Eighth Eurographics Workshop on Virtual Environments (2002)
S. Müller, W. Stürzlinger (Editors)

Collaborative stretcher carrying: a case study

Roger J. Hubbold

Advanced Interfaces Group, Department of Computer Science, University of Manchester. (roger@cs.man.ac.uk)

Abstract
This paper describes a simulation of a collaborative task in a shared virtual environment – two users carrying a shared object (a stretcher) in a complex chemical plant. The implementation includes a haptic interface for each user, so that forces transmitted through the stretcher from one user to the other can be experienced. Preliminary experiments show that the addition of haptic feedback significantly enhances the sense of sharing and each user's perception of the actions of the other user. The implementation is described, and some conclusions about the value of haptics, and plans for future work are given.

Keywords:
Collaborative virtual environments, shared virtual environments, haptics, force feedback.

1 Introduction

This paper describes a prototype implementation of a complex task in a shared, collaborative virtual environment – a simulation of two people carrying a stretcher. The environment is a detailed model of a real-world chemical processing plant, with several floor levels connected by stairways, and obstacles in the form of pipework, vessels, pumps, valves, other machinery, and handrails. The task is to determine whether the two carriers can manoeuvre the stretcher past these obstacles to get from the location of a hypothetical accident to an evacuation point, such as a helicopter pad.

In a previous paper [1] we described first attempts at building such a simulation. However, in that work the user interface was controlled by a single user, and the focus was on techniques for locomotion, including ascending and descending stairs, managing the constraints between the carriers and the stretcher, and collision detection.

This paper describes the extension of these techniques to two users, each with his own view of, and interaction with, the environment. As well as exploring methods for each user to locomote and manipulate one end of the stretcher, a preliminary investigation was made of the role of haptic feedback, so that, as well as being able to see one another, each carrier could feel the pushing and pulling forces exerted by the other.

The trial implementation was constructed while the author was visiting the University of North Carolina at Chapel Hill, from May to November 2000, and utilised their extensive laboratory facilities, including a desktop PHANToM, and the Argonne Arm – an 8-foot long, six degree of freedom force feedback device. The software has also been implemented on our equipment in Manchester, using Polhemus trackers to control 3D motion. Preliminary results suggest that the additional cues provided by force feedback lead to a richer, qualitatively different interface, compared to one with no forces. We are planning further experiments to quantify these effects.

The remainder of this paper is organised as follows. First, a little more background and related work are described, followed by details of the orginal single user version of the software. Next, the extension to two collaborating users is described, including the experimental set-up, the implementation, and the addition of haptic feedback. Finally, some observations are made about the resulting simulation, with pointers for future work.

2 Background and related work

Early work on haptics for graphical interfaces grew out of the use of remote manipulators for handling hazardous materials, leading to widespread interest in telerobotics. Here, the primary goal was to enable an operator to feel remote objects being manipulated by a robot arm. Brooks *et al* were the first to realise that the remote manipulator could be replaced by a simulation, and the pioneering GROPE project [9] used the Argonne Arm to explore forces during molecular docking. Interest in haptics in virtual environments has grown considerably, with the advent of small-scale force feedback devices, such as the desktop PHANToM, manufactured by Sensable Technologies (www.sensable.com). Attention has focused mainly on using such devices to simulate collisions and to "feel" the surfaces of objects [11]. An example where such techniques have been used in a collaborative setting is the Nanomanipulator project, in which scientists share a visualization of data captured with a scanning electron microscope; although haptic feedback is provided, it is limited to local interaction with the model [10, 8].

Little seems to have been reported in the VR literature on manipulation of shared objects, with or without haptics. Basdogan et al [4] reported an experiment in which two users attempted a hoop and wire experiment, where both hold the hoop. In a current project Slater *et al* are extending such ideas to collaboration over larger distances (Collaboration in Tele-Immersive Environments [5]). Sallnas *et al* [13] describe an experiment in which users can collaboratively manipulate cubes using PHANToM devices. The task is more restricted than that described here, as the cubes were limited to translational motion and a desktop environment was employed. However, their results show a significant benefit from adding haptic cues to a manipulation task. Oakley *et al* also describe using haptics for collaborative interaction [12], but do not consider immersive environments, or manipulation of shared objects.

The work reported here describes first attempts at exploring the use of haptic feedback for the complex task of carrying a stretcher. Our goal is to explore this for geographically distributed users, using wide area networks, but this paper focuses an implementation which operates on local machines where the delays are negligibly small.

