

Cooperative Object Manipulation in Immersive Virtual Environments: Framework and Techniques

Márcio S. Pinho

Instituto de Informática, UFRGS
Faculdade de Informática, PUCRS
Porto Alegre, RS, BRAZIL
pinho@inf.pucrs.br

Doug A. Bowman

Department of Computer Science
Virginia Polytechnic Institute and
State University
Blacksburg, Virginia, USA
bowman@vt.edu

Carla M.D.S. Freitas

Instituto de Informática, UFRGS
Porto Alegre, RS, BRAZIL
carla@inf.ufrgs.br

ABSTRACT

Cooperative manipulation refers to the simultaneous manipulation of a virtual object by multiple users in an immersive virtual environment. This paper describes a framework supporting the development of collaborative manipulation techniques, and example techniques we have tested within this framework. We describe the modeling of cooperative interaction techniques, methods of combining simultaneous user actions, and the awareness tools used to provide the necessary knowledge of partner activities during the cooperative interaction process. Our framework is based on a *Collaborative Metaphor* concept that defines rules to combine user interaction techniques. The combination is based on the separation of degrees of freedom between two users. Finally, we present novel combinations of two interaction techniques (Simple Virtual Hand and Ray-casting).

Categories and Subject Descriptors

I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - *Virtual reality*.

I.3.6 [Computer Graphics] Methodology and Techniques - *Interaction Techniques*

H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces - *Synchronous interaction*.

GENERAL TERMS

Algorithms, Human Factors.

Keywords

Interaction in Virtual Environments, Cooperative Interaction.

1. INTRODUCTION

Some object manipulation tasks in immersive virtual environments (VEs) are difficult for a single user to perform with typical 3D interaction techniques. One example is when a user, using a *Ray-casting* technique [15] has to place an object far from its current position.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

VRST'02, November 11-13, 2002, Hong Kong

Copyright 2002 ACM 1-58113-530-0/02/0011...\$5.00

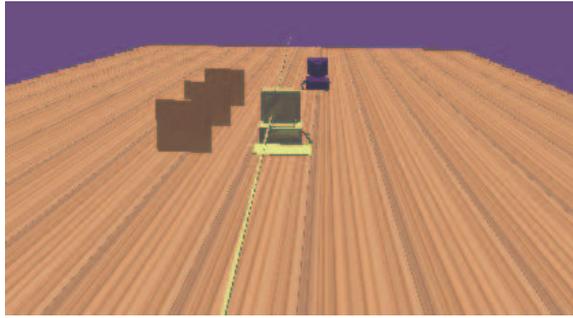
In Figure 1a, we can observe that placing the computer (in the center of the image) between the “walls” (on the left) can be difficult depending on the distance between the user and these structures. In this case, a second user, standing next to these walls, and able to slide the object along the ray, could easily help the first user position the object.

Another example is the manipulation of an object through a narrow opening. This problem can be illustrated by the situation where it is necessary to move a couch through a door or a window (Figure 1b). In this case, if we place a user on each side of the door, the task can be performed more easily because they can both advise each other and perform cooperative movements they are not able to perform alone.

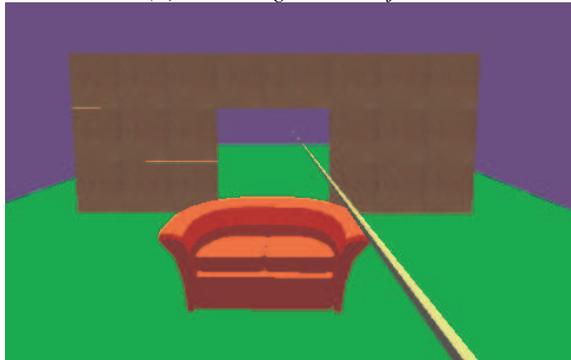
Besides task-related problems, limitations inherent to the interaction technique can also decrease user performance. Some techniques do not allow performing some movements. In Figure 1c, for example, if the task is to orient the computer on the right so that it matches the three on the left, the user will have difficulty because ray-casting does not afford rotation around the vertical axis. In this case, if a second user can control the object's orientation, the task becomes much simpler.

Some problems of this type can be addressed without cooperative manipulation; that is, by simply allowing one user to advise his partner. For this situation existing architectures (described in section 2) are sufficient to support the collaboration. If, however, it is necessary or desired that more than one user be able to act at the same time on the same object, new interaction techniques and support tools need be developed.

