Cooperative Object Manipulation in Immersive Virtual Environments

MÁRCIO S. PINHO
Instituto de Informática, UFRGS
Faculdade de Informática, PUCRS
Porto Alegre, RS, BRAZIL

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

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, has to place an object far from its current position. 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. 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. 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 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 to 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. 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. We also aim to switch 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.

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 [5].

2. Related Work

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. [3], for example, many users can manipulate the same object at the same time, but the object must be modeled with NURBS surfaces, so that when one user selects the object, he actually gets exclusive access to the shape, position and orientation of only one patch. The ICOME system [7], 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 a few examples of actual cooperative manipulation in VEs, most of them using force feedback devices [8]. These devices are used to constrain a user's hand movements by simulating the forces one would feel based on the partner's actions. Margery [4] presents the only work we found that actually employs cooperative manipulation. His architecture supports 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 field of collaborative virtual environments (CVEs) systems like AVOCADO, Bamboo , DIVE, MASSIVE , RAVEL, Urbi et Orbi and NPSNET ensure that 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;
- How to combine two interaction techniques;
- How to show to one user what his partner is doing.

The main difference between our technique and the methodology presented by Margery [4] 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 [1] or Go-Go [6]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 modules described in the following sections (see Figure 1).

3.1 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.

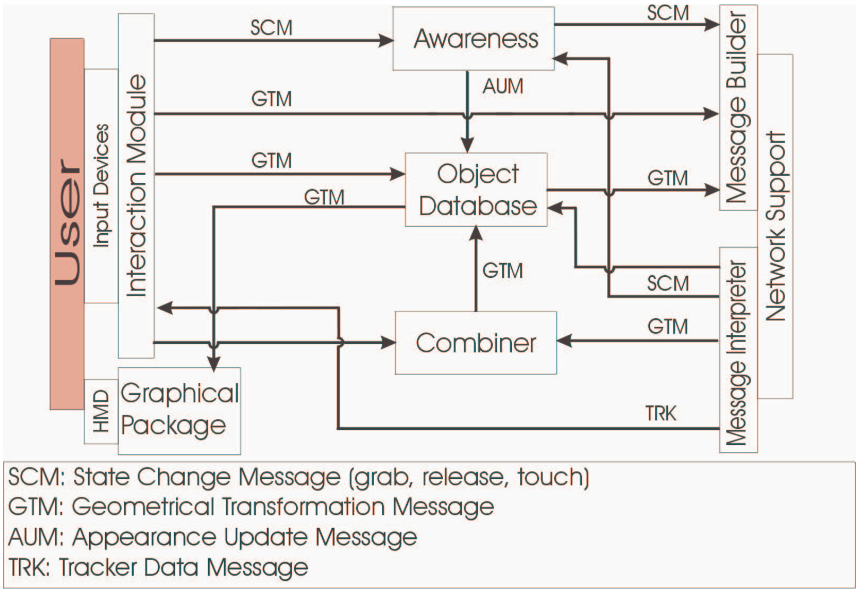


Figure 1 - Cooperative manipulation framework

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 [2] 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.

3.2 Command Combiner

The *Command Combiner* combines 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, based on the idea of ***separating 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 (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.

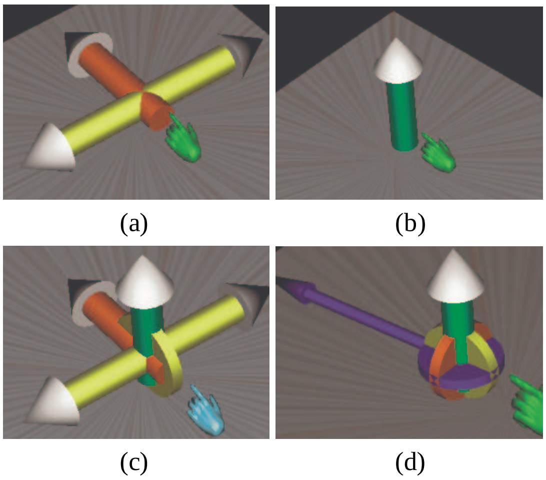
3.3 Awareness Generator

This module is responsible for providing information about the partner and his activities inside the VE. In our system we subdivide the awareness information into three categories: *user information*, *interaction information* and *object state information*.

The *user information* is generated from the user position and orientation and is used to produce understanding and awareness of the other user. The *interaction information* generates the necessary visual information in such a way that one 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. Another important information about interaction is which degrees of freedom each user controls. Our architecture provides this information by changing the pointer's geometry displaying arrows and circles based on the axis the user can controls. Figure 2 shows some pointers have been used to produce the awareness about this. In (a) the pointer shows a situation where the user can move object on plane XZ. The image (b) shows that the only possible displacement is along Y axis. In (c) the translation is possible along all three axis and rotation is possible only around X axis. The last image (d) represents a situation where the user can slide (see the thin cylinder) the object along the ray, translate on Y and apply rotations along all three axis.