3 The original, single-user simulation

Figure 1 illustrates the original stretcher simulation, described in detail in [1]. Two simulated users, represented by avatars, carry a third person on a stretcher. The two carriers and the stretcher are linked by constraints, which govern how high or low they can raise their arms, and the maximum permitted distance between their bodies and the stretcher ends. The operator drives the program by 'steering' the stretcher, to try to guide it past obstacles. This can be done with either a 2D mouse and modifier keys, or with a 3D mouse and Polhemus Long Ranger tracking system.

As the stretcher is manoeuvred, each carrier attempts to remain at a 'sweet spot' – an ideal position relative to his stretcher-end. Experiments with a real stretcher showed that this position is quite important. It is very difficult – almost impossible – to hold a heavy stretcher away from one's body.

An interesting problem arises when negotiating stairways: how to get each carrier to move up and down from step to step. This is resolved by using forces to control motions, rather than absolute positions. User input from the input device (a 2D or 3D mouse) is transformed into a force acting upon the stretcher, causing it to move. These movements are transmitted to the carriers, who attempt to remain at their respective sweet spots. A foot-probing technique is applied so that when low obstacles, such as stair treads, are encountered, the carrier will step up. Simulated gravity is used to drop from tread to tread when moving down stairs. To help avoid collisions, objects in the environment emit forces whose function is to guide the carriers around obstacles. A detailed explanation of these techniques can be found in [3]. Full collision detection is also activated, for the stretcher and all three avatars, to prevent interpenetration of solid objects in cases where the forces are insufficient to prevent collisions, such as when the carriers are moving very fast.

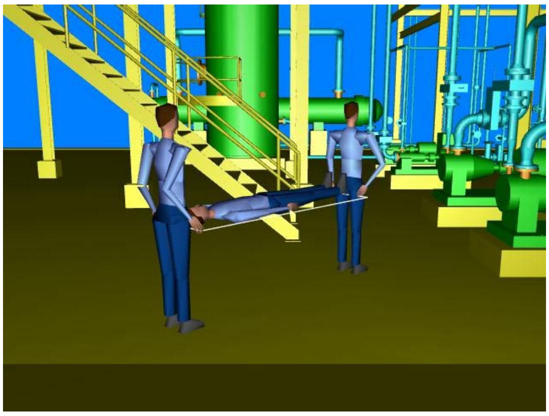

Figure 1: Stretcher simulation

Computing the final motion of the stretcher and carriers entails checking that none of the positioning constraints have been violated. Where a violation is detected movement of the stretcher is deemed invalid and is prevented.

Several interesting observations were made with this single-user version of the program. First, the program was useful for checking valid paths – finding out whether a given route was feasible. With constraints on lifting heights set appropriately, the program is a good predictor of places where negotiating a route is difficult. For example, at sharp turns on stairways it is necessary to raise the stretcher quite high in order to clear the handrails. This is actually impossible to do if both carriers face forwards – the normal situation for carrying on level surfaces, as seen in Figure 1 – because the front carrier cannot raise the stretcher high enough behind his back. Only by having both carriers face inwards (towards each other) can such obstacles be overcome.

Experience with carrying a real stretcher confirmed this was indeed the case. Second, it is very difficult to manoeuvre the stretcher single-handedly with a 3D mouse. In our initial attempt, the user manipulated the position and orientation of the centre of the stretcher. This made it very difficult to correctly simulate the movements of the two independent carriers. It was clear that for accurate simulation, independent control would be required for each end, and this is what the two-user implementation described here provides.

4 The two-user study

The task of simulating two independent users carrying a common object is a particularly challenging one for collaborative virtual environment research. Problems to be overcome include finding mechanisms for locomotion, techniques for propagating motions between the two carriers, and dealing with the effects of network latency and jitter. In this paper we ignore latency and jitter, which are the subject of on-going research, and focus on the basic simulation and interaction control.

4.1 Laboratory environment

The study was carried out using the facilities of the Graphics Laboratory at UNC Chapel Hill. The software was implemented on their SGI RealityMonster graphics supercomputer, using the MAVERIK system [2]. Graphical output was directed through two projection screens, one for each user.