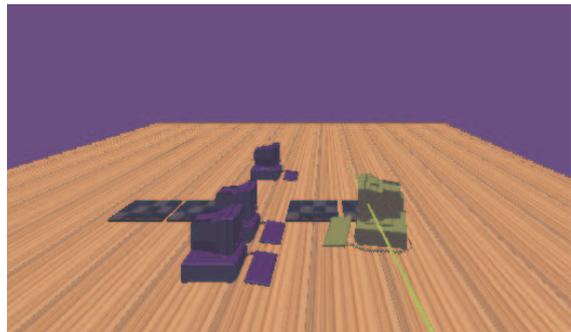
Our work is focused on these specific problems: how to support cooperative interaction and how to modify existing interaction techniques to fulfill the needs of cooperative tasks. In this paper we describe the modeling of cooperative interaction techniques, methods of combining simultaneous user actions, and the awareness tools used to provide the necessary comprehension of partner activities during the cooperative manipulation process. To support the development of such techniques, we have built a framework that allows us to explore various ways to separate degrees of freedom (DOFs) and to provide awareness for two users performing a cooperative manipulation task.



(a) Positioning distant objects



(b) Reduced moving space



(c) Limitations of existing techniques

Figure 1 –Tasks that would benefit from cooperative manipulation

Our goal is the development of *usable* and *useful* cooperative manipulation techniques. Designing such techniques requires us to consider the following issues:

- **Awareness:** Showing to one user the actions his partner is performing;
- **Evolution:** Building cooperative techniques as natural extensions of existing single-user techniques, in order to take advantage of prior user knowledge;
- **Transition:** Moving between a single-user and a collaborative task in a seamless and natural way without any sort of explicit command or discontinuity in the interactive process, preserving the sense of immersion in the VE;
- **Reuse:** Facilitating the implementation of new cooperative interaction techniques, allowing the reuse of existing code.

We base our technique design efforts on the concept of a *Collaborative Metaphor*: a set of rules that define how to combine individual interaction techniques in order to allow multiple users to manipulate the same object at the same time.

This paper is organized as follows. First, we present related work. Then, we present some important definitions and describe the main characteristics of the developed framework. Finally, we describe the cooperative manipulation techniques we have developed and preliminary results from usability studies of these techniques.

2. RELATED WORK

The use of collaborative virtual environments (CVEs) has become more and more popular due to the cheaper, faster and more reliable facilities provided by personal computer systems and network resources. Our work is in the general field of CVEs, but simultaneous manipulation of an object by two users is beyond the scope of most CVEs; thus our use of the term *cooperative* rather than simply collaborative.

Much CVE research is devoted to the development of support tools and the minimization of network traffic. Some examples include AVOCADO [25], Bamboo [27], DIVE [8], and MASSIVE [9].

Another important issue that is commonly addressed in this field is user-to-user communication (also known as computer-mediated-communication). Researchers in this area try to enhance and evaluate the communication between users. Viullème and Thalmann [26], for example, describe a system based on the VLNET framework in which the user can select a gesture and a facial expression from a set of options presented on a screen. After the selection, the choices are incorporated into an avatar that represents the user inside the CVE. In Spin [7] the aim is to create a kind of “conference table”. This is built on a computer screen as a set of panels placed side by side around a circular table. The panels can be rotated as if they were around the user’s head. To each panel, a user can associate another user or an application. To select one application to be executed or other user to talk with, one has simply to rotate the panels until the desired choice is in the middle of the screen. Other research has addressed the evaluation of user-to-user communication ([21] and [22]).

Collaborative augmented reality (AR) systems [1] often include object manipulation. In such systems the users are physically located in the same space and are able to see each other using see-through glasses. Virtual objects are superimposed on real-world objects. This setting can provide the same type of collaborative information that people have in face-to-face interaction such as communication by object manipulation and gesture. Such a setup has been used in games like AR² Hockey [18], in scientific visualization systems (Studierstube), in discussion support systems (Shared Space [2]; Virtual Round Table [5]) and in object modelers like SeamlessDesign [11]. None of these systems, however, allow cooperative manipulation.

Although some research addresses interaction in CVEs, in most of them cooperative manipulation is not possible. Usually, when one user selects an object for manipulation, the other cannot participate in the same procedure. In fact, most existing research specifically forbids this simultaneity. In the work of Li et al. [12], for example, many users can manipulate the same object at the same time, but the object must be modeled with NURBS surfaces and when one user selects the object he actually gets exclusive

access to the shape, position and orientation of only one patch. The ICOME system [20], a geometric modeling framework, organizes the object in a hierarchical way allowing users to act simultaneously on different hierarchical structural levels of the same object.

