participants that took part in our studies, Dr. James Marsh for valuable technical discussions, and Professor Alan Chalmers and Greg Ward for their proof reading efforts. Thanks also to Piet Lammertse and Wim Lam at FCS Control systems for the kind loan of a HapticMASTER device used at UNC for the studies. The provision of models from SENER INGENIERIA Y SISTEMAS, and funding from the UK Engineering and Physical Sciences Research Council (grant number GR/523087/01) and US NIH National Institute for Biomedical Imaging and Bioengineering are gratefully acknowledged.

REFERENCES

[1] R. J. Adams, D. Klowden, and B. Hannaford. Virtual training for a manual assembly task. *Haptics-e*, 2(2), October 2001.

[2] C. Basdogan, C. Ho, M. A. Shrinivasan, and M. Slater. An experimental study on the role of touch in shared virtual environments. *ACM Transactions on Computer Human Interaction*, 7(4):443–460, December 2000.

[3] B. Baxter, V. Scheib, M. C. Lin, and D. Manocha. Dab: Interactive haptic painting with 3D virtual brushes. In *ACM SIGGRAPH Computer Graphics and Interactive Techniques*, pages 461–468, Los Angeles, California, 2001.

[4] F. P. Brooks, M. Ouh-Young, J. J. Batter, and P. J. Kilpatrick. Project GROPE – haptic displays for scientific visualization. In *ACM SIGGRAPH Computer Graphics and Interactive Techniques*, volume 24 (4), pages 177–185, August 1990.

[5] G. C. Burdea. *Force and Touch Feedback for Virtual Reality*. John Wiley and Sons, Inc., 1996.

[6] P. Buttolo, R. Oboe, and B. Hannaford. Architectures for shared haptic virtual environments. *Special Issue of Computers and Graphics*, 21:421–429, 1997.

[7] M. Capps, P. McDowell, and M. Zyda. A future for entertainment-defense research collaboration. *IEEE Computer Graphics and Applications*, 21(1):37–43, 2001.

[8] W. R. Ferrell. Delayed force feedback. *Human Factors*, 8:449–455, 1966.

[9] M. Foskey, M. A. Otaduy, and M. C. Lin. ArtNova: Touch-enabled 3D model design. In *IEEE Virtual Reality Conference*, pages 119–126, 2002.

[10] D. G. Giancoli. *General Physics (Chapter Nine)*, volume 1. Prentice-Hall, 1984.

[11] C. Gunn, M. Hutchins, D. Stevenson, and M. Adcock. Using collaborative haptics in remote surgical training. In *worldHAPTICS First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 481–482, Pisa, Italy, March 2005. IEEE Computer Society.

[12] G. S. Guthart and K. J. Salisbury. Intuitive telesurgery system: Overview and application. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 618–621, 2000.

[13] C. Gutwin. The effects of network delays on group work in real-time groupware. In *European Conference on CSCW*, pages 299–318, Bonn, 2001.

[14] J. P. Hespanha, M. McLaughlin, G. S. Sukhatme, M. Akbarian, R. Garg, and W. Zhu. Haptic collaboration over the internet. In *The Fifth PHANTOM Users Group Workshop*, 2000.

[15] K. Hikichi, H. Morino, and Y. Yasuda. The evaluation of adaptation control for haptic collaboration over the internet. In *IEEE Communications Quality and Reliability (CQR)*, 2002.

[16] T. C. Hudson, A. T. Helser, D. H. Sonnenwald, and M. C. Whitton. Managing collaboration in the nanomanipulator. *Presence: Teleoperators and Virtual Environments*, 13(2):193–210, April 2004.

[17] D. R. Jefferson. Virtual time. *ACM Transactions on Programming Languages and Systems*, 7(3):404–425, 1985.

[18] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, 82(Series D):35–45, March 1960.

[19] J. Kim, H. Kim, B. K. Tay, M. Muniyandi, M. A. Srinivasan, J. Jordan, J. Mortensen, M. Oliveira, and M. Slater. Transatlantic touch: A study of haptic collaboration over long distance. *Presence: Teleoperator and Virtual Environments*, 13(3):328–337, 2004.

[20] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, July 1978.

[21] M. R. Macedonia and M. J. Zyda. A taxonomy for networked virtual environments. *IEEE MultiMedia*, 4(1):48–56, 1997.

[22] W. R. Mark, S. C. Randolph, M. Finch, J. M. Van Verth, and R. M. Taylor, II. Adding force feedback to graphics systems: Issues and solutions. In *ACM SIGGRAPH Computer Graphics and Interactive Techniques*, pages 447–452, New Orleans, Louisiana, 1996.

[23] J. Marsh. *A Software Architecture for Interactive Multiuser Visualisation*. PhD thesis, The University of Manchester, 2002.

[24] J. Marsh, M. Glencross, S. Pettiffer, and R. J. Hubbold. A robust network architecture supporting rich behaviour in collaborative interactive applications. *To appear IEEE Transactions on Visualization and Computer Graphics TVCG*, 2006.

[25] M. Mauve. Consistency in replicated continuous interactive media. In *ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 181–190, Philadelphia, PA, December 2000.

[26] M. Mauve, J. Vogel, V. Hilt, and W. Effelsberg. Local-lag and timewarp: Providing consistency for replicated continuous applications. *IEEE Transactions on Multimedia*, 6(1):47–57, February 2004.

[27] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *ACM SIGGRAPH Computer Graphics and Interactive Techniques*, pages 401–408, Los Angeles, California, 1999.

[28] A. E. Michotte. *The Perception of Causality*. Basic Books, 1963.

[29] M. Minsky, O. Ming, O. Steele, F. P. Brooks, and M. Behensky. Feeling and seeing: Issues in force display. In *Symposium on Interactive 3D Graphics*, pages 235–241, Snowbird, Utah, United States, 1990.

[30] K. S. Park and R. V. Kenyon. Effects of network characteristics on human performance in a collaborative virtual environment. In *IEEE Virtual Reality*, pages 104–111. IEEE Computer Society, 1999.

[31] S. Pettifer, J. Cook, J. Marsh, and A. West. DEVA3: Architecture for a large-scale distributed virtual reality system. In *ACM Symposium on Virtual Reality Software and Technology*, pages 33–40, Seoul, Korea, 2000.

[32] E. Sallnäs, K. Rassmus-Groehn, and C. Sjoeström. Supporting presence in collaborative environments by haptic force feedback. *ACM Transactions on Computer-Human Interaction*, 7(4):461–476, 2000.

[33] G. Sankaranarayanan and B. Hannaford. Virtual coupling schemes for position coherency in networked haptic environments. In *In Proceedings of the BioRob Conference*, Pisa, Italy, February 2006.

[34] X. Shen, F. Bogsanyi, L. Ni, and N. D. Georganas. A heterogeneous scalable architecture for collaborative haptics environments. In *IEEE International Workshop on Haptic Audio and Visual Environments (HAVE)*, pages 113–118, September 2003.

[35] S. Shirmohammadi and N. D. Georganas. An end-to-end communication architecture for collaborative virtual environments. *Computer Networks*, 35(2-3):351–367, 2001.

[36] S. Singhal and M. Zyda. *Networked Virtual Environments Design and Implementation*. ACM Press SIGGRAPH Series, Addison Wesley, 1999.

[37] R. M. Taylor II, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser. VRPN: A device-independent, network-transparent VR peripheral system. In *VRST*, pages 55–61, Banff, Alberta, Canada, November 2001.