Input at one end was provided through the Argonne Arm – a six degree of freedom, eight-foot long, force feedback device, seen in Figure 2. This cannot simulate the weight of a real stretcher, but its output of up to 8 lbs force is sufficient to give a good feel of the pushing and pulling forces transmitted between users at each end of the stretcher. Full scale surrogate handles were attached to the end of the Arm to give a more realistic feel, similar to holding real stretcher handles. These can be seen more clearly in Figure 3. At the other end a standard desktop PHANToM was used, with small surrogate handles attached. The haptic devices were attached to server computers connected to the departmental LAN.

Inter-program communication was implemented using the UNC Virtual Reality Peripherals Network (VRPN) software [6, 7], as was communication with the haptic servers. VRPN, which is freely available, provides a number of functions for haptic device control; use of these will be detailed later.

4.2 The implementation

The original simulation was modified so that two independent copies of the program could be run in a master-slave relationship. Each provides an independent interface for its user, and a shared set of data files ensures that the simulation commences with the users positioned sensibly with the respect to the stretcher, at a known position in the model. The role of the *master* is to receive inputs from both users, compute the allowable

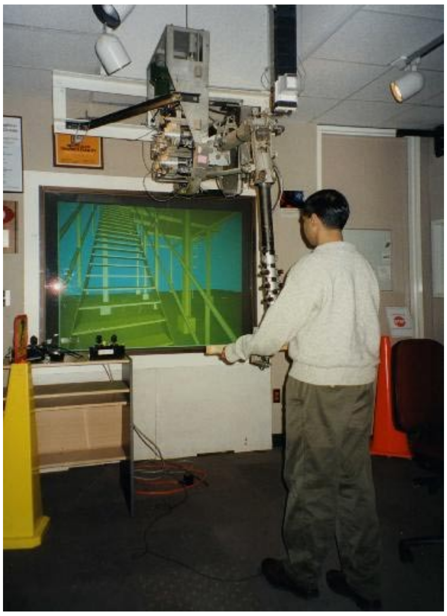

Figure 2: Projection screen and Argonne Arm

Figure 3: Argonne Arm with surrogate handles

movement of the stretcher and carriers, update its own view of the environment, and send the resulting positions back to the slave for it to update the second view. This communication was implemented via the VRPN libraries, which are based on TCP-IP and UDP, using sockets.

During the study, graduate student volunteers and staff carried a real stretcher, loaded with 100 pounds of sandbags, around the rooms, corridors, stairways and fire escapes of Sitterson Hall at UNC. The exercise was video-taped for subsequent analysis. It was hard work! In a genuine emergency, a victim might weigh twice as much as the sandbags. Even so, manoeuvring the stretcher was difficult, and demonstrated that it was impossible to hold it at any distance from one's body. Much of the time, the carriers' arms would be held straight at the sides of the body, with up and down movement limited to situations where it was strictly necessary to avoid obstacles. The only practical way to control movement was to walk in the desired direction, and to raise or lower the stretcher when necessary.

To simulate this in the VE, the interface allowed the user to move the haptic input device in the intended direction of motion: forwards, backwards, or sideways, or some combination. Vertical movements were used directly to control the height of the respective stretcher end.

Movements of the input device were mapped to forces in the corresponding directions. Computing the allowable motion of the stretcher involves:

1. Computing forces from surrounding objects and combining them with the user's inputs.
2. Calculating where the carriers would move under the influence of these combined forces.
3. Using these new positions of the carriers to compute the distance of each from his sweet spot, and mapping these to additional pushing or pulling forces transmitted through the stretcher. These forces are then added to those acting on the carriers.
4. Moving the carriers under the influence of the forces; this includes computing and making vertical movements (e.g. on stairs).
5. Recomputing the stretcher position from the new carrier positions.
6. Checking for collisions for both the carriers and the stretcher, and preventing movement if a collision is detected.

Although the above procedures may seem elaborate and computationally expensive, they are executed efficiently using optimised spatial searching capabilities and intersection functions. Generally, collisions of the carriers with their environment are avoided because of the guiding effect of the force fields. The stretcher, however, is not subject to these forces, and it becomes stuck when a collision is detected.

4.3 Introducing haptic feedback

Because all the motions are controlled and computed using forces, applied to the stretcher and to the carriers, introducing haptic feedback is relatively easy. The major role of such feedback is to allow each user to feel pushing and pulling forces transmitted through the stretcher. These forces are already computed during motion control, so all that remains is to output them through the haptic interface.