We have found only two examples of actual cooperative manipulation in VEs. Noma [17] presents a study of cooperative manipulation where two users manipulate an object using force feedback devices. These devices are used to constrain a user's hand movements by simulating the forces they would feel based on the partner's actions. Margery [14] presents an architecture to support cooperative interaction based on physical laws. In this work the users, using a VRML browser, can move an object that is controlled by a simulator. This simulator, replicated on each node, is able to receive simultaneous movement commands, combine them, and generate the resultant movement. These commands are expressed by physical entities such as direction vectors, application points on the object, intensity, etc. To produce the same movement at all sites, every simulator must be fed the same data in the same order. To guarantee this, the architecture has an ordering sub-system.

3. SOFTWARE FRAMEWORK

In the systems presented in section 2 (with the exceptions of [14] and [17]) at each moment the object (or part of it) will receive only one action selected among all users' actions. In our work, instead of *choosing* between two actions that come from different users we *combine* them so as to allow the cooperative manipulation of an object inside a VE. To do so, we use the concept of a *Collaborative Metaphor*. This metaphor is a set of rules that addresses the following issues:

- What to do in each phase of the interaction process when the users are collaborating (section 3.2);
- How to combine two interaction techniques (section 3.3);
- How to show to one user what his partner is doing (section 3.4).

The main difference between our technique and the methodology presented by Margery [14] is that instead of using physical laws to combine user actions we focus on *combining interaction techniques*. In other words, we take existing techniques with which users are familiar and from them we build cooperative ways to manipulate an object. "Magic" interaction techniques such as HOMER [3] or Go-Go [19] can be more powerful than the simple use of physical movements. Moreover, we can use the users' previous knowledge about these single-user techniques to improve their performance.

To support this combination we have developed a software framework consisting of the following modules (Figure 2):

- **Graphics package:** renders the scene (section 3.1);
- **Object Database:** stores all the geometric data that represents the VE (section 3.1);
- **Interaction Technique Module:** interprets the user input based on the interaction technique rules (section 3.2);

- **Command Combiner:** combines the user actions and creates a new command to be applied to the object being manipulated (section 3.3);
- **Awareness Generator:** provides information about the partner and his activities inside the VE (section 3.4);
- **Message Handling and Network Support:** builds, sends, receives, and interprets messages exchanged with the partner (section 3.5).

The system is currently designed to connect two machines, or *nodes*, each of which is running a copy of the same CVE application. The tracking system is connected just to one node, but the tracking information is sent to the other node as well. In the following sections we describe how each framework module supports the Collaborative Metaphor.

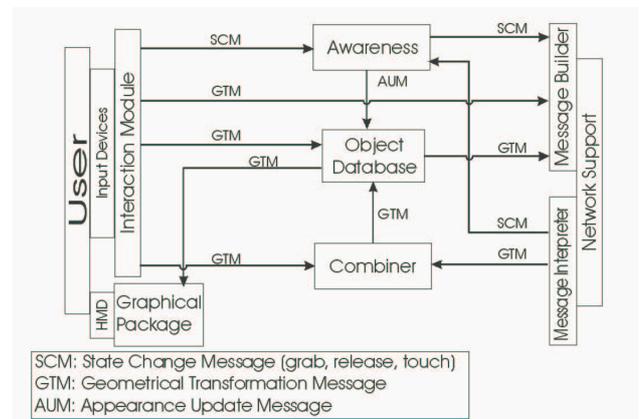


Figure 2 - Cooperative manipulation framework

3.1 Graphics Package and Object Database

The VE is described by a set of geometric objects rendered using the Simple Virtual Environment Library (SVE) [10]. The geometric transformations to be applied to the objects in each frame are built based on the users' interaction commands and on the transformations pre-defined in the system code. In each user's version of the VE, the partner is represented by simple head and hand avatars that reflect the movements of the head and hand trackers.

3.2 Interaction Technique Module

In our framework, interaction with virtual objects is performed through a tracker and button device that the user holds in one hand – we call this the *pointer*. The position and the orientation of this pointer are obtained from the tracking system. The role played by the pointer in the interaction process is defined by the interaction technique that is being used by each user. The *Interaction Technique Module* is responsible for translating the pointer movements and commands generated by a user into transformations to be applied to the virtual object.