The *object state information* helps 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*.

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 1).



(a) (b)

(c) (d)

Figure 2 – Some possible pointer's Geometry

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 1 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.

4. Cooperative manipulation techniques

We have investigated types of collaborative metaphors. In this section we describe some of them.

The first configuration we tested allowed one user to control rotations and the other to control translations and sliding along the ray. The results with 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.

Another configuration tested was to allow the primary 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). 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.

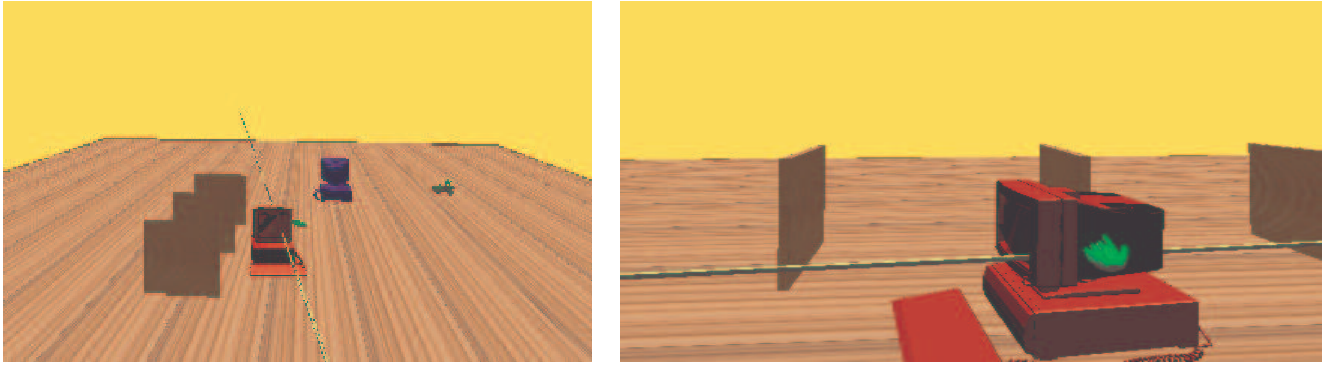
Combining techniques based on the Ray-casting metaphor can also be an interesting way to form a cooperative technique. Again, we can allow one user to control the object position and the other to control its orientation. This cooperative technique makes it simpler to place an object far from the first user. It also facilitates rotations that are difficult to perform using single-user Ray-casting (e.g. rotation about the object's vertical axis).

Another interesting configuration we tested, is the one where the user with the Simple Virtual Hand technique can 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). 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.

User A (Texture)	User B (Color)	Object Color/Texture	
No Touching	No Touching	Object original color	
No Touching	Touching	User B (light) color	
No Touching	Grabbing	User B color	
Touching	No Touching	User A (light) texture	
Touching	Touching	User A (light) texture	+ User B (light) color
Touching	Grabbing	User A (light) texture	+ User B color
Grabbing	No Touching	User A texture	
Grabbing	Touching	User A texture	+ User B (light) color
Grabbing	Grabbing	User A texture	+ User B color

Table 1 - Possible object states and system feedback

.



Distant user viewpoint

Near user viewpoint

Figure 3 – Positioning distant objects

4.1 User Studies

To test our cooperative techniques we performed a study involving 60 users grouped into 30 pairs. We asked each pair to execute tasks such as placing a set of objects on a set of platforms, moving a couch through a door with the users on opposite sides of the wall, and placing a set of objects between some walls.

In these experiments 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?

The first question we tried to answer by making a set of tests in which the users were asked to perform some task individually and collaboratively. For “individual” test each user works on a different object, but they can work in parallel to perform the task. For “collaborative” test they must work together on the same object all the time. In Table 2 we show the results of a test in which the users were asked to place computers over some tables inside a classroom. The Figure 4 shows the results in a graphical way.

The statistical analysis using *t-Test* proved that the mean difference (02:09) between the times spent to perform the task alone and cooperatively is very significant (p=0.000052613).

Concerning the ease of learning, we concluded it depends more on the individual user skills 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.

Our observations of the users’ actions and subsequent interviews led us to the following additional 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 better control some DOFs that cannot be easily controlled with the single-user technique;

Users adapted to the system and learned the appropriate times to manipulate objects individually or cooperatively. Users had no trouble with the transition between single-user mode and cooperative mode because of our careful design and implementation.