VRPN provides a convenient interface for programming the forces. The haptic devices are connected to server machines – either Windows or Linux PCs – which execute a tight control loop. This allows them to meet the 1KHz target update rate for proper haptic simulation of contact forces. Force parameters are computed by the master copy of the program in its normal rendering loop. This typically executes at 15Hz for the models used in the study. (The target frame rate can be set, and the rendering software adapts automatically to maintain this.)

This decoupling of the 1KHz haptic update rate from the rendering loop is important. For forces through the stretcher a Hooke's Law spring model is employed, and updating the spring stiffness coefficient at 15Hz is quite adequate. However, a different technique is needed to correctly simulate surface contact between the stretcher and an obstacle. When a collision is detected, a surface normal vector is computed at the point of contact. The corresponding plane equation is transmitted to the VRPN server, which then simulates the surface contact in its 1KHz loop. Thus, firm surface collisions can be felt. This technique allows the stretcher to slide along the plane of contact, but not to pass through it. Pulling the stretcher away from the surface breaks the collision condition, so that it can again be moved freely.

5 Discussion

Only limited time was available during the original study for experimentation with the working system. However, a number of qualitative observations can be made, and these are discussed below.

First, as indicated previously, the virtual environment is a good predictor of situations in which manoeuvring the stretcher past obstacles is difficult, confirmed by carrying a real stretcher. The purpose of the study was not so much to create a program for training stretcher carriers, as to explore the limitations of virtual environments, in the hope that this will give insights into improving subsequent simulations. Stretcher carrying is a real-world task, and therefore offers an opportunity for comparing real-world task performance with that in a VE.

Second, and most strikingly, the addition of haptic cues adds a completely new dimension to the experience. Without these, the only cues are visual, plus a small audible cue when collisions occur. Without haptic feedback, it is extremely difficult to manoeuvre – even for something as simple as following a curved path in order to thread one’s way past obstacles such as columns. The reason is that each carrier overcompensates for the other. When carrying the rear of the stretcher there is a natural tendency to attempt to steer the stretcher, and the person in front, by making exagerated sideways movements. When at the front this is disconcerting, because one finds oneself being directed against one’s will. A typical reaction in such situations is to overcompensate, resulting in an oscillatory behaviour. The reverse situation also holds: at the rear, one’s attempts at steering are thwarted by the person at the front counteracting one’s own actions. So, for example, when following an arc to the right, there is a tendency for both carriers to move to the right. The correct course of action would be for the rear carrier to follow a similar path to the one in front, but this does not seem to happen in practice.

An interesting question is whether such behaviour is caused, or at least compounded, by the limited field of view available on the projection screen, and the consequent lack of awareness of objects at one’s side. Evidence from other studies [3] suggests that this is very likely, so it would be interesting to try the same task in a CAVE, or using a head-mounted display. We plan to do this in future.

When haptic cues are added, each carrier can feel, as well as see, the actions of the other. When carrying the real stretcher, we experimented by asking the carriers to close their eyes and report what they could feel. It was observed that forces between the two carriers are very subtle. Thus, although a stretcher is a heavy object, it is not necessary to have very high forces to have an indication of the motion of the other person.

In the simulation, quite subtle pushing and pulling forces are experienced as the carriers move up and down stairs – the movement of the person at the other end can be felt quite clearly, much as one experiences in a real situation, such as carrying a piece of furniture on a stairway.

A completely different effect was observed as a result of the large difference of scale between the PHANToM and the Arm. The latter is eight feet long. Its operational height and range (as seen in Figure 3) make it ideally suited to simulating the task reported here. In contrast, the PHANToM has a very small working volume. It is therefore difficult to select a suitable mapping (scaling) from the PHANToM to the real world. An aggressive user manipulating the PHANToM could very easily throw off balance the person holding the Arm. This happened despite the simulation permitting differential scaling of the inputs. It is easy to speculate that if both users have similar devices then this problem would be solved, but the unequal scales certainly warrant some further study.

Haptic feedback from collisions was very effective, particularly with the limited field of view. It meant that collisions with obstacles out of view could be sensed unambiguously, making control much easier.

6 Future work

Several other experiments could be conducted with the existing implementation. Specifically, using a CAVE or HMD (both of which are available to us) would allow us to explore further some of the effects of a limited field of view. It would also be possible to devise controlled tasks to compare performance in a measurable way – for example, by measuring route completion times, and numbers of collisions – with and without haptic feedback, and we plan to do this.