As mentioned above, one important requirement of a cooperative interaction system is to combine interaction techniques naturally, giving the users the possibility to act individually or cooperatively with smooth transitions between these modes of interaction. To support smooth transitions we subdivided individual interaction techniques into simpler sub-components that can be easily modified and replaced without having to modify the entire implementation. To accomplish this goal we used Bowman's

model [4] in which a manipulation technique can be divided into four sub-components, as follows:

- **Selection technique:** the method of indicating an object to be manipulated;
- **Attachment technique:** how the object is attached to the user;
- **Position and Orientation Technique:** how the pointer movement affects the object position/orientation;
- **Release technique:** what happens when the user releases the object.

This subdivision allows the analysis of each step of the interaction process separately, which facilitates combining techniques for cooperative manipulation. Moreover, the use of this kind of organization facilitates the construction of new interaction techniques from existing components. Table 1 shows how our framework deals with each of the components for both single-user and cooperative techniques.

Our current system uses two individual manipulation techniques: *Simple Virtual Hand* and *Ray-casting* [15]. The former maps pointer motion directly to the motion of the object. The latter uses a ray emanating from the pointer to select a distant object and allows the user to manipulate it by attaching it to the ray. An extension to the basic ray-casting technique, called “reeling,” allows the user to slide the object along the ray [3]. The Simple Virtual Hand technique allows the user to easily control all six DOFs, but only within an arm’s reach. The Ray-casting technique allows manipulation at-a-distance, but it also makes it difficult to separately control the translational and rotational DOFs.

3.3 Command Combiner

The *Command Combiner* is responsible for combining the transformations generated by both users through the interaction techniques. Based on the Collaborative Metaphor, it generates a new transformation to be applied to the object.

In our work, these combination rules are based on two possible approaches. The first one *separates* the technique’s DOFs between the partners. Using this approach each user is able to manipulate only some of the technique’s DOFs. For example, one user can move the object on the horizontal plane and the other can adjust its height. Another example is the case where one user (using the Ray-casting technique), controls the object position, and the other one (using the Simple Virtual Hand) controls the orientation and can slide the object along the ray. Currently, we specify the DOFs each user will control in a configuration file, before the beginning of the session. More details on how these DOFs are being used to combine interaction techniques are presented in section 4.

TECHNIQUE COMPONENT	SINGLE USER	COOPERATIVE
Selection	Highlight the object	Highlight and send a message to the Awareness Module
Attachment	Attach the object to the pointer	Send a message to the Combiner Module
Position	Update the object position	Send a position to the Combiner Module
Release	Un-highlight the object and detach the object from the pointer	Un-highlight and send a message to the Awareness Module

Table 1 – System actions for technique components in single-user and cooperative modes

The second approach *composes* the actions of the two users and generates a new transformation. In this approach we are trying to find ways to combine the 6 DOF transformations generated by the users and apply the result to the object. If we have, for example, two users using Ray-casting, and both try to slide the object along their respective rays at the same time, we can take these displacements as direction vectors, add them and apply the resultant vector to the object.

3.4 Awareness Generator

The *Awareness Generator* is responsible for showing to one user the actions and object transformations performed by his partner. In the physical world, when we work cooperatively on an object, the forces applied to the object are transferred to the other user through the object itself. In VEs without force feedback devices, this transmission is not feasible. So, we need alternative ways to convey this information from one user to the other. Curry [6] calls this the *action metaphor*.

In our system we subdivide the awareness information into three categories: user information, interaction information and object state information. The following sections describe these three types of information.

3.4.1 User Information

This type of information is generated from the user position and orientation and is used to produce understanding and awareness of the other user. We use a 3D model of a head-mounted display (HMD) that is displayed in the VE at the current position/orientation of the other user’s head. This information allows the user to know where his partner is (locus) and what he is looking at (focus). It also provides information about positioning relative to the other user (i.e. where is left, right, front, behind, up and down, with respect to the partner). In the future, we intend to evaluate the effectiveness (for the collaboration process) of using a whole body model for the partner’s representation instead of just an HMD model.

3.4.2 Interaction Information

The geometric object that represents the pointer position in the VE depends on the interaction technique being used. For example, if one is using the Simple Virtual Hand technique, the system should generate the necessary visual information in such a way that the other user can understand that his partner has a hand, where it is, what is its orientation and whether it is holding an object or not. On the other hand, if the collaborator is using Ray-casting, it is more important to show a representation of the ray, its orientation and position within the VE.