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 Pinho was supported by grant number BEX0316/01-6 from the Brazilian Foundation for the Coordination of Higher Education and Graduate Training (CAPES).

Couple	With Single User Technique	With Cooperative Technique	Difference	% Difference
1	06:15	03:20	02:55	47%
2	03:29	03:20	00:09	4%
3	09:37	06:41	02:56	31%
4	09:41	07:30	02:11	23%
5	03:50	01:36	02:14	58%
6	07:10	04:30	02:40	37%
7	06:40	04:05	02:35	39%
8	08:50	06:10	02:4	30%
9	07:30	04:40	02:50	38%
10	04:30	04:10	00:20	7%
Mean	**06:45**	**04:36**	**02:09**	**32%**

Table 2 – User Study Results

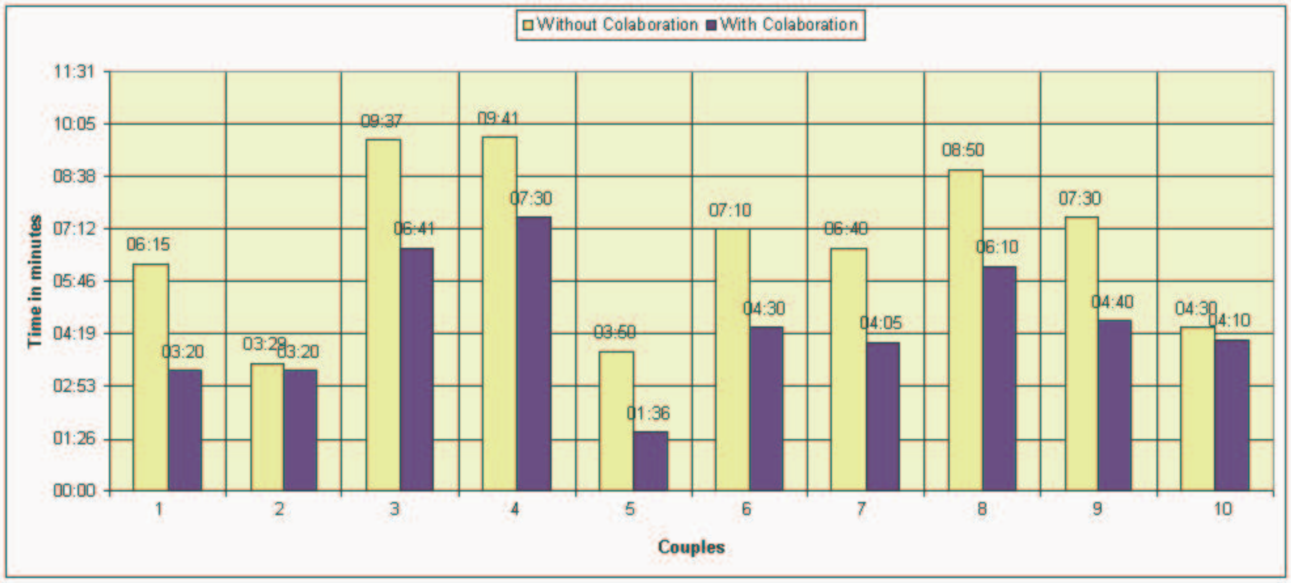


Figure 4 - User Studies Graph

References

[1] 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

[2] 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.

[3] Li, F. W.B.; Lau, R.W.H.; Ng, F.F.C. “Collaborative Distributed Virtual Sculpting”, Proceedings of IEEE VR 2001, 2001, Yokohama, Japão, pp 217-224.

[4] Margery, D., Arnaldi, B. Plouzeau, N., “A General Framework for Cooperative Manipulation in Virtual Environments”. Virtual Environments 1999, Springer, pages 169-178, 1999.

[5] Pinho, Marcio Serolli; BOWMAN, Doug A; FREITAS, Carla M D S. Cooperative Object Manipulation in Immersive Virtual Environments: Framework and Techniques. In: VIRTUAL REALITY SOFTWARE AND TECHNOLOGY, 2002, Hong Kong. Virtual Reality Software and Technology. 2002.

[6] Poupyrev, I., M. Billinghurst, S. Weghorst and T. Ichikawa. “The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR”. In: ACM Symposium on User Interface Software and Technology, 1996, Seattle, WA Proceedings…, New York, NY: ACM Press, 1996, p.79-80.

[7] 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.

[8] Sallnäs, E-L., Rassmus-Gröhn, K., Sjöström, C. (2000). Supporting Presence in Collaborative Environments by Haptic Force Feedback. ACM Transactions on Computer-Human Interaction. 7(4): 461-476.