Even with haptic feedback, some oscillatory behaviour was observed. In part, this is believed to be due to the unequal scales of the two input devices. Nonetheless, it would be useful to add an inertial component to the stretcher, in effect simulating its heavy mass; at present, the virtual stretcher is weightless, which is clearly unrealistic. Adding inertia should remove the high frequency components which cause oscillations.

The simulation has also been implemented in our laboratory in Manchester, where it runs between different computers, but without haptic feedback. This has demonstrated that an unconstrained 3D mouse is definitely not a good interface in the two-user case. We therefore plan to explore whole body movements to control motions.

Finally, it would be instructive to run experiments across a wide area network with inherent latency and jitter. This can be simulated, using a tool such NIST Net Tools [14], and we have plans to do this, as well as to conduct Internet trials.

Acknowledgements

The author would like to offer his warmest thanks to

Professor Fredrick P. Brooks Jr. and Professor Mary Whitton at the University of North Carolina at Chapel Hill. Their enthusiasm and keen interest was a real source of inspiration. Thanks must also go to many other members of staff and graduate students in the Computer Science Department at UNC who helped directly and indirectly: Samir Naik, Sharif Razzaque, Russell Taylor, David Harrison, Murray Anderegg, and to just about the whole of the rest of the department for making me so welcome. In Manchester, I would like to thank my colleagues in the Advanced Interfaces Group for their patience, and particularly Jon Cook for helping to resolve many small system problems.

References

1. Roger Hubbold and Martin Keates. Real-time simulation of a stretcher evacuation in a large-scale virtual environment. *Computer Graphics Forum*, 19(2), pages 123–134, June 2000.

2. Roger Hubbold, Jon Cook, Martin Keates, Simon Gibson, Toby Howard, Alan Murta, Adrian West and Steve Pettifer. GNU/MAVERIK: A micro-kernel for large-scale virtual environments. *Presence* 10(1), MIT Press, February 2001. (First appeared in *Proc. ACM Virtual Reality Software & Technology Conference*, pages 66–73, London, December 1999, ACM Press, New York, 1999. ISBN 1-58113-141-0.)

3. Dongbo Xiao and Roger Hubbold. Navigation guided by artificial force fields. In *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, volume 1, pages 179–186. ACM SIGCHI, Addison Wesley, April 1998. ISBN 0-201-30987-4.

4. Cagatay Basdogan, Chih-hao Ho, Mandayam A. Srinivasan and Mel Slater. An experimental study on the role of touch in shared virtual environments, *ACM Transactions on Computer-Human Interaction, 7 (4), Dec. 2000, pages 443–460.*

5. www.cs.ucl.ac.uk/research/vr/Projects/Internet2

6. http://www.cs.unc.edu/Research/vrpn/

7. William R. Mark, Scott C. Randolph, Mark Finch, James M. Van Verth and Pussel M. Taylor II. Adding force feedback to graphics systems: issues and solutions, *Proc. ACM SIGGRAPH '96*, pages 447–452.

8. Diane H. Sonnenwald, Ronald E. Bergquist, Kelly L. Maglaughlin, Eileen Kupstas-Soo and Mary C. Whitton. Designing to support collaborative scientific research across distances: the nanoManipulator environment, in *Collaborative Virtual Environments, Digital Places and Spaces for Interaction*, Elizabeth E. Churchill, David N. Snowdon and Alan J. Munro (Eds), Springer-Verlag, 2001, pages 202–224.

9. J. Batter and F. Brooks Jr., GROPE-I: A computer display to the sense of feel, *Proceedings of IFIP Congress*, 1971, pp.759–763.

10. Russell M. Taylor II, Warren Robinett, Vernon L. Chi, Frederick P. Brooks Jr., William V. Wright, R. Stanley Williams, Erik J. Snyder. The Nanomanipulator: A virtual reality interface for a scanning tunneling microscope, *Proc. SIGGRAPH'93, pp. 127–134.*

11. Grigore C. Burdea. *Force and Touch Feedback for Virtual Reality*, John Wiley and Sons, Inc., 1996.

12. Ian Oakley, Stephen Brewster and Philip Gray. Can you feel the force? An investigation of haptic collaboration in shared editors, in *Proc. Eurohaptics 2001*, University of Birmingham, UK, July 2001.

13. E. Sallnas, K. Rassmus-Grohn and C. Sjostrom. Supporting presence in collaborative virtual environments by haptic force feedback, ACM Transaction on CHI, 7(4), December 2000, 461–476.

14. http://snad.ncsl.nist.gov/itg/nistnet.