The pointer representation provides an understanding of which technique one's partner is using and, more importantly, which functions he can perform during the interaction process. In other words, it represents the partner's interaction capabilities.

3.4.3 Object State Information

During cooperative manipulation, it is also important for the users to understand which object is being manipulated and by which user. There are, in this context, three possible states that show the relationship between an object and a user: *free*, *touched* and *grabbed*.

The free state means that there is no interaction between the user and the object. The touched state means that the user is ready to grab the object but has not yet done so. Once the object is selected, it passes to the grabbed state. This state means that the user is interacting with the object.

In a cooperative manipulation system, of course, these three states do not represent all the possible states for an object. We can have situations where one user is touching the same object that the other one is grabbing, or where both are touching the same object, among other situations. Since each object is in one of the three states with respect to each user, there are actually nine different states we need to consider (Table 2).

For each of these states the Awareness Generator module has to provide feedback to the users. In our system we are using colors and textures to inform users of the correct object state.

The colors and textures we use correspond to the colors and textures of the users' avatars. The right column of Table 2 shows the feedback we provide in each of the nine states (Note that a "light" version of the color/texture is used when the user is simply touching (not grabbing) the object).

3.5 Communication system

To support the communication between two collaborating nodes, we use a simple message protocol built on TCP/IP. At the beginning of each frame messages are received, and at the end they are sent. Messages fit into three categories, based on their semantics: *Position Information*, *Commands* and *Tracker Data*. The Position messages contain the position and orientation of the user and his pointer at each frame. If there is an object that is being manipulated these messages also carry information about it. The Commands inform one user of the occurrence of an event on the other node.

These events correspond to object state transitions:

User A (Texture)	User B (Color)	Object Color/Texture
Free	Free	Object original color
Free	Touched	User B (light) color
Free	Grabbed	User B color
Touched	Free	User A (light) texture
Touched	Touched	User A (light) texture + User B (light) color
Touched	Grabbed	User A (light) texture + User B color
Grabbed	Free	User A texture
Grabbed	Touched	User A texture + User B (light) color
Grabbed	Grabbed	User A texture + User B color

Table 2 - Possible object states and system feedback

- TOUCH: the user has touched an object;
- UNTOUCH: the user has ceased to touch an object;
- GRAB: the user has selected an object;
- RELEASE: the user has released an object.

These events, generated by a user A, will force some modifications in the state of user B's VE. These modifications are accomplished by the Awareness Module, and can be seen in section 0.

Depending on which interaction technique is being used, some other commands can be sent to the partner. Using Ray-casting, for example, we include *SLIDE FORWARD* and *SLIDE BACK* commands that move the object along the ray.

The Tracker Data messages are passed from the node that is reading the tracking device to the other one. The second node uses this information to update its user position and to perform the appropriate interaction based on the technique used locally.

4. COOPERATIVE MANIPULATION TECHNIQUES

Our framework allows two types of rules for combining user actions during cooperative manipulation: separation of DOFs and composition of user actions (section 3.3).

Our first experiments are using the separation of DOFs approach. Using this approach, we have investigated two types of collaborative metaphor, based on the interaction techniques we are combining. We classify our cooperative manipulation techniques into *Homogeneous* and *Heterogeneous* techniques. The first class includes cooperative techniques built from the same single-user interaction technique, while the second class contains cooperative techniques built from two different single-user techniques. In the following sections we present both homogeneous and heterogeneous techniques based on Simple Virtual Hand and Ray-casting.

4.1 Homogeneous Cooperative Techniques

To evaluate the combination of two Simple Virtual Hand techniques, we first allowed one user to control rotations and the other translations. This cooperative technique has proven very interesting when small adjustments are necessary to place the object in a small space such as a box or a hole. In such cases, while one user places the object in the desired position, the other can adjust its orientation, to make the placement easier. We have also noticed that this technique is very useful when the user that is controlling the rotations is able to see parts of the manipulated object (or of the docking object) that the other user cannot.

One example of this situation is when the users have to pass a couch through a door or a window and they are positioned on either side of the wall.

Another way to combine two Simple Virtual Hand techniques is to allow the primary user to translate the object left/right and up/down, while a second user translates the object in/out (the depth dimension relative to the primary user).

It can be difficult for a single user to manipulate an object's depth, especially when the VE system does not support stereo images.

This technique works best when the second user faces in a direction perpendicular to that of the primary user, so that the in/out direction for the primary user is the left/right direction for the second user.

Two Ray-casting techniques can also be combined to form a cooperative technique. Again, we can allow one user to control the object position and the other to control its orientation. The only difference when Ray-casting is used is that the object can be selected at a distance. This cooperative technique makes it simpler to place an object far from the first user. It also facilitates rotations that are difficult to perform using single-user Ray-casting. In Figure 1c, for example, if a single user (using Ray-casting) needs to perform a rotation around the object's vertical axis, many movements will be necessary

The second configuration we have tested to combine Ray-casting techniques is exactly the same as single-user Ray-casting, except that a second user controls the sliding of the object along the first user's ray. This technique has proven to be useful for tasks in which it is necessary to place the object at a precise depth far from the first user, or when the first user is not able to see the final object position because there are other objects between him and the manipulated object.

4.2 Heterogeneous Cooperative Techniques

We have also tested heterogeneous techniques in which Ray-casting is combined with Simple Virtual Hand in various ways. The first configuration we tested allowed the Simple Virtual Hand user to control rotations and the Ray-casting user to control translations and sliding along the ray. The results with this technique were similar to those obtained when two users used Ray-casting with the same configuration.

In the second configuration we tested, the user with the Simple Virtual Hand technique also controls the object sliding along his partner's ray. This sliding is controlled by moving the pointer along the X-axis in the user's coordinate system. The possibility of moving the object along the partner's ray is quite helpful in those cases where the desired position is far from the Ray-casting user.

Figure 3, for example, shows a situation where the user with Ray-casting needs to place an object between two distant walls. In this case, the user with the Simple Virtual Hand technique can easily adjust the object along the ray and also set the correct orientation for the object.

4.3 Summary

Table 3 summarizes the cooperative techniques we have developed and tested. In this table **RC** stands for *Ray-casting* technique, **SVH** for *Simple Virtual Hand*, **IT** for *Interaction Technique* and **DOF** for *Degrees Of Freedom*.

5. PRELIMINARY RESULTS

To test our collaborative techniques we performed a study involving twelve users grouped into six pairs. We asked each pair to execute one of the following tasks (see Figure 1):

- To place a set of objects on a set of platforms. The objects must be placed with a specific orientation;
- To move a couch through a door, with the users on opposite sides of the wall;
- To place a set of objects between some walls. These walls are far from one user and near to the other one.

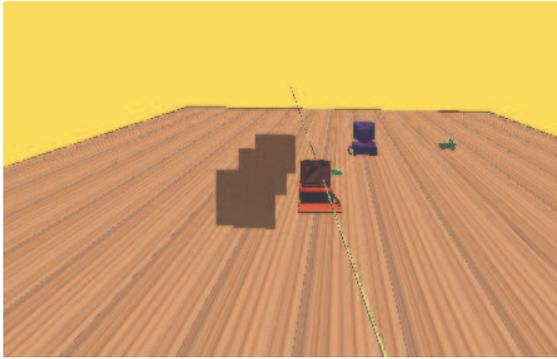
Both users wore a Virtual Research V8 head-mounted display (HMD), and had their head and one hand tracked with a Polhemus Fastrak tracking system. Each user also had a button that was used to indicate grabbing and releasing virtual objects. Users were seated, facing one another, and each had a unique point of view in the shared VE.

First, we asked each user to perform the task alone. Next, we told them to try to help each other do the task, without manipulating the object cooperatively. In other words, users were allowed to manipulate the object sequentially, but not simultaneously. Finally, we asked them to execute the task cooperatively.

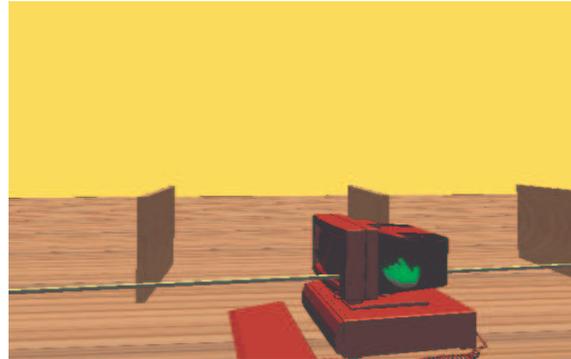
In this preliminary experiment our goal was to evaluate two main issues. First, does cooperative manipulation lead to greater efficiency or ease of use as compared to single-user manipulation or sequential manipulation? Second, is it possible to quickly learn how to use a cooperative technique, once one knows the single-user technique?

Our observations of the users and subsequent interviews led us to the following preliminary conclusions:

- Cooperative techniques can provide increased performance and usability in difficult manipulation scenarios. However, single-user manipulation is simpler to use and understand for most manipulation tasks;
- The use of a cooperative technique is applicable to those situations in which cooperation allows the users to control some DOFs that cannot be controlled with the single-user technique;
- The ease of learning for cooperative techniques depends more on the individual user than on the technique itself, i.e., those users who learned quickly how to use an individual technique also learned quickly how to use the cooperative one. On the other hand, those who had difficulty in learning the individual technique also took more time to learn the cooperative one;
- Users adapted to the system and learned the appropriate times to manipulate objects individually and cooperatively. Users had no trouble with the transition between single-user mode and cooperative mode because of our careful design and implementation.



Distant user viewpoint



Near user viewpoint

Figure 3 – Positioning distant objects

IT A	DOF User A	IT B	DOF User B	Comments
SVH	Position	SVH	Rotation	Useful for docking tasks and small adjustments. Good when one user cannot see parts of the object.
SVH	X, Y	SVH	Z	Facilitates precise positioning
RC	Position	RC	Rotation	Useful for rotations that are difficult with RC
RC	Position	RC	Rotation Slide	Useful for distant placement and rotations
SVH	Rotation	RC	Position	Useful for rotations that are difficult with RC
SVH	Rotation Slide	RC	Position	Useful for distant placement and rotations

Table 3 - Cooperative techniques tested in our framework

6. CONCLUSIONS AND FUTURE WORK

This paper has presented a framework to allow cooperative manipulation – the simultaneous interaction of two users with a single object. The system is based on the Collaborative Metaphor concept that allows the combination of multiple manipulation techniques. Unlike prior work, our system combines existing “magic” interaction techniques instead of combining user movements based on physical laws. Our system also incorporates awareness information that shows a user the activity and the capabilities of his partner. Currently the system shows only simple information about the object state, the user and the interaction technique. In the future we intend to incorporate more complex feedback that also shows object activity.

We also plan to implement and investigate more powerful manipulation techniques such as HOMER [3], WIM [23] and Go-Go [19] for cooperative manipulation. Another interesting issue is how to allow two users to control the same DOF. Preliminary tests have shown that, for example, if both users can slide the object in the direction of their rays, the sum of these movements can be useful for translation tasks such as moving a piece of furniture inside a house. We are also looking for ways to define dynamically, during the interaction process, which DOFs each user can control.

Finally, we are planning a more extensive, formal, and empirical evaluation of the cooperative techniques. In order to validate the need for cooperative manipulation, we will compare cooperative techniques to the best single-user techniques, with the aim of demonstrating that for certain tasks, cooperation significantly increases performance and usability.

7. ACKNOWLEDGEMENTS

The authors would like to thank the subjects in the experiment for their time and effort. We also acknowledge Drew Kessler for his help with the SVE toolkit. Marcio S. Pinho was supported by grant number BEX0316/01-6 from the Brazilian Foundation for the Coordination of Higher Education and Graduate Training (CAPES).

8. REFERENCES

- [1] Azuma, R., Baillet, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B., “Recent Advance in Augmented Reality”. *IEEE Computer Graphics and Applications*, 21 (6):34-47, November 2001
- [2] Billinghurst, M. “Shared Space: An Augmented Reality Approach for Computer Supported Cooperative Work”. *Virtual Reality*. 1998. Vol. 3, No. Springer.
- [3] Bowman, D. and Hodges, L. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, 1997, pp. 35-38
- [4] Bowman, D., & Hodges, L. “Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments”. *The Journal of Visual Languages and Computing*, 10(1), 37–53.

- [5] Broll, W., Meier, E., Schardt, T. "The Virtual Round Table - A Collaborative Augmented Multi-User Environment". ACM Collaborative Virtual Environments, 2000, pp. 39-46.
- [6] Curry, K. M. "Supporting Collaborative Awareness in Tele-immersion". Master Thesis. Virginia Polytechnic Institute. June, 1999.
- [7] Dumas, C., Degrande, S., Saugis, G., Chaillou, C., Viaud, M.-L., and Plénacoste, P. "Spin: a 3d Interface for Cooperative Work". Virtual Reality. 1999. Vol. 4, No.1. Springer. pp. 15-25.
- [8] Frécon, E. Stenius, M. "DIVE: A scaleable network architecture for distributed virtual environments". Distributed Systems Engineering Journal (special issue on Distributed Virtual Environments), 5(3):91-100, September 1998.
- [9] Greenhalgh, C. Benford, S. MASSIVE: A Virtual Reality System for Tele-conferencing, ACM Transactions on Computer Human Interfaces (TOCHI), Volume 2, Number 3, pp. 239-261, ISSN 1073-0516, ACM Press, September 1995.
- [10] Kessler, G., Bowman, D., and Hodges, L. The Simple Virtual Environment Library: An Extensible Framework for Building VE Applications. Presence: Teleoperators and Virtual Environments, vol. 9, no. 2, 2000, pp. 187-208.
- [11] Kiyokawa, K. Takemura, H. Yokoya, N. "Seamless Design for 3D object Creation". IEEE Multimedia. Jan 2000.
- [12] Li, F. W.B.; Lau, R.W.H.; Ng, F.F.C. "Collaborative Distributed Virtual Sculpting", Proceedings of IEEE VR 2001, 2001, Yokohama, Japan, pp 217-224.
- [13] Macedonia, Michael R., Zyda, Michael J., Pratt, Donald P., and Barham, Paul T., Zeswitz, Steven, "NPSNET: A Network Software Architecture for Large-Scale Virtual Environments", in Presence, Vol. 3, No. 4, Fall 1994, pp. 265-287, MIT Press.
- [14] Margery, D., Arnaldi, B. Plouzeau, N., "A General Framework for Cooperative Manipulation in Virtual Environments". Virtual Environments 1999, M. Gervautz, A. Hildebrand, D. Schmalstieg (eds.), Springer, pages 169-178, 1999.
- [15] Mine, M., Virtual environment interaction techniques. UNC Chapel Hill CS Dept.: Technical Report TR95-018. 1995.
- [16] Nishino, H., Utsumiya, K., Sakamoto A., Yoshida, K., Korida, K. "A method for sharing interactive deformations in collaborative 3D modeling", Proceedings of ACM VRST, pp. 116-123, Dec 1999.
- [17] Noma, Miyasato, "Cooperative Object Manipulation in Virtual Space using Virtual Physics", ASME-DSC-Vol.61, pp.101-106, 1997.
- [18] Ohshima, T., Satoh, K., Yamamoto, H. and Tamura, H. "AR² Hockey: A case study of collaborative augmented reality". IEEE Virtual Reality 1998. pp 268-275.
- [19] Poupyrev, I., M. Billinghurst, S. Weghorst and T. Ichikawa. "The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR". In: ACM UIST, 1996, Seattle, WA Proceedings..., New York, NY: ACM Press, 1996, p.79-80.
- [20] Raymaekers C., De Weyer T., Coninx K., Van Reeth F., Flerackers E. "ICOME: An Immersive Collaborative 3D Object Modeling Environment", Virtual Reality, 1999 Vol 1. No 4. pp. 265-274.
- [21] Schroeder, R. Steed, A., Axelsson, A-S., Heldal, I., Abelin, A., Wideström, J., Nilsson, A., Slater, M. "Collaborating in networked immersive spaces: as good as being there together?". Computer & Graphics, 2001, No.25. pp. 781-788.
- [22] Slater M, Sadagic A, Usuh M, Schroeder R. "Small group behaviour in a virtual and real environment: a comparative study". Presence: Journal of Teleoperators and Virtual Environments. 2000;9(1):37-51.
- [23] Stoakley, R., Conway, M., Pausch, R., Virtual reality on a WIM: interactive worlds in miniature. Proceedings of CHI'95. 1995. pp. 265-272.
- [24] Szalavari, Z., Schmalstieg, D., Fuhrmann, A., Gervautz, M. "Studierstube: An environment for collaboration in augmented reality". Virtual Reality. 1998. Vol. 3, No.1. Springer. pp. 37-48.
- [25] Tramberend, H. "AVOCADO - A distributed Virtual Environment Framework". ". Proceedings of IEEE Virtual Reality 1999. Houston, Texas. pp. 14-21.
- [26] Vuillème, A. Capin T., Pandzic, I., Magnenat, N., "Nonverbal Communication Interface for Collaborative Virtual Environments". Virtual Reality. 1999. Vol. 4, No.1. Springer. pp. 49-59.
- [27] Watson, K.; Zyda, M.. "Bamboo - a portable system for dynamically extensible, real time, networked, virtual environments". In 1998 IEEE VRAIS, pages 252-260. IEEE Computer Society Press, March 1998.